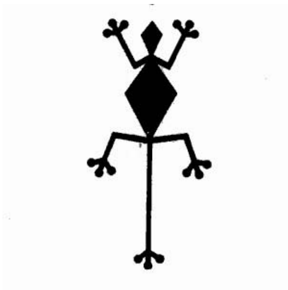
# Modalities Through the Looking Glass

A study on coalgebraic modal logics and their applications

**Raúl Andrés Leal**

# Modalities Through the Looking Glass

A study on coalgebraic modal logics and their applications



ILLC Dissertation Series DS-2011-09

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

# Modalities Through the Looking Glass

A study on coalgebraic modal logics and their applications

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Aula der Universiteit
op vrijdag 18 November 2011, te 11.00 uur

door

Raúl Andrés Leal Rodríguez

geboren te Bogota, Colombia

Promotiecommissie:

Promotor: Prof. dr. Y. Venema

Overige leden:
Prof. dr. J.F.A.K. van Benthem
Prof. dr. J. Adámek
Prof. dr. B.P. F. Jacobs
Dr. D. Pattinson
Dr. A. Kurz
Dr. A. Baltag

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam

The work comprising this dissertation was made possible by VICI grant 639.073.501 of the Netherlands Organization for Scientific Research (NWO).



*Cover:*
Design: Elske Verdoorn.
Image in patron: Pectoral Zoomorfo, Altiplano Nariense - Nario Tardo, 15,1 x 16,2 cm 600 d.C. - 1700 d.C. ©COLECCION MUSEO DEL ORO - BANCO DE LA REPBLICA, Bogota - Colombia. Photograph: CLARK M. RODRIGUEZ

Printed and bound by Ponsen & Looijen.

ISBN: 978-90-5776-231-4

*para Lena*

# Contents

# Acknowledgments

On Monday 29 of August 2005 at 7:35am a Delta Airlines plane coming from Atlanta-USA landed at Schiphol airport. Around 8:05am, I crossed gate number 1 and that was the beginning of my Holland experience. This dissertation represents the culmination of my basic scientific education at the ILLC. I would have never gotten to this point without the teachings, encouragement, advice, and criticisms of many. During my studies I was lucky enough to meet many that made the journey worth it. In the next pages, I would like to thank them. By construction the list here is condemned to be incomplete.

On the top of the list is my supervisor Yde Venema. You would not have been reading these lines if he would not have believed in me. We have a long history together as long as my footprints in the polders can witness. He first welcomed me as an MSc student. Right after that, he accepted the challenge of keeping me on his side as a PhD student. I have learned a lot from him both in the academic and personal. In particularly, I really appreciate his honesty in the good and the bad times. I thank him for being a very caring supervisor and a very good friend.

At least half of this manuscript is influenced by Alexander Kurz. He was almost a second supervisor. Always ready to show me a way out when I would get myself lost between diagrams, colimits, and deadlines. As shown in the cover, most of this manuscript is the influence of Alexander and Yde. Yde drew a framework and Alexander showed me the techniques to organise it and fill it with beauty.

The only person that has been on my side longer than Yde is Lena. We literally met during my first sunshines in Amsterdam. At some point during our master studies the stars were right and our paths have been together since. She has been my unbounded support 24/7. Her patience, love, arepitas, and coffee give me the daily motivation to be where I am now.

Thanks to my coauthors this manuscript has content. As said before, Alexander walked with me through the Stone age. Clemens was happy to explore the final simplicity that took us from basic modal logic to the realm of solution sets. Helle was adventurous enough to get modalities in movement. Gaelle and Yde were courageous enough to build bridges between our fields. Thanks to all of you as the content belongs to you.

I would like to thank to the illustrious members of the committee for having devoted their time and energy to read my dissertation. So thanks to Alexander Kurz, Bart Jacobs, Jiří Adámek, Dirk Pattinson, Alexandru Baltag, and Johan van Benthem.

I would like to thank NWO for having supported my research. I would also like to thank the Beth foundation and Colfuturo who were the first to believe that I could complete an MSc and eventually a PhD. Without their financial support I would have never landed at Schiphol that day.

Being a Colombian learning Dutch with a lot of organisational energy requires a enormous logistic support. The secretaries and staff members of the ILLC/UvA were always there to support me. Tanja, Peter, Karin and René were always helpful no matter whether it was an official letter for some obscure Dutch immigration procedure, a letter for some embassy, booking of a room, simply and opinion on my writing, or the need for a drink. I thank them also because they were among the first that supported me on moving beyond the English speaking culture of the ILLC. Among those, I should also mention Ingrid, and Leen. I also thank them for supporting me during my naturalizatie procedure. I thank Niels Molenaar for always being helpful on explaining the Kafkaian castle of Dutch bureaucracy. In several occasions, I was saved from sleeping on the street thanks to him and the housing office of the UvA.

I would like to thank Alexandra en Helle for their good balance of work and side life so that we could have very enjoyable times at conferences talking about live, coalgebras, and everything. I also than them and Clemens for reading and commenting on some parts of the final version of this manuscript.

I thank Frank for his help with mijn wetenschappelijke taal.

I met many wonderful people outside the university who have made my life very enjoyable. Most of those gravitate around climbing. I first would like to thank the Boulder Gang of Sloterdijk. In particular I would like to thank Casper and Remco who showed me how to avoid the elephant climbing style. Casper also taught me Dutch language beyond what any dictionary could do. There were also Fleur and Megan who showed me the girl way to climb. Still Fleur would climb better than me with an 8 months pregnancy belly; I will never beat that. I do not forget Daan, Koen, and Rudi among others. I learned to climb with them, this shaped my whole stay in the Netherlands.

Later I met the ASAC (Amsterdam Student Alpine Club) and learned even more about climbing. Within the ASAC Elout shines for various reasons. I learned most of my of my rope technique from him. During those many hours that we spent in his car driving to the rocks he also introduced to many faces of Dutch culture among which I will never forget Guus Meeuwis with “Brabant” and “Het is een nacht”. I also have met many friends in the ASAC which have made my stay in Amsterdam wonderful. Among those, Bas, Willem, Lieke D, Kimberly, Vicky, Marianne, Adam, Regien, Dorien, Anne, Eva, Jeroen, Lieke T, Carel, Jacobien, etc...

In my last year I moved from ice skating to rollerblading. I joined the Survivors and learned the basics from Anders and Martine. Indirectly, thanks to them this thesis came to an end. The reason is that after I learned to slide on wheels and stop without crashing I joined the Friday Night Skate Amsterdam (FNS). Many chapters and sections of this manuscript where finished thanks to the FNS as every Friday, when there was no rain, I just wanted to finish to join the skate.

Elske is one my oldest friends. Beside contemplating her beautiful eyes, I have always enjoyed hanging out with her talking about life, the universe and everything. I specially thank her for the beautiful design of the cover.

I would also like to thank Laura for all the coffees that gave a well deserved rest during the writing. Thanks to the Carnival gang, Spiros, Roberto, Felix, etc. for the crazy carnival trips and lunches that allowed me to drive my mind away of work.

I thank to my paraninfs, Lena and Willem, for their very professional job.

I finally would like to express my gratitude to my parents and family in Colombia for they support.

Raul Andres Leal
Amsterdam, October, 2011

# Chapter 1

# Introduction

*Life is what happens to you while you're busy making other plans*
**John Lennon**

This thesis hovers over the interaction of coalgebras and modal logics. Coalgebras arise from computer science as a promising mathematical foundation for computer systems. Modal logic has its origins in philosophy but thanks to the so-called relational semantics it has found its way to areas such as linguistics, artificial intelligence, and computer science [19]. In this thesis we study the offspring of the encounter of coalgebras and modal logics. More concretely, this manuscript has two parts: *Modalities in de Stone age* and *Coalgebraic modal logics at work*. In the first part of this manuscript we investigate *coalgebraic modal logics* which have become one of the main currents of modal logics for coalgebras. Coalgebraic modal logics bring uniformity to the rising wave of modal logics in computer science and provide generality to the interactions of coalgebras and modal logics. In the second part of this manuscript we investigate how these logics can be used to study coalgebras.

In this introduction we tell the story of how coalgebras meet modal logics. The structure of the introduction is as follows. In the next section we will elaborate on the relation of coalgebras and transition systems. In Section 1.2, we will give a brief overview of the historical framework that lead modal logics to become the language for coalgebras. We finish with Section, 1.3 where we sketch the outline and contributions of this thesis.

## 1.1 Coalgebras and Systems

We begin by discussing coalgebra in more detail. Widely speaking, coalgebras provide a mathematical theory of states and observations. The definition of coalgebra strikes by its simplicity.

A coalgebra consists of a state space, or set of states $X$; and a function

$$\xi : X \longrightarrow \mathcal{T}(X),$$

where the elements of $\mathcal{T}(X)$, depending on the contex, can be read as: the transitions from, or in, $X$, or the observations from $X$, or the computations from $X$. Formally speaking, $\mathcal{T}$ is given by a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$, where $\mathsf{Set}$ is the category of sets and functions.

Coalgebras provide a perspective to study different state-based systems where the set of states can be understood as a black box to which one has limited access. Universal coalgebra is a mathematical theory of systems which we understand via observations [96].

For a more intuitive illustration of the coalgebraic view, the reader can think of a coffee vending machine. Most people do not really know what the inner mechanism of the machine is, or even have ever seen such mechanism. Nevertheless they can use the machine efficiently. Here we have an interaction with a system from the black box perspective. We could even say that coalgebras are machines from the point of view of the user. With this in mind, we can stress that we all have had *coalgebraic experiences* and that they are more common than what we would think. Moreover, it is not rare to be confronted with situations where we make *coalgebraic inferences*. Here are two examples: The first example involves two machines that look identical but one pours coffee for free. In such a situation, after having observed the behaviour of the two machines, many of us will take advantage and use the the free machine as far as it keeps the behaviour. Here we have made the coalgebraic inference that in reality the inner mechanisms of the two machines are not identical, or at least do not behave the same, and the free machine is more advantageous for us. Elaborating a bit further, for most of us the inner mechanism of the machine is irrelevant. Most of us do not care why one machine gives free beverages and the other does not. We just care that as far as when we press the button we obtain the chosen beverage. The second example of coalgebraic inference is when this does not happen. In case we press the button and no beverage comes out of the machine we then make a coalgebraic inference and conclude that there is something wrong with the inner mechanism of the machine.

Next we show two more formal examples of coalgebras as systems to illustrate this perspective on coalgebras.

The first example is the so-called systems with termination, or one-button machines [58]. A one button machine is a function $\xi : X \longrightarrow 1 + X$, where 1 is a singleton set disjoint from $X$; an evaluation of the function accounts for pressing the button. We say that $x \in X$ terminates if $\xi(x) \in 1$; this is written $x \not\rightarrow$. If $\xi(x) = y$ then we write $x \longrightarrow y$ and say *"there is a transition from x to y"*.

The second example is deterministic automata on an alphabet $A$. These can be seen as coalgebras for the functor $2 \times (-)^A$. A coalgebra $\xi : X \longrightarrow 2 \times X^A$ is described by two functions $\xi_1 : X \longrightarrow 2$ and $\xi_2 : X \longrightarrow X^A$. The former function is the characteristic function of the set of accepting states of the automaton. The latter function describes the transition of the the system, i.e. if $\xi_2(x)(a) = y$ we write $x \xrightarrow{a} y$ and read *"there is a transition a from x to y"*.

Historically, the formal definition of coalgebras, as the dual of algebras, can be traced back to the origins of categorical algebra, see e.g [82]. However, coalgebras did not receive much attention as such. The field did not begin its rise until computer scientists realised that coalgebras could be used as an unifying environment to model different computer systems. Among the first illustrations of this we have the modelling of: linear dynamic systems [4, 10], data types of infinite objects [111, 11], and the behaviour of systems [93].

The crucial breakthrough that made coalgebra a promising field for a mathematical foundation of computer systems was achieved by Aczel ([2], and [3]). In those papers, Aczel introduced coalgebras for a functor $\mathcal{T}$ as a generalisation of transition systems, showing that Kripke frames, or non-deterministic transition systems, could be presented as coalgebras. On the top of this he made three crucial observations: 1) coalgebras come with a canonical notion of observational or *behavioural equivalence* (induced by the functor $\mathcal{T}$); 2) this notion of behavioural equivalence generalizes the notion of *bisimilarity* from computer science and modal logic; 3) any 'domain equation' $X \cong \mathcal{T}(X)$ has a canonical solution, namely the *final coalgebra*.

The idea of a type of dynamic systems being represented by a functor $\mathcal{T}$ and an individual system being a $\mathcal{T}$-coalgebra, led Rutten [96] to the theory of universal coalgebra which, parametrized by $\mathcal{T}$, applies in a uniform way to a large class of different types of systems. In particular, final semantics and the associated proof principle of coinduction (which are dual to, respectively, initial algebra semantics and induction) find their natural place here. These ideas have been proved very successful. For example, nowadays, coalgebras encompass such diverse systems as, for example, labelled transition systems [2], deterministic automata [95], $\pi$-calculus processes [38], HD-automata [36], stochastic systems [32], neighborhood frames [45], among others.

## 1.2 Coalgebras and Modal Logics

Modal logics began as a formalisation of modalities. We can trace its origins to philosophy almost a century ago, though the *informal* study of modalities can be traced to the work of medieval logicians and back to the ancient Greeks [19]. In the middle of the XX century a perspective of Modal Logics as a fine tuning of the structure of classical logics was developed [19]. This second perspective has its roots in the invention of graph based relational semantics (by J. Hintikka, S. Kanger, and S. Kripke), usually know as *Kripke semantics*. This perspective emphasises modal languages as means that bring to light the inner structure of classical systems. Or as said in [20], modal languages provide an internal, local perspective on relational structures. This view has helped to place modal languages as the appropriate choice of languages to describe coalgebras. Moreover, nowadays, it is also fair to say that *modal logics are coalgebraic* [28].

The use of modal logic for coalgebras can be seen as an attempt to open the black boxes of coalgebras. The reason for this is that modal formulas are, traditionally, evaluated on the states of a coalgebra. Hence, somehow coalgebraic modal logics require, and give, access to the state space of a coalgebra. The reason for this is duality; we will elaborate on this in Section 1.2.4.

We will now explain how the encounter of modal logic and coalgebras took place, and how from this modal logics became the language for coalgebras. We identify four steps, more details can be found below.

### 1.2.1 Basic modal logic & coalgebras

The first important step towards modal logics for coalgebras was achieved by Aczel in [2] where he noticed that Kripke frames are coalgebras for the covariant power set functor. We now explain this.

A Kripke frame, or non-deterministic transition system, is a pair $(X, R)$, where $X$ is a set and $R$ is a binary relation on $X$. To see these as coalgebras for the covariant power set functor notice that a binary relation $R \subseteq X \times X$ can be seen as a function

$$\xi_R : X \longrightarrow \mathsf{Pow}(X),$$

where $\mathsf{Pow}$ denotes the covariant power set functor. The function $\xi_R$ maps a set $x$ to its set of successors, i.e. $\xi_R(x) = \{y \in X \mid xRy\}$. Conversely, given a function $\xi : X \longrightarrow \mathsf{Pow}(X)$ we can define a binary relation $R_\xi$, on $X$ by $xR_\xi y$ iff $y \in \xi(x)$. In the future, if $y \in \xi(x)$ we write $x \longrightarrow y$ and read *"there is a transition from $x$ to $y$"*.

As we mentioned before, Kripke frames or graph based relational semantics was invented to study basic modal logic. Thus, the work of Aczel gave a first glimpse to the use of modal logics for coalgebras. Using the coalgebraic perspective on Kripke frames, the modalities $\Box$ *(universal modality)* and $\Diamond$ *(existential modality)*[1]

[1]Through this thesis we call $\Box$ the universal modality and call $\Diamond$ the existential modality, as

have the following interpretations on a coalgebra $\xi : X \longrightarrow \mathsf{Pow}(X)$.

$$[\![\Box\varphi]\!]_\xi = \Big\{x \in X \mid \xi(x) \subseteq [\![\varphi]\!]_\xi\Big\}, \text{ and } [\![\Diamond\varphi]\!]_\xi = \Big\{x \in X \mid \xi(x) \cap [\![\varphi]\!]_\xi \neq \emptyset\Big\}.$$

Since $\xi(x)$ is the set of successors of $x$, and by definition $xR_\xi y$ iff $y \in \xi(x)$. We can write the previous interpretations in the standard relational notation [20]; more explicitly this is

$$[\![\Box\varphi]\!]_\xi = \Big\{x \in X \mid (\forall y)(xR_\xi y \Rightarrow y \in [\![\varphi]\!]_\xi)\Big\}, \text{ and}$$
$$[\![\Diamond\varphi]\!]_\xi = \Big\{x \in X \mid (\exists y)(xR_\xi y \wedge y \in [\![\varphi]\!]_\xi)\Big\}.$$

### 1.2.2 Moss logic

A bit before Aczel, Moss & Barwise [16] made a step that would prove to be crucial for the development of coalgebraic modal logic. They introduced another perspective on basic modal logic. Their intention here was to account for non well-founded sets, and various phenomena involving circularity and self-reference. This logic is called the *Nabla logic* or the *Moss Logic*.

The Moss logic has one modality: the nabla $\nabla$. This modality is a bit exotic in the sense that it takes sets of formulas, instead of just formulas, as parameters. In particular, this means that the structure of the formulas in Moss logic is out of the scope of (standard) universal algebra as in [24]. The interpretation of $\nabla$ in a coalgebra $\xi : X \longrightarrow \mathsf{Pow}(X)$ is as follows: Given a set of formulas $\Phi$

$$[\![\nabla\Phi]\!]_\xi = \Big\{x \in X \mid (\forall y \in \xi(x))(\exists \varphi \in \Phi)(y \in [\![\varphi]\!]_\xi) \text{ and}$$
$$(\forall \varphi \in \Phi)(\exists y \in \xi(x))(y \in [\![\varphi]\!]_\xi)\Big\}.$$

The modality $\nabla$, and the usual modalities are interdefinable as follows:

$$\nabla\Phi = \Box \bigvee_{\varphi\in\Phi} \varphi \wedge \bigwedge \Diamond\Phi, \quad \Box\varphi = \nabla\emptyset \vee \nabla\{\varphi\}, \quad \Diamond\varphi = \nabla\{\top, \varphi\};$$

where $\Diamond\Phi = \{\Diamond\varphi \mid \varphi \in \Phi\}$, and $\top$ denotes truth.

The second braketrough towards modal logics as the appropriate language for coalgebras was achieved by Moss [84]. Moss noticed that his work on the $\nabla$ modality for basic modal logic [16] could be presented parametric in the functor $\mathcal{T}$ and lifted to the coalgebraic level of generality.

above. The reader should be aware that such terminology is sometimes used for the modalities *"for all states..."* and *"there exists a state..."*, respectively. We refer to those modalities as the *global universal modality* and the *global existential modality*, respectively.

His idea was to take $\mathcal{T}$ itself as a modality; this modality is denoted as $\nabla_{\mathcal{T}}$, or just $\nabla$. More precisely, if $\mathcal{M}$ is the set of formulas of the language and $t \in \mathcal{T}(\mathcal{M})$ then $\nabla t \in \mathcal{M}$. We postpone details of the semantics because in the general case the $\nabla$ modality can be quite involved. We treat this logic in full detail in Section 3.3.1.

The crucial contribution of [84], for the development of modal logics for coalgebras, was that it demonstrated that each coalgebra type comes equipped with a generic notion of modality. Hence it was quite natural to ask whether the uniformity that coalgebra brings to transitions systems could also be achieved for modal logics. This opened the door to the use of modal logics to describe, or specify, the behaviour of coalgebras. However, the immediate developments on modal logics for coalgebras did not follow Moss approach; an exception to this is [12]. Interest on the Moss logic came back to the research arena when Venema [108, 106] pointed out that Janin and Walukiewicz [60], had already, independently, observed that, in basic modal logic, the connectives $\nabla$ and $\vee$ may replace the connectives $\Box, \Diamond, \wedge, \vee$. This observation, which is closely linked to fundamental automata-theoretic constructions, lies at the heart of the theory of the modal $\mu$-calculus, and has many applications, see for instance [30, 97]. In [71] Kupke & Venema generalized the link between fix-point logics and automata theory to the coalgebraic level of generality by showing that many fundamental results in automata theory are really theorems of universal coalgebra. Future generalizations of Moss idea were investigated in [29]. A complete and sound system for the Moss logic was introduced in [69]. Recent work on the Moss Logic can be found in [17, 18].

### 1.2.3 Modal logics for coalgebras

After Moss [84] it was clear that the connection between modal logic and coalgebras was worth to develop. It was by then not totally clear how the general Moss modality could be seen as a modality in the standard extension of the word [20]. At a more technical level the modality $\nabla$ is not easily studied with the techniques of modal logic or universal algebra. Subsequently [72] and [52] proposed a more standard modal logic for a restricted class of coalgebras. Although both of these works presented generic methods to define standard modalities, they needed some involved machinery to describe the fundamental example of basic modal logic as in Section 1.2.1. It was Pattinson [88], taking ideas from Jacobs & Hermida [50], Jacobs [52] and Rößiger [94], who explicitely proposed a general *simple pattern* behind modalities like $\Box$ and $\Diamond$, and for coalgebras in general. Their idea was to use the so-called *predicate liftings* (Definition 3.1.1). Even more important was that Pattinson's approach presented modalities in the standard tradition of universal algebra i.e. without the use of a multi-sorted language as in [52]. The logic of all predicate liftings was first investigated in [99, 64]. We detail this perspective in Chapter 3.

Using predicate liftings, the motto *modal logics are coalgebraic* really gained strength. It has been shown that rank-1 modal logics are coalgebraic [100]. Some PSPACE bounds for coalgebraic logics have been established [102]. There are results on the finite model property [98], correspondence theory [103], and cut elimination in coalgebraic logics [90, 89]. But even more important are the innumerable applications of predicate liftings. In this thesis we will encounter predicate liftings in the following modal logics: basic modal logic, propositional dynamic logic, probabilistic modal logic, graded modal logic, non-normal modal logics, and game logic. To mention a few more, nowadays we can find predicate liftings for conditional logics [90], hybrid logics [86], description logics [101], and $\mu$-calculus [27], among others. There is even a coalgebraic logic satisfiability solver [25]. A more detailed survey of the scope of predicate liftings and coalgebraic logics in general can be found in [28].

### 1.2.4 From modalities to functors

The last step in the development of modal logics for coalgebras was to try to generalize the theory of boolean algebras with operators for modal logics, as in [20], to the coalgebraic level of generality. In fact, this development was done almost in parallel with the development of predicate liftings.

Here is an intuitive explanation of the idea. There are two key relations between concepts and mathematical structures which interest us for the moment. On the one hand we have coalgebras as a theory of systems [96]. On the other hand, we have the well known perspective of algebras as presentations of logics. The question is, how modalities fit into this picture. Modal logic can be seen as extension of classical logic. We rephrase this by saying that the intuitive concept of modality provides a link between logics and systems. Hence on the formal side, coalgebraic modal logics, should provide a link between algebras and coalgebras. The following picture depicts the situation:

$$\begin{array}{ccc} \textbf{systems} & \longleftrightarrow & \mathit{coalgebras} \\ \mathit{modalities} \updownarrow & & \updownarrow \text{modal logics} \\ \textbf{logics} & \longleftrightarrow & \mathit{algebras} \end{array} \tag{1.1}$$

On the left side we have the intuitions, or concepts; on the right side we have the formal mathematical structures. The horizontal lines represent the formalisation of the concept on the left side with the structure on the right side. Because of this interaction we can say that using modal logics we open the black box of coalgebra. More precisely, modalities transform coalgebras into algebras and the latter are well known to have a state space to which we have full access.

In the case of basic modal logic, this picture presents the theory of modal algebras, or algebraic semantics for modal logic. This perspective makes use of Stone duality, see [20] for details. A similar picture was also seen in domain theory [1] where a duality is used to connect systems and logics. The idea of semantics and syntax being connected by an adjunction can be traced back to Lawvere [78].

Modal algebras, or more generally boolean algebras with operators, are, nowadays, a well developed area in (classical) modal logic. They allow the use of powerful algebraic techniques to bear on modal logic problems, see [107] for a survey and more uses of algebra in modal logic. For example, the algebraic semantics is better behaved than the frame based semantics; every normal modal logic is complete with respect to its class of algebras, see [20] for details. Given the prosperity of Boolean Algebras with Operators in (classical) Modal Logic, it was quite natural to generalize those to arbitrary coalgebras to give an algebraic account of the rising mass of logics for coalgebras. First attempts towards a general framework can be found in [51, 52].

It was first explicitly noticed by A. Kurz and M. Bonsangue, in a talk at TACL(2003) [21], using ideas from [1], that the right vertical connection in Diagram (1.1) could be extended to all coalgebras by liftings, in a manner to be precise later, of the functors

$$\mathsf{Set}^{op} \underset{S}{\overset{P}{\rightleftarrows}} \mathsf{BA} \tag{1.2}$$

in Stone duality. Here $P$ is the contra variant power set functor and $S$ is the ultrafilters functor. This was later used in [68] to present coalgebraic logics as functors over $\mathsf{BA}$. More specifically, the situation is described by the following picture

$$\mathcal{T} \circlearrowright \mathsf{Set} \underset{S}{\overset{P}{\rightleftarrows}} \mathsf{BA} \circlearrowleft L \tag{1.3}$$

where the functor $\mathcal{T}$ provides the type of transition and the functor $L$ provides the modal logic. This is called the functorial framework for coalgebraic modal logics. This is the approach we follow in Part I of this dissertation.

## 1.3 Outline and contributions of this thesis

This thesis is divided in two parts. Part I, *Modalities in the Stone age*, is dedicated to show how Diagram (1.3) generalizes the theory of Boolean algebras with operators to arbitrary coalgebras and how using this we can lay a general map of coalgebraic logics. Part II, *Coalgebraic modal logics at work*, is dedicated to further investigate the uses of coalgebraic modal logics to describe coalgebras.

In the Historical context, Part I finds its natural place as follows: Both the Moss logic and the logic of all predicate liftings have their own merits to be called a generic logic of coalgebras. A natural question/task was to compare them to explicitly reveal their similarities and differences. A first systematic study for a restricted class of functors was started in [80, 79]. There the relation between the Moss logic and logics of predicate liftings was not clarified mainly because the theory of Boolean algebras with operators was not used in its full power. Thus to depict the general map of coalgebraic modal logics a more general approach that could account for the algebraic semantics of modal logic was needed. More explicitly, Part I has its origins when trying to lay a general picture of modal logics in the functorial framework.

Part II finds its natural place as follows: One of the insights of Part I is to show that all coalgebraic logics in the functorial framework are logics of predicate liftings. Hence a natural task is to investigate some concrete boundaries of the framework. In Part II we study three cases where we try to see how far can we push the use of coalgebraic modal logics.

We now outline the chapters of this dissertation and the contributions of the material.

Chapter 2 is a technical introduction to the material of this dissertation. Here we fix our basic notation. We begin by introducing algebras and coalgebras. Quite some space is dedicated to algebras for a functor and their relation to algebraic signatures. This will be of crucial importance to show how to arrive to the functorial framework for coalgebras. We also discuss varieties and more specifically the finitely presentability of varieties of finitary signatures. Those will be important technical tools in our work. We finish the Chapter by formally introducing the modal logics that will make the running examples trough this manuscript, in particular, we also describe the coalgebraic semantics for those logics.

The following three chapters constitute Part I of the manuscript.

Chapter 3 introduces the functorial framework for modal logics. Most of the ideas and definitions here are taking from the literature. Novel to the chapter is the presentation. We first introduce modal similarity types in the tradition of universal algebra using predicate liftings. We then develop in detail how from here we can arrive to the functorial framework in Diagram (1.3). We deliberately do not use Stone duality to introduce modal logics because we want to stress that the duality is not essential to define modal logics and compare them. In this Chapter we try to find a balance between categorical abstraction and concrete presentation. Since coalgebras are more naturally presented in the categorical language, the Chapter is more inclined to the categorical perspective. In particular, we introduce the notion of coalgebraic modal logic based on a category of power set algebras (Definition 3.2.12). This notion allows us to handle various modal logics not based on modal logic.

Chapter 4 introduces the basic translation techniques to compare coalgebraic modal logics. The contributions of the chapter can be summarised as follows: we introduce the notion of *one-step translation* (Definition 4.1.1). We develop the notions of *singleton lifting* (Definition 4.2.5) and *translator* (Definition 4.2.1), from [79], into the functorial framework for coalgebraic modal logics. In particular we show that every singleton lifting has a translator. We develop the notion of *logical translator* (Definition 4.3.2) for any category of power set algebras (Definition 3.2.12). The most important contribution is the use of the structural properties of the category of boolean algebras to show that every translator induces a one-step translation. We give conditions on the type of coalgebras for a translation between the Moss logic and logics of predicate liftings to exist.

Chapter 5 illustrates various uses of translators and translations introduced in the previous chapter. The main contributions of this chapter can be summarised as follows: We introduce a new type of predicate liftings called the Moss liftings (Definition 5.1.12). Using these we can define a new translation of the Moss Logic into the language of predicate liftings (Proposition 5.1.21). We prove two representations theorems that exhibit any coalgebraic modal logic (Definition 3.2.13) as a logic of predicate liftings (Theorems 5.2.2 and 5.2.17). We develop a novel equational coalgebraic logic (Section 5.3) with a sound and complete axiomatization for it.

The following three chapters form Part II of this dissertation.

Chapter 6 discusses the relation between final coalgebras and languages for coalgebras. Our main contribution is a systematic study of the relationship between following three characterisations of behavioural equivalence for coalgebras: the structural characterisation using final coalgebras, the logical characterisation using coalgebraic languages, the structural characterisation using logical congruences. The gain of this study is that we simplify various of the existing proofs in the literature. In particular we show that the relation above mostly relies on structural properties of the base category.

In Chapter 7 we concentrate on a coalgebraic framework for modal logics which encompasses (test free) Propositional Dynamic Logic and Game Logic. Our key idea is to consider extra structure on the functor $\mathcal{T}$. We exploit this structure to give the desired outer perspective on programs/games. For example, for sequential composition we assume $\mathcal{T}$ to be a monad (Section 7.3.1). The main contributions of this chapter can be summarised as follows. We provide a general notion of dynamic structure which describes the algebraic structure on programs, and their interpretation as $\mathcal{T}$-coalgebras. Once this view is in place, *labelled modalities* arise in a natural way by a generic process of labelling (Definition 7.1.1). We then proceed to investigate the nature of PDL and GL axioms such as $[a;b]\varphi \iff [a][b]\varphi$ and $[a \cup b]\varphi \iff [a]\varphi \wedge [b]\varphi$ in our general setting. We show that such axioms hold if the underlying $\mathcal{T}$-modality preserves the extra structure on $\mathcal{T}$ in a manner that we make precise in Theorem 7.3.7 (sequential composition) and Theorem 7.3.18 (pointwise operations).

In Chapter 8 we introduce automata for an *arbitrary* type of coalgebras (Definition 8.2.1). More precisely, given a set of monotone predicate liftings $\Lambda$, we introduce $\Lambda$-automata as devices that accept or reject pointed $\mathcal{T}$-coalgebras on the basis of so-called *acceptance games.* The main technical contribution of this chapter concerns a *small model property* for $\Lambda$-automata (Theorem 8.3.4). We show that any $\Lambda$-automaton $\mathbb{A}$ with a non-empty language recognises a pointed coalgebra $(\xi, x)$ that can be obtained from $\mathbb{A}$ via some uniform construction involving a satisfiability game (Definition 8.3.2) that we associate with $\mathbb{A}$. The size of $(X, \xi)$ is exponential in the size of $\mathbb{A}$. We also provide some hints of how coalgebra automata could be treated within the functorial approach to modal logics.

Our final chapter is Chapter 9. Here we discuss further paths for research.

### The origins of the material

Parts of the material in this thesis have been previously published or are currently awaiting publication. More specifically the relation between chapters and publications is as follows:

- Part I is based on [80] and joint work with Alexander Kurz [74, 73].
- Chapter 6 is based on joint work with Clemens Kupke [70].
- Chapter 7 is based on joint work with Helle Hansen [46]
- Chapter 8 is based on joint work with Gaëlle Fontaine and Yde Venema [41].

# Chapter 2

# Algebras & Coalgebras

This chapter is a technical introduction to the material that will be covered through this thesis. Here we fix our notation and terminology. With this chapter we also intend to point the reader to the relevant literature in the background of our work.

We have tried to make this manuscript understandable for both the modal logic community and to the category theory community. Of course we expect the manuscript to be understandable for the coalgebra community. Nevertheless, in several occasions we will use some advanced categorical techniques.

We do not expect the reader to be an expert in category theory but still some previous exposure to the subject is required. The reader should at least be familiar with basic concepts like functor, natural transformation, diagram, limit, colimit, and adjunction.

We do not require background in modal logic but familiarity with one or two modal logics will help to give body to the concepts introduced here.

We try to give a survey on coalgebras but it is far from complete and didactic. Some familiarity with the subject is expected.

This chapter is not intended to be self contained or present a state of the art of the subjects.

## 2.1 Coalgebras and Algebras

We begin with the general definition of algebras and coalgebras for a functor.

**Definition 2.1.1.** Let $\mathbb{C}$ be a category and $\mathcal{T} : \mathbb{C} \to \mathbb{C}$ be an endofunctor.

1. A $\mathcal{T}$-coalgebra is a pair $(X, \xi)$ where $X$ is an object in $\mathbb{C}$ and $\xi$ is an arrow

$$\xi : X \to \mathcal{T}(X) \qquad \text{(in } \mathbb{C}.)$$

We call $X$ the *carrier* or the *state space* and $\xi$ is called the *structural map*. A *pointed coalgebra* is a pair $(\xi, x)$ where $x$ is a point[1] in the state space of $\xi$.

2. A $\mathcal{T}$-algebra is a pair $(A, \alpha)$ where $A$ is an object in $\mathbb{C}$ and $\alpha$ is an arrow

$$\alpha : \mathcal{T}(A) \to A \qquad \text{(in } \mathbb{C}.)$$

We call $A$ the *carrier* and $\alpha$ is called the *structural map*.

The functor $\mathcal{T}$ is called the *signature functor* and the category $\mathbb{C}$ is called *base category*. Often, we denote $\mathcal{T}$-coalgebras and $\mathcal{T}$-algebras by using only the structural map, from which the carrier can be deduced. If there is no risk for confusion $\mathcal{T}$-coalgebras and $\mathcal{T}$-algebras will be simply called coalgebras and algebras, respectively.

The following will be our convention to denote algebras and coalgebras.

**Notation.** Structural maps will always be denoted by small Greek letters. Coalgebras will always have carriers denoted by $X, Y$ and $Z$; the structural maps will be $\xi, \gamma$ and $\zeta$, respectively. Algebras will always have carriers denoted by $A$ and $B$; the structural maps will be denoted by $\alpha$ and $\beta$ respectively. In some occasions, we will also use the letters $A, B, X$, $Y$ and $Z$ to denote objects in a category.

We now introduce morphism of algebras and coalgebras.

**Definition 2.1.2.** Let $\mathcal{T} : \mathbb{C} \to \mathbb{C}$ be an endofunctor.

1. A morphism between $\mathcal{T}$-coalgebras $(X, \xi)$ and $(Y, \gamma)$ is a morphism $f : X \to Y$, in the base category, such that the following diagram

$$\begin{array}{ccc} X & \xrightarrow{\quad f \quad} & Y \\ {\scriptstyle \xi}\downarrow & & \downarrow{\scriptstyle \gamma} \\ \mathcal{T}(X) & \xrightarrow[\mathcal{T}(f)]{} & \mathcal{T}(Y) \end{array}$$

commutes i.e. $\gamma f = \mathcal{T}(f)\xi$.

The category of $\mathcal{T}$-coalgebras, morphisms of $\mathcal{T}$-coalgebras, and usual composition is denoted by $\mathsf{Coalg}(\mathcal{T})$.

[1]Concerning points: In the case $\mathbb{C} = \mathsf{Set}$, points of the state space are just the elements of $X$. However, in a general category points are regarded as morphisms from an initial object into $X$.

2. A morphism between $\mathcal{T}$-algebras $(A, \alpha)$ and $(B, \beta)$ is a morphism $f : A \longrightarrow B$, in the base category, such that the following diagram

$$\begin{array}{ccc} \mathcal{T}(A) & \xrightarrow{\mathcal{T}(f)} & \mathcal{T}(B) \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ A & \xrightarrow[f]{} & B \end{array}$$

commutes i.e. $f\alpha = \beta\mathcal{T}(f)$. Morphism of $\mathcal{T}$-algebras are also called $\mathcal{T}$-homomorphisms.

The category of $\mathcal{T}$-algebras, morphisms of $\mathcal{T}$-coalgebras, and usual composition is denoted by $\mathsf{Alg}(\mathcal{T})$.

If there is no risk for confusion we simply refer to morphisms of coalgebras and algebras.

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

**Definition 2.1.3.** A *Kripke polynomial functor* [94], or KPF for short, is a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ built according to the following grammar

$$\mathcal{T} ::= \mathsf{Id} \mid \mathsf{K}_C \mid (-)^A \mid \mathsf{Pow} \mid \mathcal{T} + \mathcal{T} \mid \mathcal{T} \times \mathcal{T} \mid \mathcal{T} \circ \mathcal{T}$$

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

**Remark 2.1.4.** Notice that we can define a polynomial functor on any category $\mathbb{C}$ provided that $\mathbb{C}$ has products and coproducts.

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

1. We use $\mathsf{Pow} : \mathsf{Set} \longrightarrow \mathsf{Set}$ for the covariant powerset functor. This functor maps a set $X$ to its power set and a function $f : X \longrightarrow Y$ to its direct image.

2. The finite distribution functor $\mathcal{D}$ maps a set $X$ to the set of probability distributions on $X$, i.e. $\mathcal{D}(X)$ is the set of functions $\mu : X \longrightarrow [0, 1]$ such that $\sum_{x \in X} \mu(x) = 1$, with finite support. On functions, $\mathcal{D}$ maps a function $f : X \longrightarrow Y$ to the function $\mathcal{D}(f) : \mathcal{D}(X) \longrightarrow \mathcal{D}(Y)$ which maps a probability distribution $\mu : X \longrightarrow [0, 1]$ the function $\mathcal{D}(f)(\mu) : Y \longrightarrow [0, 1]$ given by

$$y \mapsto \sum_{x \in f^{-1}(\{y\})} \mu(x).$$

Since $\mu$ is a probability distribution with finite support so is $\mathcal{D}(f)(\mu)$. A similarly functor is the subdistribution functor, written $\mathcal{D}_{\leq}$, which maps $X$ to the set of sub-probability distributions, i.e. $\{\mu : X \longrightarrow [0,1] \mid \mu \text{ has fin. sup. and } \Sigma_{x\in X}\mu(x) \leq 1\}$.

3. We write $\mathcal{B}_{\mathbb{N}} : \mathsf{Set} \longrightarrow \mathsf{Set}$ for the finite multiset functor. The idea follows the same spirit used in the example of distributions. $\mathcal{B}_{\mathbb{N}}$ maps a set $X$ to $\mathcal{B}_{\mathbb{N}}(X)$ which consists of all maps ('bags') $B : X \longrightarrow \mathbb{N}$ with finite support. For $f : X \longrightarrow Y$, the function $\mathcal{B}_{\mathbb{N}}(f)$ maps $B : X \longrightarrow \mathbb{N}$ to the function $\mathcal{B}_{\mathbb{N}}(f)(B) : Y \longrightarrow \mathbb{N}$ given by $y \mapsto \sum_{x\in f^{-1}(\{y\})} B(x)$.

4. We use $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ for the contravariant power set functor. This functor maps a set $X$ to its power set and a function $f : X \longrightarrow Y$ to its inverse image. Recall that $\mathcal{P}(X) = 2^X = \mathsf{Set}(-, 2)$.

5. We use $\mathcal{P}^{op} : \mathsf{Set} \longrightarrow \mathsf{Set}^{op}$ for the dual of contravariant power set functor. If there is no risk of confusion we also denote this functor by $\mathcal{P}$.

6. The composition of $\mathcal{P}$ after $\mathcal{P}^{op}$, i.e. $\mathcal{P}\mathcal{P}^{op}$, is called the neighborhood functor.

7. We use $\mathsf{Mon} : \mathsf{Set} \longrightarrow \mathsf{Set}$ to denote the monotone neighborhood functor. This functor maps a set $X$ to its set of monotone neighbourhoods. More explicitly,

$$\mathsf{Mon}(X) = \{N \in \mathcal{P}\mathcal{P}^{op}(X) \mid \text{if } U \in N \wedge U \subseteq V \text{ then } V \in N\}$$

This functor maps a function $f$ to $(f^{-1})^{-1}$.

In several occasions, in order for everything to work properly, we will need some extra technical assumptions on functors. The next definition makes some of these assumptions explicit and precise.

**Definition 2.1.5.** Let $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ be a functor.

1. the functor $\mathcal{T}$ is *standard* if $\mathcal{T}$ preserves (non-empty) inclusions and the equalizer $0 \longrightarrow 1 \rightrightarrows 2$. The latter condition ensues $\mathcal{T}$ preserves monomorphism with empty domain.

2. Given a standard functor $\mathcal{T}$ and a regular cardinal $\kappa$, we can define the *$\kappa$-bounded* version of $\mathcal{T}$, written $\mathcal{T}_\kappa$, as follows: The functor $\mathcal{T}_\kappa$ maps a set $X$ to $\mathcal{T}_\kappa(X) = \bigcup\{\mathcal{T}(Y) \mid Y \subseteq X, |Y| < \kappa\}$; an arrow $f : X \longrightarrow Y$ is mapped to the restriction of $\mathcal{T}(f)$ to $\mathcal{T}_\kappa(X)$.

3. A standard functor is $\kappa$-accessible iff $\mathcal{T} = \mathcal{T}_\kappa$. A functor is accessible if it is $\kappa$-accessible for some regular cardinal $\kappa$.

4. A standard functor is said to be *finitary* iff $\mathcal{T} = \mathcal{T}_\omega$ i.e. if it is $\omega$-accessible.

5. A functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ *preserves finite sets* if it maps finite sets to finite sets.

Here are some illustrations of the previous definition.

**Example 2.1.6.** 1. All Kripke polynomial functors, as in Definition 2.1.3, preserve finite sets whereas the multiset functor and finite distributions functor do not.

2. The $\kappa$ bounded version of $\mathsf{Pow}$ maps a set $X$ to the set of subsets of $X$ of cardinality less than $\kappa$. In particular $\mathsf{Pow}_\omega$ maps $X$ to its set of finite subsets.

3. The finite multiset functor $\mathcal{B}_\mathbb{N}$ is the finitary version of the functor which maps a set $X$ to the set of functions $\mathsf{Set}(X, \bar{\mathbb{N}})$, where $\bar{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$; the action on arrows is the same of $\mathcal{B}_\mathbb{N}$.

4. The finite distribution functor is the finitary version of the functor which maps a set $X$ to the set of probability distributions on $X$, i.e. functions $\mu : X \longrightarrow [0, 1]$ such that $\sum_{x \in X} \mu(x) = 1$; the action on arrows is the same of $\mathcal{D}$.

The next remark shows that in all our investigations we can assume $\mathsf{Set}$-functors to be standard.

**Remark 2.1.7.** In all our investigations, without lose of generality, we can assume functors $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ to be standard. Indeed, given any $\mathcal{T}$ we can define $\mathcal{T}'(X) = \mathcal{T}(X)$ for $X \neq 0$ and $\mathcal{T}(0)$ as the equaliser $\mathcal{T}(0) \longrightarrow \mathcal{T}(1) \rightrightarrows \mathcal{T}(2)$. Further, given $\mathcal{T}'$ we can find a naturally isomorphic $\mathcal{T}''$ that preserves inclusions. The details can be found in [9]. The important point for us is that the categories of $\mathcal{T}$-coalgebras and $\mathcal{T}''$-coalgebras are (concretely) isomorphic.

The following examples illustrate the situation.

**Example 2.1.8.** Consider the functor $(-)^2$ which maps a set $X$ to the set of functions from 2 to $X$. This functor is not standard because functions can only be equal if their codomains are equal hence if $X \subseteq Y$ to say $X^2 \subseteq Y^2$ does not even make sense unless $X = Y$. However, $(-)^2$ is isomorphic to $\mathsf{Id} \times \mathsf{Id}$ which is standard.

A similar remark applies to $\mathcal{B}$ and $\mathcal{D}$. For example, $\mathcal{D}$ becomes standard if we replace a function $\mu : X \longrightarrow [0, 1]$ with the set $\{(x, \mu(x)) \mid \mu(x) \neq 0\}$.

The next remark concerns some of the general categorical definitions of accessibility.

**Remark 2.1.9.** In the general categorical world a functor is finitary iff it preserves filtered colimits and accessible iff it preserves $\kappa$-filtered colimit; see [7] for details.

In case $\mathcal{T}$ is not standard we can define the $\kappa$-bounded version of $\mathcal{T}$ by mapping a set $X$ to $\mathcal{T}_\kappa(X) = \bigcup\{\mathcal{T}(i_Y)[\mathcal{T}(Y)] \mid Y \subseteq X, |Y| < \kappa\}$, where $i_Y : Y \to X$ is the inclusion. Compare this with the computation of directed colimits in Proposition A.0.6.

## 2.2 More on Universal Coalgebra...

Coalgebras are generalized transition systems. The states of the system are the elements of the set $X$, the type of transitions are described by the functor $\mathcal{T}$ and the transitions of the system are given by the function $\xi : X \to \mathcal{T}(X)$.

In this section we introduce some of the basic theory of universal coalgebra. This includes bisimulations, relation lifting, and colimits. We begin by illustrating some examples of coalgebras as generalized transition system. The first three examples were already mentioned in the introduction but we repeat them for the sake of completeness.

**Example 2.2.1.**

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

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

3. Coalgebras for the covariant power set functor are Kripke frames, also known as non-deterministic (unlabelled) transitions systems [2]. For this, recall that a function $\xi : X \to \mathsf{Pow}(X)$ can be seen as a binary relation $R_\xi$, on $X$, defined as $xR_\xi y$ iff $y \in \xi(x)$. If $y \in \xi(x)$ we write $x \to y$ and read *"there is a transition from x to y"*.

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

5. Coalgebras for the finite distribution functor are discrete-time Markov chains [15], also known as probabilistic transition systems. This can be seen as

follows. Given a coalgebra $\xi : X \longrightarrow \mathcal{D}(X)$ and a state $x \in X$, we obtain a probability distribution $\xi_x = \xi(x) : X \longrightarrow [0,1]$. If $\xi_x(y) = p$, we write $x \xrightarrow{p} y$ and read *"the probability of having a transition from $x$ to $y$ is $p$"*.

6. Coalgebras for the finite multiset functor are directed graphs with $\mathbb{N}$-weighted edges, often referred as multigraphs [110]. The idea follows the same spirit used in the example of distributions. Given a coalgebra $\xi : X \longrightarrow \mathcal{B}_{\mathbb{N}}(X)$ we can describe the transitions as follows: if $\xi(x)(y) = n$ we write $x \xrightarrow{n} y$ and read *"there is a transition from $x$ to $y$ and the multiplicity, or weight, of $y$ is $n$"*.

7. Coalgebras for the neighborhood functor, i.e. $\mathcal{P}\mathcal{P}^{op}$-coalgebras, are known as neighborhood frames in modal logic and are investigated as coalgebras in [45]. A coalgebra $\xi : X \longrightarrow \mathcal{P}\mathcal{P}(X)$ can be interpreted as a two player game where a move in state $x_1$ consists of the first player choosing a set $S \in \xi(x_1)$ and the second player then the successor-state $x_2 \in S$.

### Bisimulations

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

**Definition 2.2.2.** Two states $x_i, (i = 1, 2)$, in two coalgebras $(X_i, \xi_i)$ are $\mathcal{T}$-*behaviourally equivalent*, written $x_1 \sim x_2$, if there is a coalgebra $(Z, \zeta)$ and two coalgebra morphisms $f_i : (X_i, \xi_i) \longrightarrow (Z, \zeta)$ such that $f_1(x_1) = f_2(x_2)$.

Going back to Example 2.2.1, one finds that this notion of behavioural equivalence coincides with the standard notions found in computer science. In detail: in Example 2.2.1 two states are behavioural equivalent in (1), iff they do precisely the same number of steps before terminating; in (2), iff they accept the same language [95]; in (3-7), iff they are behavioural equivalent in the sense of process algebra and modal logic [2, 96, 31, 45].

The next remark describes the relation of behavioural equivalence with the usual notion of bisimulation.

**Remark 2.2.3.** A bisimulation between two coalgebras $(X_1, \xi_1)$ and $(X_2, \xi_2)$ is a relation $B \subseteq X_1 \times X_2$ such that there is a coalgebra $B \longrightarrow \mathcal{T}(B)$ making the two projections $B \longrightarrow X_i$ into coalgebra morphisms. In case the functor $\mathcal{T}$ preserves weak pullbacks (see Proposition 2.2.10), to say that there is a bisimulation relating $x_1$ and $x_2$ is the same [96] as to say that $x_1$ and $x_2$ are behavioural equivalent according to Definition 2.2.2. In case $\mathcal{T}$ does not preserve weak pullbacks , the notion of bisimulation is problematic, e.g. it is not transitive, but the notion of behavioural equivalence still works fine.

One of the key contributions of universal coalgebra was to characterisation of behavioural equivalence (and bisimilarity) as a structural property of categories of coalgebras. As we mentioned, it was noted [2, 96] that behavioural equivalence (bisimilarity) could be characterised using final systems, also called final coalgebras.

**Definition 2.2.4.** A final coalgebra for an endofunctor $\mathcal{T}$ is a terminal object in $\mathsf{Coalg}(\mathcal{T})$. Explicitly, a final coalgebra is a coalgebra $\zeta : Z \longrightarrow \mathcal{T}(Z)$ such that for any coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ there exists a unique morphism $f_\xi : \xi \longrightarrow \zeta$. This morphism is called the *final map of* $\xi$.

Final coalgebras are to coalgebra what initial algebras or term algebras are to algebra, see e.g. [58]. The key result to keep in mind is the following:

**Proposition 2.2.5.** *If a final $\mathcal{T}$-coalgebra exists, two states $x_i$, $(i = 1, 2)$, in coalgebras $\xi_i$ are behavioural equivalent if and only if they are mapped to the same state of the final coalgebra, i.e. $f_{\xi_1}(x_1) = f_{\xi_2}(x_2)$.*

The previous result can be also read as follows: In case a final coalgebra exists, we can define the behaviour of a state as its image in the final coalgebra. In other words, the states of a final $\mathcal{T}$-coalgebra represent all the possible behaviours of states in $\mathcal{T}$-coalgebras, i.e. transition system of type $\mathcal{T}$.

Unfortunately, final coalgebras do not always exist. One way to show this is using Lambek's lemma.

**Lemma 2.2.6** (Lambek's Lemma)**.** *If a final $\mathcal{T}$-coalgebra exists, its structural map is an isomorphism.*

Notice that the previous lemma implies that there is no final $\mathsf{Pow}$-coalgebra because of cardinality reasons. In Chapter 6 we will discuss how to describe final coalgebras using logics for coalgebras.

Another reason to pay attention to final coalgebras is that they can also be used to formalise the notions of coinduction, which is dual to the notion of induction from algebra. We do not treat this here.

### Relation Lifting

Given relations $R \subseteq X \times Y$, $R' \subseteq Y \times Z$ we write their composition as $R;R'$ and the converse of a relation is denoted by $R^o$.

**Definition 2.2.7.** Given a binary relation $R \subseteq X \times Y$ with projections $X \xleftarrow{p_1} R \xrightarrow{p_2} Y$, the *relation lifting* $\overline{\mathcal{T}}(R) \subseteq \mathcal{T}(X) \times \mathcal{T}(Y)$ of $R$ is the set

$$\overline{\mathcal{T}}(R) = \Big\{(t, t') \in \mathcal{T}(X) \times \mathcal{T}(Y) \mid (\exists r \in \mathcal{T}(R))(T(p_1)(r) = t \text{ and } \mathcal{T}(p_2)(r) = t'\Big\}.$$

There are at least two other perspectives on relation lifting:

1. If we identify functions with their graphs, we can show $\overline{\mathcal{T}}(R) = (\mathcal{T}(p_1))^o;\mathcal{T}(p_2)$.
2. $\overline{\mathcal{T}}(R)$ is the image of $\mathcal{T}(R) \xrightarrow{\langle \mathcal{T}(p_1), \mathcal{T}(p_2)\rangle} \mathcal{T}(X) \times \mathcal{T}(Y)$.

Here are some concrete examples of relation lifting.

**Example 2.2.8.** Let $R$ be binary relation between $X$ and $Y$.

1. In the case of $\mathcal{T} = \mathsf{Id}$, for every relation $R$ we have $\overline{\mathcal{T}}(R) = R$.
2. For $\mathcal{T} = \mathsf{Pow}$, the lifting of a relation $R \subseteq X \times Y$ is the set

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

   Compare this with the description of basic modal logic using $\nabla$ (Section 1.2.1). Relation lifting is closely related to bisimulation. A binary relation $B$ between Kripke frames $(X, R_0)$ and $(Y, R_1)$ is a bisimulation iff for all $(x, y) \in B$ we have $(R_0[x], R_1[y]) \in \overline{\mathsf{Pow}}(B)$; here $R[x]$ denoted the set of $R$-successors of $x$.

3. Using the distribution functor the lifting of a relation $R \subseteq X \times Y$ can be described as follows: Recall that a distribution $\mu : X \longrightarrow [0, 1]$ can be seen as a finite list $\{(x_i, p_i) \mid i \in n\}$ (Example 2.1.8); the idea is that $\mu(x_i) = p_i$. Using this perspective, we see that $\{(x_i, p_i) \mid i \in n\}\overline{D}(R)\{(y_j, q_j) \mid j \in m\}$ holds iff there exists $(r_{ij})_{1\leq i\leq n, 1\leq j\leq m}$, $r_{ij} \in [0, 1]$ such that if $\neg(x_i R y_j)$ then $(r_{ij} = 0)$ and $\sum_i r_{ij} = q_j$ and $\sum_j r_{ij} = p_i$. As said in the previous item, relation lifting is related to bisimulation; in [31] a presentation like the one above is used to describe bisimulation of probabilistic systems.

In the case of Kripke polynomial functors, relation lifting can be described inductively as follows:

**Proposition 2.2.9** ([53]). *Let $R \subseteq X \times Y$ be a binary relation. The following induction presents the relation lifting $\overline{\mathcal{T}}(R) \subseteq \mathcal{T}(X) \times \mathcal{T}(Y)$, for each kripke polynomial functor.*

- $\overline{\mathsf{Id}}(R) = R$,
- $\overline{\mathsf{K}_C}(R) = \Delta_C$,
- $\overline{\mathcal{T}_1 \times \mathcal{T}_2}(R) = \big\{\big((t_1, t_2), (t_1', t_2')\big) \mid (t_1, t_1') \in \overline{\mathcal{T}_1}(R)$ *and* $(t_2, t_2') \in \overline{\mathcal{T}_2}(R)\big\}$,
- $\overline{\mathcal{T}_1 + \mathcal{T}_2}(R) = \big\{(\kappa_1(t), \kappa_1(t') \mid (t, t') \in \overline{\mathcal{T}_1}(R)\big\} \cup \big\{(\kappa_2(t), \kappa_2(t') \mid (t, t') \in \overline{\mathcal{T}_2}(R)\big\}$,
- $\overline{\mathsf{Pow}\mathcal{T}}(R) = \big\{(\Phi, \Psi) \mid (\forall \varphi \in \Phi)(\exists \psi \in \Psi)((\varphi, \psi) \in \overline{\mathcal{T}}(R))$ *and* $(\forall \psi \in \Psi)(\exists \varphi \in \Phi)((\varphi, \psi) \in \overline{\mathcal{T}}(R))\big\}$.

### Relation lifting and functors

The process of relation lifting described in Definition 2.2.7 determines a function $\overline{\mathcal{T}}$ mapping relations to relations. In case $\mathcal{T}$ preserves weak pullbacks, $\overline{\mathcal{T}}$ is a functor

$$\overline{\mathcal{T}} : \mathsf{Rel} \longrightarrow \mathsf{Rel},$$

where Rel is the category with sets as objects and relations as arrows.

More explicitly, $\overline{\mathcal{T}}$ maps a set $X$ to $\mathcal{T}(X)$ and a relation $R : X \longrightarrow Y$ to $\overline{\mathcal{T}}(R) : \mathcal{T}(X) \longrightarrow \mathcal{T}(R)$.

The following result will be used very often

**Proposition 2.2.10.** *For a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ the following are equivalent:*

1. *$\mathcal{T}$ preserves weak pullbacks (Definition A.0.1, page 215)*
2. *$\overline{\mathcal{T}} : \mathsf{Rel} \longrightarrow \mathsf{Rel}$ is a functor*
3. *the $\mathcal{T}$ relation lifting preserves the composition of relations i.e. $\overline{\mathcal{T}}(R \circ S) = \overline{\mathcal{T}}(R) \circ \overline{\mathcal{T}}(S)$.*

A proof of this fact appears in [13] although it is not explicitly stated there.

### Some structural properties of coalgebras

Two constructions of coalgebras are important for us. Those are disjoint unions and quotients. It is well known that to construct the disjoint union of two Kripke frames we first take the disjunction of the carriers and then "extend" the relational structure. This procedure works for any type of coalgebras. The same principle applies to the formation of quotients of coalgebras i.e. we first form the quotient of the carrier and then extend the "relational structure", see Chapter 6 for more on quotients of coalgebras. In fact, every (finite) colimit can be constructed using disjoint unions and and quotients. The property to remember is that colimits of coalgebras are build by first performing the respective operation on the carrier and then extending the coalgebraic structure using the universal property. In the categorical language this means that the forgetful functor creates colimits.

**Proposition 2.2.11.** *The forgetful functor $U : \mathsf{Coalg}(\mathcal{T}) \longrightarrow \mathbb{C}$, where $\mathbb{C}$ is the base category, creates colimits.*

Products, or limits in general, of coalgebras are wild creatures. In general, the product of two coalgebras might even fail to exists [?]. Another example of bad behaved limits of coalgebras are final coalgebras. As mentioned, those do not exist in general. Chapter 6 is devoted to the construction of final coalgebras. During most of this manuscript limits of coalgebras will not torment us.

## 2.3 Algebras, algebraic signatures, and functors

We recall some familiar definitions from Universal algebra.

**Definition 2.3.1.** A *(finitary) algebraic signature* is a set of symbols $\Sigma$ together with a function $ar : \Sigma \longrightarrow \mathbb{N}$. A symbol $p \in \Sigma$ is called an operation; we refer to $ar(p)$ as the arity of the operation. The subset of $n$-ary operations of $\Sigma$ is denoted by $\Sigma_n$.

An algebra of type $\Sigma$ is a set $A$ together with a function $p^A : A^{ar(p)} \longrightarrow A$ for each $p \in \Sigma$. We denote a $\Sigma$- algebra as a tuple $(A, p^A)_{p\in\Sigma}$.

A homomorphism between $\Sigma$-algebras $(A, p^A)_{p\in\Sigma}$ and $(B, p^B)_{p\in\Sigma}$ is a function $f : A \longrightarrow B$ preserving the given operations, i.e. for each $n$-ary operation $p$, $fp^A(a_1, \dots, a_n) = p^B(f(a_1), \dots, f(a_n))$.

### 2.3.1 Algebraic signatures as functors

Algebras for an algebraic signature correspond to algebras for a polynomial functor and viceversa. This fact will be of vital importance during all this manuscript.

Given an algebraic signature $\Sigma$ we call the functor

$$\mathcal{T}_\Sigma = \coprod_{n<\omega} \Sigma_n \times (-)^n \tag{2.1}$$

**the associated functor** of the signature. Given a polynomial functor as above, the **associated signature** is given by the set $\Sigma = \coprod_{n<\omega} \Sigma_n$.

$\mathcal{T}_\Sigma$-algebras and $\mathcal{T}_\Sigma$-morphisms coincide with $\Sigma$-algebras and homomorphism of $\Sigma$-algebras.

Indeed, given a $\mathcal{T}_\Sigma$-algebra $\alpha : \mathcal{T}_\Sigma(A) \longrightarrow A$ for each operation $p \in \Sigma_n$ induces a function $\alpha(p, -) : A^n \longrightarrow A$, we regard this function as $p^A$; in other words each $\mathcal{T}_\Sigma$-algebra is a $\Sigma$-algebra. By the universal property of coproducts, $\alpha$ is univocally determined by these functions, this means that each $\Sigma$-algebra is a $\mathcal{T}_\Sigma$-algebra.

In order to see the correspondence between homomorphisms of $\Sigma$-algebras and $\mathcal{T}_\Sigma$-morphisms, notice that by the universal property of coproducts, a $\mathcal{T}_\Sigma$-morphism between algebras $(A, \alpha)$ and $(B, \beta)$ is a function $f : A \longrightarrow B$ such that for each $n$ and each $p \in \Sigma_n$ the following diagram

$$\begin{array}{ccc} A^n & \xrightarrow{\quad f^n \quad} & B^n \\ {\scriptstyle \alpha(p,-)}\downarrow & & \downarrow{\scriptstyle \beta(p,-)} \\ A & \xrightarrow[\quad f \quad]{} & B \end{array}$$

commutes. Recall that $f^n(a_1, \ldots, a_n) = (f(a_1) \ldots f(a_n))$. Therefore, the diagram above literately codes the defining property of $\Sigma$-homomorphisms.

**Notation.** Given the perfect correspondence between algebras for an algebraic signature and algebras for a polynomial functor, we will denote the associated functor, Equation (2.1), of an algebraic signature $\Sigma$ also by $\Sigma$.

The next remark provides another description of the associated functor of a signature.

**Remark 2.3.2.** The associated functor of a signature $\Sigma$ can also be described as $\coprod_{p \in \Sigma} (-)^{ar(p)}$.

It is important to notice that algebras for a functor do not cover the whole landscape of universal algebra. In order to give account for varieties using functors we need monads and algebras for a monad (see Section 2.3.3). Algebras for a functor correspond to varieties that can be axiomatized by axioms of rank 1, see Section 5.1, on page 95, for more details.

### Some structural properties of algebras

The situation for algebras is dual to that of coalgebras. It is well known that the product of two algebras $(A, \alpha)$ and $(B, \beta)$, for an algebraic signature, is obtained/defined by taking the cartesian product of the carriers and then extending the operations componentwise. This procedure also works for algebras for a functor in general. In full generality, we can show that (finite) limits, i.e. products and congruences (subalgebras), are obtained by taking the respective operations on the carrier and then extending them using universal properties. In the categorical language this means that the forgetful functor creates limits.

**Proposition 2.3.3.** *The forgetful functor $U : \mathsf{Alg}(\mathcal{T}) \to \mathbb{C}$, where $\mathbb{C}$ is the base category, creates limits.*

Quotients, homomorphic images, and congruences of algebras are very well understood in universal algebra. We refer the reader to a standard text like [24] for details or to a text like [8] for a categorical perspective.

### Term Algebras

Term algebras will play a crucial role in our development of coalgebraic modal logic.

In Universal Algebra the terms of a signature $\Sigma$ are defined as the smallest subset closed under the operations in $\Sigma$. The next definition makes this precise.

**Definition 2.3.4.** Let $\Sigma$ be a (finitary) algebraic signature; let $\Sigma_n$ be the set of operations of arity $n$ in $\Sigma$. Let $X$ be a set; call the elements of $X$ *propositional variables.* The set $\mathsf{T}_\Sigma(X)$ of *terms of type* $\Sigma$*, over* $X$*,* is the smallest set such that:

1. $X \cup \Sigma_0 \subseteq \mathsf{T}_\Sigma(X)$,
2. if $t_1, \ldots t_n \in \mathsf{T}_\Sigma(X)$ and $p \in \Sigma_n$ then $p(t_1, \ldots, t_n) \in \mathsf{T}_\Sigma(X)$.

The *term algebra of type* $\Sigma$, over $X$, written $\mathbb{T}_\Sigma(X)$, has as carrier the set $\mathsf{T}_\Sigma(X)$; an operation $p \in \Sigma_n$ is interpreted as the function $p^{\mathbb{T}_\Sigma(X)} : \mathsf{T}_\Sigma(X)^n \longrightarrow \mathsf{T}_\Sigma(X)$ defined by the clauses above. More explicitly, a tuple $t_1, \ldots t_n \in \mathsf{T}_\Sigma(X)$ is mapped to $p^{\mathbb{T}_\Sigma(X)}(t_1, \ldots t_n) = p(t_1, \ldots t_n)$. The term algebra over $\Sigma$ is also referred as the *absolutely free* $\Sigma$*-algebra.*

In the language of Category Theory, term algebras define a left adjoint to the forgetful functor.

**Proposition 2.3.5.** *Let* $\Sigma$ *be a polynomial functor. The forgetful functor* $U_\Sigma : \mathsf{Alg}(\Sigma) \longrightarrow \mathsf{Set}$ *has a left adjoint* $\mathbb{T}_\Sigma : \mathsf{Set} \longrightarrow \mathsf{Alg}(\Sigma)$*. The functor* $\mathbb{T}_\Sigma$ *maps a set* $X$ *to the* $\Sigma$*-term algebra over* $X$*.*

### 2.3.2 Varieties

In this section we discuss some properties of varieties, of algebras, which will play an important role trough the dissertation. More concretely, these properties concern the finite presentability of variates of algebras for a finitary signature. The standard text in the subject is [7].

**Definition 2.3.6.** Let $\Sigma$ be an algebraic signature. A class $\mathcal{A} \subseteq \mathsf{Alg}(\Sigma)$ is said to be a variety if it is closed under products, homomorphic images, and subalgebras. The signature $\Sigma$ is called the *algebraic signature of* $\mathcal{A}$ and is often written as $\Sigma_\mathcal{A}$.

**Remark 2.3.7.** By definition all our varieties are finitary, this means that they are varieties over a finitary algebraic signature. Often we will add some redundancy referring to *(finitary) varieties.* This is to stress that things could go wrong if the operations of the signature are not of finite arity.

A well known theorem of Birkhoff shows that the varieties of algebras are precisely the equationally definable classes of algebras, see [24] for details and [8] for a categorical version. Example of varieties are boolean algebras and distributive lattices. We denote these categories with the usual morphism by $\mathsf{BA}$ and $\mathsf{DL}$, respectively.

Varieties for an algebraic signature can in fact be described by operations of finite arity and equations. Every variety $\mathcal{A}$ comes equipped with a forgetful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$, which has a left-adjoint $F : \mathsf{Set} \longrightarrow \mathcal{A}$.

In a finitary variety, every algebra $A \in \mathcal{A}$ is a colimit of finitely generated free algebras in a canonical way [8], more explicitly it is the colimit of all valuations $n \longrightarrow U(A)$. The next definition and proposition make this precise.

**Definition 2.3.8.** Let $\mathcal{A}$ be a variety of algebras and let $\mathcal{A}_0$ be the full subcategory of finitely generated free algebras. For each algebra $A \in \mathcal{A}$, the *canonical diagram of* $A$, denoted by $D_0^A$, with respect to $\mathcal{A}_0$, is the natural forgetful functor $D_0^A : \mathcal{A}_0 \downarrow A \longrightarrow \mathcal{A}$, where $\mathcal{A}_0 \downarrow A$ denotes the comma category over $A$.

More explicitly, the objects in the comma category $\mathcal{A}_0 \downarrow A$ are all homomorphism $i : F(n_i) \longrightarrow A$, i.e valuations $n_i \longrightarrow U(A)$, where $n_i$ is a finite set. The morphisms are substitution of variables in a valuation; more explicitly, given valuations $i : n_i \longrightarrow U(A)$ and $j : m_j \longrightarrow U(A)$ a morphism from $i$ to $j$ is a function $f : n_i \longrightarrow m_j$ such that $i = j \circ f$. Hence the canonical diagram of $A$ is given by the family $\big(f_i^j : F(n_i) \longrightarrow F(m_j)\big)$ where $i : n_i \longrightarrow U(A)$ and $j : m_j \longrightarrow U(A)$ range over all valuations; and $f_i^j : n_i \longrightarrow m_j$ is a function such that $i = f_i^j \circ j$.

In case $\mathcal{A} = \mathsf{Set}$ the canonical diagram of a set $A$ correspond to all finite subsets of $A$. Morphism are then given by permutations of the elements of those. With this example in mind, the canonical diagram will generalise the following fact to algebras in a variety. Every set is the join of its finite subsets. The next proposition makes this precise and present other properties of the canonical diagram that will be relevant for us.

**Proposition 2.3.9.** *For any (finitary) variety $\mathcal{A}$ the following holds.*

1. *Every algebra in $\mathcal{A}$ is the colimit of its canonical diagram, i.e. for every $A \in \mathcal{A}$ we have $A \cong \mathsf{colim}(D_0^A)$.*
2. *The canonical colimit is directed.*
3. *The forgetful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$ preserves the canonical colimit, thus it can be computed as in $\mathsf{Set}$. In general $U$ preserves all directed colimits.*
4. *Moreover, the canonical colimit commutes with finite products. More precisely, this means that given a finite number of algebras $\{A_i \mid i \in m\}$ in $\mathcal{A}$, we have*
$$\mathsf{colim}(D_0^{\prod_{i\in m} A_i}) \cong \prod_{i\in m} A_i \cong \prod_{i\in m} \mathsf{colim}(D_0^{A_i}).$$

The details can be found in [8].

**Remark 2.3.10.** In case the variety does not correspond to a finitary algebraic signature it is not enough to consider finitely generated algebras. We refer the reader to [8] for a detail account of those situations.

The next remark addresses some divergence of our terminology with the terminology used in the literature.

**Remark 2.3.11.** The property in Item 4 in the previous proposition tells us that the the canonical colimit is in fact a so-called *sifted colimit*. In this dissertation the only relevant sifted colimit is the canonical colimit.

More formally a category $\mathbb{D}$ is sifted if finite products in $\mathsf{Set}$ commute with colimits over $\mathbb{D}$. A sifted colimit is the colimit of a diagram whose domain is a sifted category. More explicitly, a category $\mathbb{D}$ is sifted iff, for every diagram

$$D : \mathbb{D} \times \mathbb{I} \longrightarrow \mathsf{Set}$$

where $\mathbb{I}$ is a finite discrete category we have

$$\mathsf{colim}\left(\prod_{i \in \mathbb{I}} D(d,i)\right) \cong \prod_{i \in \mathbb{I}}(\mathsf{colim}(D(d,i)))$$

as we said this property will only matter for the canonical colimit.

One of the important properties of finitary varieties is that to define a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$, it is enough to describe $L$ on $\mathcal{A}_0$ and then extend it to general $A \in \mathcal{A}$ via colimits. As mentioned before, this colimit is preserved by $U$ and thus it is calculated as in $\mathsf{Set}$. The next definition makes this precise.

**Definition 2.3.12.** We say that a functor $L$ on a variety $\mathcal{A}$ is determined by finitely generated free algebras if for every algebra $A \in \mathcal{A}$ we have $L(A) \cong \mathsf{colim}(L \circ D_0^A)$.

In more detail, a functor $L$ is determined on finitely generated algebras if for every algebra $A$, the algebra $L(A)$ is the colimit of the application of $L$ to the canonical diagram of $A$. More explicitly, if $A$ is the colimit of $\left(f_i^j : F(n_i) \longrightarrow F(m_j)\right)$ then $L(A)$ is the colimit of $\left(LF(n_i) \xrightarrow{L(f_i^j)} LF(m_j)\right)$.

The next remark addresses the general definition of functor generated by finitely generated algebras.

**Remark 2.3.13.** A functor is determined by finitely generated free algebras iff it preserves, so-called, sifted colimits [7]. It was proved in [76] that a functor preserves sifted colimits iff it can be described by operations and equations [22].

We will see examples of such presentations in Chapter 5

In particular the forgetful functor of any finitary variety is determined by finitely generated algebras [76].

We can also describe natural transformations by describing them on finitely generated algebras.

**Proposition 2.3.14.** *Let $L$ and $L'$ be functors determined by finitely generated algebras. Let $\{(\tau_\omega)_n : LF(n) \longrightarrow L'F(n)\}_{n\in\omega}$ be a family natural on $n$, i.e. a natural transformation on finitely generated algebras.*

*There exists a (unique) natural transformation $\tau : L \longrightarrow L'$ extending $\tau_\omega$, i.e. $\tau$ and $\tau_\omega$ coincide on finitely generated algebras.*

*Proof.* This is immediate by the universal property of colimits. More explicitly, the $A$-component $\tau_A : L(A) \longrightarrow L'(A)$ is the unique arrow such that the following diagram

$$\begin{array}{ccc} L(A) & \overset{\tau_A}{\dashrightarrow} & L'(A) \\ {\scriptstyle L(i)}\uparrow & & \uparrow{\scriptstyle L'(i)} \\ LF(n_i) & \underset{(\tau_\omega)_n}{\longrightarrow} & L'F(n_i) \end{array}$$

commutes for every $i : F(n_i) \longrightarrow A$. □

### 2.3.3 Monads and Algebras

As we mentioned, to describe varieties using functors we need monads.

**Definition 2.3.15.** A monad on a category $\mathbb{C}$ is a triple $\mathbb{M} = (M, \eta, \mu)$ where $M$ is a functor on $\mathbb{C}$, $\eta$ is a natural transformation $\eta : \mathsf{Id} \longrightarrow M$ called the *unit*, and $\mu$ is a natural transformation $\mu : M^2 \longrightarrow M$ called the *multiplication*; such that the following diagrams

$$\begin{array}{ccccc} M & \xrightarrow{M(\eta)} & M^2 & \xleftarrow{\eta_M} & M \\ & {\scriptstyle id}\searrow & \downarrow{\scriptstyle \mu} & \swarrow{\scriptstyle id} & \\ & & M & & \end{array} \qquad \begin{array}{ccc} M^3 & \xrightarrow{M(\mu)} & M^2 \\ {\scriptstyle \mu_M}\downarrow & & \downarrow{\scriptstyle \mu} \\ M^2 & \xrightarrow[\mu]{} & M \end{array}$$

commute.

**Example 2.3.16.** The following are examples of monads.

1. The functor $1 + \mathsf{Id}$ is a monad. The unit $\eta_X : X \longrightarrow 1 + X$ is given by inclusion. The multiplication $\mu_X : 1 + 1 + X \longrightarrow 1 + X$ maps $x \in 1 + 1 + X$ to $*$, the only element of 1, if $x \in 1 + 1$ and to itself if $x \in X$.

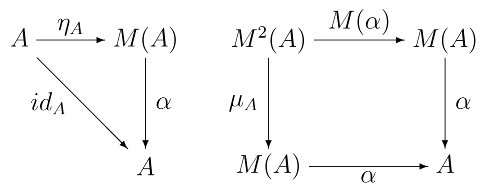
2. The covariant power set functor $\mathsf{Pow}$ is a monad with unit $\eta_X(x) = \{x\}$ and multiplication $\mu_X(\{U_i \mid i \in I\}) = \bigcup_{i\in I} U_i$.

3. The distribution functor $\mathcal{D}$ is a monad. The unit $\eta_X : X \longrightarrow \mathcal{D}(X)$ maps $x$ to the probability Dirac distribution $d_x : X \longrightarrow [0, 1]$, i.e. $d_x(x) = 1$ and $d_x(y) = 0$ in any other case. The multiplication $\mu : \mathcal{D}\mathcal{D}(X) \longrightarrow \mathcal{D}(X)$ maps $D \in \mathcal{D}\mathcal{D}(X)$ to the probability distribution $\mu(D) : X \longrightarrow [0, 1]$ which maps $x$ to $\sum_{d\in\mathcal{D}(X)}(D(d) \cdot d(x))$.

4. The neighbourhood functor $\mathcal{P}\mathcal{P}^{op}$ is a monad with unit $\eta_X(x) = \{U \in \mathcal{P}X \mid x \in U\}$ and multiplication $\mu$ defined for all $W \in (\mathcal{P}\mathcal{P}^{op})(\mathcal{P}\mathcal{P}^{op})(X)$ by
$$\mu_X(W) = \{U \in \mathcal{P}(X) \mid \{H \in \mathcal{P}\mathcal{P}^{op}(X) \mid U \in H\} \in W\}.$$
5. The functor $\mathsf{Mon}$ is also a monad. The unit $\eta$ and multiplication $\mu$ are obtained by restricting the ones for $\mathcal{P}\mathcal{P}^{op}$. In particular, for $W \in \mathsf{Mon}(\mathsf{Mon}(X))$, the multiplication is defined by $\mu_X(W) = \{U \in \mathcal{P}(X) \mid \{H \in \mathsf{Mon}(X) \mid U \in H\} \in W\}$.
6. Note that there is no natural way to define a monad structure on the functors $\mathsf{Pow}(-)^A$, for an arbitrary set $A$, neither for the functor $(B \times \mathsf{Id})^A$ where $A$ and $B$ are arbitrary sets, unless $A = B$.

We can also define algebras for a monad.

**Definition 2.3.17** (Eilenberg-Moore algebras)**.** Let $\mathbb{M} = (M, \eta, \mu)$ be a monad on a category $\mathbb{C}$. An *Eilenberg-Moore algebra* for $\mathbb{M}$, or $\mathbb{M}$-algebra for short, is an arrow $\alpha : M(A) \longrightarrow A$ such that the following diagrams




commute. Morphisms of $\mathbb{M}$-algebras are defined as morphisms of algebras for the functor $M$. The respective category is denoted by $\mathsf{Alg}(\mathbb{M})$, or $(\mathbb{C}^{\mathbb{M}}, U^{\mathbb{M}})$; where $U^{\mathbb{M}}$ is the (natural) forgetful functor. Clearly this category is concrete over $\mathbb{C}$.

An important property of monads is that every monad comes from an adjunction and every adjunction defines a monad.

**Proposition 2.3.18.** *In any category $\mathbb{C}$ we have:*

1. *Every adjunction $(F, U, \varphi, \eta, \varepsilon)$, where $U : \mathcal{A} \longrightarrow \mathbb{C}$, induces a monad $(M, \eta, \mu) = (UF, \eta, U\varepsilon_F)$.*
2. *For every monad $\mathbb{M} = (M, \eta, \mu)$, on $\mathbb{C}$, there exists a category $\mathcal{A}$, a functor $U : \mathcal{A} \longrightarrow \mathbb{C}$, and an adjunction $(F, U, \varphi, \eta, \varepsilon)$, such that $(M, \eta, \mu) = (UF, \eta, U\varepsilon_F)$.*

The first item is a straight forward computation. For the second item we can take $\mathcal{A}$ to be the category of Eilenberg-Moore algebras for $\mathbb{M}$ and $U$ the natural forgetful functor. However this is not the only way. In Section 7.2 we discus another manner of obtaining the mentioned adjunction.

Putting the previous proposition together with proposition 2.3.5 we can see that for every polynomial functor $\Sigma : \mathsf{Set} \longrightarrow \mathsf{Set}$ the term algebras define a monad $\mathsf{T}_\Sigma : \mathsf{Set} \longrightarrow \mathsf{Set}$, this is called the *free monad* of $\Sigma$. A important property is that the categories $\mathsf{Alg}(\Sigma)$ and $\mathsf{Alg}(\mathsf{T}_\Sigma)$ are equivalent. In general, given a functor $\mathcal{T} : \mathbb{C} \longrightarrow \mathbb{C}$ such that the forgetful functor $U : \mathsf{Alg}(\mathcal{T}) \longrightarrow \mathbb{C}$ has a left adjoint, $F$, the monad $\mathbb{C} \xrightarrow{\mathbb{T}=UF} \mathbb{C}$ is called the *free monad* of $\mathcal{T}$. In case the free monad exists, the categories $\mathsf{Alg}(\mathcal{T})$ and $\mathsf{Alg}(\mathbb{T})$ are equivalent.

Using monads we can characterise categories of algebras abstractly as follows:

**Definition 2.3.19.** A category $\mathcal{A}$ is said to be *monadic*, or *algebraic*, over a category $\mathbb{C}$, if (1) it is concrete over $\mathbb{C}$, i.e there is a faithful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$, and $\mathcal{A}$ is concretely isomorphic to a category $(\mathbb{C}^{\mathbb{M}}, U^{\mathbb{M}})$ for some monad on $\mathbb{C}$. In case $(\mathcal{A}, U)$ is monadic, we say that the functor $U$ is monadic.

The gain of this is that now we have a categorical presentation of categories of algebras and varieties.

**Example 2.3.20.** The following categories are monadic.

1. Let $\Sigma$ be a (finitary) algebraic signature. Every variety $\mathcal{A} \subseteq \mathsf{Alg}(\Sigma)$ is monadic. It can be shown that the category $\mathcal{A}$ is isomorphic to the category of Eilenberg-Moore algebras for the monad $UF$, where $U : \mathcal{A} \longrightarrow \mathsf{Set}$ is the forgetful and $F$ is its lefts adjoint.

2. In fact, any category $\mathsf{Alg}(\mathcal{T})$ for an accessible functor is monadic. In such case the monad $UF$ is called the *free monad* generated by $\mathcal{T}$.

The previous example shows that monadic categories subsume varieties for finitary signatures. However, monads can also account for (some) infinitary algebraic theories.

**Example 2.3.21.** The following categories are monadic:

1. The category of complete atomic Boolean Algebras complete boolean algebra homomorphisms is monadic.

2. The category $\mathsf{Frm}$, of frames and frame homomorphism, is monadic. This is usually proven by factoring the forgetful functor $U : \mathsf{Frm} \longrightarrow \mathsf{Set}$ via some intermediate category as $\mathsf{Pos}$ or $\mathsf{DL}$. A survey of such factorisations can be found in [62].

In Section 5.1 we show which monads correspond to finitary categories of algebras.

## 2.4 A first glance at logics for coalgebras

Modal languages have been widely studied as simple yet expressive languages which provide an internal, local perspective on relational structures, see [20]. As we discussed in the introduction, modal logics have a coalgebraic nature. In this section we present the coalgebraic semantics for various well known modal logics. In Part I, we will show how all these systems can be studied under the single uniform framework of *coalgebraic modal logic*.

In the introduction we discussed the case of basic modal logic (Section 1.2.1). As a remainder we recall that Kripke frames correspond to Pow-coalgebras. Morphism of Pow-coalgebras correspond to bounded morphisms [2].

### 2.4.1 Probabilistic modal logic & coalgebras

Probabilistic modal logic [49] has modalities of the following type: For each real number $p \in [0,1]$ there is a modality $\Diamond_p$ with the following reading *"the probability of $\varphi$ is at least $p$"*. Another common probabilistic modality is $\Diamond^p$ with the reading *"the probability of $\varphi$ is at most $p$"*.

Recall that coalgebras for the finite distribution functor are discrete time Markov chains [15], also known as probabilistic transition systems. Probabilistic modal logic can be interpreted over $\mathcal{D}$-coalgebras as follows. The semantics of $\Diamond_p$ and $\Diamond^p$ on a coalgebra $\xi : X \longrightarrow \mathcal{D}(X)$ is given by:

$$[\![\Diamond_p \varphi]\!]_\xi = \left\{ x \in X \mid p \leq \sum_{y \in [\![\varphi]\!]_\xi} \xi(x)(y) \right\}, \text{ and}$$

$$[\![\Diamond^p \varphi]\!]_\xi = \left\{ x \in X \mid \sum_{y \in [\![\varphi]\!]_\xi} \xi(x)(y) \leq p \right\}.$$

#### Graded modal logic & coalgebras

Another example of modal logics for coalgebras is *graded modal logic* [37]. In this logic, for each natural number $n$ there is a modality $\Diamond_n$ with the following reading: *"there are at least $n$ successors satisfying $\varphi$"*.

Using coalgebras we can interpret graded modal logic can be interpreted over directed graphs with $\mathbb{N}$-weighted edges, often referred as multigraphs or graded Kripke frames; these can be see as coalgebras for the finite multiset functor [110].

The idea follows the same spirit used in the example of distributions. The semantics of $\Diamond_n$ on a coalgebra $\xi : X \longrightarrow \mathcal{B}_{\mathbb{N}}(X)$ is given by

$$[\![\Diamond_n \varphi]\!]_\xi = \left\{ x \in X \mid n \leq \sum_{y \in [\![\varphi]\!]_\xi} \xi(x)(y) \right\}.$$

Since $\xi(x) : X \longrightarrow \mathbb{N}$ assigns a weight to each element of $X$. This shows that the equation above gives the usual interpretation of graded modalities .

### 2.4.2 Propositional dynamic logic & coalgebras

Slight variations of this example allow us to add labels to transitions of states and the modalities. As we saw, coalgebras for $\mathsf{Pow}^L$ are labelled transition systems. We elaborate on this.

Notice that a coalgebra $\xi : X \longrightarrow \mathsf{Pow}(X)^L$ curries into a function

$$\widehat{\xi} : X \times L \longrightarrow \mathsf{Pow}(X).$$

hence for each $a \in L$ we have a function $\xi_a : X \longrightarrow \mathsf{Pow}(X)$ which, by the discussion on Kripke frames, corresponds to a binary relation $R_a \subseteq X \times X$. In other words, a $\mathsf{Pow}^L$ coalgebra can be presented as a a relational structure $(X, \{R_a\}_{a \in L})$ where $xR_ay$ iff $y \in \xi(x)(a)$. We can then describe the transitions of the systems as follows: if $y \in \xi(x)(a)$, we write $x \stackrel{a}{\to} y$ and read *"there is a transition a from x to y"*.

Propositional Dynamic Logic (PDL) [39, 92], see [47] for a more detailed account, started as a modal logic for reasoning about program correctness. Modalities are indexed by programs; a formula $[a]\varphi$ should be read as *"after all halting executions of a, $\varphi$ holds"*. PDL programs are built inductively using the operations of *choice* ($\cup$), *sequential composition* (;) and *iteration* ($^*$). Moreover, a formula $\varphi$ can be turned into a program $\varphi?$ by the *test* operation ?. Complex programs are interpreted by operations on relations, i.e $\mathsf{Pow}$-coalgebras as follows: sequential composition is interpreted as relation composition, choice as union, iteration as reflexive, transitive closure, and a test $\varphi?$ as the relation $\{(x, x) \mid x \in [\![\varphi]\!]\}$.

Using the relational presentation above, we can now see that the interpretation of $[a]$ on a coalgebra $\xi : X \longrightarrow \mathsf{Pow}(X)^L$ is as follows:

$$[\![ [a]\varphi ]\!]_\xi = \left\{ x \in X \mid \xi(x)(a) \subseteq [\![\varphi]\!]_\xi \right\}.$$

In terms of relational structures this can be written as

$$[\![ [a]\varphi ]\!]_\xi = \left\{ x \in X \mid (\forall y)(xR_ay \Rightarrow y \in [\![\varphi]\!]_\xi) \right\}$$

We will discuss more on PDL and labelled transition systems in Chapter 7.

### 2.4.3 Non-normal modal logic

Non-normal modal logics [26] are modal logics for which the **K** axiom from basic modal logic may fail. Their semantics is usually given by the so called *neighborhood frames.* Those can be seen as coalgebras for the double contravariant power set functor, also known as the neighbourhood functor [45].

More explicitly, a neighborhood frame is a function

$$\xi : X \to \mathcal{P}\mathcal{P}^{op}(X)$$

where $\mathcal{P} : \mathsf{Set}^{op} \to \mathsf{Set}$ is the contravariant power set functor, and $\mathcal{P}^{op} : \mathsf{Set} \to \mathsf{Set}$ is its dual. In other words, the function $\xi$ assigns to each $x \in X$ a family of subsets of $X$. This can be interpreted as a two player game where a move in state $x_1$ consists of the first player choosing a set $S \in \xi(x_1)$ and then the second player chooses a successor-state $x_2 \in S$.

The non normal modality $\Box$ has the following semantics over a coalgebra $\xi : X \to \mathcal{P}\mathcal{P}^{op}(X)$.

$$[\![\Box\varphi]\!]_\xi = \Big\{x \in X \mid [\![\varphi]\!]_\xi \in \xi(x)\Big\}.$$

Logics like coalition logic, also called logics for social software, are labelling of non-normal modal logics, i.e. they are modal logics interpreted over $(\mathcal{P}\mathcal{P}^{op})^L$-coalgebras; in this case, the elements of $L$ are usually interpreted as agents, coalitions, or games; see [91] for a survey.

It is very common to require a bit more of structure and consider the *monotone neighborhood* functor $\mathsf{Mon}$ instead of the neighborhood functor $\mathcal{P}\mathcal{P}^{op}$. Later, for various purposes we will assume non-normal modal logic is interpreted over $\mathsf{Mon}$ coalgebras. The semantics of $\Box$ on $\mathsf{Mon}$-coalgebras is the same as for $\mathcal{P}\mathcal{P}^{op}$.

#### Game Logic

Among non-normal modal logics, we will pay particular attention to *game logic.* Game logic is in some sense the non-normal (monotone) version of PDL, see Chapter 7. Game Logic (GL) [87, 91] is a non-normal modal logic for reasoning about strategic ability in determined 2-player games. Informally, a modal formula $[\gamma]\varphi$ should be read as *"player 1 has a strategy in the game $\gamma$ to ensure an outcome in which $\varphi$ holds".*

Game operations extend the program operations of PDL with the operation *dual* ( $^d$) which corresponds to a role switch of the two players. Formally, let $\Gamma_0$ be a set of atomic games. The games and formulas of Game Logic are defined as follows:

$$\begin{array}{llll} \text{games } \gamma \in \Gamma, & \gamma \;::=\; g \mid \gamma \cup \gamma \mid \gamma;\gamma \mid \gamma^* \mid \gamma^d \mid \varphi? & & \text{where } g \in \Gamma_0, \\ \text{formulas } \varphi \in \Phi, & \varphi \;::=\; \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid [\gamma]\varphi & & \end{array}$$

Game Logic semantics is given by multi-modal monotone neighbourhood models [26]. For simplicity we leave out atomic propositional variables from the language, but these can easily be included and interpreted by valuations as usual. A *GL model* $\mathcal{M} = (X, \{E_a | \, g \in \Gamma_0\})$ consists of a set of states $X$, and a monotone neighbourhood function $E_g \colon X \longrightarrow \mathsf{Mon}(X)$ for each atomic game $g \in \Gamma_0$. Truth of formulas and interpretations of complex games are again defined by simultaneous induction. The clauses for the Boolean operations are as usual, and $[\![[\gamma]\varphi]\!]\mathcal{M} = \{x \in X \mid [\![\varphi]\!]\mathcal{M} \in E_\gamma(s)\}$. Complex game semantics is defined in [87, 91] in terms of the transposed neighbourhood functions.

Recall that the transpose of $\nu : X \longrightarrow \mathcal{P}\mathcal{P}^{op}(X)$ is given by $\widehat{\nu}(U) = \{x \in X \mid U \in \nu(s)\}$ for all $U \subseteq X$. Note that $\widehat{\nu} \colon \mathcal{P}(X) \longrightarrow \mathcal{P}(X)$ is a monotonic map whenever $\nu$ is a monotonic neighbourhood function. For $\gamma, \chi \in \Gamma$ and $U \subseteq X$,

$$\begin{array}{rclrcl} \widehat{E}_{\gamma \cup \chi}(U) & = & \widehat{E}_\gamma(U) \cup \widehat{E}_\chi(U), & \widehat{E}_{\gamma ; \chi}(U) & = & \widehat{E}_\gamma(\widehat{E}_\chi(U)), \\ \widehat{E}_{\gamma^*}(U) & = & \mu Y.U \cup \widehat{E}_\gamma(Y), & \widehat{E}_{\gamma^d}(U) & = & X \setminus (\widehat{E}_\gamma(X \setminus U)). \\ \widehat{E}_{\varphi ?}(U) & = & [\![\varphi]\!]_{\mathcal{M}} \cap U, & & & \end{array} \tag{2.2}$$

### 2.4.4 Translations

We finish this chapter by introducing the concept of translation. This will be important trough Part I of this thesis.

**Definition 2.4.1.** Let $\mathcal{T}$ be set functor and let $\mathcal{L}_1$ and $\mathcal{L}_2$ be language that can be interpreted on $\mathcal{T}$-coalgebras. A *translation* is a function $tr : \mathcal{L}_1 \longrightarrow \mathcal{L}_2$ that preserves the semantics, i.e. for every $\varphi \in \mathcal{L}_1$ and every coalgebra $\xi$, we have $[\![\varphi]\!]_\xi^{\mathcal{L}_1} = [\![tr(\varphi)]\!]_\xi^{\mathcal{L}_2}$.

The domain of *tr* is called the *source* language and the codomain of *tr* is called the *target* language.

# Part I

# Modalities in the Stone age

# Chapter 3

# Coalgebraic Modal Logics

The jungle of modal logics is dense and vast. Nowadays, modalities appear in innumerable shapes and backgrounds; a general framework covering all incarnations of modalities does not strike us in the face. In the quest for such general framework, the first issue rises when we try to formalise the concept of "modality". One thing that seems essential to several modalities is a notion of successor state; as we have mentioned, the notion of successor can be formalised using coalgebras. With this perspective we can then intuitively say:

*A modality transforms properties of states into properties of successors.*

In this chapter, and this thesis in general, we present a general mathematical framework for modalities that fall under the scope of the intuition above. An important insight from this perspective is that it suggests that the concept of modality should be independent of any particular coalgebra and only rely on the functor $\mathcal{T}$, i.e. modalities should only rely on the type of successor. In fact all modalities in Section 1.2 display this format. In Table 3.1 bellow the most right column presents some of these the modalities using a component independent of any coalgebra and only dependent on the functor $\mathcal{T}$.

We will use this observation to define a general framework for modal logics. This will lead us to the concept of concrete modality which is a formalisation of the intuition above. One gain of this is that an algebraic semantics for modalities will rise naturally.

Novel to this chapter is the presentation. In this chapter we try to find a balance between categorical abstraction and concrete presentation. Since coalgebras are more naturally presented in the categorical language, the chapter is more inclined to the categorical perspective.

The structure of the chapter is as follows. Firstly (Section 3.1), we introduce modal similarity types in the tradition of universal algebra using predicate liftings. We develop in detail how from this we can arrive to the functorial framework

| Functor | Modality | Semantics | Decomposition |
|---|---|---|---|
| $\mathsf{Pow}$ | $\Box$ | $\left\{x \in X \mid \xi(x) \subseteq [\![\varphi]\!]_\xi\right\}$ | $\xi^{-1}\left\{U \in \mathsf{Pow}(X) \mid U \subseteq [\![\varphi]\!]_\xi\right\}$ |
| $\mathsf{Pow}$ | $\Diamond$ | $\left\{x \in X \mid \xi(x) \cap [\![\varphi]\!]_\xi \neq \emptyset\right\}$ | $\xi^{-1}\left\{U \in \mathsf{Pow}(X) \mid U \cap [\![\varphi]\!]_\xi \neq \emptyset\right\}$ |
| $\mathcal{B}_{\mathbb{N}}$ | $\Diamond_n$ | $\left\{x \in X \mid n \leq \sum_{y\in[\![\varphi]\!]_\xi} \xi(x)(y)\right\}$ | $\xi^{-1}\left\{\delta \in \mathcal{B}_{\mathbb{N}}(X) \mid n \leq \sum_{y\in[\![\varphi]\!]_\xi} \delta(y)\right\}$ |
| $\mathcal{D}$ | $\Diamond^p$ | $\left\{x \in X \mid \sum_{y\in[\![\varphi]\!]_\xi} \xi(x)(y) \leq p\right\}$ | $\xi^{-1}\left\{\delta \in \mathcal{D}(X) \mid \sum_{y\in[\![\varphi]\!]_\xi} \delta(y) \leq p\right\}$ |
| $\mathsf{Pow}^L$ | $[a]$ | $\left\{x \in X \mid \xi(x)(a) \subseteq [\![\varphi]\!]_\xi\right\}$ | $\xi^{-1}\left\{\delta \in \mathsf{Pow}(X)^L \mid \delta(a) \subseteq [\![\varphi]\!]_\xi\right\}$ |

Table 3.1: Some concrete modalities and their semantics

in Diagram (1.3) as a generalisation of boolean algebras with operators. We deliberately do not use Stone duality to introduce modal logics because we want to stress that the duality is not essential to define modal logics and compare them; the latter is the main topic in Part I of this manuscript. In Section 3.2 we introduce the abstract functorial framework formally. In Section 3.2.1 we show how the usual modal algebras fit into this framework. In Section 3.2.2 we introduce the notion of coalgebraic modal logic based on a category of power sets algebras (Definition 3.2.12). This category might not be that of Boolean algebras. Using this, we present modal logics which are not based in classical logic.

We finish the chapter with Section 3.3 where we introduce two generic modal logics for coalgebras. The first one is the Moss Logic (Section 3.3.1) and the second one is a the logic of all predicate liftings (Section 3.3.2).

## 3.1 Concrete Modalities

As mentioned, for us, a modality transforms properties of states, in a coalgebra, into properties of their successors. This can be restated as follows: given a coalgebra $\xi : X \to \mathcal{T}(X)$ a modality transforms subsets of $X$ into subsets of $\mathcal{T}(X)$; the issue is to describe this without using the coalgebra $\xi$. Pattinson [88] formalised this intuition with the notion of *predicate lifting*, a concept that we now define.

**Definition 3.1.1** ([88])**.** Given a functor $\mathcal{T} : \mathsf{Set} \to \mathsf{Set}$, an $n$-ary predicate lifting, for $\mathcal{T}$, is a natural transformation $\lambda : \mathcal{P}^n \to \mathcal{P}\mathcal{T}$.

Given a predicate lifting $\lambda$, we write $\Box_\lambda$ for the *associated modality*. We refer to modalities given by predicate liftings as *concrete modalities*.

It is important to notice that the previous definition of concrete modality only depends on the type of transition, i.e. the functor $\mathcal{T}$. In other words it really captures the idea that a modality is independent of any particular coalgebra. The use of natural transformations accounts for the intuition that states are transformed into successors always in a uniform maner.

The next remark addresses some technicalities in the previous definition.

**Remark 3.1.2.** For us the contravariant power set functor has domain and codomain as follows $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$. This means that technically speaking in the previous definition a predicate lifting should be defined as a natural transformation $\mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}^{op}$, where $\mathcal{T}^{op} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}^{op}$ is the dual of the functor $\mathcal{T}$. To simplify our notation we avoid the use of the exponent $^{op}$ here.

Another issue in the previous definition is that we implicitly assume predicate liftings to have only finite arity. This gives a smooth correspondence with standard Universal Algebra. However, this condition can be avoided, i.e. we can consider predicate liftings of possibly infinite arity. When relevant, we will mention this explicitly.

Before proceeding to discuss modal signatures and the semantics of predicate liftings we illustrate the concept with examples.

**Example 3.1.3.** In the following examples the semantics on a coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ will be given by post-composing with $\xi^{-1}$.

1. Let $\mathcal{T} = \mathsf{K}_C$ be a constant functor with value $C$. Any subset $P$ of $C$ defines a predicate lifting $\lambda_P : \mathcal{P} \longrightarrow \mathcal{P}\mathsf{K}_C$; it is has constant value $P$.

2. The previous example can be modified to provide propositional information. For this we consider the functor $\mathcal{P}(Q) \times \mathcal{T}$, where $Q$ is a fixed set of propositional letters. The semantics of the proposition letter $q \in Q$ is given by the predicate liftings $\lambda^q_X(\varphi) = \{(U,t) \in \mathcal{P}(Q) \times \mathcal{T}(X) \mid q \in U\}$, and $\lambda^{\neg q}_X(\varphi) = \{(U,t) \in \mathcal{P}(Q) \times \mathcal{T}(X) \mid q \notin U\}$.

3. Let $\mathcal{T}$ be the covariant power set functor. The existential modality $\Diamond$ can be presented using an homonymous predicate lifting $\Diamond : \mathcal{P} \longrightarrow \mathcal{P}\mathsf{Pow}$ which maps a set $\varphi \subseteq X$ to $\Diamond_X(\varphi) = \{\psi \subseteq X \mid \varphi \cap \psi \neq \emptyset\}$. Similarly, the universal modality $\Box$ can be presented as a predicate lifting which transforms a set $\varphi \subseteq X$ into $\Box_X(\varphi) = \{\psi \subseteq X \mid \psi \subseteq \varphi\}$.

4. Consider the neighborhood functor, i.e. $\mathcal{P}\mathcal{P}^{op}$. The standard modalities, used in Non-Normal Modal logic, can be seen as predicate liftings. For example, the universal modality transforms a set $\varphi \subseteq X$ into the set $\Box_X(\varphi) = \{N \in \mathcal{P}\mathcal{P}^{op}(X) \mid \varphi \in N\}$.

5. Consider the multiset functor $\mathcal{B}_{\mathbb{N}}$ and let $k$ be a natural number. The graded modality $\Diamond_k$ can be seen as a predicate lifting for this functor; a set $\varphi \subseteq X$ is mapped to $(\Diamond_k)_X(\varphi) = \{B : X \longrightarrow \mathbb{N} \mid \sum_{x\in\varphi} B(x) \geq k\}$. In this case $x \Vdash_\xi \Diamond_k\varphi$ holds iff $x$ has at least $k$ many successors satisfying $\varphi$.

6. Let $\mathcal{T}$ be the finite distribution functor. The modality $\Diamond_p \varphi$ specifies a probability of at least $p$ for the sum of transitions to a successor satisfying $\varphi$. This can be described by the predicate lifting : $\Diamond_p : \mathcal{P}X \longrightarrow \mathcal{PD}(X)$ mapping $\varphi \subseteq X$ to $(\Diamond_p)_X(\varphi) = \{d \in \mathcal{D}(X) \mid \mu_d(\varphi) \geq p\}$, where $\mu_d(\varphi) = \sum_{x \in \varphi} d(x)$, i.e. $\mu$ is the measure associated with $d$. Similarly, the predicate lifting $\Box_p = \neg \Diamond_p \neg$ maps a set $\varphi \subseteq X$ to the set $(\Box_p)_X(\varphi) = \{d \in \mathcal{D}(X) \mid \mu_d(\varphi) > 1 - p\}$. Another common modality in probability logic is given by $\Diamond^p = \Diamond_{1-p}\neg$; this modality corresponds to the predicate lifting which maps a set $\varphi \subseteq X$ to $(\Diamond^p)_X(\varphi) = \{d \in \mathcal{D}(X) \mid \mu_d(\varphi) \leq p\}$. These modalities give the usual language from [49].

We can now introduce modal signatures.

### Modal Signatures

We identify modal signatures, or modal similarity types [20], with sets of predicate liftings.

**Definition 3.1.4.** Given a set of predicate liftings $\Lambda$, for a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$, the *modal signature*, or *modal similarity type*, associated with $\Lambda$ is given by $\Sigma_\Lambda = \{\Box_\lambda \mid \lambda \in \Lambda\}$, where the arity of $\Box_\lambda$ is that of $\lambda$.

To obtain a usual modal language [20] associated with a set of predicate liftings we add the boolean connectives.

**Definition 3.1.5** ([88]). Let $\Lambda$ be a set of predicate liftings, for a functor $\mathcal{T}$, let $\Sigma_\Lambda$ be the modal signature associated with $\Lambda$, and let $\Sigma_{\mathsf{BA}}$ be the the boolean signature $\{\top, \bot, \neg, \wedge, \vee, \longrightarrow\}$. The (boolean) modal language, written $\mathcal{L}_\Lambda$, associated with $\Lambda$ is defined by the grammar

$$\varphi := \bot \mid \top \mid \neg\varphi \mid \varphi_1 \heartsuit \varphi_2 \mid \Box_\lambda(\varphi_1, \dots \varphi_{ar(\lambda)})$$

where $\heartsuit$ is one of the connectives $\wedge, \vee, \longrightarrow$; $\lambda$ ranges over the elements of $\Lambda$, and $ar(\lambda)$ denotes the arity of $\lambda$.

The semantics of formulas is defined in the usual inductive manner; the modal clause for an $n$-ray predicate lifting $\lambda$ on a coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ is given by

$$[\![\Box_\lambda(\varphi_1, \dots, \varphi_n)]\!]_\xi = \xi^{-1}\lambda_X([\![\varphi_1]\!]_\xi, \dots [\![\varphi_n]\!]_\xi).$$

A first important property of these languages is that formulas are invariant under bisimulation.

**Proposition 3.1.6** ([88]). *Let $\mathcal{L}_\Lambda$ be a (boolean) modal language associated with a set of predicate liftings $\Lambda$. All formulas in $\mathcal{L}_\Lambda$ are invariant under behavioural equivalence (and bisimulation).*

The previous proposition also depicts the limits of the framework. Namely that predicate liftings can only account for modalities which are invariant under behavioural equivalence, this leaves out modalities like the global universal modality or the global existential modality[1].

### Complex algebras

Using the functorial perspective to algebraic signatures, Section 2.3, the (boolean) modal language associated with $\Lambda$ can also be seen as an initial algebra of the functor $\Sigma_\Lambda + \Sigma_{BA} : \mathsf{Set} \longrightarrow \mathsf{Set}$, where $\Sigma_{\mathsf{BA}}$ and $\Sigma_\Lambda$ are the functors associated with the signatures. From here, we will now elaborate towards an algebraic semantics; and eventually, to generalize boolean algebras with operators. This path will lead us to the abstract functorial framework.

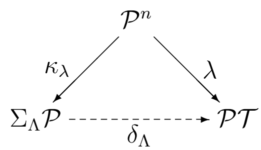
By definition, the semantics of a modality on a coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ is given by composing the associated predicate lifting with the inverse image of $\xi$. Since $\mathcal{P}(\xi) = \xi^{-1}$ this definition corresponds to define an $n$-ary operation over $\mathcal{P}(X)$ via the following composite:

$$\mathcal{P}(X)^n \xrightarrow{\lambda_X} \mathcal{P}\mathcal{T}(X) \xrightarrow{\mathcal{P}(\xi)} \mathcal{P}(X). \tag{3.1}$$

This can be generalized to any set of predicate liftings $\Lambda$, for $\mathcal{T}$. In order to see this, first notice that the functor associated associated with $\Lambda$, can be presented by

$$\Sigma_\Lambda = \coprod_{n<\omega} \Lambda_n \times (-)^n$$

where $\Lambda_n$ is the set of $n$-ary predicate liftings in $\Lambda$. Second notice that, by the universal property of coproducts, we can combine all predicate liftings in $\Lambda$ into a single natural transformation $\delta_\Lambda : \Sigma_\Lambda\mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$. More precisely, $\delta_\Lambda$ is the only natural transformation which makes the following diagram



(3.2)

commute for each predicate lifting $\lambda \in \Lambda$; here $\kappa_\lambda$ denotes the corresponding coproduct inclusion. More concretely, a pair $(\lambda, \varphi) \in \Lambda_n \times \mathcal{P}(X)^n$ is mapped via $\delta_\Lambda$ to $\lambda(\varphi_1, \ldots, \varphi_n)$, i.e. the image of the sequence $\varphi$ under $\lambda$.

Thus, using $\delta_\Lambda$ we can associate a $\Sigma_\Lambda$ algebra with each coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$. Such an algebra is given by the following composite

$$\Sigma_\Lambda\mathcal{P}(X) \xrightarrow{(\delta_\Lambda)_X} \mathcal{P}(X) \xrightarrow{\mathcal{P}(\xi)} \mathcal{P}(X) \tag{3.3}$$

[1]The global universal modality is *"for all states..."* the global existential modality is *"there is a state..."*

The gain here is that we have a general method to define the usual (full) complex algebras from modal logic [20] hence we can use the same idea on arbitrary coalgebras. The next example illustrates that in the case that $\lambda$ is the predicate lifting associated with the existential modality $\Diamond$, for $\mathsf{Pow}$, the composition in Equation (3.3) gives the usual modal algebra.

**Example 3.1.7.** Consider the language with only the existential modality for $\mathsf{Pow}$, i.e. $\Lambda = \{\Diamond\}$. Recall from Example 3.1.3 that this predicate lifting maps a set $\varphi \subseteq X$ to $\{\psi \in \mathsf{Pow}(X) \mid \psi \cap \varphi \neq \emptyset\}$. In this case, we can describe the function on Equation (3.3) on a coalgebra $\xi : X \longrightarrow \mathsf{Pow}(X)$ as follows: A set $\varphi \subseteq X$ is mapped to

$$\begin{aligned}\mathcal{P}(\xi)\delta_X(\varphi) &= \xi^{-1}\Diamond_X(\varphi)\\ &= \xi^{-1}\{\psi \in \mathsf{Pow}(X) \mid \psi \cap \varphi \neq \emptyset\}\\ &= \{x \in X \mid \xi(x) \cap \varphi \neq \emptyset\}\\ &= \Big\{x \in X \mid (\exists y)(xR_\xi y \wedge y \in \varphi)\Big\}\end{aligned}$$

Which is the usual description of the complex algebra associated with the Kripke frame $(X, R_\xi)$, see [20].

When we are dealing with only one predicate lifting, the composition in Equation (3.1), i.e. the complex algebra associated with the predicate lifting, is also referred to as a predicate transformer. We now fix some notation for this.

**Notation.** Given a predicate lifting $\lambda : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ and a coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$, we write $[\![\Box_\lambda -]\!]_{(X,\xi)}$ for the composite $\xi^{-1} \circ \lambda_X : \mathcal{P}(X)^n \longrightarrow \mathcal{P}(X)$, i.e. the structural map of the complex algebra. We call this function the *predicate transformer* associated with $\lambda$.

The idea is that for a sequence $(\varphi_1, \ldots, \varphi_n) \in \mathcal{P}(X)^n$ we have

$$[\![\Box_\lambda -]\!]_\xi(\varphi_1, \ldots \varphi_n) = [\![\Box_\lambda(\varphi_1, \ldots, \varphi_n)]\!]_\xi,$$

In other words, the image of $(\varphi_1, \ldots, \varphi_n)$ under $[\![\Box_\lambda -]\!]_\xi$ is the semantics of the formula $\Box_\lambda(\varphi_1, \ldots, \varphi_n)$.

The natural transformation in Equation (3.2) provides us a mean to define complex algebras, this construction is sometimes denoted by $(-)^+$, e.g. [20]. However, here we denoted the construction by $\widehat{P}_\delta$, one reason for this change is that this is in fact a lifting of the functor $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$. More explicitly, $\widehat{P}_\delta : \mathsf{Coalg}(\mathcal{T})^{op} \longrightarrow \mathsf{Alg}(\Sigma_\Lambda + \Sigma_{\mathsf{BA}})$ maps a coalgebra $(X, \xi)$ to the algebra $(\mathcal{P}(X), \mathcal{P}(\xi) \circ \delta_X)$; and a morphism $f : (X, \xi) \longrightarrow (Y, \gamma)$ to its inverse image. The lifting part means that the following diagram

$$\begin{array}{ccc}\mathsf{Coalg}(\mathcal{T})^{op} & \xrightarrow{\widehat{P}_\delta} & \mathsf{Alg}(\Sigma_\Lambda + \Sigma_{\mathsf{BA}})\\ \downarrow & & \downarrow\\ \mathsf{Set}^{op} & \xrightarrow[\mathcal{P}]{} & \mathsf{Set}\end{array}$$

commutes; in the diagram the vertical arrows are the respective forgetful functors. All this provides an algebraic perspective to modal logic and lead us to our next topic: the abstract functorial framework.

## 3.2 The abstract functorial framework

Until here we have shown that modal signatures, and modal languages in general, can be presented by functors on $\mathsf{Set}$. This is in harmony with the standard Universal Algebra approach, see Section 2.3. However, this approach is not yet using its full potential. To make this more clear consider the algebra

$$\Sigma_\Lambda \mathcal{P}(X) \xrightarrow{(\delta_\Lambda)_X} \mathcal{P}(X) \xrightarrow{\mathcal{P}(\xi)} \mathcal{P}(X)$$

in Equation (3.3). This algebra has more structure than just a function between sets. In particular its carrier set is a boolean algebra. Moreover, the structural map itself might preserve some of this structure. For example, the modalities might satisfy some extra properties like $\Box a \wedge b = \Box a \wedge \Box b$. This is to say that the complex algebra above is a boolean algebra with some extra operations. On top of this, notice that also the language $\mathcal{L}_\Lambda$ is an algebra in the boolean signature with some extra operations, namely the modalities. Recall that intuitively, modal logics are (usually) regarded as extensions of classical propositional logic. All this suggests that in order to take full advantage of this algebraic perspective on modalities, we should rather consider functors on *boolean algebras* instead of just on $\mathsf{Set}$. Indeed this will prove to be a very fruitful idea which we will illustrate through this manuscript. To account for such extra structure also corresponds to generalize boolean algebras with operators to the coalgebraic level of generality.

Another force that drives us to consider logics as functors on $\mathsf{BA}$, instead of on $\mathsf{Set}$, is that BAOs will be a sub-variety of $\mathsf{Alg}(\Sigma_\Lambda + \Sigma_{BA})$. The issue resides in how we present such variety. One option is to add all the axioms at once. The key insight in [68] is not to do this but to add the modalities, and their axioms, over an axiomatization of Boolean algebras. This accounts to present modal logics, or modal signatures, as functors over $\mathsf{BA}$ instead of $\mathsf{Set}$.

Among the advantages of this approach we highlight the following ones:

1. We can use structural properties of the category of algebras to define, compare, and study modal logics.
2. Modularity in the presentation of modal logics.
3. Uniform framework to study different types of logics which allows natural generalisations.

(a) In particular, as we will later see, Section 3.3.1, Moss logic [84] does not comfortably fit into the paradigm of Universal Algebra but it does find its natural place in our framework.

4. We follow the intuitive motto that modal logics are extensions of classical logic.

5. For more advantages concerning translations see page 72.

Without any more ado we now present the definition.

**Definition 3.2.1** ([68])**.** Let $\mathcal{T}$ be a $\mathsf{Set}$ endofunctor and let $P : \mathsf{Set}^{op} \longrightarrow \mathsf{BA}$ be the contravariant powerset functor. A (boolean) modal logic for $\mathcal{T}$-coalgebras, also called a (boolean) *coalgebraic modal logic* for $\mathcal{T}$ coalgebras, is a pair $(L, \delta)$ where $L$ is a functor $L : \mathsf{BA} \longrightarrow \mathsf{BA}$ determined by finitely generated free algebras (Definition 2.3.12) and $\delta$ is a natural transformation

$$\delta : LP \longrightarrow P\mathcal{T}. \tag{3.4}$$

The natural transformation $\delta$ is called the *semantics* of the logic. The *language of the logic* is given by the initial $L$-algebra (see Remark 3.2.6).

The next remark concerns the nature of formulas in this functorial framework.

**Remark 3.2.2.** Notice that given a coalgebraic logic $(L, \delta)$, the elements of the initial $L$-algebra are not formulas in the standard sense of terms in an algebraic signature. Instead, they are equivalence classes of formulas. The initial $L$-algebra corresponds to the Lindenbaum-Tarski algebra of the modal logic and not the absolutely free algebra of the signature. Still we call the elements of the initial $L$-algebra the *formulas* of the logic.

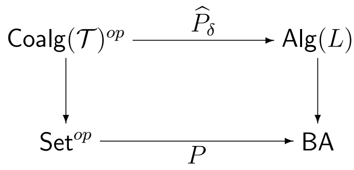
In fact, to extract an algebraic signature and a standard language (Definition 3.1.5) from a coalgebraic modal logic is not a trivial issue. This was first addressed in [22, 76]. We address the issue in Chapter 5 (Section 5.2) where we show that every coalgebraic modal is a modal logic where the modalities are given by predicate liftings (Theorem 5.2.17).

We now explain the terminology used for $\delta$. As it was done with modal signatures, see Equation (3.3), using $\delta$ we can assign to each coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ an $L$-algebra with carrier $P(X)$ and structural map given by the following composite

$$LP(X) \xrightarrow{\delta_X} P\mathcal{T}(X) \xrightarrow{P(\xi)} \mathcal{P}(X). \tag{3.5}$$

As in the case of modal signatures this induces a functor from the category of $\mathcal{T}$-coalgebras into the category of $L$-algebras giving the complex algebras. The next fact and definition makes the terminology precise.

**Fact 3.2.3.** *Every (boolean) coalgebraic modal logic* $(L, \delta)$ *induces a lifting of* $P : \mathsf{Set}^{op} \longrightarrow \mathsf{BA}$ *as shown in the following diagram*




*where the vertical arrows are the "obvious" forgetful functors.*

*More explicitly, the functor* $\widehat{P}_\delta$ *maps a coalgebra* $(X, \xi)$ *to* $(P(X), P(\xi) \circ \delta_X)$*, and a coalgebra morphism* $f$ *to its inverse image, i.e.* $\widehat{P}_\delta(f) = P(f) = f^{-1}$.

We now fix some terminology.

**Definition 3.2.4.** Let $(L, \delta)$ be a (boolean) coalgebraic modal logic. The functor $\widehat{P}_\delta$ is called the *$\delta$-lifting of $P$*. The image of a coalgebra $(X, \xi)$ under $\widehat{P}_\delta$, i.e. $(P(X), P(\xi) \circ \delta_X)$, is called the *complex $(L, \delta)$-algebra* associated with $(X, \xi)$. If there is no risk for confusion we drop the subindex.

The interpretation of a formula, in the language of $(L, \delta)$, on a coalgebra $(X, \xi)$ is given by the unique arrow from the initial $L$-algebra (see Remark 3.2.6), written $(I, \iota)$, into the complex algebra $\widehat{P}(X, \xi)$. More explicitly, the interpretation is given by the unique function

$$[\![-]\!]_\xi^\delta : I \longrightarrow P(X) \tag{3.6}$$

which is a homomorphism between $(I, \iota)$ and $\widehat{P}(X, \xi)$. In particular, a formula $\varphi \in I$ is mapped to a set $[\![\varphi]\!]_\xi^\delta \subseteq X$. This justifies our terminology for $\delta$ and motivates the following convention.

**Definition 3.2.5.** Let $(L, \delta)$ be a (Boolean) logic for $\mathcal{T}$-coalgebras and let $(X, \xi)$ be a $\mathcal{T}$-coalgebra. The function in Equation (3.6), i.e. the initial morphism from the initial $L$-algebra into the complex algebra $\widehat{P}(X, \xi)$, is called *the interpretation into* $(X, \xi)$. If $x \in [\![\varphi]\!]_\xi^\delta$, we say that $x$ *satisfies* $\varphi$ and write $x \Vdash_\xi \varphi$. The relation $\Vdash$ is called the satisfaction relation. If the logic is clear from the context, we will just write $[\![\varphi]\!]_\xi$. We will use the same convention for logics defined in Definition 3.2.13 and 3.2.22.

The next remark addresses other issues concerning Definition 3.2.1.

**Remark 3.2.6.** Recall Definition 3.2.1.

1. Formally speaking, in Equation (3.4) we should write $\mathcal{T}^{op}$ instead of $\mathcal{T}$. We write just $\mathcal{T}$ to keep the notation simple.

2. The requirement that the functor $L$ is determined by finitely generated free algebras has two main purposes: one to ensure that free $L$-algebras exist and two to allow a description of $L$ by modal operators of finite arity. A proof of this can be found in [76], but we will see examples in the next subsections.

3. It is important to understand that $L$ only describes how to add one layer of modalities: If $A$ consists of Boolean formulas, then $L(A)$ consists of modal formulas in which each formula $a \in A$ is under the scope of precisely one modal operator. The initial $L$-algebra is obtained by iterating this construction and contains modal formulas of arbitrary depth. Moreover, $L$ can take into account not only the syntax, but also the axiomatisation of the logic; to capture the axiomatization by a functor, it is essential to consider $L$ on $\mathsf{BA}$ and not simply on $\mathsf{Set}$, see Section 3.2.1.

4. In case $L$ comes from an usual modal signature, see examples below, the interpretation of a formula $\varphi$ in the signature is given by applying the function in Equation (3.6) to the equivalence class of $\varphi$.

5. We do not require any type of normality or additivity of modal operations; in this sense our approach differs from that in [20]. Hence we can also account for non-normal modal logics.

### 3.2.1 Modal Signatures as functors on $\mathsf{BA}$

We now illustrate how the usual boolean algebras with operators from modal logics fit into the picture. The idea is that the algebras in $\mathsf{Alg}(L)$ are the boolean algebras with operators of the logic. We can state this by saying that the functor $L$ adds the operators and their axiomatizations.

In this section we will consider the case of modal signatures. More explicitely, given a set of predicate liftings $\Lambda$, i.e. a modal signature, we want to obtain a functor $L : \mathsf{BA} \longrightarrow \mathsf{BA}$ from the functor $\Sigma_\Lambda + \Sigma_{\mathsf{BA}} : \mathsf{Set} \longrightarrow \mathsf{Set}$ such that $\mathsf{Alg}(L)$ is the category of boolean algebras with operators for $\Sigma_\Lambda$. Functors on boolean algebras take care of the factor $\Sigma_{\mathsf{BA}}$. The problem then reduces to that of "moving" the functor $\Sigma_\Lambda : \mathsf{Set} \longrightarrow \mathsf{Set}$ to the category $\mathsf{BA}$.

First of all notice that algebras for a functor $L : \mathsf{BA} \longrightarrow \mathsf{BA}$ are of the form $\alpha : L(A) \longrightarrow A$, where $A$ is a boolean algebra and $\alpha$ is a boolean homomorphism. Here, it is important to notice that the problem of moving $\Sigma_\Lambda$ to $\mathsf{BA}$ is not solved by simply considering a signature given by a polynomial functor on $\mathsf{BA}$, as it was done in Section 2.3 in the case of $\mathsf{Set}$ and is usually done in categorical algebra.

We now explain why polynomial functors do not work. For example, within such framework, to add a unary $\Box$ to classical propositional logic corresponds to consider algebras for $\mathsf{Id}_{\mathsf{BA}} : \mathsf{BA} \longrightarrow \mathsf{BA}$. In such case an $\mathsf{Id}_{\mathsf{BA}}$-algebra is a boolean algebra $(A, \alpha)$ together with a boolean homomorphism $\mathsf{Id}_{\mathsf{BA}}(A, \alpha) = (A, \alpha) \stackrel{\Box}{\longrightarrow} (A, \alpha)$.

This does not subsume the standard approach in Modal Logic [20], because there is no reason why $\Box$ should be a $\mathsf{BA}$-morphism; in general $\Box$ does not preserve $\vee$ or $\neg$. In other words, BAOs, as in [20], are not $\mathsf{Alg}(\mathsf{Id}_{\mathsf{BA}})$. In general, the technique from categorical algebra of adding operators using polynomial functors on $\mathsf{BA}$ does not work.

To overcome these difficulties, here is a little trick we can use: We define a functor $\mathsf{BA} \longrightarrow \mathsf{BA}$, call it $\bar{L}$, such that algebras $\bar{L}(A, \alpha) \longrightarrow (A, \alpha)$ are in 1-1 correspondence with maps $U(A, \alpha) \longrightarrow U(A, \alpha)$, i.e. functions $A \longrightarrow A$. The next example illustrates this.

**Example 3.2.7.** Let $(A, \alpha)$ be a boolean algebra. To add a unary operator $\Box$, we define $\bar{L}(A, \alpha)$ to be the free boolean algebra generated by $\blacksquare a$, $a \in A$. Note that the $\blacksquare a$'s are just formal symbols and we have

$$\bar{L}_{\Box} \cong FU, \tag{3.7}$$

where $U : \mathsf{BA} \longrightarrow \mathsf{Set}$ is the forgetful functor and $F$ its left adjoint. Using properties of adjoints, it is now clear that an $\bar{L}_{\Box}$-algebra

$$FU(A, \alpha) = \bar{L}_{\Box}(A, \alpha) \longrightarrow (A, \alpha)$$

corresponds to a function $A \longrightarrow A$.

In fact, the previous idea works for any modal signature.

**Example 3.2.8.** In general given a modal signature $\Sigma_{\Lambda}$ we define $\bar{L}_{\Lambda} = F\Sigma_{\Lambda}U$. Again using properties of adjoints, we can see that an $\bar{L}_{\Lambda}$-algebra

$$F\Sigma_{\Lambda}U(A, \alpha) = \bar{L}_{\Lambda}(A, \alpha) \longrightarrow (A, \alpha)$$

correspond to a function $\Sigma_{\Lambda}(A) \longrightarrow A$. In other words, an $\bar{L}_{\Lambda}$-algebra amounts to have a function function $A^{ar(\lambda)} \longrightarrow A$ for each modality $\Box_{\lambda} \in \Sigma_{\Lambda}$.

Next, we observe that certain axioms of a special form can be incorporated into the definition of the functor. In particular, the axioms defining the basic modal logic **K**, i.e. BAOs, are of this form.

**Example 3.2.9.** Continuing Example 3.2.7, define $L_{\Box} : \mathsf{BA} \longrightarrow \mathsf{BA}$ to map an algebra $(A, \alpha)$ to the boolean algebra $L_{\Box}(A, \alpha)$ generated by $\blacksquare a$, $a \in A$, and quotiented by the relation stipulating that $\Box$ preserves finite meets, that is,

$$\Box\top = \top \qquad\qquad \Box(a \wedge b) = \Box a \wedge \Box b \tag{3.8}$$

Compare this with the construction of term algebras for varieties on page 25. It follows from the definition that $\mathsf{BA}$-morphisms $L_{\Box}(A, \alpha) \longrightarrow (A, \alpha)$ are in 1-1 correspondence with meet-preserving maps $A \longrightarrow A$, see [22], therefore, that $\mathsf{Alg}(L)$ is isomorphic to the category of modal algebras [20]. Also notice that there is a surjective natural transformation $q : \bar{L}_{\Box} \longrightarrow L_{\Box}$.

A more formal description of the quotient in the previous example can be found in Section 5.2.2. We avoid the details by now because we are more interested on illustrating the framework.

The next step is to describe the semantics of such a logic without referring to Kripke frames, but directly in terms of the functor $\mathcal{T}$. This allows us to generalise the relationship between algebras and coalgebras to arbitrary functors. The following property will play a crucial role: The contravariant power set functor $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ factors via $\mathsf{BA}$, i.e. the following diagram

$$\begin{array}{ccc} \mathsf{Set}^{op} & \xrightarrow{\quad P \quad} & \mathsf{BA} \\ & {\scriptstyle \mathcal{P}} \searrow \quad \swarrow {\scriptstyle U} & \\ & \mathsf{Set} & \end{array} \tag{3.9}$$

commutes. With this observation in mind we can continue with Example 3.2.9.

**Example 3.2.10.** Continuing Example 3.2.9, consider $\mathcal{T} = \mathsf{Pow}$. In order to define the semantics $\delta_X : L_\Box P(X) \longrightarrow P\mathsf{Pow}(X)$ we first define the semantics $\bar{\delta}_\Box : \bar{L}P(X) \longrightarrow P\mathsf{Pow}(X)$ by mapping a generator $\blacksquare a$, with $a \in P(X)$, as follows

$$\blacksquare a \mapsto \{b \in \mathsf{Pow}(X) \mid b \subseteq a\}. \tag{3.10}$$

It is now easy to check that $\bar{\delta}$ satisfies the equalities in Equation (3.8). From this we conclude that $\bar{\delta}$ extends to the quotient $L_\Box$ via $q : \bar{L}_\Box \longrightarrow L_\Box$ giving $\delta$ as in the following diagram

$$\begin{array}{ccc} \bar{L}_\Box P & \xrightarrow{\quad q_P \quad} & L_\Box P \\ & {\scriptstyle \bar{\delta}} \searrow \quad \swarrow {\scriptstyle \delta} & \\ & P\mathcal{T} & \end{array}$$

More explicitly, $\delta$ maps an equivalence class $\bar{\varphi} \in L_\Box P(X)$ to $\{b \in \mathsf{Pow}(X) \mid b \subseteq \varphi\}$.

Since $\mathcal{P} = UP$ (Diagram 3.9) the predicate lifting associated with the modality $\Box$ can be written $\Box : UP(X) \longrightarrow UP\mathsf{Pow}(X)$. Now notice Equation (3.10) is then just the predicate lifting $\Box$. Hence $\bar{\delta}$ is its transpose of $\Box$, i.e. we have $FUP(X) \xrightarrow{\bar{\delta}=\widehat{\Box}} P\mathsf{Pow}(X)$. Clearly this can be done for any predicate lifting and more generally for any set of predicate liftings $\Lambda$. The next example shows this.

**Example 3.2.11.** Let $\Sigma_\Lambda$ be a modal signature associated with a set of predicate liftings $\Lambda$. Recall from Equation (3.2) that we have a natural transformation $\delta_\Lambda : \Sigma_\Lambda \mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ combining all predicate liftings in $\Lambda$, since $UP = \mathcal{P}$ we can rewrite this as a natural transformation

$$\delta_\Lambda : \Sigma_\Lambda UP \longrightarrow UP\mathcal{T}$$

the transpose of which gives the semantics, i.e. $F\Sigma_\Lambda UP = \bar{L}_\Lambda P \xrightarrow{\bar{\delta}_\Lambda = \widehat{\delta}_\Lambda} P\mathcal{T}$.

Now we detail how to obtain from Equation (3.4), by using Equations (3.5) and (3.6), the usual semantics of $\Box$. First recall that the basic modal language $\mathcal{L}_\Box$ is defined by the grammar

$$\varphi := \bot \mid \top \mid \neg\varphi \mid \varphi_1 \heartsuit \varphi_2 \mid \Box\varphi$$

where $\heartsuit$ is one of the connectives $\wedge, \vee, \rightarrow$. Secondly notice that the initial $\bar{L}_\Box$-algebra is the smallest set closed under the boolean operations and $\Box$ modulo the axioms for Boolean algebras; in other words, the initial $\bar{L}_\Box$ algebra is the term algebra over the language $\mathcal{L}_\Box$ modulo the usual axioms for Boolean algebras. The initial $L_\Box$-algebra is obtained from further quotienting by the modal axioms in Equation (3.8).

According to Equations (3.5) and (3.6), the interpretation of a formula is defined by initiality as in the following diagram

$$\begin{array}{ccccc} L_\Box(I) & & \xrightarrow{\quad\Box\quad} & & I \\ {\scriptstyle L_\Box([\![-]\!]_\xi)}\downarrow & & & & \downarrow{\scriptstyle [\![-]\!]_\xi} \\ L_\Box P(X) & \xrightarrow[\delta_X]{} & P\mathcal{T}(X) & \xrightarrow[\xi^{-1}]{} & P(X) \end{array} \tag{3.11}$$

where $\Box : L_\Box(I) \rightarrow I$ is the initial $L_\Box$-algebra. This means that given a formula $\bar{\varphi} \in I$, which is an equivalence hence the bar, we have $[\![\Box\bar{\varphi}]\!]_\xi = \xi^{-1}(\delta_X(L[\![\bar{\varphi}]\!]_\xi))$. Now we can compute (eliding the subindex $\xi$)

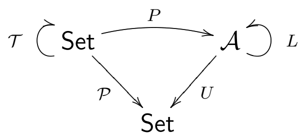
$$\begin{array}{llr} x \Vdash \Box\varphi & \text{iff } x \in [\![\Box\bar{\varphi}]\!] & (\text{Definition } \Vdash) \\ & \text{iff } x \in \xi^{-1}(\delta_X(L_\Box[\![\bar{\varphi}]\!])) & (\text{Diagram (3.11)}) \\ & \text{iff } \xi(x) \in \delta_X(L_\Box[\![\bar{\varphi}]\!]) & (\text{Definition } \xi^{-1}) \\ & \text{iff } \xi(x) \subseteq [\![\varphi]\!] & (\text{Example (3.2.10)}) \\ & \text{iff } (\forall y)(y \in \xi(x) \Rightarrow y \in [\![\varphi]\!]) & (\text{Definition } \Vdash) \\ & \text{iff } (\forall y)(y \in \xi(x) \Rightarrow y \Vdash \varphi) & \end{array}$$

which gives the usual semantics of $\Box$ in terms of a satisfaction relation $\Vdash$. All this shows that the basic theory of algebraic modal logic fits into the abstract functorial framework.

### 3.2.2 Coalgebraic modal logics beyond BA

Yet another advantage of the functorial approach is that it immediately suggests important generalisations. For example, in the following chapters, to construct certain counterexamples, we will need to replace the category BA by other categories corresponding to other base logics. For example, the category of distributive lattices which corresponds to the positive fragment of propositional logic; the category of frames which corresponds to geometric logic; even the category of sets will be used to describe the modalities that need no extra structure to be translated. The key idea to remember here is that the basic propositional logic corresponds to a category of algebras. We can also think of the algebras of the category as propositional, algebraic, theories and of morphisms as truth preserving translations between theories. In the examples mentioned above there are two essential ingredients: Free algebras and powerset algebras. From this we conclude that the category $\mathsf{BA}$ can be replaced by any category $\mathcal{A}$ with a forgetful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$ which has a left adjoint and a functor $P : \mathsf{Set}^{op} \longrightarrow \mathcal{A}$ such that $UP = \mathcal{P}$.

The new situation is depicted in the following diagram




(3.12)

and formalised in the next definition.

**Definition 3.2.12.** A category $\mathcal{A}$ is said to be a category *with powerset algebras* if it satisfies the following conditions:

1. It is monadic over $\mathsf{Set}$ (Definition 2.3.19) i.e. it is a category of algebras;
2. there exists a functor $P : \mathsf{Set}^{op} \longrightarrow \mathcal{A}$ such that $UP = \mathcal{P}$ called the *predicate functor*.

where $U : \mathcal{A} \longrightarrow \mathsf{Set}$ is the forgetful functor.

The generalisation of (boolean) coalgebraic modal logics, Definition 3.2.1, is now immediate.

**Definition 3.2.13.** Let $\mathcal{T}$ be a $\mathsf{Set}$ endofunctor and let $\mathcal{A}$ be a category with power set algebras. A logic for $\mathcal{T}$-coalgebras, or *coalgebraic modal logic*, is a pair $(L, \delta)$ where $L$ is a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$ determined by finitely generated free algebras, and $\delta$ is a natural transformation

$$\delta : LP \longrightarrow P\mathcal{T}. \tag{3.13}$$

The natural transformation $\delta$ is called the *semantics* of the logic. The category $\mathcal{A}$ is called the *base category of the coalgebraic modal logic*. The algebraic theory induced by $\mathcal{A}$ is refereed as the *basic propositional logic*, or the *base structure*, of the coalgebraic modal logic.

A coalgebraic modal logic is said to be *finitary* if $\mathcal{A}$ is finitely presentable and $L$ is determined by finitely generated free algebras (Definition 2.3.12).

**Assumption:** Unless explicitly stated we assume all our coalgebraic modal logics to be finitary.

As in the case of boolean coalgebraic modal logics, the natural transformation $\delta$ is used to associate a dual, or complex, $L$-algebra with each $\mathcal{T}$-coalgebra, see Definition 3.2.4. We use the same convention as in Definition 3.2.5 for the satisfaction relation.

The next remark comments on the extension of the previous definition.

**Remark 3.2.14.** The definition of category of power set algebra might seem ad hoc. However, as we will later see (Chapter 6, Example 6.1.3 and Remark 6.1.4) these allow us to fit the language given by a coalgebraic modal logic in the tradition of abstract model theory where languages are sets and theories are subsets of those.

The next remark illustrates how in the case of a coalgebraic modal logic given by predicate liftings, we can define a language in the usual sense.

**Remark 3.2.15.** In the case that the coalgebraic modal logic $(L, \delta)$, over $\mathcal{A}$, corresponds to a logic of predicate liftings $\Lambda$, i.e. $L = F\Sigma_\Lambda U$ and $\delta$ is the transpose of the function in Equation (3.2), we can define the language more in harmony with Definition 3.1.5 as follows:

Let $\Sigma_\mathcal{A}$ be the algebraic signature of $\mathcal{A}$ (Definition 2.3.6), and let $X$ be a set of propositional variables. The language of predicate liftings, over $\mathcal{A}$, written $\mathcal{L}^\mathcal{A}_\Lambda(X)$, with variables from $X$, is given by the grammar

$$\varphi := x \mid p(\varphi_1, \ldots \varphi_{ar(p)}) \mid \Box_\lambda(\varphi_1, \ldots \varphi_{ar(\lambda)})$$

where $x \in X$, $p \in \Sigma_\mathcal{A}$, and $\lambda \in \Lambda$. Note that $\mathcal{L}^\mathcal{A}_\Lambda(X)$ is the carrier of the free algebra over $X$ for the functor $L^{\mathsf{Alg}(\Sigma_A)}_\Lambda = \mathbb{T}_{\Sigma_\mathcal{A}}\Sigma_\Lambda U_{\Sigma_\mathcal{A}}$, where $U_{\Sigma_\mathcal{A}} : \mathsf{Alg}(\Sigma_\mathcal{A}) \longrightarrow \mathsf{Set}$ is the forgetful functor and $\mathbb{T}_{\Sigma_\mathcal{A}}$ is its left adjoint, i.e. $\mathbb{T}_\Sigma(X)$ is the absolutely free algebra for the signature $\Sigma_\mathcal{A}$.

A simple illustration is the pure modal language.

**Example 3.2.16.** The category $\mathsf{Set}$ is a category of power set algebras. The functor $P$ is given by the contravariant power set functor $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$. The category $\mathsf{Set}$ is monadic over $\mathsf{Set}$ via the identity. Given a modal signature $\Sigma_\Lambda$ we can use the procedure in Example 3.2.8 to obtain $\mathsf{Id}\Sigma_\Lambda\mathsf{Id} = \Sigma_\Lambda$; the semantics is given by Equation (3.2). In this case the language is the smallest set closed under $\Box_\lambda$ for all $\lambda \in \Lambda$, i.e. the pure modal language.

Clearly Examples 3.2.8 and 3.2.11 generalize to any category of power set algebras. This means that every modal signature can be interpreted on a category of power set algebras.

This framework can also be used to present positive modal logic as a coalgebraic modal logic. In this case we replace $\mathsf{BA}$ by $\mathsf{DL}$. The next example shows this.

**Example 3.2.17.** Elaborating on Example 3.2.10 but now letting $\mathcal{A} = \mathsf{DL}$ and considering two modalities $\{\Box, \Diamond\}$.

*Positive modal logic* is given by the functor $L : \mathsf{DL} \to \mathsf{DL}$ that maps a distributive lattice $(A, \alpha)$ to distributive lattice $L(A, \alpha)$ generated by $\blacksquare a$ and $\blacklozenge a$ for all $a \in A$, and quotiented by the relations stipulating that 1) $\Box$ preserves finite meets, 2) $\Diamond$ preserves finite joins, and

$$3) \quad \Box a \wedge \Diamond b \leq \Diamond(a \wedge b) \qquad \Box(a \vee b) \leq \Diamond a \vee \Box b \tag{3.14}$$

More explicitly, we quotient the functor $FU + FU : \mathsf{DL} \to \mathsf{DL}$, where $U : \mathsf{DL} \to \mathsf{Set}$ and $F$ is the left adjoint, with the axioms 1), 2), and 3) above.

We assume the first component of $\bar{L}$ deals with $\Box$ and the second one deals with $\Diamond$; with this convention, the semantics $\bar{\delta}_X : \bar{L}P(X) \to P\mathsf{Pow}(X)$ is defined componentwise as follows: for $a \in \bar{L}PX$, if $a$ belongs to the first component, i.e. it corresponds to the formal symbol $\blacksquare a$, it is mapped as in Example 3.2.10; if $a$ belongs to the second component, i.e. it corresponds to the formal symbol $\blacklozenge a$, it is mapped by

$$\blacklozenge a \mapsto \{b \in \mathsf{Pow}(X) \mid b \cap a \neq \emptyset\}. \tag{3.15}$$

This is just what was described in Example 3.2.11. It is now standard to show that $\bar{\delta}$ extends via the quotient to the semantic $\delta_X : LP(X) \to P\mathcal{T}(X)$. This gives positive modal logic. Positive modal logic was introduced in [33], but also appeared in [61, 1]. The above construction is a variation of the Plotkin power domain, also called the Vietories locale, see e.g. [109].

We fix some conventions, which have been implicit, concerning logics of predicate liftings.

**Definition 3.2.18.** Let $\Lambda$ be a set of predicate liftings for a functor $\mathcal{T}$. Let $\mathcal{A}$ be a category of power set algebras, and write $U : \mathcal{A} \to \mathsf{Set}$ for the forgetful functor and $F$ for its left adjoint.

The coalgebraic modal logic, over $\mathcal{A}$, associated with $\Lambda$ is given by the functor $\bar{L}^{\mathcal{A}}_{\Lambda} = F\Sigma_{\Lambda}U$, and the natural transformation $\bar{\delta}_{\Lambda} : \bar{L}^{\mathcal{A}}_{\Lambda}P \to P\mathcal{T}$ given by the $F$-transpose of the natural transformation $\Sigma_{\Lambda}\mathcal{P} \to \mathcal{P}\mathcal{T}$ in Equation 3.2. If there is no risk for confusion we will avoid the superscript and/or subindex.

### 3.2.3 Other features of the functorial framework

As we have seen, the functorial framework allows us to present a large variety of modal logic for coalgebras uniformly. In particular, as we will see in Section 3.3.1, we can study Moss logic at the same level of classical modal logic. Another advantage, we will illustrate in Section 3.3, is that we can present generic logics for coalgebras. In this section we discuss some other properties and insights.

### Depth-one formulas & one-step semantics

We should still clarify which axioms can be incorporated into a functor. This will be addressed in detail in Section 5.2.2. We provide some highlights here.

To account for axioms we introduce *depth-one formulas*, or depth one modal formulas; these will also play an important role in Chapter 8.

In the case of a logic of predicate liftings, the depth-one modal formulas, over $X$, are the formulas of $\mathcal{L}^{\mathcal{A}}_{\Lambda}(X)$ for which each propositional variable is under the scope of exactly one modal operator. Notice that the depth one formulas are the elements of the set

$$U_{\Sigma_{\mathcal{A}}}\mathbb{T}_{\Sigma_{\mathcal{A}}}\Sigma_{\Lambda}U_{\Sigma_{\mathcal{A}}}\mathbb{T}_{\Sigma_{\mathcal{A}}}(X)$$

where $\mathbb{T}_{\Sigma_{\mathcal{A}}} : \mathsf{Set} \longrightarrow \mathsf{Alg}(\Sigma_{\mathcal{A}})$ maps a set $X$ to the free $\Sigma_{\mathcal{A}}$-algebra over $X$, $U_{\Sigma_{\mathcal{A}}}$ is the right adjoint, or forgetful functor. Indeed, an element in $U_{\Sigma_{\mathcal{A}}}\mathbb{T}_{\Sigma_{\mathcal{A}}}(X)$ is just a term in the signature $\Sigma_{\mathcal{A}}$ with variables in $X$. Then an element in $\Sigma_{\Lambda}U_{\Sigma_{\mathcal{A}}}\mathbb{T}_{\Sigma_{\mathcal{A}}}(X)$ is a modality in $\Sigma_{\Lambda}$ together with a term in $\Sigma_{\mathcal{A}}$, e.g. $\Box_{\lambda}t$. Then the elements in $U_{\Sigma_{\mathcal{A}}}\mathbb{T}_{\Sigma_{\mathcal{A}}}\Sigma_{\Lambda}U_{\Sigma_{\mathcal{A}}}\mathbb{T}_{\Sigma_{\mathcal{A}}}(X)$ are the depth-one modal formulas over $X$.

In fact $L^{\Sigma_{\mathcal{A}}}_{\Lambda} = \mathbb{T}_{\Sigma_{\mathcal{A}}}\Sigma_{\Lambda}U_{\Sigma_{\mathcal{A}}}$ and the language $\mathcal{L}^{\mathcal{A}}_{\Lambda}(X)$ is the free $L^{\Sigma_{\mathcal{A}}}_{\Lambda}$ algebra over $X$. Clearly this can be generalized to any coalgebraic modal logic.

**Definition 3.2.19.** Let $(L, \delta)$ be a coalgebraic modal logic over $\mathcal{A}$. The *depth-one modal formulas* of $(L, \delta)$, over $X$, written $\mathsf{Form}^1_{(L,\delta)}(X)$ are the elements of $ULF(X)$. If the logic is clear from the context we simply write $\mathsf{Form}^1(X)$.

An equation between depth-one modal formulas is called a *rank-1* axiom, or rank-1 equation.

In other words, a **rank-1** formula is a formula where all variables are under the scope of precisely one modal operator. Without going into the technical details here, we note that for all equations of rank 1 we can quotient the functor by the axioms as in Equations (3.8) and (3.14). It was shown in [22, 76] that we can quotient $L^{\mathcal{A}}_{\Lambda}$ with any rank 1 axiomatisation and that every quotient of $L^{\mathcal{A}}_{\Lambda}$ corresponds to a rank 1 axiomatization. It is important to notice that this is not a restriction of coalgebraic logic as such: First use a rank-1 logic to describe properties of all $\mathcal{T}$-coalgebras; then further non-rank-1 axioms can be used to single out $\mathcal{T}$-coalgebras with particular properties. We give more details in Chapter 5.

Together with with depth-one formulas is the so-called one-step semantics. Roughly speaking, one-step semantics is the interpretation of depth-one formulas over the set $\mathcal{T}(X)$. As we discussed, a coalgebraic modal logic $(L, \delta)$ for a functor $\mathcal{T}$ allows us to transform $\mathcal{T}$-coalgebras into $L$-algebras. However, notice that in several occasions we are interested in just describing the successors of a state i.e. the elements in some $\mathcal{T}(X)$. It was noted in [88, 102] that in the case of languages of predicate liftings $\mathcal{T}$-coalgebras are not essential for this; this is referred as the *one-step semantics* of coalgebraic modal logics. We now explain how this works for any coalgebraic modal logic.

To illustrate the procedure we fix a set of propositional variables $Q$ and a category of power set algebras $\mathcal{A}$ with forgetful functor $U : \mathcal{A} \to \mathsf{Set}$; let $F$ be the left adjoint of $U$. We first explain how define the one step semantics for a single predicate liftings, for a functor $\mathcal{T}$.

Every valuation $V : Q \to UP(X) = \mathcal{P}(X)$ can be extended to a meaning function $[\![-]\!]_V : F(Q) \to P(X)$. We can compose this function with any predicate lifting $\lambda : UP \to UP\mathcal{T}$ and obtain the following function.

$$UF(Q) \xrightarrow{U([\![-]\!]_V)} UP(X) \xrightarrow{\lambda_X} UP\mathcal{T}(X)$$

Taking the transpose of this function and replacing $FU$ by $L_\lambda$ we obtain the one step semantics, for $\lambda$, as $[\![-]\!]^1_V = \delta_\lambda \circ L_\lambda([\![-]\!]_V)$; more explicitly this is:

$$L_\lambda F(Q) \xrightarrow{L_\lambda([\![-]\!]_V)} L_\lambda P(X) \xrightarrow{\delta_\lambda} P\mathcal{T}(X)$$

Applying one more time $U$ we obtain an interpretation of all depth-one modal formulas (Definition 3.2.19) on the language of $\Box_\lambda$. This clearly work for any set of predicate liftings and more generally for any coalgebraic logic.

**Definition 3.2.20.** Let $(L, \delta)$ be a coalgebraic modal logic over $\mathcal{A}$, let $Q$ be a set of propositional variables and let $V : Q \to UP(X)$ be a valuation.

The *one-step semantics* of depth-one modal formulas over $Q$ (Definition 3.2.19), relative to $V$, written $[\![-]\!]^1_V$, is given by the following function

$$\mathsf{Form}^1_{(L,\delta)}(Q) = ULF(Q) \xrightarrow{[\![-]\!]^1_V = U(\delta \circ L([\![-]\!]_V))} P\mathcal{T}(X).$$

Given $t \in \mathcal{T}(X)$ and a formula $\psi \in \mathsf{Form}^1_{(L,\delta)}(Q)$, we write $\mathcal{T}(X), t \Vdash^1_V \psi$ to indicate $t \in [\![\psi]\!]^1_V$.

Depth-one formulas were used in [88, 68] to show that issues of soundness and completeness of a coalgebraic modal logic $(L, \delta)$ rely on properties of $\delta$. We will discuss this in more detail in Section 5.2.2. As a preview we mention that a coalgebraic modal logic $(L, \delta)$ is always sound and it is complete if $\delta$ is injective (Proposition 5.2.16).

### Further generalizations

The functorial framework suggests various generalisations.

**Definition 3.2.21.** Let $\mathcal{A}$ be a monadic category over $\mathsf{Set}$ and let $\mathcal{T}$ be an endofunctor on $\mathsf{Set}$. Also assume there is a functor $P : \mathsf{Set}^{op} \to \mathcal{A}$ with left adjoint $S$.

A *logic for $\mathcal{T}$-coalgebras*, also called a *coalgebraic modal logic* for $\mathcal{T}$, based on $\mathcal{A}$ and relative to $P$, is a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$ together with a natural transformation

$$\delta : LP \to P\mathcal{T}. \tag{3.16}$$

We use the same conventions as in Definition 3.2.13.

The highest level of generality that we can achieve with this framework is to simply consider a general contravariant adjunction.

**Definition 3.2.22.** Let $P : \mathbb{C}^{op} \longrightarrow \mathcal{A}$ be a functor with a left adjoint $S$. Let $\mathcal{T}$ be an endofunctor on $\mathbb{C}$. A coalgebraic modal logic for $\mathcal{T}$-coalgebras, or coalgebraic modal logic, is a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$ together with a natural transformation $\delta : LP \to P\mathcal{T}$.

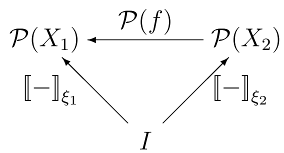
### Expressivity

A key property of coalgebraic modal logics is that formulas are invariant under behavioural equivalence. This will follow from the following proposition.

**Proposition 3.2.23.** *Let $(L, \delta)$ be coalgebraic modal logic, over a category of power set algebras, for a functor $\mathcal{T}$. The interpretation of the $(L, \delta)$-formulas is invariant under coalgebra morphisms.*

*More precisely, if $f : (X_1, \xi_2) \longrightarrow (X_2, \xi_2)$ is a coalgebra morphism and $\varphi$ is a formula in $(L, \delta)$ then for every $x \in X_1$ if $x \in [\![\varphi]\!]_{\xi_1}$ then $f(x) \in [\![\varphi]\!]_{\xi_2}$.*

*Proof.* Let $(I, \iota)$ be the initial $L$-algebra. Notice that since $UP(f) = \mathcal{P}(f) = f^{-1}$ the statement proposition can be rephrased as follows: for every morphism of coalgebras $f : (X_1, \xi_2) \longrightarrow (X_2, \xi_2)$ the following diagram




commutes. The commutativity of the diagram follows by initiality. Indeed, since $f$ is a morphism of coalgebras then $f^{-1} = \mathcal{P}(f)$ is a homomorphism between the complex algebras $(P(X_2), P(\xi_2) \circ \delta_{X_1})$ and $(P(X_1), P(\xi_1) \circ \delta_{X_1})$, see Fact 3.2.3. Therefore its composition with $[\![-]\!]_{\xi_2}$ is a homomorphism from the initial $L$-algebra into $(P(X_1), P(\xi_1) \circ \delta_{X_1})$ which by initiality must be equal to $[\![-]\!]_{\xi_1}$. This concludes the proof. $\square$

Another important property is the so-called Hennessy-Milner property which states that the language can distinguish states that are not behavioural equivalent. The Hennessy-Milner property has been studied using the so-called mate of the natural transformation transformation $\delta$, see [59].

We first have to notice that the Stone duality adjunction can be generalised to any category with power set algebras.

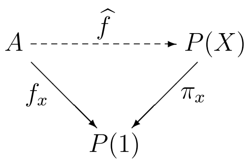
**Proposition 3.2.24.** [2] *For every category of power set algebras $\mathcal{A}$, the functor $P : \mathsf{Set}^{op} \longrightarrow \mathcal{A}$ has a left adjoint $S : \mathcal{A} \longrightarrow \mathsf{Set}^{op}$ given by $S(A) = \mathcal{A}(A, P1)$.*

*Proof.* We will describe the counit of the adjunction. The counit is given by a function $\varepsilon_X : X \longrightarrow SPX = \mathcal{A}(P(X), P(1))$; this function maps $x$ to $P(i_x)$ where we write $i_x : 1 \longrightarrow X$ for the map picking $x$. From here, to show that $P$ is right adjoint of $S$, it suffices to prove that given a function $f : X \longrightarrow S(A)$ we can find an appropriate morphism $\widehat{f} : A \longrightarrow P(X)$ in $\mathcal{A}$ such that $S(\widehat{f}) \circ \varepsilon_X = f$.

We now show how to define $\widehat{f}$. First recall that $\mathcal{P}(\coprod A_i) = \prod \mathcal{P}(A_i)$. By assumption $UP = \mathcal{P}$, hence for every set $X$ we have

$$\begin{aligned}
UP(X) = \mathcal{P}(X) = \mathcal{P}(\coprod_{x\in X} 1) && (\coprod_{x\in X} 1 = X)\\
= \prod_{x\in X} \mathcal{P}(1) && \text{(prev. observation)}\\
= \prod_{x\in X} UP(1) = U \prod_{x\in X} P(1). && (U \text{ is a right adjoint})
\end{aligned}$$

Since $U$ is monadic it reflect limits hence we conclude $\prod_{x\in X} P(1) \cong P(X)$. Now notice that we can see $f : X \longrightarrow S(A) = \mathcal{A}(A, P(1))$ as a family $\{f_x : A \longrightarrow P(1)\}_{x\in X}$. Using the previous observation we define $\widehat{f}$ as the product of all those maps. In other words, $\widehat{f}$ is the only map that makes all the following diagrams




commute, where $\pi_x$ is the projection given by $\prod_{x\in X} P(1) \cong P(X)$. More concretely $\pi_x = P(i_x)$.

We now show $S(\widehat{f}) \circ \varepsilon_X = f$.

$$\begin{aligned}
S(\widehat{f}) \circ \varepsilon_x(x) &= S(\widehat{f}) \circ P(i_x) && \text{(def. } \varepsilon_X)\\
&= P(i_x) \circ \widehat{f} && \text{(def. } S)\\
&= \pi_x \circ \widehat{f} && (\pi_x = P(i_x))\\
&= f_x. && \text{(above diagram)}
\end{aligned}$$

[2]At the moment of writing we do not know whether this result is known

The arrow $\widehat{f}$ is unique because any arrow $g$ such that $S(g) \circ \varepsilon_X = f$ we must have $\pi_x \circ g = f_x$ and then by the universal property of products $g = \widehat{f}$. This concludes the proof. □

Let $(L, \delta)$ be a coalgebraic modal logic as in Definition 3.2.21. The key observation to study the Hennessy-Milner property with in this framework, is that there is bijective correspondence between natural transformation $\delta : LP \to P\mathcal{T}$ and natural transformations $\tau : \mathcal{T}S \to SL$. We now show this, Let $\eta$ and $\varepsilon$ be the unit and counit, respectively, of the adjunction given by the functors $P$ and $S$. Given $\delta : LP \to P\mathcal{T}$ we define $\tau$ via the following composite:

$$\mathcal{T}S \xrightarrow{\varepsilon_{\mathcal{T}S}} SP\mathcal{T}S \xrightarrow{S(\delta_S)} SLPS \xrightarrow{SL(\eta)} SL.$$

On the converse direction, given $\tau : \mathcal{T}S \to SL$ we obtain $\delta$ via

$$LP \xrightarrow{\eta_{LP}} PSLP \xrightarrow{P(\tau_S)} P\mathcal{T}SP \xrightarrow{P\mathcal{T}(\varepsilon)} P\mathcal{T}.$$

The the key result is

**Proposition 3.2.25** ([59])**.** *A coalgebraic modal logic $(L, \delta)$ (Definition 3.2.21) has the Hennessy-Milner property iff each of the components of the mate of $\delta$, i.e. $\tau : \mathcal{T}S \to SL$, are injective.*

Other approaches, see e.g [88], use the so called final-sequence to prove the Henessy-milner property; we do not discuss them here because it goes beyond the functorial framework. In Chapter 6 we present a study on the Hennessy-Milner property which can be seen as an alternative to those approaches using the final sequence.

## 3.3 Two generic coalgebraic modal logics

We have seen that coalgebras come equipped with a generic notion of behavioural equivalence (Definition 2.2.2). In the same spirit, the quest for the generic modal language to describe coalgebraic systems has played a key role in the development of logics for coalgebras. Two major currents have been successfully in claiming the title for themselves: The Moss logic and the logic of all predicate liftings. Both proposals can be elegantly presented within the functorial framework of coalgebraic modal logics introduced in the previous section.

### 3.3.1 Moss Logic

The Moss logic [84] was the first proposal of a generic coalgebraic modal logic parametric in the functor $\mathcal{T}$. On the technical side, Moss logic requires the functor

$\mathcal{T}$ to preserve weak pullbacks (Section 2.2); examples of such functors include all KPFs and composition of those with $\mathcal{B}_{\mathbb{N}}$ and $\mathcal{D}$, but not the functor $\mathcal{P}\mathcal{P}^{op}$.

The idea of the Moss logic is to use the functor $\mathcal{T}$ itself as a modality. From the perspective of modal signatures, Example 3.2.8, this amounts to define the functor $L$ using $\mathcal{T}$ instead of $\Sigma_\Lambda$. It is important to notice that from the perspective of the functorial framework, Definition 3.2.13, this is more natural than the use of $\Sigma_\Lambda$. The next definition introduces the functor for the Moss logic.

**Definition 3.3.1.** Let $\mathcal{A}$ be a category with power set algebras, let $U$ be the forgetful functor and let $F$ be its left adjoint; Given a weak pullback preserving $\mathsf{Set}$-endofunctor $\mathcal{T}$. *Moss (finitary) logic for $\mathcal{T}$, on $\mathcal{A}$*, is given by the functor

$$F\mathcal{T}_\omega U = M_\mathcal{T} : \mathcal{A} \to \mathcal{A},$$

where $\mathcal{T}_\omega$ is the finitary version of $\mathcal{T}$. If there is no risk for confusion, we will simply write $M$.

Notice that in the previous definition we used $\mathcal{T}_\omega$ instead of $\mathcal{T}$. The next remark explains this divergence.

**Remark 3.3.2.** In the original version [84], Moss showed that his coalgebraic logic characterizes bisimilarity of $\mathcal{T}$-coalgebras. However, $\mathcal{T}$ may permit unbounded branching, e.g. $\mathcal{T} = \mathsf{Pow}$, therefore a general result requires infinitary conjunctions in the logic (but does not need negation). Here our interests are different: We want to specify properties of coalgebras using only finitary Boolean connectives. In [108] it was shown that for modal formulas to have only finite depth we should work with the finitary version $\mathcal{T}_\omega$ instead of $\mathcal{T}$.

In the case of the Moss logic we can still concretely present a "standard" language, i.e. formulas are not equivalence classes; however, the language is multi-sorted.

**Definition 3.3.3.** The Moss language $\mathcal{M}_\mathcal{T}$ is the smallest set closed under boolean operations and under the formation rule

$$\text{if } t \in \mathcal{T}_\omega(\mathcal{M}_\mathcal{T}) \text{ then } \nabla t \in \mathcal{M}_\mathcal{T}.$$

Quotienting $\mathcal{M}_\mathcal{T}$ by Boolean axioms yields the carrier of the initial $M_\mathcal{T}$-algebra, compare this with Example 3.2.7. If there is no risk for confusion we will drop the subscript $\mathcal{T}$.

The Moss language can also be characterised as the carrier of the initial algebra for the functor $\mathcal{T}_\omega + \Sigma_{\mathsf{BA}} : \mathsf{Set} \to \mathsf{Set}$. Of course we can also present the Moss language concretely for any category of power set algebras $\mathcal{A}$ for which we have a concrete description of the signature, e.g. $\mathsf{DL}$ or $\mathsf{Frm}$.

By definition, the semantics of the Moss logic will be given by a natural transformation $M_{\mathcal{T}}P \longrightarrow P\mathcal{T}$, as in Equation (3.13). Unravelling the definitions, we see that the semantics ought to come from a natural transformation $F\mathcal{T}_\omega UP \longrightarrow P\mathcal{T}$. As it was seen in the case of modal signatures, i.e. languages of predicate liftings, it is then enough to give a natural transformation

$$\mathcal{T}_\omega\mathcal{P} \longrightarrow \mathcal{P}\mathcal{T};$$

the next definition makes this explicit.

**Definition 3.3.4.** Let $\mathcal{T}$ be a weak pullback preserving functor. The semantics $M_{\mathcal{T}}P \longrightarrow P\mathcal{T}$ of Moss's logic is induced by $\blacktriangledown : \mathcal{T}_\omega\mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ mapping $\Phi \in \mathcal{T}_\omega\mathcal{P}(X)$ to

$$\blacktriangledown(\Phi) = \{t \in \mathcal{T}(X) \,|\, t \ \overline{\mathcal{T}}(\in_X)\ \Phi\}, \tag{3.17}$$

where $\overline{\mathcal{T}}(\in_X)$ is the relation lifting of $\in_X$ (Section 2.2).

Let $\mathcal{A}$ be category of power set algebras. Let $U$ be the forgetful functor and let $F$ be its left adjoint. The *(finitary) Moss coalgebraic modal logic*, or Moss logic for short, is given by $(M_T, \nabla)$, where $M_{\mathcal{T}} = F\mathcal{T}_\omega U$ and $\nabla : F\mathcal{T}_\omega UP \longrightarrow P\mathcal{T}$ is the transpose of $\blacktriangledown$, i.e. $\nabla = \varepsilon \circ F(\blacktriangledown)$ where $\varepsilon : FU \longrightarrow \mathsf{Id}_{\mathcal{A}}$ is the counit of the adjunction.

The fact that $\blacktriangledown : \mathcal{T}_\omega\mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ is natural uses the fact that $\mathcal{T}$ preserves weak pullbacks; it follows from Proposition 2.2.10.

**Remark 3.3.5.** The fact that $\blacktriangledown$ is natural is essentially the observation that Moss logic is invariant under bisimilarity. However, should the functor not preserve weak-pullbacks naturality might fail.

The next examples illustrate the semantics given by $\blacktriangledown$.

**Example 3.3.6.**

1. In the case of the identity functor $\mathsf{Id}$, we have that $\blacktriangledown : \mathsf{Id}\mathcal{P} \longrightarrow \mathcal{P}\mathsf{Id}$ is the identity. The Moss logic is just that of deterministic transition systems ($\nabla\varphi \equiv \Box\varphi \equiv \Diamond\varphi$), i.e. in this case the Moss modality is the modality "next time". Explicitly, a state $x$ in a coalgebra $\xi$ satisfies $\nabla\varphi$ iff $\xi(x) \in [\![\varphi]\!]$.

2. In the case of a constant functor $\mathsf{K}_C$, we have that $\blacktriangledown : \mathsf{K}_C\mathcal{P} \longrightarrow \mathcal{P}\mathsf{K}_C$ maps an element $c \in C = \mathsf{K}_C\mathcal{P}(X)$ to the set $\{c\}$. A state $x$ in a coalgebra $\xi$ satisfies $\nabla c$ iff $\xi(x) = c$, i.e. $\nabla c$ holds on $x$ iff the colour of $x$ is $c$.

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

$$t \in \blacktriangledown(\Phi) \text{ iff } \pi_1(t) = \pi_1(\Phi) \text{ and } \pi_2(t) \in \pi_2(\Phi).$$

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

4. In the case of the covariant power set functor, we have that for $\Phi \in \mathsf{Pow}\mathcal{P}(X)$ the set $\blacktriangledown(\Phi)$ is given by

$$t \in \blacktriangledown(\Phi) \text{ iff } (\forall x \in t \;.\; \exists \varphi \in \Phi \;.\; x \in \varphi) \text{ and } (\forall \varphi \in \Phi \;.\; \exists x \in t \;.\; x \in \varphi).$$

   As we said before, page 4, and not difficult to check, in this case the Moss logic (over BA or DL) is equivalent to classical modal logic, that is, there are *translations* in both directions:

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

   Hence, the Moss logic for Pow is equivalent to standard modal logic.

5. To describe $\blacktriangledown$ in the case of the finite distribution functor recall that each $b \in \mathcal{D}(X)$ and $B \in \mathcal{D}\mathcal{P}(X)$ can be presented as finite sequences $b = (x_i, p_i)_{1\leq i\leq n}$ for some $x_i \in X, p_i \in [0,1], p_i > 0, n \in \mathbb{N}$; and $B = (\varphi_j, q_j)_{1\leq j\leq m}$ for $\varphi_j \in \mathcal{P}(X), q_j \in [0,1], q_j > 0, m \in \mathbb{N}$, see Example 2.2.8. The relation $b\,\overline{\mathcal{D}}(\in_X)\,B$ can be then described as follows: $b\,\overline{\mathcal{D}}(\in_X)\,B$ iff there are $(r_{ij})_{1\leq i\leq n, 1\leq j\leq m}$, $r_{ij} \in [0,1]$ such that $x_i \notin \varphi_j \;\Rightarrow\; r_{ij} = 0$ and $\sum_i r_{ij} = q_j$ and $\sum_j r_{ij} = p_i$.

   For example, a state $x$ in a coalgebra $\xi$ satisfies $\nabla\{(\varphi, q), (\top, 1-q)\}$ iff the probability of going to a successor satisfying $\varphi$ is larger or equal to $q$. That is, $\nabla$ (together with Boolean operators) can express the modal operators $\Diamond_p$ of probability logic [49], see page 31 here.

6. In the case of the finite multiset functor we have the same description as in the case of the distribution functor by just replacing $[0,1]$ by $\mathbb{N}$. For example, a state $x$ in a coalgebra $\xi$ satisfies

   - $\nabla\{(\top, n)\}$ iff $x$ has exactly $n$ successors;
   - $\nabla\{(\varphi, m), (\top, n)\}$ iff $x$ has at least $m$ successors satisfying $\varphi$ and exactly $m+n$ successors in total.

   In fact, each $\nabla$-formula specifies the total number of successors; this means that the usual graded modalities, introduced on page 31, can therefore not be expressed using $\nabla$.

The following properties will be useful later.

**Proposition 3.3.7** ([84]). *The formulas in Moss Logic, for a weak pullback preserving functor $\mathcal{T}$, are invariant under bisimulation and behavioural equivalence.*

The natural transformation $\blacktriangledown : \mathcal{T}\mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ is very particular in the sense that we can change $\mathcal{P}$ by Pow and it will still be a natural transformation. The next remark makes this more explicitly.

**Remark 3.3.8.** In case $\mathcal{T}$ preserves weak pullbacks we can also consider $\blacktriangledown$ using $\mathsf{Pow}$ instead of $\mathcal{P}$, i.e. as a natural transformation $\blacktriangledown : \mathcal{T}\mathsf{Pow} \longrightarrow \mathsf{Pow}\mathcal{T}$. In fact, in this case we can say a bit more; namely that $\blacktriangledown$ is a distributive law over the monad $(\mathsf{Pow}, \eta, \mu)$, see Example 2.3.16. This means that $\blacktriangledown : \mathcal{T}\mathsf{Pow} \longrightarrow \mathsf{Pow}\mathcal{T}$ is natural and the following diagrams

$$\begin{array}{ccc} & \mathcal{T} & \\ {\scriptstyle \mathcal{T}(\eta)} \swarrow & & \searrow {\scriptstyle \eta_{\mathcal{T}}} \\ \mathcal{T}\mathsf{Pow} & \xrightarrow[\blacktriangledown]{} & \mathsf{Pow}\mathcal{T} \end{array} \qquad \begin{array}{ccccc} \mathcal{T}\mathsf{PowPow} & \xrightarrow{\blacktriangledown_{\mathsf{Pow}}} & \mathsf{Pow}\mathcal{T}\mathsf{Pow} & \xrightarrow{\mathsf{Pow}(\blacktriangledown)} & \mathsf{PowPow}\mathcal{T} \\ {\scriptstyle \mathcal{T}(\mu)} \downarrow & & & & \downarrow {\scriptstyle \mu_{\mathcal{T}}} \\ \mathsf{PowPow}\mathcal{T} & & \xrightarrow[\blacktriangledown]{\qquad\qquad} & & \mathsf{Pow}\mathcal{T} \end{array}$$

commute. In fact, $\blacktriangledown : \mathcal{T}\mathsf{Pow} \longrightarrow \mathsf{Pow}\mathcal{T}$ is a distributive law iff $\mathcal{T}$ preserves weak pullbacks, for more detail see [56]. The axioms of distributive laws will come back in the axiomatization of Moss Logic [69], see Chapter 5 here.

### 3.3.2 Logics of Predicate Liftings

Although the Moss logic has a natural description within the functorial framework for coalgebraic modal logics, It is not totally clear how $\nabla$ is a direct generalisation of the logics in Section 1.2. Concrete modalities given by predicate liftings are a direct generalisation of the modal logics of Section 1.2. However, it might seem that there are too many predicate liftings. In this section we show that this is not the case and that we can describe all of them concretely. On top of this, we illustrate how for any functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ there is a canonical way of extracting the modal operators and their semantics from $\mathcal{T}$.

We will assume that the basic propositional logic corresponds to a variety with powerset algebras (Definition 3.2.12). We will make use of the fact that every algebra is the colimit of finitely generated free ones, i.e. the colimit of its canonical diagram.

#### Predicate liftings via Yoneda

In order to describe concrete modalities, i.e. predicate liftings, concretely we need to use some categorical machinery. Before going into the details recall that $\mathcal{P}(X) = \mathsf{Set}(X, 2)$ and more generally $\mathcal{P}(X)^n = \mathsf{Set}(X, 2^n) = \mathsf{Set}(X, \mathcal{P}(n))$. The categorically minded reader will recognise the soil to apply the Yoneda Lemma. This is the path that we follow. The following version of Yoneda Lemma will be enough for the purposes of this manuscript.

**Proposition 3.3.9.** *For every functor $K : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ and every $n$ there is a natural isomorphism (natural in $n$ and $K$)*

$$Y_{(n,K)} : K\mathcal{P}(n) \longrightarrow \mathsf{Nat}(\mathcal{P}^n, K). \tag{3.18}$$

*Proof.* As we mentioned before $\mathcal{P}^n(X) = \mathsf{Set}(X, \mathcal{P}(n))$.

We only define the bijection between $K\mathcal{P}(n)$ and natural transformations $\mathcal{P}^n \longrightarrow K$. This is done as follows: Assign to an element $p \in K\mathcal{P}(n)$ the natural transformation $Y(p) : \mathcal{P}^n(X) \longrightarrow K(X)$ which maps a function $v : X \longrightarrow \mathcal{P}(n)$ to the image of $p$ under the function $K(v) : K\mathcal{P}(n) \longrightarrow K(X)$, i.e. $K(v)(p)$; note that the arrow $v$ changes the direction because $K$ is contra variant.

Conversely, for a natural transformation $\lambda_X : \mathcal{P}^n(X) \longrightarrow K(X)$, taking $X = \mathcal{P}(n)$, notice $\mathcal{P}(\mathcal{P}(n))^n = \mathsf{Set}(\mathcal{P}(n), \mathcal{P}(n))$; hence $Y(\lambda) := \lambda_{\mathcal{P}(n)}(id_{\mathcal{P}(n)})$ is an element of $K\mathcal{P}(n)$. □

The next remark will be used in the prove of Theorem 4.2.10, it can be skipped for now.

**Remark 3.3.10.** The naturality of $Y$ in the previous proposition implies that for any natural transformation $\hbar : K' \longrightarrow K$ and any $n$, the following diagram

$$\begin{array}{ccc} \mathsf{Nat}(\mathcal{P}^n, K) & \xleftarrow{\;Y_{(n,K)}\;} & K\mathcal{P}(n) \\ {\scriptstyle \hbar \circ -}\uparrow & & \uparrow{\scriptstyle \hbar_{\mathcal{P}(n)}} \\ \mathsf{Nat}(\mathcal{P}^n, K') & \xleftarrow[\;Y_{(n,K')}\;]{} & K'\mathcal{P}(n) \end{array}$$

commutes.

The case of predicate liftings follows by taking $K = \mathcal{P}\mathcal{T}$.

**Corollary 3.3.11.** *The set of $n$-ary predicate liftings for a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ is in natural bijection with the set $\mathcal{P}\mathcal{T}\mathcal{P}(n)$. In case $n = 1$ the set of unary predicate liftings for $\mathcal{T}$ is in natural bijection with the subsets of $\mathcal{T}(2)$, i.e. $\mathcal{P}\mathcal{T}\mathcal{P}(1)$.*

More explicitly, Proposition 3.3.9 depicts a procedure to convert natural transformations $\mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ into subsets of $\mathcal{T}\mathcal{P}(n)$ and viceversa. This is done as follows: Given a set $P \subseteq \mathcal{T}\mathcal{P}(n)$ we define a predicate lifting $\lambda_P : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ which maps a sequence $\varphi : n \longrightarrow \mathcal{P}(X)$ to the set

$$(\lambda_P)_X(\varphi) = \{t \in \mathcal{T}(X) \mid \mathcal{T}(\chi_\varphi)(t) \in P\} \tag{3.19}$$

where $\chi_\varphi : X \longrightarrow \mathcal{P}(n)$ is the transpose of $\varphi$. In the converse direction, a predicate lifting $\lambda : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ corresponds to the image of $id_{\mathcal{P}(n)}$. In case $n = 1$, the set corresponding to $\lambda$ is the image of $\varphi = \{\top\} \in \mathcal{P}(2)$ under $\lambda_2$. This presentation of predicate liftings was first used in [99].

We now illustrate how the predicate liftings in Example 3.1.3 can be described from this perspective.

**Example 3.3.12.** Recall Example 3.1.3. In the following $\mathcal{P}(1) = 2 = \{\top, \bot\}$.

1. For a constant functor $\mathsf{K}_C$, the predicate lifting $\lambda_P : \mathcal{P} \longrightarrow \mathcal{P}\mathsf{K}_C$ with constant value $P$, where $P \subseteq C = \mathsf{K}_C(2)$, correspond to the image of $\{\top\}$ under $\lambda_P$ which is the set $P$.

2. For the functor $\mathcal{P}(Q) \times \mathcal{T}$, where $Q$ is a fixed set propositional letters, the predicate liftings providing the propositional information of $q \in Q$ correspond to the sets $U_q \times \mathcal{T}(2)$ and $U_{\neg q} \times \mathcal{T}(2)$ respectively, where we write $U_q$ for the set of subsets of $Q$ containing $q$ and $U_{\neg q}$ for its complement. Indeed, we have $\lambda_2^q(\{\top\}) = \{(U,t) \in \mathcal{P}(Q) \times \mathcal{T}(2) \mid q \in U\} = U_q \times \mathcal{T}(2)$, and $\lambda_2^{\neg q}(\{\top\}) = \{(U,t) \in \mathcal{P}(Q) \times \mathcal{T}(2) \mid q \notin U\}$.

3. For the covariant power set functor $\mathsf{Pow}$, the existential modality corresponds to $\Diamond_2(\{\top\}) = \{\psi \in \mathsf{Pow}(2) \mid \psi \cap \{\top\} \neq \emptyset\}$, i.e. the set $\{\{\top\}, \{\top, \bot\}$. For the universal modality we have $\Box_2(\{\top\}) = \{\psi \in \mathsf{Pow}(2) \mid \psi \subseteq \{\top\}\}$ which means that it corresponds to the set $\{\emptyset, \{\top\}\}$.

4. For the neighbourhood functor $\mathcal{P}\mathcal{P}^{op}$ the non monotone modality $\Box$ corresponds to the set $\Box_2(\{\top\}) = \{N \in \mathcal{P}\mathcal{P}^{op}(2) \mid \{\top\} \in N\}$, this is the ultrafilter generated by $\{\top\}$.

5. For the multiset functor the graded modality $\Diamond_k$ corresponds to the set $\{B : 2 \longrightarrow \mathbb{N} \mid B(\top) \geq k\}$; in this case since the value of $\bot$ is irrelevant, we can say that $\Diamond_k$ corresponds to the set $[k, \infty)$. In general, a predicate lifting for $\mathcal{B}_{\mathbb{N}}$ can be described by two subsets of $\mathbb{N}$; one describing the target of $\top$ and other describing the target of $\bot$.

6. For the distribution functor $\mathcal{D}$, notice that we can describe a probability distribution $d : 2 \longrightarrow [0,1]$ by its value on $\top$ ($d(\bot) = 1 - d(\top)$), we can then say that unary predicate liftings correspond to subsets of $[0,1]$. More precisely, $P \subseteq [0,1]$ corresponds to the set of distributions $d : 2 \longrightarrow [0,1]$ such that $d(\top) \in P$. In particular, $\Diamond_p$ corresponds to the set $[p,1]$. Similarly, the predicate lifting $\Box_p = \neg\Diamond_p\neg$ correspond to the set $(1-p, 1]$; the modality $\Diamond^p = \Diamond_{1-p}\neg$ corresponds to $[0,p]$. In general, the predicate lifting associated with an interval $(q,q') \subseteq [0,1]$ maps a set $\varphi \subseteq X$ to the set of probability distributions over $X$ that assign a probability between $q$ and $q'$ to the set $\varphi$.

### A generic description of the logic of all predicate liftings

Using the perspective of predicate liftings given by the Yoneda Lemma we can present the signature of all (finitary) predicate liftings by

$$\Sigma_{\mathcal{T}} = \coprod_{n<\omega} \mathcal{P}\mathcal{T}\mathcal{P}(n) \times (-)^n.$$

Consequently, we obtain a coalgebraic modal logic $(\bar{L}_{\mathcal{T}}, \bar{\delta}_{\mathcal{T}})$ on any category of power set algebras (Definition 3.2.18). More specifically, recall that $\bar{L}_{\mathcal{T}} = F\Sigma_{\mathcal{T}}U$, and the semantics $\bar{\delta}_{\mathcal{T}} : \bar{L}_{\mathcal{T}} P \longrightarrow P\mathcal{T}$ is given by the coproduct of the transposes of the predicate liftings. However, this logic is far from complete. It was noted in [75, 76] that using the finitely presentability of $\mathsf{BA}$, i.e. that every Boolean Algebra is the colimit of finite Boolean Algebras, we can present a complete logic of predicate lifting. In fact the construction works for any category of power set algebras. This section requires some fitness in category theory.

We now introduce the construction in [76] formally.

Let $\mathcal{A}$ be a category of power set algebras. Recall that every algebra in $\mathcal{A}$ is the colimit of its canonical diagram (Definition 2.3.8). In particular, for each set $X$, the algebra $P(X)$ can be presented as the colimit of its canonical diagram. More explicitly, $P(X)$ is the colimit of all maps $(c_i : F(n_i) \longrightarrow P(X))$, where $i$ ranges over valuations $\{n \longrightarrow UP(X) \mid n < \omega\}$, $n_i$ denotes the domain of $i$ and $c_i : F(n_i) \longrightarrow P(X)$ is the transpose of $i$.

**Definition 3.3.13.** Let $\mathcal{A}$ be a category with power set algebras, and let $\mathcal{T}$ be a $\mathsf{Set}$-endofunctor. Let $S : \mathcal{A} \longrightarrow \mathsf{Set}^{op}$ be the left adjoint of $P : \mathsf{Set}^{op} \longrightarrow \mathcal{A}$ (Proposition 3.2.24), i.e. $S(A) = \mathcal{A}(A, P(1)$.

The logic $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$, over $\mathcal{A}$, is defined as follows:

The functor $L_{\mathcal{T}} : \mathcal{A} \longrightarrow \mathcal{A}$ is defined on finitely generated free algebras as

$$L_{\mathcal{T}}(F(n)) = P\mathcal{T}\mathcal{P}(n)$$

and extended to arbitrary $A \in \mathcal{A}$ via colimits (Definition 2.3.12).

The semantics $(\delta_{\mathcal{T}})_X : L_{\mathcal{T}} P(X) \longrightarrow P\mathcal{T}(X)$ is the unique arrow making the following diagram

$$\begin{array}{ccc} L_{\mathcal{T}} P(X) & \overset{(\delta_{\mathcal{T}})_X}{\dashrightarrow} & P\mathcal{T}(X) \\ {\scriptstyle L_{\mathcal{T}}(c_i)} \uparrow & & \uparrow {\scriptstyle P\mathcal{T}(\widehat{c_i})} \\ L_{\mathcal{T}} F(n_i) & \underset{id}{\longrightarrow} & P\mathcal{T}\mathcal{P}(n_i) \end{array} \tag{3.20}$$

commute for each $i : n_i \longrightarrow UP(X)$; where $\widehat{c_i}$ comes from applying the sequence of isomorphisms $\mathcal{A}(F(n_i), P(X)) \cong \mathsf{Set}(n_i, UPX) \cong \mathsf{Set}(n_i, \mathcal{P}(X)) \cong \mathsf{Set}(X, \mathcal{P}(n_i))$ to $c_i$, i.e. the transpose of $i$.

We now explain the construction above in more detail. We first describe $UL_{\mathcal{T}}(A)$, i.e. the carrier of $L_{\mathcal{T}}(A)$, more concretely. The key point is that $UL_{\mathcal{T}}(A)$ is the colimit of a directed diagram hence we can use the description of directed colimits in $\mathsf{Set}$ (Proposition A.0.6) to compute it. We now elaborate on this.

For any algebra $A \in \mathcal{A}$ the canonical diagram is a directed diagram (Proposition 2.3.9). Therefore, $L_{\mathcal{T}}(A)$ is also the colimit of a directed diagram, more precisely, it is the colimit of the diagram $\left(L_{\mathcal{T}} F(n_i) \xrightarrow{L_{\mathcal{T}}(f_i^j)} L_{\mathcal{T}} F(m_j)\right)$. Since $\mathcal{A}$

is a (finitary) variety the forgetful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$ preserves this colimit (Proposition 2.3.9). This means that $UL_T(A)$ is the colimit of the directed diagram $\left(UL_{\mathcal{T}}F(n_i) \xrightarrow{UL_{\mathcal{T}}(f_i^j)} UL_{\mathcal{T}}F(m_j)\right)$.

Now we can use Proposition A.0.6 to describe $UL_T(A)$. Since $UL_{\mathcal{T}}F(n) = UP\mathcal{T}\mathcal{P}(n) = \mathcal{P}\mathcal{T}\mathcal{P}(n)$ we have that $UL_{\mathcal{T}}(A)$ is (isomorphic) to the quotient of $\coprod_{n<\omega} \mathcal{P}\mathcal{T}\mathcal{P}(n)$ modulo the the following relation:

$\lambda_i \in \mathcal{P}\mathcal{T}\mathcal{P}(n_i)$ and $\lambda_j \in \mathcal{P}\mathcal{T}\mathcal{P}(m_j)$ are equivalent if there exists valuations $c_i : F(n_i) \longrightarrow A$ and $c_j : F(m_j) \longrightarrow A$ such that

$$UL_{\mathcal{T}}(c_i)(\lambda_i) = UL_{\mathcal{T}}(c_j)(\lambda_j). \tag{3.21}$$

This finishes the description of $UL_{\mathcal{T}}(A)$.

Now we describe the action of $\delta_{\mathcal{T}}$. From the above characterization we have that each element of $UL_{\mathcal{T}}P(X)$ is of the form $UL_{\mathcal{T}}(c_\varphi)(\lambda)$ for some $c_\varphi : F(n_\varphi) \longrightarrow P(X)$ and some $\lambda \in \mathcal{P}\mathcal{T}\mathcal{P}(n_\varphi)$; we use $\varphi$ for the index because it corresponds to a list of subsets $\varphi : n \longrightarrow UP(X) = \mathcal{P}(X)$. Let us write $\lambda(\varphi)$ for $UL_{\mathcal{T}}(c_\varphi)(\lambda)$. From Diagram 5.6 we have

$$(\delta_{\mathcal{T}})_X(\lambda(\varphi)) = P\mathcal{T}(\widehat{c}_\varphi)(\lambda).$$

Now recall, from Definition 3.3.13, that $\widehat{c}_\varphi$ was obtained by applying the following chain of isomorphisms $\mathcal{A}(F(n_i), P(X)) \cong \mathsf{Set}(n_i, UPX) \cong \mathsf{Set}(n_i, \mathcal{Q}X) \cong \mathsf{Set}(X, \mathcal{P}(n_i))$ to $c_\varphi : F(n_\varphi) \longrightarrow P(X)$. In this particular case we have that $X \xrightarrow{\widehat{c}_\varphi = \chi_\varphi} \mathcal{P}(n_\varphi)$, where $\chi_\varphi$ is the exponential transpose of $\varphi : n_\varphi \longrightarrow \mathcal{P}(X)$. Gathering all this we have

$$\begin{aligned}
(\delta_{\mathcal{T}})_X(\lambda(\varphi)) &= P\mathcal{T}(\widehat{c}_\varphi)(\lambda) && \text{(Diagram 5.6)}\\
&= P\mathcal{T}(\chi_\varphi)(\lambda) && (\widehat{c}_\varphi = \chi_\varphi)\\
&= \{t \in \mathcal{T}(X) \mid \mathcal{T}(\chi_\varphi)(t) \in \lambda\} && \text{(Definition of } P)
\end{aligned}$$

Which is precisely the description of the predicate lifting associated with $\lambda$ (Equation (3.19)).

An important point to notice in this construction is that the logic is $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$ is a quotient of the logic of all predicate liftings, i.e. $(\bar{L}_{\mathcal{T}}, \bar{\delta}_{\mathcal{T}})$. More explicitely, $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$ is the quotient of $(\bar{L}_{\mathcal{T}}, \bar{\delta}_{\mathcal{T}})$ via the equivalence relation described in Equation 3.21. In the case of $\mathsf{BA}$ this logic is sound and complete.

**Proposition 3.3.14** ([76]). *The logic $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$, over* $\mathsf{BA}$*, is sound and complete.*

We refer the reader to [76] for a detailed proof.

# Chapter 4

# Comparing Coalgebraic Modal Logics

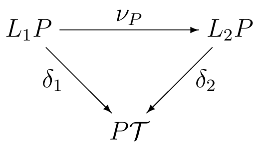
In this chapter we investigate the use of the functorial framework to find a translations between the Moss logic ($\mathcal{M}$) and the logic of all predicate liftings ($\mathcal{L}$). Our main result states that the Moss logic and the logic of all predicate liftings are equivalent, that is, can be translated into each other, in case that the following conditions are fulfiled: 1) the functor $\mathcal{T}$ preserves weak pullbacks, 2) the functor $\mathcal{T}$ preserves finite sets, and 3) the basic propositional logic is Boolean. Recall that the first condition is needed because otherwise Moss logic is not defined; Examples 4.3.1 and 4.4.6 explain the other conditions.

Let us emphasise that we are not interested in only showing only that every formula in $\mathcal{L}$ has an equivalent formula in $\mathcal{M}$ and viceversa. Rather we want an inductive definition of the translation, which respects the one-step nature (see Remarks 3.2.6 and 4.1.5). This stronger property of translations is captured by the existence of natural transformations $\bar{L} \to M$ and $M \to L$.

## 4.1 One step translations

We start by introducing translations between coalgebraic modal logics within the functorial framework. Recall that our notion of coalgebraic modal logic assumes a category $\mathcal{A}$ of power-set algebras, a functor $L : \mathcal{A} \to \mathcal{A}$ and a natural transformation $\delta : LP \to P\mathcal{T}$, as explained in the previous chapter (Definition 3.2.13). Also remember that a translation maps formulas in one language into formulas in another languages preserving the semantics. In the functorial framework for coalgebraic modal logics this intuition is captured by a natural transformation between the functors on the algebra side. The next definition makes this precise.

**Definition 4.1.1.** Let $(L_1, \delta_1)$ and $(L_2, \delta_2)$ be coalgebraic modal logics, over the same base category, for a functor $\mathcal{T}$. A *one-step translation*, written $\nu : (L_1, \delta_1) \to (L_2, \delta_2)$, is a natural transformation $\nu : L_1 \to L_2$ which commutes with the semantics, i.e. the following diagram




commutes. We say that $\nu$ *translates* the logic $(L_1, \delta_1)$ into the logic $(L_2, \delta_2)$.

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

**Example 4.1.2.** All the coalgebraic logics below are over $\mathsf{BA}$; we write $U : \mathsf{BA} \to \mathsf{Set}$ for the forgetful functor and $F$ for its left adjoint; a boolean algebra is denoted as a pair $(A, \alpha)$.

Let $\Box$ and $\Diamond$ be the predicate liftings associated with the universal modality and the existential modality, for $\mathsf{Pow}$, respectively (Example 3.1.3). Let $(L_{\{\Box,\Diamond\}}, \Box + \Diamond)$, $(L_\Box, \Box)$, and $(L_\Diamond, \Diamond)$ be logics of predicate liftings (Example 3.2.11) for the signatures as indicated. Let $(M, \nabla)$ be the Moss logic for $\mathsf{Pow}$ (Example 3.3.6)

1. The interdefinability of the modalities $\Box$ and $\Diamond$ over $\mathsf{BA}$ can be illustrated using one-step translations as follows. We define a one step translation $\nu : (L_\Box, \Box) \to (L_\Diamond, \Diamond)$ by presenting a natural transformation $\tau : U \to UFU = UL_\Diamond$ and then extend it freely to a natural transformation $L_\Box = FU \xrightarrow{\nu} FU = L_\Diamond$. The natural transformation $\tau$ maps $a \in U(A, \alpha) = A$ to $\neg_F \neg_A a \in UFU(A, \alpha)$, where $\neg_A$ is the negation of $(A, \alpha)$ and $\neg_F$ is the negation of $FU(A, \alpha)$. Using $\blacksquare$ and $\blacklozenge$ as formal symbols, see Examples 3.2.7 and 3.2.17, $\tau$ is just the usual translation i.e. $\tau(\blacksquare a) = \neg_F \blacklozenge \neg_A a$, the transpose of $\tau$ gives the one-step translation $\nu : (L_\Box, \Box) \to (L_\Diamond, \Diamond)$. It is standard to show that $\nu$ gives a one-step translation, i.e. it commutes with the semantics. We now compute this explicitly to illustrate the technique.

   Recall that the semantics of the logics are given by the transposes of the predicate liftings (Example 3.2.11), in this particular case they are given by $FUP = L_\Box P \xrightarrow{\widehat{\Box}} P\mathsf{Pow}$ and $FUP = L_\Diamond P \xrightarrow{\widehat{\Diamond}} P\mathsf{Pow}$. To show that that $\nu$ is a one-step translation we ought to argue that $\widehat{\Diamond} \circ \nu = \widehat{\Box}$. By properties of adjoints (Lemma A.1.3) to show this, it is enough to show $U(\widehat{\Diamond}) \circ \tau = \Box$. We can now compute, given $\varphi \in UP(X)$ we have:

$$
\begin{aligned}
U(\widehat{\Diamond}) \circ \tau(\varphi) &= U(\widehat{\Diamond}) \neg_F \neg_{P(X)} \varphi && \text{(Def. } \tau) \\
&= \neg_{P\mathsf{Pow}(X)} \Diamond \neg_{P(X)} \varphi && \text{(Def. } \widehat{\Diamond}) \\
&= \neg_{P\mathsf{Pow}(X)} \left\{ \psi \subseteq X \mid \psi \cap \neg_{P(X)} \varphi \neq \emptyset \right\} && \text{(Def. } \Diamond) \\
&= \left\{ \psi \subseteq X \mid \psi \cap \neg_{P(X)} \varphi = \emptyset \right\} && \text{(Def. } \neg_{P(X)}) \\
&= \{ \psi \subseteq X \mid \psi \subseteq \varphi \} && \text{(Proper. } \neg_{P(X)}) \\
&= \Box(\varphi). && \text{(Def. } \Box)
\end{aligned}
$$

It should now be clear how to define a one-step translation $\nu' : (L_\Diamond, \Diamond) \to (L_\Box, \Box)$ corresponding to $\Diamond = \neg\Box\neg$. By properties of BA, more specifically the fact the $\neg\neg = id$, it is easy to see that $\nu'$ is the inverse of the $\nu$ defined above; hence the two logics $(L_\Diamond, \Diamond)$ and $(L_\Box, \Diamond)$ are isomorphic.

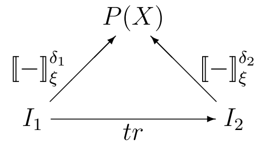
It is important to notice that to define a translation it is not enough to find a natural transformation, or isomorphism, between the functors; for example, over DL or Set the functors $L_\Box$ and $L_\Diamond$ are isomorphic but clearly there are no translations between these coalgebraic modal logics.

2. The usual translation of $\Diamond$ into the Moss logic can be presented as follows: In this particular case, we can define a one-step translation $\nu_\Diamond : FU \to F\mathsf{Pow}_\omega U$ by just presenting a natural transformation $\tau_\Diamond : U \to \mathsf{Pow}_\omega U$ and then applying $F$ to it. We define $\tau_\Diamond$ as follows: an element $a \in U(A, \alpha)$ is mapped to $\tau_\Diamond(a) = \{a, \top_A\}$, where $\top_A$ is the top element of $(A, \alpha)$. This correspond to the translation $\Diamond\varphi = \nabla\{\varphi, \top\}$. The fact that we can just apply $F$ is particular to $\Diamond$; see Section 4.2 for more on this.

3. The usual translation of $\Box$ into Moss logic is given by a natural transformation $\nu_\Box : FU \to F\mathsf{Pow}_\omega U$. In this case, we define the translation by presenting a natural transformation $\tau_\Box : U \to UF\mathsf{Pow}_\omega U$. An element $a \in U(A, \alpha)$ is mapped to $\tau_\Box(a) = \{a\} \vee_F \bot_A$, where $\vee_F$ is the disjunction of $F\mathsf{Pow}_\omega U(A, \alpha)$ and $\bot_A$ is the bottom element of $\mathsf{Pow}_\omega U(A, \alpha)$; this corresponds to the usual translation $\Box a = \nabla\{a\} \vee \nabla\emptyset$.

4. The well known translation of $\nabla$ into basic modal logic is done as follows: we want to define a natural transformation $\nu_\nabla : F\mathsf{Pow}_\omega U \to F(U_\Box + U_\Diamond)$, here we write $U_\Box + U_\Diamond$ to indicate that one factor deals with $\Box$ and the other with $\Diamond$. One more time, using properties of free algebras it is enough to define a natural transformation $\tau : \mathsf{Pow}_\omega U \to UF(U_\Box + U_\Diamond)$. Let $\Phi$ be an element in $\mathsf{Pow}_\omega U(A, \alpha)$. Since $\Phi$ is finite there are elements in $U_\Box(A, \alpha)$ and $FU_\Diamond(A, \alpha)$ corresponding to $\bigvee_A \Phi$ and $\bigwedge_F \Phi$ respectively. We now define $\tau$ as expected, i.e. $\tau(\Phi) = \bigvee_A \Phi \wedge \bigwedge_F \Phi$. On the one hand, since $\bigwedge_F$ is a conjunction of elements in $FU_\Diamond(A, \alpha)$ it correspond to a conjunction of elements $\Diamond a$, $a \in \Phi$. On the other hand, $\bigvee_A$ is a conjunctions of elements in $U_\Box(A, \alpha)$ and is itself an element of $FU_\Box(A, \alpha)$ hence it corresponds to $\Box \bigvee_{a\in\Phi} a$. In summary, the translation above corresponds to the usual translation $\nabla\Phi = \Box \bigvee\limits_{a\in\Phi} a \wedge \bigwedge\limits_{a\in\Phi} \Diamond a$.

We now explain how one-step translations can be understood as inductive definitions of translations between the associated languages. More explicitly, we will show that given a one-step translation $\nu : (L_1, \delta_1) \to (L_2, \delta_2)$ we can define a function $tr : I_1 \to I_2$ such that for every coalgebra $\xi : X \to \mathcal{T}(X)$ the following diagram




(4.1)

commutes, where $I_i$ is the carrier of the initial $L_i$ algebra. In other words, a function $tr$ such that for each formula $\varphi \in I_1$ we have $[\![\varphi]\!]_\xi^{\delta_1} = [\![tr(\varphi)]\!]_\xi^{\delta_2}$.

Recall (Remark 3.2.2) that the elements of $I_i$ are not "formulas" in the standard sense i.e. elements of the term algebra for a signature, but equivalence classes of formulas. The function $tr$ translates those equivalence classes. It is not absolutely trivial to obtain a translation, in the sense of Definition 2.4.1, between formulas. Before going into more details we explain how to define the function $tr : I_1 \longrightarrow I_2$ and explain why the diagram above commutes.

Before defining $tr$ we notice that a one-step translation induces a functor between the corresponding algebras. The next definition makes this precise.

**Definition 4.1.3.** Let $\nu : (L_1, \delta_1) \longrightarrow (L_2, \delta_2)$ be a one step translation. The *translation functor*, induced by $\nu$ is given by

$$- \circ \nu =: Tr_\nu : \mathsf{Alg}(L_2) \longrightarrow \mathsf{Alg}(L_1).$$

More explicitly, an $L_2$-algebra, $\alpha : L_2(A) \longrightarrow A$, is mapped to the following composite $L_1(A) \xrightarrow{\nu_A} L_2(A) \xrightarrow{\alpha} A$, i.e. $Tr_\nu(A, \alpha) = (A, \alpha \circ \nu_A)$; the functor $Tr_\nu$ is the identity on arrows.

Notice that since $\nu$ is a natural transformation the definition above indeed defines a functor, because any morphism $f : A \longrightarrow A'$, which is a morphism of $L_2$-algebras is also a morphism between the corresponding $L_1$-algebras.

We now show how to define $tr : I_1 \longrightarrow I_2$ in Diagram 4.1. Denote by $\iota_i : L_i(I_i) \longrightarrow I_i$ the initial $L_i$-algebra, $(i = 1, 2)$. The function $tr : I_1 \longrightarrow I_2$ is given by the initial morphism of $L_1$-algebras $(I_1, \iota_1) \longrightarrow Tr_\nu(I_2, \iota_2)$. Indeed, from the definition of $Tr_\nu$, this morphism is given by a function $tr : I_1 \longrightarrow I_2$.

Now notice that since $Tr_\nu$ is the identity on arrows, the function $[\![-]\!]_\xi^{\delta_2} : I_2 \longrightarrow P(X)$ is a homomorphism between the $L_1$-algebras $Tr_\nu(I_2, \delta_2)$ and $Tr_\nu \widehat{P}_{\delta_2}(X, \xi)$. This means that the semantics for $\delta_2$ is a morphism of $L_1$-algebras.

Now we show how to obtain Diagram 4.1, i.e. $[\![-]\!]_\xi^{\delta_1} = [\![-]\!]_\xi^{\delta_2} \circ tr$. First we show that $Tr_\nu \widehat{P}_{\delta_2}(X, \xi)$ is the complex $(L_1, \delta_1)$-algebra of $(X, \xi)$. To show this we will use the fact that $\nu$ commutes with the semantics, i.e. $\delta_2 \circ \nu_P = \delta_1$. With this in mind we obtain

$$\begin{aligned}
Tr_\nu \widehat{P}_{\delta_2}(X,\xi) &= Tr_\nu(P(X), P(\xi)\circ\delta_2) && (\text{Def. } \widehat{P}_{\delta_2})\\
&= (P(X), P(\xi)\circ\delta_2\circ\nu_{P(X)}) && (\text{Def. } Tr_\nu)\\
&= Tr_\nu(P(X), P(\xi)\circ\delta_1) && (\delta_2\circ\nu_P=\delta_1)\\
&= \widehat{P}_{\delta_1}(X,\xi). && (\text{Def. } \widehat{P}_{\delta_1})
\end{aligned}$$

As we wanted to show. In particular, this implies that the carrier set of $Tr_\nu \widehat{P}_{\delta_2}(X,\xi)$ is $P(X)$ and that the initial morphism is given by $[\![-]\!]^{\delta_1}_\xi : I_1 \longrightarrow P(X)$. From this, since $tr : I_1 \longrightarrow I_2$ is a morphism of $L_1$-algebras and so is $[\![-]\!]^{\delta_2}_\xi : I_2 \longrightarrow P(X)$, by initiality, we conclude that for any coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ we have $[\![-]\!]^{\delta_1}_\xi = [\![-]\!]^{\delta_2}_\xi \circ tr$. In other words, $tr$ is a translation.

A gain of the functorial approach is that $tr$ is a morphism in the base category of the logics, which means that it preserves the basic structure, i.e the basic propositional logic. In other words, the translation is inductively defined over the operations of the basic structure. For example, if the basic propositional logic is that of boolean algebras and we already know how to translate formulas $\varphi$ the formulas like $\neg\varphi$ are translated by $tr(\neg\varphi) = \neg tr(\varphi)$.

We now address the issue that $tr$ is a function between equivalence classes of formulas and not between formulas. First we make the problem precise. At this point, a first concern is that it is not clear how to obtain a signature from a coalgebraic modal logic (Remark 3.2.2). We can go around this because in Chapter 5 we will see that every coalgebraic logic can be seen as a rank 1 axiomatization of a logic of predicate liftings. More precisely, for every coalgebraic modal logic $(L,\delta)$ there exists a set of predicate liftings $\Lambda$ and a surjection $q : L_\Lambda \twoheadrightarrow L$ such that $\bar{\delta} = \delta \circ q$, where $\bar{\delta}$ is the semantics of $L_\Lambda$. Under these assumptions the problem of defining a translation between formulas can be made precise as follows: First recall (Remark 3.2.15) that $\mathcal{L}^{\mathcal{A}}_\Lambda$ denotes the language of predicate liftings over $\mathcal{A}$,

> Given coalgebraic modal logics $(L_1,\delta_1)$ and $(L_2,\delta_2)$, over a category $\mathcal{A}$, which are rank 1 axiomatizations of logic of predicate liftings $(L_{\Lambda_1},\bar{\delta}_1)$ and $(L_{\Lambda_2},\bar{\delta}_2)$, respectively. Assume there is a one step translation $\nu : (L_1,\delta_1) \longrightarrow (L_2,\delta_2)$. Can we always obtain a usual translation (Definition 2.4.1), i.e. a function $tr' : \mathcal{L}^{\mathcal{A}}_{\Lambda_1} \longrightarrow \mathcal{L}^{\mathcal{A}}_{\Lambda_2}$ which preserves the interpretation of formulas?

The short answer is NO; in general we can not obtain a usual translation from a one-step translation. We now explain this. The idea is that $tr'$ should make the following diagram commute

$$\begin{array}{ccc} \mathcal{L}^{\mathcal{A}}_{\Lambda_1} & \overset{tr'}{\dashrightarrow} & \mathcal{L}^{\mathcal{A}}_{\Lambda_2} \\ \downarrow & & \downarrow \\ I_1 & \underset{tr}{\longrightarrow} & I_2 \end{array} \tag{4.2}$$

where the lower horizontal edge is the translation described in Diagram 4.1 and the vertical arrows are the respective quotient maps. By following the lower edge, we clearly see that there is always a function $\bar{tr} : \mathcal{L}^{\mathcal{A}}_{\Lambda_1} \longrightarrow I_2$. However, to define $tr' : \mathcal{L}^{\mathcal{A}}_{\Lambda_1} \longrightarrow \mathcal{L}^{\mathcal{A}}_{\Lambda_2}$ amounts to make a choice of representants in the equivalence classes (formulas) in $I_2$; moreover such choice should be somehow inductively defined on the complexity of the formula. The problem is that there is no general canonical method to make such choice. The following examples illustrate this.

Consider the case of $\Box$ and $\Diamond$ for $\mathsf{Pow}$. It is well known that one translation is $\Box = \neg\Diamond\neg$, however we could also translate $\Box$ via $\neg\neg\neg\Diamond\neg$ or $\neg\Diamond\neg\neg\neg$. The question is how can we guarantee to always choose the first translation? Here we could argue that the first translation is preferred because it is shorter. In other words, we can use length to make a canonical choice. This is not always possible, consider the signature where we add a new unary operator $\heartsuit$ to the boolean signature, call this signature $\Sigma_{\mathsf{BA}^{\heartsuit}}$. Boolean algebras can be axiomatized with this signature putting the axiom $\heartsuit = \neg\neg\neg$ on top of the usual axiomatization. The one step-translation in Example 4.1.2 still works, but now to define a translation $tr' : \mathcal{L}^{\mathsf{BA}^{\heartsuit}}_{\Box} \longrightarrow \mathcal{L}^{\mathsf{BA}^{\heartsuit}}_{\Diamond}$ in addition to the choices above we could also translate $\Box$ via $\heartsuit\Diamond\neg$ or $\neg\Diamond\heartsuit$ and the length argument does not work anymore. If we insists on making a choice based on length, we should then try using a minimal functional complete sets of operators and then present $\mathsf{BA}$ by a single connective like NAND($\uparrow$)[1]. In conclusion there is no general canonical manner to define $tr' : \mathcal{L}^{\mathcal{A}}_{\Lambda_1} \longrightarrow \mathcal{L}^{\mathcal{A}}_{\Lambda_2}$. It is important to notice that previous examples also show that the problem of choosing a particular presentation for a translation is also present in the classical approach to modal logic. Here the issue is addressed by making a "smart" choice of the signature before hand. The choice of an appropriate signature hovers over the realms of sociology of mathematics; we do not go into such discussion in this manuscript.

Until here we have addressed the issue of obtaining usual translations (Definition 2.4.1) from one-step translations (Definition 4.1.1). We argued that to define a translation there is a choice problem that can not be solved within the functorial framework but neither it is solvable in the standard approach. We now proceed to mention some advantages of one-step transaltions and the functorial approach.

1. One-step translations are independent of the chosen base signature. One-step translations use its own "weakness" into its favour, they avoid the need of

[1]The NAND connective is the negation of AND, i.e. $\uparrow (p, q) = \neg(p \wedge q)$. It is well known that all other Boolean connectives can be expressed using only $NAND$, see [35] for details.

making any choice at all. For example, in the case of translating $\Box$ into $\Diamond$ there is no need to choose between $\neg\Diamond\neg$, $\neg\neg\neg\Diamond\neg$, or $\neg\Diamond\heartsuit$ because all of them are in the same equivalence class, hence they are one formula from the functorial perspective.

2. One-step translation are canonical in the following sense: If a coalgebraic modal logic $(L, \delta)$ is (one-step) complete, then for any other coalgebraic modal logic $(L', \delta')$ there is at most one one-step translation $\nu : (L', \delta') \longrightarrow (L, \delta)$.

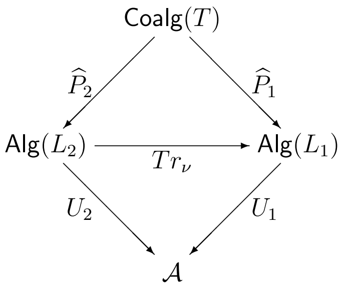
3. We can use structural properties of the base category to produce transaltions. This will be illustrated in Section 4.4 where we show how, using Stone duality, every predicate lifting can be translated into Moss logic.

4. We can use the categorical description of the functor $L$ to define generic translations. We will illustrate this in Chapter 5 where we use the presentation of the language of all (finitary) predicate liftings $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$, see Definition 3.3.13, to show that every other coalgebraic modal logic is translatable into $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$.

5. Assuming a signature has been given, the functorial approach allows us to reduce the choice for the translation to the simplest depth-one formulas of $(L_{\Lambda_1}, \bar{\delta_1})$. Recall that the translation $tr : I_1 \longrightarrow I_2$ in Diagram 4.1 is a morphism in the base category hence to define a translation $tr' : \mathcal{L}^{\mathcal{A}}_{\Lambda_1} \longrightarrow \mathcal{L}^{\mathcal{A}}_{\Lambda_2}$ as indicated in Diagram 4.2, it is enough to choose a representant for $tr(\Box_\lambda x)$ and then extend inductively to define $tr'$.

The following proposition summarises what is needed to define the function $tr : I_1 \longrightarrow I_2$ in Diagram 4.1.

**Proposition 4.1.4.** *Let $\nu : (L_1, \delta_1) \longrightarrow (L_2, \delta_2)$ be a one step translation. The translation functor, $Tr_\nu : Alg(L_2) \longrightarrow Alg(L_1)$ makes the following diagrams*




*commute, where $\widehat{P_i}$ is the $\delta_i$-lifting of $P$ as described in Definition 3.2.4.*

*Proof.* The details can be found in the discussion after Definition 4.1.3. We just recall that the lower triangle commutes because $\nu$ is natural and that the upper triangle commutes because $\nu_P$ commutes with $\delta_1$ and $\delta_2$. $\square$

In the previous proposition, the lower triangle is used to define the translation, i.e. the function $tr : I_1 \to I_2$. The upper triangle is used to show that this translation preserves the interpretation of formulas i.e. Diagram 4.1 commutes.

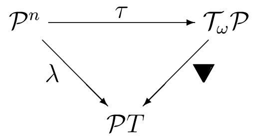
The next remark discusses why we do not follow a more standard categorical approach using free monads.

**Remark 4.1.5.** Another idea that might come to mind is to define translations between coalgebraic modal logics using the free monads generated by $L_1$ and $L_2$. More concretely, the idea would be to define translations as natural transformations, monad morphism, between those. Such a more general notion would allow, for example, to express an $L_1$-formula $\Box_1\varphi_1$ as a combination of $L_2$-formulas with nested modal operators such as e.g. $\Box_2\Diamond_2\varphi_2$. Moreover, it is well known that any functor making the lower triangle, in the previous proposition, commute is induced by such type of natural transformation. However, as we will later see, all those "advantages" would still not allow us to always find a translation. Examples 4.3.1 and 4.4.6 show that translations simply do not exist. On top of this, our stronger notion preserves the one step nature of the modalities, hence the name.

## 4.2 Decomposing predicate liftings

As we illustrated in the previous chapter, the Moss modality can be technically involved. For this reason, we first show how to translate predicate liftings. Our long term aim is to find a one-step translation $(\bar{L}_{\mathcal{T}}, \delta_{\mathcal{T}}) \to (M_{\mathcal{T}}, \nabla)$. This can be simplified by considering one predicate lifting at the time. In order to tailor the desired translations, we will first introduce the concept of translators for a predicate lifting (Definition 4.2.1). Technically speaking, a translator factors a predicate lifting $\lambda$ via $\blacktriangledown$. Unfortunately, not all predicate liftings have translators (Example 4.2.4). We can overcome this by showing that all the so-called singleton liftings (Definition 4.2.5) do have translators and in fact every predicate lifting is a union of singleton liftings (Proposition 4.2.8).

**Definition 4.2.1.** A *translator* for an $n$-ary predicate lifting $\lambda$ is a natural transformation $\tau : \mathcal{P}^n \to \mathcal{T}_\omega\mathcal{P}$ such that the following diagram




(4.3)

commutes, where $\blacktriangledown$ is the semantics of the Moss Logic (Definition 3.3.4).

We illustrate the concept with some examples.

**Example 4.2.2.** The following are examples of translators.

1. Consider the predicate lifting associated with the existential modality $\Diamond$ of the covariant power set functor (Example 3.1.3). The following natural transformation is a translator for $\Diamond$; we define $\tau_X : \mathcal{P}(X) \longrightarrow \mathsf{Pow}_\omega \mathcal{P}(X)$ mapping an element $\varphi \subseteq X$ to $\tau_X(\varphi) = \{\varphi, X\}$. Compare this with the equivalence $\Diamond\varphi = \nabla\{\varphi, \top\}$ discussed in Examples 3.3.6 and 4.1.2.

2. Consider the usual probability modality $\Diamond_p$, i.e. *"the probability of* $\varphi$ *is at least* $p$*"*. This predicate lifting has a translator $\tau_p : \mathcal{P} \longrightarrow \mathcal{D}\mathcal{P}$ defined as follows: A set $\varphi \subseteq X$ is mapped to to the probability distribution $D^\varphi_p : \mathcal{P}(X) \longrightarrow [0,1]$ which assigns $p$ to the set $\varphi$ and $1-p$ to the set $X$. Compare this with the description in Example 3.3.6.

3. We can use the same idea of the previous item to translate the probability modality $\Diamond^p$, i.e. *"the probability of* $\varphi$ *is at most* $p$*"* (Example 3.1.3). The translator is given by the natural transformation $\tau^p : \mathcal{P} \longrightarrow \mathcal{D}\mathcal{P}$ which maps a set $\varphi \subseteq X$ to the probability distribution, $D^p_\varphi : \mathcal{P}(X) \longrightarrow [0,1]$, assigning $1-p$ to the set $\neg\varphi$ and $p$ to the set $X$, is a translator for $\Diamond^p$.

The next remark presents translators in terms of relation lifting.

**Remark 4.2.3.** Using relation lifting we can describe translators as follows: a natural transformation $\tau : \mathcal{P}^n \longrightarrow \mathcal{T}_\omega\mathcal{P}$ is a translator for a predicate lifting $\lambda : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ iff for every $\varphi : n \longrightarrow \mathcal{P}(X)$ and every $t \in \mathcal{T}(X)$ the following holds

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

where $\overline{\mathcal{T}}$ is as in Definition 2.2.7.

Now we explain the intuition behind translators. The idea of a translator is to define a translation $tr$ via

$$tr(\Box_\lambda\varphi) = \nabla\tau\big(tr(\varphi)\big). \tag{4.4}$$

The key issue is to somehow show that $\tau(tr(\varphi)$ is definable in the base logic. We come back to this in Section 4.3. A more immediate concern is that not all predicate liftings have translators. This can be rephrased by saying that not all predicate liftings can be translated using only $\nabla$ without propositional connectives. The following example illustrates this.

**Example 4.2.4.** The following predicate liftings fail to have translators.

1. Let $K_C$ be a constant functor where $C$ has at least two distinct elements $c_1$ and $c_2$. Using Proposition 3.3.9 (see also Example 3.1.3), predicate liftings correspond to subsets of $C$. The predicate lifting $\lambda_E$ corresponding to $E = \{c_1, c_2\}$ does not have a translator. This is because the components of a natural transformation $\tau : \mathcal{P} \longrightarrow \mathsf{K}_C$ ought to be constant functions, hence the cardinality of $\blacktriangledown\tau(X)$ is always 1, but $\lambda_E(X) = E$. Nevertheless, notice that the formula $\nabla c_1 \vee \nabla c_2$ translates the associated modality, i.e. $\Box_E$.

2. Consider de graded modality $\Diamond_k$ for the finite multiset functor, i.e. *there are at least $k$ successors satisfying $\varphi$.* Recall from Example 3.3.6 that each $\nabla$ formula for $\mathcal{B}_{\mathbb{N}}$ specifies the total number of successors. Since $\Diamond_k$ does not declare a specific number of successors, we conclude that $\Diamond_k$ can not have a translator.

3. Let $\Diamond_{>p}$ be a modality for the finite distribution functor corresponding to the set $(p, 1]$; in natural language this modality says *the probability of $\varphi$ is strictly larger than $p$.* In terms of natural transformations this modality maps a set $\varphi \subseteq X$ to $\Diamond_{>p}(\varphi) = \{d \in \mathcal{D}(X) \mid \mu_d(\varphi) > p\}$, where $\mu_d(\varphi) = \sum_{x\in\varphi} d(x)$.

   Each of these modalities fail to have a translator. The reason for this is that each natural transformation $\tau : \mathcal{P} \longrightarrow \mathcal{DP}$ specifies a probability for each set $\varphi$, as an element of $\mathcal{P}(X)$, say $q$; since we want $\tau$ to be a translator we must have $p < q$. Consequently, as seen in Example 3.3.6, $\nabla\tau(\varphi)$, which is an element of $\mathcal{PD}(X)$, can only contain probability distributions $d : X \longrightarrow [0, 1]$ such that $\sum_{x\in\varphi} d(x) = q$. Hence no single natural transformation can factor $\Diamond_{>p}$ via $\blacktriangledown$; in our terminology, this means that no predicate liftings corresponding to a modality $\Diamond_{>p}$ has a translator. Notice that this argument does not work for $\Diamond_p$ because there we can pick $q = p$; in fact that is how we define the translator $\tau_p$ there.

   In particular the predicate lifting corresponding to the modality $\Box_p$, the dual to $\Diamond_p$ in Example 3.1.3, does not have a translator because it corresponds to the set $(1-p, 1]$. Nevertheless, notice $\Box_p$ can be translated into Moss language using negations because $\Diamond_p$ is translatable, see the previous example.

### 4.2.1 Singleton Liftings

Translators are one of our primary means to define transaltions from languages of predicate liftings into the Moss logic. Before explaining how this is done we address possible concerns risen from the previous example by showing that for every functor there are enough predicate liftings that do have translators. For this purpose, we now introduce singleton liftings. Informally speaking, singleton liftings are the simplest predicate liftings. As the name indicates, they are associated, via Proposition 3.3.9, with singleton sets. Here is the formal definition.

**Definition 4.2.5.** **([80])** An $n$-ary predicate lifting $\lambda$ is called a *singleton predicate lifting*, or a *singleton lifting* for short, if it is associated (via Proposition 3.3.9) with a singleton set in $\mathcal{PTP}(n)$. More explicitly, a predicate lifting $\lambda : \mathcal{P}^n \longrightarrow \mathcal{PT}$ is a singleton lifting if there exists $p \in \mathcal{TP}(n)$ such that for every $\varphi : n \longrightarrow \mathcal{P}(X)$ the following holds

$$\lambda_X(\varphi) = \{t \in \mathcal{T}(X) \mid \mathcal{T}(\chi_\varphi)(t) = p\}, \tag{4.5}$$

where $\chi_\varphi : X \longrightarrow \mathcal{P}(n)$ is the transpose of $\varphi$. If $\lambda$ is a singleton lifting, we write it $\lambda_p$ or just $p$, where $p$ is the associated element of $\mathcal{TP}(n)$.

Here are some examples of singleton liftings and the corresponding translators.

**Example 4.2.6.** In the following $2 = \{\top, \bot\}$.

1. If $\mathcal{T}$ is a constant functor with value $C$, then the singleton liftings for $\mathcal{T}$ are associated with elements $c \in C = \mathcal{T}(2)$. The $X$-component of a singleton lifting $\lambda_c$ is the function $\lambda_c : \mathcal{P}X \longrightarrow \mathcal{P}K_C$ with constant value $\{c\}$. A translator for $\lambda_c$ is given by the natural transformation $\tau : \mathcal{P} \longrightarrow \mathsf{K}_C\mathcal{P}$ whose components are functions with constant value $c$.

2. If $\mathcal{T}$ is the identity functor we have $\mathcal{T}(2) = 2 = \{\top, \bot\}$, then there are two singleton liftings of arity 1 for $Id$. The $X$-component of $\lambda_\top$ is the identity. Similarly, the $X$-component of $\lambda_\bot$ is the function $(\lambda_\bot)_X : \mathcal{P}(X) \longrightarrow \mathcal{P}(X)$ mapping a set $\varphi \subseteq X$ to to its complement i.e. $\lambda_\bot(\varphi) = \neg_X\varphi$. In this case $\blacktriangledown : \mathsf{Id}\mathcal{P} \longrightarrow \mathcal{P}\mathsf{Id}$ is the identity, using this it is easy to see that $id : \mathcal{P} \longrightarrow \mathsf{Id}\mathcal{P}$ is a translator for $\lambda_\top$ and $\neg : \mathcal{P} \longrightarrow \mathsf{Id}\mathcal{P}$, complement, is a translator for $\lambda_\bot$.

3. Let $\mathcal{T} = 1 + \mathsf{Id}$, we have $\mathcal{T}(2) = 1 + 2$. Consider the set $\{*\} \subseteq 1 + 2$, where $* \in 1$. The associated singleton lifting $\lambda_* : \mathcal{P} \longrightarrow \mathcal{P}(1 + \mathsf{Id})$ maps a set $\varphi \subseteq X$ to $\{*\}$. This modality indicates termination, i.e. $x \Vdash_\xi \lambda_*\varphi$ iff a transition from $x$ leads the system to halt. The natural transformation $\tau_* : \mathcal{P} \longrightarrow 1 + \mathcal{P}$ with constant value $*$ is a translator for $\lambda_*$. The other singleton liftings for $\mathcal{T}$ are similar to those of $\mathsf{Id}$.

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

$$\begin{aligned}
\lambda_\emptyset(\varphi) &= \{\emptyset\};\\
\lambda_{\{\top\}}(\varphi) &= \{U \in \mathsf{Pow}(X) \mid \emptyset \neq U \subseteq \varphi\};\\
\lambda_{\{\bot\}}(\varphi) &= \{U \in \mathsf{Pow}(X) \mid \emptyset \neq U \subseteq \neg_X\varphi\};\\
\lambda_{\{\top,\bot\}}(\varphi) &= \{U \in \mathsf{Pow}(X) \mid U \cap \neg_X\varphi \neq \emptyset \neq U \cap \varphi\};
\end{aligned}$$

   Note that they all have translators, corresponding to $\nabla\{\varphi\}$, $\nabla\{\neg_X\varphi\}$, $\nabla\emptyset$, $\nabla\{\varphi, \neg_X\varphi\}$, respectively. We work out the case of $\lambda_{\{\top,\bot\}}$ in more detail to illustrate what we mean with "corresponding to". The natural transformation $\tau : \mathcal{P} \longrightarrow \mathsf{Pow}\mathcal{P}$ which maps $\varphi \in \mathcal{P}(X)$ to the set $\{\varphi, \neg_X\varphi\} \in \mathsf{Pow}\mathcal{P}(X)$ is a translator for $\lambda_{\{\top,\bot\}}$.

5. If $\mathcal{T}$ is the finite multiset functor we have $\mathcal{T}(2) = \mathbb{N}^2$. This means that a singleton lifting is given by a pair of natural numbers $(n, m)$, we write $f^n_m : 2 \longrightarrow X$ for the function which maps $\top$ to $n$ and $\bot$ to $m$. we use $(n, m)$ to denote the associates predicate lifting. We now compute the predicate lifting associated with $(n, m)$ explicitly. From Equation (4.5) we know that the $X$-component, $(n, m)_X : \mathcal{P}(X) \longrightarrow \mathcal{P}\mathcal{B}_{\mathbb{N}}(X)$, maps a set $\varphi \subseteq X$ to

$$\begin{aligned}(n, m)_X(\varphi) &= \{B : X \longrightarrow \mathbb{N} \mid \mathcal{B}_{\mathbb{N}}(\chi_\varphi) = f^n_m\} \\ &= \left\{ b : X \longrightarrow \mathbb{N} \mid \sum_{x \in \chi_\varphi^{-1}(\top)} b(x) = n \text{ and } \sum_{x \in \chi_\varphi^{-1}(\bot)} b(x) = m \right\} \\ &= \left\{ b : X \longrightarrow \mathbb{N} \mid \sum_{x \in \varphi} b(x) = n \text{ and } \sum_{x \in \neg_X \varphi} b(x) = m \right\}\end{aligned}$$

In words, $(n, m)_X(\varphi)$ is the set of bags over $X$ with $n + m$ elements, $n$ of which are in $\varphi$ and $m$ are in the complement of $\varphi$. Such a predicate lifting has a translator as it corresponds to $\nabla\{(\varphi, n), (\neg\varphi, m)\}$, see Example 3.3.6. More explicitly this means that the natural transformation $\tau^n_m : \mathcal{P} \longrightarrow \mathcal{B}_{\mathbb{N}}\mathcal{P}$ which maps a set $\varphi \subseteq X$ to the bag $B : \mathcal{P}(X) \longrightarrow \mathbb{N}$ assigning $n$ to $\varphi$, $m$ to $\neg_X\varphi$, and 0 to any other set, is a translator for $(n, m)$.

6. If $\mathcal{T}$ is the finite distribution functor, a singleton lifting is given by a probability distribution $d : 2 \longrightarrow [0, 1]$. Since we require $d(\top) + d(\bot) = 1$, a singleton lifting for the finite distribution is then determined by a single real number $q \in [0, 1]$. The $X$-component of $q$ maps a set $\varphi \subseteq X$ to the set of probability distributions over $X$ that assign probability $q$ to the set $\varphi$. As in the case of the multiset functor, such predicate liftings have translators corresponding to $\nabla\{(\varphi, q), (\neg_X\varphi, 1 - q)\}$, see Example 3.3.6; compare this formula with the one in the mentioned example.

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

**Definition 4.2.7.** The set of, finitary, singleton liftings is denoted by $\Lambda_s$; we write $(\bar{L}_s, \bar{\delta}_s)$ for the corresponding coalgebraic modal logic (Definition 3.2.18), over a category of power set algebras $\mathcal{A}$.

Singleton liftings appeared for the first time in [80]. Some reasons to consider singleton liftings are:

1. They always have translators, see Theorem 4.2.10.

2. In the case of KPF's they can be inductively presented over the complexity of the functor, see Section 4.2.2; hence,

3. their translators, and eventual translations, can also be presented inductively over the complexity of the functor.

4. They generate all other predicate liftings, see Proposition 4.2.8.

The items 2 and 3 will be developed in Section 4.2.2. The next proposition presents the last item more formally.

**Proposition 4.2.8** ([80]). *If $\lambda$ is an $n$-ary predicate lifting associated with a set $P \subseteq \mathcal{TP}(n)$, then for every set $X$ and every $n$-sequence $\varphi : n \longrightarrow \mathcal{P}(X)$ we have:*

$$\lambda_X(\varphi) = \bigcup_{p \in P} (\lambda_p)_X(\varphi).$$

*In other words, every $n$-ary predicate lifting can be obtained as a (possibly infinite) join of singleton predicate liftings.*

*Proof.* The proof is an application of Proposition 3.3.9. Recall the description of a predicate lifting given by Equation 3.19, on page 62. Recall the instantiation of this same equation for a singleton lifting (Definition 4.2.5). Using those we can show that the action of $\lambda$, over a $n$-sequence $\varphi : n \longrightarrow \mathcal{P}X$, can be described as follows

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

□

The next example illustrates how the previous proposition can be used to define translations.

**Example 4.2.9.** Let $\Box$ and $\Diamond$ be the predicate liftings associated with the universal modality and the existential modality for Pow, Example 3.1.3.

1. In the case of the universal modality, we saw that the predicate lifting for $\Box$ is $\lambda_{\{\emptyset,\{\top\}\}}$, see Example 3.3.12. This predicate lifting does not have a translator. From the previous proposition we know it is the union of singleton liftings; more concretely $\Box = \lambda_\emptyset \cup \lambda_{\{\top\}}$. In Example 4.2.6 we saw that these singleton liftings have translators. We can then recover the usual translation via

$$tr(\Box\varphi) = tr(\Box_{\lambda_\emptyset}\varphi) \vee tr(\Box_{\lambda_{\{\top\}}}\varphi) = \nabla\emptyset \vee \nabla\{\varphi\}.$$

2. In the case of the existential modality, the predicate lifting for $\Diamond$ corresponds to $\lambda_{\{\{\top,\bot\},\{\top\}\}} = \lambda_{\{\top,\bot\}} \cup \lambda_{\{\top\}}$. Incidentally, $\Diamond$ does have a translator, see Example 4.2.2, which induces the usual translation $tr(\Diamond\varphi) = \nabla\{\top,\varphi\}$. However, we could also translate $\Diamond$ using the translators for $\lambda_{\{\top,\bot\}}$ and $\lambda_{\{\top\}}$; in such perspective we have

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

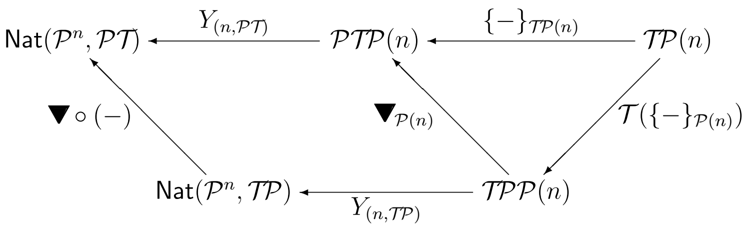
It can be checked, by long direct computations, that this is indeed equivalent to the usual translation.

The starting point for the enterprise of comparing coalgebraic logic in the functorial framework was the discovery that singleton liftings always have translators. The next theorem is the main result of this section and states this fact precisely.

**Theorem 4.2.10.** *Let $\mathcal{T}$ be a weak pullback preserving functor. Every (finitary) singleton lifting $\lambda_p$ for $\mathcal{T}$ has a translator.*

*More explicitly, the translator is associated with $\mathcal{T}(\{-\}_{\mathcal{P}(n)})(p) \in \mathcal{TPP}(n)$, where $\{-\}_X : X \longrightarrow \mathcal{P}(X)$ maps an element $x \in X$ to its singleton*[2]*, and elements in $\mathcal{TPP}(n)$ can be identified with natural transformation $\mathcal{P}^n \longrightarrow \mathcal{TP}$ because of Proposition 3.3.9.*

*Proof.* Consider the following diagram




In the diagram, $Y$ denotes the isomorphism given in Proposition 3.3.9. Recall that since $\mathcal{T}$ preserves weak pullbacks $\blacktriangledown$ is natural (Remark 3.3.5).

First we discuss the commutativity of the diagram. The parallelogram on the left commutes by Remark 3.3.10, page 62. Since $\mathcal{T}$ preserves weak pullbacks, the triangle on the right commutes by Remark 3.3.8, page 61.

The commutativity of the diagram implies that the natural transformation associated with $\mathcal{T}(\{-\}_{\mathcal{P}(n)})(p)$ is a translator for $\lambda_p$. We now explain why this is the case. Call $\tau_p : \mathcal{P}^n \longrightarrow \mathcal{TP}$ the natural transformation corresponding to $Y_{(n,\mathcal{TP})}(\mathcal{T}(\{-\}_{\mathcal{P}(n)})(p))$. An element $p \in \mathcal{TP}(n)$ is mapped by the lower edge of the diagram to $\tau_p \circ \blacktriangledown$ whereas the upper edge maps it to $\lambda_p$. Since the diagram commutes we have $\lambda_p = \blacktriangledown \circ \tau_p$ this means that $\tau_p$ is a translator fro $\lambda_p$ as we wanted to show. □

[2]The functions $\{-\}_X : X \longrightarrow \mathcal{P}(X)$ do not form a natural transformation. For this reason we use the bracket notation instead of $\eta$ which is usually used to indicate the unit of $\mathsf{Pow}$.

**Remark 4.2.11.** In the previous theorem we used $\mathcal{T}$ instead of $\mathcal{T}_\omega$; the reader may worry that we do not obtain a translator as in Definition 4.2.1. This is not a problem because $\mathcal{T}$ and $\mathcal{T}_\omega$ coincide on finite sets and we are only considering predicate liftings of finite arity, i.e. elements (subsets) of $\mathcal{T}\mathcal{P}(n)$ for some finite $n$. More formally, for a finite $n$, we use the following chain of isomorphisms/equalities:

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

The reason to restrict to singleton liftings of finite arity is that we only consider the finitary version of the Moss logic (Definition 3.3.1). If we define the Moss logic using $\mathcal{T}$ instead of $\mathcal{T}_\omega$, the previous theorem holds for all singleton liftings.

### 4.2.2 Translators, Singletons and Inductive Presentations

This section is a technical intermezzo where we discuss how to present predicate liftings and their translators inductively on the complexity of the functor. Such presentations are worth to mention because they show the translation procedure in a modular fashion enlightening the power of the functorial approach. This is particularly appealing to develop actual implementations of the mentioned coalgebraic languages and their transaltions. We do not use the inductive presentations further on; the reader not interested in a potential implementation may skip this section.

Recall the definition of Kripke polynomial functor (Definition 2.1.3). We want to inductively describe predicate liftings and their translators. The cases for the identity and constant functors can be found in Example 4.2.6. In this section, we will only consider the inductive cases corresponding to coproducts, products, and composition with the power set functor.

#### Coproducts of functors

Let $\mathcal{T}_1 + \mathcal{T}_2$ be the coproduct of two functors. We want to describe the predicate liftings for the coproducts in terms of the predicate liftings for $\mathcal{T}_1$ and $\mathcal{T}_2$. For this purpose, we use Yoneda Lemma (Proposition 3.3.9) and the fact that $\mathcal{P} : \mathsf{Set}^{op} \to \mathsf{Set}$ preserves products i.e $\mathcal{P}(X+Y) \cong \mathcal{P}(X) \times \mathcal{P}(Y)$.

Using the fact that $\mathcal{P}$ preserves products we have

$$\mathsf{Nat}(\mathcal{P}^n, \mathcal{P}(\mathcal{T}_1+\mathcal{T}_2)) \cong \mathsf{Nat}(\mathcal{P}^n, \mathcal{P}(\mathcal{T}_1)\times\mathcal{P}(\mathcal{T}_2)) \cong \mathsf{Nat}(\mathcal{P}^n, \mathcal{P}(\mathcal{T}_1))\times\mathsf{Nat}(\mathcal{P}^n, \mathcal{P}(\mathcal{T}_2)),$$

the last isomorphism is just the universal property of products. This last equation tells us that a predicate lifting for the coproduct $\mathcal{T}_1 + \mathcal{T}_2$ is univocally determined by a predicate lifting for $\mathcal{T}_1$ and one for $\mathcal{T}_2$. In other words, a predicate lifting for the coproduct is presented by describing its action on each of the components, of the coproduct, independently. More explicitly, given predicate liftings $\lambda_i$ for $\mathcal{T}_i$, $(i = 1, 2)$, we define a predicate lifting $\lambda$ for $\mathcal{T}_1 + \mathcal{T}_2$ as follows: A sequence $\varphi : n \to \mathcal{P}(X)$ is mapped to

$$\lambda(\varphi) = \lambda_1(\varphi) \cup \lambda_2(\varphi). \tag{4.6}$$

The decomposition of a predicate lifting for $\mathcal{T}_1 + \mathcal{T}_2$ is done by post-composing with $\mathcal{P}(\kappa_i) : \mathcal{P}(\mathcal{T}_1 + \mathcal{T}_2) \to \mathcal{P}\mathcal{T}_i$, the inverse image of the coproduct inclusion. More explicitly, $\lambda$ is decomposed into $\mathcal{P}(\kappa_1) \circ \lambda$ and $\mathcal{P}(\kappa_2) \circ \lambda$.

Using Proposition 3.3.9 we can describe predicate liftings as follows:

$$\begin{aligned}\mathsf{Nat}(\mathcal{P}^n, \mathcal{P}(\mathcal{T}_1 + \mathcal{T}_2)) &\cong \mathcal{P} \circ (\mathcal{T}_1 + \mathcal{T}_2)(\mathcal{P}(n)) \\ &\cong \mathcal{P}(\mathcal{T}_1\mathcal{P}(n) + \mathcal{T}_2\mathcal{P}(n)) \\ &\cong \mathcal{P}(\mathcal{T}_1\mathcal{P}(n)) \times \mathcal{P}(\mathcal{T}_2\mathcal{P}(n))\end{aligned}$$

This means that the $n$-ary liftings for $\mathcal{T}_1 + \mathcal{T}_2$ are in natural bijection with the set $\mathcal{T}_1\mathcal{P}(n) \times \mathcal{T}_2\mathcal{P}(n)$. Using this, we can introduce the key property of predicate liftings for coproducts. Namely that we can extend predicate liftings for each of the factors; the next definition makes this precise.

**Definition 4.2.12.** Let $\lambda_i$ be a predicate lifting for a functor $\mathcal{T}_i$, $(i = 1, 2)$ associated with a set $P_i \subseteq \mathcal{T}_i(2^n)$. The *coproduct extension* of $\lambda_i$, also denoted by $\lambda_i$, is the predicate lifting for $\mathcal{T}_1 + \mathcal{T}_2$ which maps a sequence $\varphi : n \to \mathcal{P}(X)$ to the set

$$\lambda_i(\varphi) = \Big\{ t \in (\mathcal{T}_1 + \mathcal{T}_2)(X) \mid t \in \lambda_i(X) \Big\}.$$

This predicate lifting correspond to the set $(P_i, \emptyset)$.

In the case of singleton liftings we have:

**Proposition 4.2.13.** *The coproduct extension of a singleton lifting is a singleton lifting and all singleton liftings for the coproduct arise this way.*

Using the coproduct inclusions, we can also extend translators.

**Proposition 4.2.14.** *Let $\lambda_i$ be a predicate lifting for a functor $\mathcal{T}_i$, $(i = 1, 2)$, and let $\tau_i : \mathcal{P}^n \to \mathcal{T}_i\mathcal{P}$ be a translator for $\lambda_i$. The composition of $\tau_i$ with the corresponding coproduct inclusion, i.e.*

$$\mathcal{P}^n \xrightarrow{\tau_i} \mathcal{T}_i\mathcal{P} \xrightarrow{\kappa_i} (\mathcal{T}_1 + \mathcal{T}_2)\mathcal{P},$$

*is a translator for the coproduct extension of $\lambda_i$.*

The proof of the proposition is a straight forward computation using the inductive presentation of relation lifting (Proposition 2.2.9).

### Products of Functors

The situation for the product of functors is not as nice as that for coproducts; the reason being that $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ does not behave well with coproducts in $\mathsf{Set}^{op}$, i.e. $\mathcal{P}(X \times Y) \not\cong \mathcal{P}(X) + \mathcal{P}(Y)$ in general. To overcome this difficulty, we first define the "product" of predicate liftings.

**Definition 4.2.15.** Let $\lambda_i$ be an $n$-ary predicate lifting for a functor $\mathcal{T}_i$, $(i = 1, 2)$. The *product* of $\lambda_1$ and $\lambda_2$, denoted $\lambda_1 \otimes \lambda_2$, is the predicate lifting for $\mathcal{T}_1 \times \mathcal{T}_2$ defined as follows: a sequence $\varphi : n \longrightarrow \mathcal{P}(X)$ is mapped to

$$(\lambda_1 \otimes \lambda_2)(\varphi) = \lambda_1(\varphi) \times \lambda_2(\varphi),$$

i.e. the Cartesian square.

Not all predicate liftings for the product are of this form; the next example shows this.

**Example 4.2.16.** Let $\mathsf{K}_{\mathbb{R}}$ be the functor with constant value the real numbers. Let $S_1$ be the unitary ball in $\mathbb{R} \times \mathbb{R} = \mathsf{K}_{\mathbb{R}}(2) \times \mathsf{K}_{\mathbb{R}}(2)$. The predicate lifting, for the product $\mathsf{K}_{\mathbb{R}} \times \mathsf{K}_{\mathbb{R}}$, associated with $S_1$ is not the product of any pair of predicate liftings for $\mathsf{K}_{\mathbb{R}}$.

Although we can not describe all predicate liftings for product as squares, all singleton liftings for the product can be presented as the product of two predicate liftings. This will follow from the next proposition.

**Proposition 4.2.17.** *Let $\lambda_i$ be a predicate lifting for a functor $\mathcal{T}_i$ associated with a set $P_i \subseteq \mathcal{T}_i(2^n)$, $(i = 1, 2)$. The product of $\lambda_i$'s corresponds to the Cartesian product of the sets $P_i$ and viceversa.*

*Proof.* We want to show $\lambda_{P_1 \times P_2} = \lambda_1 \otimes \lambda_2$. We will follow the algorithm given by Yoneda Lemma (Proposition 3.3.9) on the set $P_1 \times P_2$. Let $\varphi : n \longrightarrow \mathcal{P}(X)$ be a sequence of sets and $\chi_\varphi : X \longrightarrow \mathcal{P}(n)$ its exponential transpose. Using Yoneda lemma we have

$$\begin{aligned}
\lambda_{(P_1 \times P_2)}(\varphi) &= \{(t_1, t_2) \in (\mathcal{T}_1 \times \mathcal{T}_2)(X) \mid (\mathcal{T}_1 \times \mathcal{T}_2)(\chi_\varphi)(t_1, t_2) \in P_1 \times P_2\} \\
&= \{(t_1, t_2) \in (\mathcal{T}_1 \times \mathcal{T}_2)(X) \mid (\mathcal{T}_1(\chi_\varphi)(t_1), \mathcal{T}_2(\chi_\varphi)(t_2)) \in P_1 \times P_2\} \\
&= \{(t_1, t_2) \in (\mathcal{T}_1 \times \mathcal{T}_2)(X) \mid (\mathcal{T}_1(\chi_\varphi)(t_1) \in P_1 \text{ and } \mathcal{T}_2(\chi_\varphi)(t_2)) \in P_2\} \\
&= \{(t_1, t_2) \in (\mathcal{T}_1 \times \mathcal{T}_2)(X) \mid t_1 \in \lambda_{P_1}(\varphi) \text{ and } t_2 \in \lambda_{P_2}(\varphi)\} \\
&= (\lambda_1 \otimes \lambda_2)(\varphi).
\end{aligned}$$

The equality $\lambda_{(P_1 \times P_2)}(\varphi) = (\lambda_1 \otimes \lambda_2)(\varphi)$ is what we wanted to prove. □

We can specialise the previous proposition to singleton liftings.

**Corollary 4.2.18.** *Let $\mathcal{T}_i$, $(i = 1, 2)$, be functors. The product of singleton liftings is a singleton liftings. And all singleton liftings for $\mathcal{T}_1 \times \mathcal{T}_2$ arise this way.*

Translators can be combined using the universal property of products. More formally,

**Proposition 4.2.19.** *Let $\lambda_i$ be predicate liftings for a functor $\mathcal{T}_i$, $(i = 1, 2)$. If $\tau_i : \mathcal{P}^n \longrightarrow \mathcal{T}_i\mathcal{P}$ is a translator for $\lambda_i$, then $(\tau_1, \tau_2) : \mathcal{P}^n \longrightarrow (\mathcal{T}_1 \times \mathcal{T}_2)\mathcal{P}$ is a translator for $\lambda_1 \otimes \lambda_2$.*

The proof of the proposition is a straight forward computation using the inductive presentation of relation lifting (Proposition 2.2.9).

### Composition with Pow

Singleton liftings are particularly useful to describe the singleton liftings for composition of the type $\mathsf{Pow}\mathcal{T}$. The key ingredient is that a set $P \in \mathsf{Pow}\mathcal{T}\mathcal{P}(n)$, i.e. a singleton lifting for $\mathsf{Pow}\mathcal{T}$, is a subset of $\mathcal{T}\mathcal{P}(n)$, i.e a predicate lifting for $\mathcal{T}$. Since all predicate liftings for $\mathcal{T}$ can be described using singleton liftings, we can then expect to describe the predicate lifting $\lambda_P : \mathcal{P}^n \longrightarrow \mathcal{P}\mathsf{Pow}\mathcal{T}$ using the singleton liftings for $\mathcal{T}$. We now explain this more formally.

To avoid confusion we fix the following convention for this section. Predicate liftings for $\mathsf{Pow}\mathcal{T}$ will be denoted using $\lambda$ whereas predicate liftings for $\mathcal{T}$ will be denoted using $\hbar$. Elements of $\mathsf{Pow}\mathcal{T}\mathcal{P}(n)$ are denoted by capital letters, elements of $\mathcal{T}\mathcal{P}(n)$ are denoted with lowercase letters. The subindexes follow the conventions established before.

We first dissemble the action of $\lambda_P : \mathcal{P} \longrightarrow \mathcal{P}\mathsf{Pow}\mathcal{T}$. Let $\varphi : n \longrightarrow \mathcal{P}(X)$ be a sequence of sets and let $\chi_\varphi : X \longrightarrow \mathcal{P}(n)$ be its exponential transpose. We have

$$\begin{aligned}\lambda_P(\varphi) &= \{U \in \mathsf{Pow}\mathcal{T}(X) \mid \mathsf{Pow}\mathcal{T}(\chi_\varphi)(U) = P\} && \text{(Prop. 3.3.9)}\\ &= \{U \in \mathsf{Pow}\mathcal{T}(X) \mid \mathcal{T}(\chi_\varphi)[U] = P\} && \text{(Def. of Pow)}\end{aligned}$$

The connection with predicate liftings for $\mathcal{T}$ appears when we want to unravel $\mathcal{T}(\chi_\varphi)[U] = P$. By definition of the direct image this equality means:

$$(\forall t \in U)(\exists p \in P)(\mathcal{T}(\chi_\varphi)(t) = p) \text{ and } (\forall p \in P)(\exists t \in U)(\mathcal{T}(\chi_\varphi)(t) = p)$$

Recall that the equation $\mathcal{T}(\chi_\varphi)(t) = p$ describes the elements of $\hbar_p(\varphi)$. Using this we can present the action of $\lambda_P$ as follows:

$$\begin{aligned}\lambda_P(\varphi) &= \left\{U \in \mathsf{Pow}\mathcal{T}(X) \mid U \subseteq \bigcup_{p \in P} \hbar_p(\varphi) \text{ and } (\forall p \in P)(U \cap \hbar_p(\varphi) \neq \emptyset)\right\}\\ &= \left\{U \in \mathsf{Pow}\mathcal{T}(X) \mid U \in \blacktriangledown_{\mathsf{Pow}}\{\hbar_p(\varphi) \mid p \in P\}\right\}\end{aligned} \tag{4.7}$$

As said before, the fact that $\mathcal{P}$ and $\mathsf{Pow}$ "coincide" in objects allows us to describe singleton liftings for $\mathsf{Pow}\mathcal{T}$ inductively. It is not clear to us how to extend this induction to composition with other functors.

Since all singleton liftings have translators (Theorem 4.2.10), we can present the translator of a singleton lifting for $\mathsf{Pow}\mathcal{T}$ inductively as follows:

**Proposition 4.2.20.** *Let $P$ be an element in $\mathsf{Pow}\mathcal{T}\mathcal{P}(n)$. For each $p \in P$, let $\tau_p : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ be a translator for $\hbar_p$ (Theorem 4.2.10). The natural transformation*

$$\mathcal{P}^n \xrightarrow{\left\{\tau_p \,\mid\, p \in P\right\}} \mathsf{Pow}\mathcal{T}\mathcal{P}$$

*is a translator for $\lambda_P$.*

*Proof.* We want to show $\blacktriangledown_{\mathsf{Pow}\mathcal{T}}\left\{\tau_p \mid p \in P\right\} = \lambda_P$. In order to show this, recall the inductive description of relation lifting (Proposition 2.2.9) and the presentation of $\lambda_P$ in Equation (4.7). It is also important to remember (Remark 4.2.3) that since $\tau$ is a translator, for every $\varphi : n \longrightarrow \mathcal{P}(X)$ we have

$$(t, \tau_p(\varphi)) \in \mathcal{T}(\in_X) \text{ iff } t \in \hbar_p(\varphi)$$

Using these observations we can detail $\blacktriangledown_{\mathsf{Pow}\mathcal{T}}\left\{\tau_p \mid p \in P\right\}$ as follows: Let $\varphi : n \longrightarrow \mathcal{P}(X)$ be a sequence of sets and let $\chi_\varphi : X \longrightarrow \mathcal{P}(n)$ be its exponential transpose; write $U$ for an element in $\mathsf{Pow}\mathcal{T}(X)$. We have

$$\begin{aligned}
\blacktriangledown_{\mathsf{Pow}\mathcal{T}}\left\{\tau_p \mid p \in P\right\}(\varphi) &= \blacktriangledown_{\mathsf{Pow}\mathcal{T}}\left\{\tau_p(\varphi) \mid p \in P\right\} \\
&= \Big\{U \mid (U, \left\{\tau_p(\varphi) \mid p \in P\right\}) \in \overline{\mathsf{Pow}\mathcal{T}}(\in_X)\Big\} \\
&= \Big\{U \mid (\forall t \in U)(\exists p \in P)((t, \tau_p(\varphi)) \in \overline{\mathcal{T}}(\in_X) \\
&\qquad\qquad \text{and } (\forall p \in P)(\exists t \in U)((t, \tau_p(\varphi)) \in \overline{\mathcal{T}}(\in_X))\Big\} \\
&= \Big\{U \mid (\forall t \in U)(\exists p \in P)((t \in \hbar_p(\varphi)) \text{ and } (\forall p \in P)(\exists t \in U)((t \in \hbar_p(\varphi))\Big\} \\
&= \Big\{U \in \mathsf{Pow}\mathcal{T}(X) \mid U \subseteq \bigcup \hbar_p(\varphi) \text{ and } (\forall p \in P)(U \cap \hbar_p(\varphi) \neq \emptyset)\Big\} \\
&= \lambda_P(\varphi)
\end{aligned}$$

This concludes the proof. □

**Remark 4.2.21.** Notice that the Propositions 4.2.14, 4.2.19 and 4.2.20 do not require any of the $\mathcal{T}_i$ to be a Kripke polynomial functor. Should we have a good description of the translators, then we can then extend them using coproducts, products and composition with $\mathsf{Pow}$.

## 4.3 Logical Translators

We have now all the material needed to address the issue of defining generic translations between logics of predicate liftings and the Moss logic. Up to now we have presented our translations under the tacit assumption that our basic propositional logic is given by Boolean algebras. As we will later see (Lemma 4.4.1), in such situation translators do define translations. However, this is not always the case. As it could be expected, the possibility of defining a translation is strongly tied to the expressive power of the base logic of the coalgebraic modal logic. Here we will show how the base category of the coalgebraic modal logic makes the difference.

We now turn back to the intuition behind translators to explain the problem more precisely. By definition, a translator $\tau$ factors a predicate lifting $\lambda$ via $\blacktriangledown$. As we mentioned in Equation (4.4), the idea is to inductively define a translation via $tr(\Box_\lambda \varphi) = \nabla \tau(tr(\varphi))$. The problem is to show that $\tau(tr(\varphi))$ is expressible in the basic propositional logic. The next example shows that this is not always possible.

**Example 4.3.1.** Consider $\mathcal{A} = \mathsf{DL}$ and $\mathcal{T} = \mathsf{Id}$ and the predicate lifting $\lambda_\perp : \mathcal{P} \to \mathcal{P}$ given by complementation. In this case $\blacktriangledown_{\mathsf{Id}} : \mathsf{Id}\mathcal{P} \to \mathcal{P}\mathsf{Id}$ is the identity. From this we see that complementation $\neg : \mathcal{P} \to \mathcal{P}$ is a translator for $\lambda_\perp$. Hence the intended translation would be $tr(\Box_{\lambda_\perp} \varphi) = \nabla \neg \varphi$. Since the base category for the coalgebraic modal logic is distributive lattices, all the operators in $\mathcal{M}_{\mathsf{Id}}$ are monotone, therefore all the definable predicate liftings are monotone, which implies that negation is not definable. In other words, we cannot translate $\lambda_\perp$ into $\mathcal{M}_{\mathsf{Id}}$.

The following definition will ensure that $\tau(tr(\varphi))$ is expressible in the base logic given by the category $\mathcal{A}$.

**Definition 4.3.2.** Let $\lambda$ be an $n$-ary predicate lifting for a functor $\mathcal{T}$; let $\mathcal{A}$ be a category with power-set algebras; Let $U : \mathcal{A} \to \mathsf{Set}$ be the forgetful functor.

An $\mathcal{A}$-*logical translator* $\tau$ for $\lambda$ is a natural transformation $\tau : U^n \to \mathcal{T}_\omega U$ such that $\tau_P : UP^n \to \mathcal{T}_\omega UP$ is a translator for $\lambda$, i.e. $\lambda = \blacktriangledown_{\mathcal{T}} \circ \tau_P$.

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

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

**Example 4.3.3.**

1. Example 4.3.1 can now be presented with this new terminology as follows: $\tau = \neg$ does not extend to a $\mathsf{DL}$-translator; but, of course,

it does extend to a BA-translator, namely the negation $\neg : U \longrightarrow U$. We can also say that $\lambda_\perp$ is not DL-translatable but it is BA-translatable.

2. Consider the predicate lifting associated with the existential modality $\Diamond$ as in Example 4.2.2. We define a BA-translator $\tau$ as follows: Given a Boolean algebra $(A, \alpha)$, the function $\tau_{(A,\alpha)} : A \longrightarrow \mathsf{Pow}(A)$ maps an element $a \in A$ to $\tau_{(A,\alpha)}(a) = \{a, \top\}$; as expected, $\tau$ induces the following translation $tr(\Diamond\varphi) = \nabla\{\varphi, \top\}$. This is also a DL-translator but not a Set-translator because we can no identify the element $\top$.

3. Consider the probabilistic modality $\Diamond_p$. We define a DL-translator $\tau : U \longrightarrow \mathcal{D}U$ as follows: let $(A, \alpha)$ be a distributive lattice. The $(A, \alpha)$ component of $\tau_{(A,\alpha)} : A \longrightarrow \mathcal{D}(A)$ maps an element $a \in A$ to the probability distribution $D^a_p : A \longrightarrow [0, 1]$ which assigns a probability $p$ to $a$ and a probability $1 - p$ to $\top$. Compare this with Example 4.2.2.

4. Consider the probabilistic modality $\Diamond^p$. We define a BA-translator $\tau : U \longrightarrow \mathcal{D}U$ as follows: let $(A, \alpha)$ be a boolean algebra. The $(A, \alpha)$ component of $\tau$ maps an element $a \in A$ to the probability distribution $D^p_a : A \longrightarrow [0, 1]$ assigning probability $p$ to $\neg a$ and $1 - p$ to $\top$. Clearly, this can not be regarded as a DL-translator. Compare this with Example 4.2.2.

5. Consider the natural transformation $\eta : \mathsf{Id} \longrightarrow \mathsf{Pow}$ which maps an element $x$ to $\{x\}$. If we precompose $\eta$ with $\mathcal{P}$ we obtain a natural transformation $\tau_\mathcal{P} : \mathcal{P} \longrightarrow \mathsf{Pow}\mathcal{P}$ which maps a set $\varphi \subseteq X$ to $\{\varphi\}$. This is a BA-translator for the predicate lifting $\lambda_\top : \mathcal{P} \longrightarrow \mathcal{P}\mathsf{Pow}$ (Example 4.2.6). The translator $\tau_\mathcal{P}$ induces the following translation $tr(\Box_{\lambda_\top}\varphi) = \nabla\{\varphi\}$. Notice that this translator is an $\mathcal{A}$-translator for any category of power set algebras.

**Remark 4.3.4.** Generalizing the last item in the previous example, we can ask which predicate liftings have $\mathcal{A}$-translators for all categories $\mathcal{A}$ of power-set algebras. These are precisely what we call the Moss liftings (Diagram (5.4) on page 102), see comments after Remark 5.1.16 on page 103.

As said before, the main property of logical translators, as suggested by the previous examples, is that they define one step translations. In other words, $\tau(tr(\varphi)$ in Equation 4.4 is definable in the base logic.

**Lemma 4.3.5.** *Every logical translator induces a one-step translation.*

*Proof.* Let $\mathcal{A}$ be a category with power set algebras with forgetful functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$, write $F$ for the left adjoint of $U$. Let $\tau : U^n \longrightarrow \mathcal{T}_\omega U$ be an $\mathcal{A}$-logical translator for an $n$-ary predicate lifting $\lambda$.

We want to define a one-step translation $\nu : (\bar{L}_\lambda, \bar{\delta}_\lambda) \longrightarrow (M_\mathcal{T}, \nabla)$. Recall from Definitions 3.2.18 and 3.3.4, that the functors are given by $L_\lambda = F(U^n)$, and $M_{\mathcal{T}} = F\mathcal{T}_\omega U$; and the semantics are given by the $F$-transposes of $\lambda$ and $\blacktriangledown$, respectively; i.e. $\delta_\lambda = \widehat{\lambda}$ and $\nabla = \widehat{\blacktriangledown}$.

The one-step translation we are looking for is given by $\bar{L}_\lambda = F(U^n) \xrightarrow{F(\tau)} F\mathcal{T}_\omega U = M_{\mathcal{T}}$. Since $\tau$ natural so is $F(\tau)$. It is only left to show that $F(\tau_P) : F(UP^n) \longrightarrow F\mathcal{T}_\omega UP$ commutes with the semantics, i.e. $\nabla \circ F(\tau_p) = \delta_\lambda$. By definition of logical translator, $\tau_P$ is a translator for $\lambda$, i.e. the following diagram

$$\begin{array}{ccc} (UP)^n & \xrightarrow{\tau_P} & T_\omega UP \\ & \lambda \searrow \qquad \swarrow \blacktriangledown & \\ & UP\mathcal{T} & \end{array}$$

commutes, recall $UP = \mathcal{P}$. By properties of adjoints (Lemma A.1.3, item 3) we can move $U$ to the left and obtain $\nabla \circ F(\tau_P) = \delta_\lambda$. In other words $F(\tau)$ is a one step translation. This concludes the proof. □

The next proposition shows how the previous argument can be extended to sets of predicate liftings.

**Proposition 4.3.6.** *Let $\Lambda$ be a set of predicate liftings, each of which has a logical translator. Then we can find a one-step translation $\nu : (\bar{L}_\Lambda, \bar{\delta}_\lambda) \longrightarrow (M_{\mathcal{T}}, \nabla)$.*

*Proof.* Recall from Definitions 3.2.18 and 3.3.4 that $\bar{L}_\Lambda = \coprod_{\lambda\in\Lambda} FU^{n_\lambda}$ and $M = F\mathcal{T}_\omega U$, where $n_\lambda$ is the arity of $\lambda$; recall that $\delta_\Lambda$ is given by the coproduct of the $F$-transposes of the predicate liftings in $\Lambda$ and $\nabla$ is the $F$-transpose of $\blacktriangledown$.

By assumption for each $\lambda \in \Lambda$ there is a logical translator $\tau_\lambda : U^{n_\lambda} \longrightarrow \mathcal{T}_\omega U$. From the previous lemma each $F(\tau_\lambda) : FU^{n_\lambda} \longrightarrow F\mathcal{T}_\omega U$ is a one step translation from $(L_\lambda, \delta_\lambda)$ into $(M_{\mathcal{T}}, \nabla)$. Since $(\bar{L}_\Lambda, \bar{\delta}_\Lambda) = (\coprod_{\lambda\in\Lambda} L_\lambda, \coprod_{\lambda\in\Lambda} \delta_\lambda)$ it is now straightforward to check, using the universal property of coproducts, that the arrow $\coprod_{\lambda\in\Lambda} F(\tau_\lambda) : \coprod_{\lambda\in\Lambda} L_\lambda \longrightarrow F\mathcal{T}_\omega U$ is a one-step translation from $(\bar{L}_\Lambda, \bar{\delta}_\lambda)$ to $(M_{\mathcal{T}}, \nabla)$. This concludes the proof. □

To summarise, all $\mathcal{A}$-logical translators give rise to one-step translations. Translators alone do not define translations; to obtain a translation from a translator we need to extend it to a logical translator. Such an extension is not always possible (Example 4.3.1); the possibility of extending a translator rests on the structural properties of the category $\mathcal{A}$. To illustrate this, we are now going to show that all translators do extend to $\mathsf{BA}$-logical translators.

## 4.4 The Boolean Paradise

Until here, we have presented two perspectives for translating. On the one hand, we have one-step translations which define translations at a syntactic level, i.e.

for each formula in the source language we find a semantically equivalent formula in the target language. On the other hand, we have translators. Translators are a semantic perspective for translating predicate liftings, they literally transform the semantics of one modality into the semantics of another modality. The bridge between these two perspectives is built with logical translators. The "trick" of replacing the contravariant functor $\mathcal{P}$ by the forgetful functor of the base category is what is needed to make the "semantic translations", given by translators, explicit in the syntax. However, as shown in Example 4.3.1, this is not always possible. At a first glance it might seem that such a definability problem is a pure concern of the syntax. Nevertheless, using the functorial approach we can show that this issue concerns the base category of the logic.

In this section, we illustrate how we may use the structural properties of the category $\mathsf{BA}$ to extend every translator to a $\mathsf{BA}$-translator. This was somehow expected because the algebraic theory of boolean algebras is finitary and the functor $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ presents the category $\mathsf{Set}^{op}$ as the category of complete atomic boolean algebras.

### 4.4.1 Translating predicate liftings

In this section, we will define a one-step translation (Definition 4.1.1) from the logic of all predicate liftings $(\bar{L}_{\mathcal{T}}, \bar{\delta}_{\mathcal{T}})$ (see page 63) to the Moss logic $\mathcal{M}_T$ (Definition 3.3.2). The main technical result is that that translators (Definition 4.2.1) can always be extended to $\mathsf{BA}$-translators (Definition 4.3.2).

**Lemma 4.4.1.** *Every translator $\tau : \mathcal{P}^n \longrightarrow \mathcal{T}_\omega \mathcal{P}$ can be extended to a BA-translator, i.e. a natural transformation $U^n \longrightarrow \mathcal{T}_\omega U$, where $U : \mathsf{BA} \longrightarrow \mathsf{Set}$ is the forgetful functor.*

*Proof.* The following facts will be used in the proof:

1. every boolean algebra is the directed colimit of finite boolean algebras, more specifically of its canonical diagram (Proposition 2.3.9),

2. every finite Boolean algebra is (isomorphic) to a power set algebra,

3. every boolean algebra morphism $A \longrightarrow B$ with $A$ finite and $B = PY$, for some possibly infinite $Y$, arises from the inverse image of a function with domain $Y$,

4. $F(m) = P\mathcal{P}(m)$ for finite sets $m$.

Let $\tau : \mathcal{P}^n \longrightarrow \mathcal{T}_\omega \mathcal{P}$ be a translator for a predicate lifting $\lambda$. We want to show that we can extend $\tau$ to all boolean algebras, i.e. we want to replace $\mathcal{P}$ by $U$. We first define a natural transformation $\tau' : U^n \longrightarrow \mathcal{T}_\omega U$ and then show that it coincides with $\tau$ on power set algebras.

First we explain how to define $\tau'$. Let $\tau_\omega : \mathcal{P}^n \longrightarrow \mathcal{T}_\omega \mathcal{P}$ be the restriction of $\tau$ to finite sets. Because of (2), we can replace $\mathcal{P}$ by the restriction of the forgetful $U$ to finite Boolean algebras, i.e. $U_\omega : \mathsf{BA}_\omega \longrightarrow \mathsf{Set}$. This gives us for each finite boolean algebra $A$ a function $(\tau_\omega)_A : U_\omega(A)^n \longrightarrow \mathcal{T}_\omega U_\omega(A)$. We want to show that these functions form a natural transformation. This follows from (3), more explicitly, if $A = P(n)$ and $A' = P(n')$ are finite Boolean algebras, then a homomorphism $h : A \longrightarrow A'$ is equal to $P(f) = f^{-1}$ for some function $f : n' \longrightarrow n$. From (4) we have that every finitely generated algebra is a power set algebra; hence we can see $\tau_\omega$ as a natural transformation $\tau_\omega : (UF_\omega)^n \longrightarrow \mathcal{T}_\omega UF_\omega$, where $F_\omega$ is the restriction of the functor $F$ to finite sets. Then by (1), see Proposition 2.3.14, we can extend $\tau_\omega$ to all boolean algebras, i.e. to a natural transformation $\tau' : U^n \longrightarrow \mathcal{T}_\omega U$. This is the logical translator we are looking for.

It is only left to check that $\tau$ and $\tau'$ coincide in power set algebras, i.e. $\tau'_P = \tau$. By definition, $\tau'_{P(Y)} = \tau_Y$ whenever $Y = \mathcal{P}(m)$ for some finite $m$. We now show that this is also the case for any other set $Y$. Since $\mathsf{BA}$ is a finitary variety, every algebra $A$ is the colimit of its canonical diagram, i.e. the colimit of all valuations into $A$. Write $F(n_i) \xrightarrow{h_i} P(Y)$ for the particular case of a power set algebra $P(Y)$.

Consider the following diagram

$$
\begin{array}{ccccc}
P(Y) & \quad & UP(Y)^n = \mathcal{P}(Y)^n & \xrightarrow{\quad ? \quad} & \mathcal{T}_\omega \mathcal{P}(Y) \\
\uparrow{\scriptstyle h_i} & & \uparrow{\scriptstyle U(h_i)} & & \uparrow{\scriptstyle \mathcal{T}U(h_i)} \\
F(n_i) & & UF(n_i)^n = \mathcal{P}\mathcal{P}(n_i)^n & \xrightarrow{\tau'_{P(n_i)} = \tau_{n_i}} & \mathcal{T}_\omega \mathcal{P}\mathcal{P}(n_i)
\end{array}
$$

the equality in the lower left corner holds because of (4); the equality in the lower edge follows by definition of $\tau'$. The equality on the upper left corner holds because $U$ is a right adjoint.

Notice that $\mathcal{T}_\omega$ preserves the colimit $F(n_i) \xrightarrow{h_i} P(Y)$ because it is finitary, and $U$ preserves the $n$-product of this colimit, i.e. $F(n_i)^n \xrightarrow{h_i^n} P(Y)^n$, because of Proposition 2.3.9. This implies that the upper horizontal arrow in the diagram is the unique arrow induced by the universal property of colimits. From this, the equality $\tau'_P = \tau$ will follow once we show that putting $\tau'_{P(Y)}$ or $\tau_Y$ in the upper row of the diagram makes it commute. Indeed, for $\tau'_{P(Y)}$ this holds by definition of $\tau'$, see Proposition 2.3.14. For $\tau_Y$, by (3), we have that each $h_i$ is $f_i^{-1}$ for some function $f_i : Y \longrightarrow \mathcal{P}(n_i)$; since $\tau$ is natural in $\mathcal{P}$, this means that $\tau_Y$ also makes the diagram commute. This concludes the proof. □

**Remark 4.4.2.** The previous lemma strongly depends on the properties of the category $\mathsf{BA}$. More precisely, in the previous proof, it is essential that every finitely generated algebra is a power set algebra, i.e. it is in the image of $P$. Otherwise we have no means to extend $\tau_\omega$ to the full variety. Extensions of the previous theorem to other categories (of power set) algebras is a topic for further research. For example, for the category of distributive lattices we should somehow modify the notion of predicate lifting or the argument will not work, the reason being that there are finite distributive lattices which are not power set algebras.

An immediate corollary is that we can translate singleton liftings.

**Corollary 4.4.3.** *Every singleton lifting, of finite arity, for a weak pullback preserving functor $\mathcal{T}$, can be translated into Moss logic for $\mathcal{T}$ on* BA.

*Proof.* Let $\lambda$ be singleton lifting. By Theorem 4.2.10 it has a translator $\tau$. By the previous theorem, $\tau$ can be extended to a BA-translator. By Lemma 4.3.5 this induces a one-step translation, i.e. $\lambda$ can be translated into the Moss logic for $\mathcal{T}$. □

The following translations illustrate the previous corollary.

**Example 4.4.4.**

1. The translations in Example 4.3.3 are instantiations of the previous theorem.

2. Let $\lambda_*$ the predicate lifting that indicates termination (Example 4.2.6). The constant natural transformation $\tau : A \longrightarrow 1 + A$ into 1 is a translator for $\lambda_*$. This is in fact an $\mathcal{A}$-translator for any category $\mathcal{A}$ of power set algebras. The induced translation is $tr(\Box_*\varphi) = \nabla *$, where $* \in \mathcal{M}_T$. Notice that $*$ is a formula in $\mathcal{M}_T$ and does not depend on the underlying propositional logic (see Remark 3.3.2).

3. Let $(n, m)$ be a singleton lifting for the finite multiset functor (Example 4.2.6). We define a BA-translator for $(n, m)$ as follows: Given a Boolean algebra $(A, \alpha)$ the function $\tau_{(A,\alpha)} : A \longrightarrow \mathcal{B}_{\mathbb{N}}(A)$ maps an element $a \in A$ to the bag $B_{(a,n,m)} : A \longrightarrow \mathbb{N}$ which maps an elements in $A$ as follows:

$$\begin{aligned} B_{(a,n,m)}(a) &= n, \\ B_{(a,n,m)}(\neg_{(A,\alpha)} a) &= m \\ B_{(a,n,m)}(x) &= 0 (\text{for any other element } x) \end{aligned}$$

   This induces the following translation $tr(\Box_{(n,m)}\varphi) = \nabla B_{(\varphi,n,m)}$.

4. Let $q \in [0, 1]$ be a singleton lifting for the distribution functor. We define a BA-translator as follows: $\tau : A \longrightarrow \mathcal{D}(A)$ maps an element $a$ to the probability distribution $\mu_a : A \longrightarrow [0, 1]$ which assigns $q$ to $a$, $1 - q$ to $\neg a$, and 0 to any other element. As shown in Example 4.2.6 the induced translation is $tr(\Box_q\varphi) = \nabla\{(\varphi, q), (\neg_X\varphi, 1 - q)\}$.

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

**Theorem 4.4.5.** *If $\mathcal{T}$ preserves finite sets and weak pullbacks, there is a one-step translation $(\bar{L}_{\mathcal{T}}, \bar{\delta}_{\mathcal{T}}) \longrightarrow (M_{\mathcal{T}}, \nabla)$.*

*Proof.* Let $(\bar{L}_s, \bar{\delta}_s)$ be the logic of singleton liftings (Definition 4.2.7). Because $\mathcal{T}$ preserves finite sets, every predicate lifting can be expressed as a finite join of singleton liftings (Proposition 4.2.8), hence we have an isomorphism $\bar{L} \cong \bar{L}_s$. Now let $\lambda$ be a singleton lifting and let $\tau$ be the corresponding translator (Theorem 4.2.10). Obtain a one-step translation $L_\lambda \longrightarrow M_\mathcal{T}$ as in the previous corollary. Doing this for each singleton lifting and combining all of these logical translators, as in Proposition 4.3.6, we obtain a translation $\bar{L}_s \longrightarrow M_\mathcal{T}$. □

Note that Examples 4.3.3, 4.3.1, and 4.2.4 show that in order to translate all predicate liftings, we need classical propositional logic. Weak pullback preservation is needed because otherwise the Moss logic is not defined. The following example shows that the condition of $\mathcal{T}$ preserving finite sets cannot be dropped.

**Example 4.4.6.** Let $\mathcal{T}$ be the constant functor with value $\mathbb{N}$, let $E \subseteq \mathbb{N}$ be the set of even numbers. If we are working over $\mathsf{BA}$, the predicate lifting $\lambda_E$ can not be translated into the Moss logic over $\mathsf{BA}$. Consider the coalgebra $N = (\mathbb{N}, 1_\mathbb{N})$ and the formula $\Box_E \top$. On the one hand, this formula defines the set of even numbers, i.e. $[\![\Box_E \top]\!] = E$. On the other hand, we can check that using the Moss logic we can only define finite and cofinite sets; therefore we conclude that the predicate lifting $\lambda_E$ can not be translated.

The following translations illustrate the previous theorem.

**Example 4.4.7.**
1. The translations in Example 4.2.9 illustrate the previous theorem. We recall them.
   (a) The predicate lifting for $\Box$ is the join of $\lambda_{\{\emptyset\}}$ and $\lambda_{\{\top\}}$ hence we can translate $\Box$ using the translators for those, more explicitly we have

   $$tr(\Box\varphi) = tr(\Box_{\lambda_\emptyset}\varphi) \vee tr(\Box_{\lambda_{\{\top\}}}\varphi) = \nabla\emptyset \vee \nabla\{\varphi\}.$$

   (b) In the case of the existential modality, the predicate lifting for $\Diamond$ is the join of $\lambda_{\{\top,\bot\}}$ and $\lambda_{\{\top\}}$. We can then translate $\Diamond$ using the translators for $\lambda_{\{\top,\bot\}}$ and $\lambda_{\{\top\}}$; in such perspective we have

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

2. Even though we can translate singleton liftings for $\mathcal{B}_\mathbb{N}$ and $\mathcal{D}$, see above, we can not use the previous theorem to conclude that the standard logics for these functors are translatable into the Moss logic because these functors do not preserve finite sets. In case of $\mathcal{D}$, Example 4.3.3 shows that sometimes we can. The case of $\mathcal{B}_\mathbb{N}$ shows that this might also fail, see Examples 3.3.6 and 4.2.4.

### 4.4.2 Translating Moss logic

Our next step is to find a translation $(M_{\mathcal{T}}, \nabla) \to (L_{\mathcal{T}}, \delta_{\mathcal{T}})$, see Definition 3.3.13. Note that we do not expect a natural transformation $M_{\mathcal{T}} \to \bar{L}_{\mathcal{T}}$ because each $\nabla$-formula corresponds to many different but equivalent formulas of $\mathcal{L}_{\mathcal{T}}$ (see also the next chapter). So we make use of the fact that $L_{\mathcal{T}}$ is a quotient of $\bar{L}_{\mathcal{T}}$.

**Theorem 4.4.8.** *For all weak pullback preserving functors $\mathcal{T}$ there exists a one-step translation $(M_{\mathcal{T}}, \nabla) \to (L_{\mathcal{T}}, \delta_{\mathcal{T}})$, where $(L_T, \delta_T)$ is as in Definition 3.3.13.*

*Proof.* Recall that for finite $n$ we have $F(n) = P\mathcal{P}(n)$ and that $L_{\mathcal{T}}F(n) = P\mathcal{T}\mathcal{P}(n)$, see Definition 3.3.13. From this, we can see that the semantics of the Moss logic $\blacktriangledown : \mathcal{T}_\omega UP \to UP\mathcal{T}$ on $\mathcal{P}(n)$ can be written $\blacktriangledown_{\mathcal{P}(n)} : \mathcal{T}_\omega UF(n) \to UL_{\mathcal{T}}F(n)$. Since $U$ is a right adjoint and $M_T = F\mathcal{T}_\omega U$ this yields $M_{\mathcal{T}}F(n) \to L_{\mathcal{T}}F(n)$. Since both $M_{\mathcal{T}}$ and $L_{\mathcal{T}}$ are determined by their action on finitely generated free algebras, this gives a natural transformation $\nu : M_{\mathcal{T}} \to L_{\mathcal{T}}$, see Proposition 2.3.14, it is now straightforward to check that $\nu$ is a one-step translation. □

Again, the theorem is specific to $\mathsf{BA}$. In particular, both translations $\bar{L}_{\mathcal{T}} \to M_{\mathcal{T}}$ and $M_{\mathcal{T}} \to L_{\mathcal{T}}$ made use of the fact that in case of $\mathsf{BA}$ we have $F(n) = P\mathcal{P}(n)$ for finite $n$.

On the other hand, Theorem 4.4.8 is a particular instance of a more general Lindström-like theorem showing that $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$ is the most expressive finitary Boolean logic for $\mathcal{T}$-coalgebras; see also [77] for more on coalgebraic Lindström theorems.

**Theorem 4.4.9.** *Assume that $\mathcal{T}$ preserves finite sets and that $(L, \delta)$ is a Boolean logic for $\mathcal{T}$-coalgebras. Then $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$ is at least as expressive as $(L, \delta)$, that is, there is a one-step translation $\nu : (L, \delta) \to (L_{\mathcal{T}}, \delta_{\mathcal{T}})$.*

*Proof.* The argument is a particular instance of Theorem 5.2.2. We only sketch the construction.

By definition of $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$ the semantics $(\delta_{\mathcal{T}})_X : L_{\mathcal{T}}P(X) \to P\mathcal{T}(X)$ is an isomorphism on finite sets $X$. To define the translation consider the following composite, $LP(X) \xrightarrow{\delta} P\mathcal{T}(X) \xrightarrow{\delta_{\mathcal{T}}^{-1}} L_{\mathcal{T}}P(X)$ on finite $X$. As in the proof of Theorem 4.4.8, this determines a natural transformation $L' \to L$ on finitely generated free Boolean algebras and hence on all Boolean algebras, Proposition 2.3.14. □

## 4.5 Conclusions

In this chapter, we presented a general theory for translating coalgebraic modal logics based on the same category (Section 4.1). We concentrated on translations between logics of predicate liftings and the Moss logic. We introduced the novel notion of a translator (Definition 4.2.1). The intuition behind translators is a simple semantic translation given by $tr(\Box_\lambda \varphi) = \nabla\tau(tr(\varphi)$. In Section 4.2.2 we developed translators by showing how they can be described inductively over the complexity of the functor. In Section 4.4 we showed that under the appropriate circumstances translators induce translations . Along the side of translators we introduced singleton liftings (Definition 4.2.5). These predicate liftings are important because they always have translators (Theorem 4.2.10) and can be inductively presented over the complexity of the functor (Section 4.2.2).

Perhaps the main point to remember from this chapter is that the base category for the modal logic makes the difference to translate the modalities. This was evidenced in Section 4.3 where we showed how translations between coalgebraic modal logic might fail to exist in case the base logic is not expressive enough (Example 4.3.1).

The key insight introduced in this chapter is the use of the structural properties of the base category to produce logical translators (Section 4.3). This technique was illustrated in Section 4.4 where we showed that all translators, for a weak-pullback preserving functor which preserves finite sets, induce translations if the base category is $\mathsf{BA}$ (Lemma 4.4.1). We also showed that all these conditions are needed (Example 4.4.6) to develop a general theory. We also showed how using the structural properties of the base category we can translate the Moss logic (Theorem 4.4.8). Moreover, we presented a Lindström like theorem for coalgebraic logics (Theorem 4.4.9).

In our opinion, none of this could have been developed without the functorial view of modalities.

# Chapter 5

# From Abstract to Concrete

In this chapter we illustrate various uses of translators and translations. Our main technical tool will be the so called presentation of functors by operations and equations. This will have various applications in our study of coalgebraic modal logics. The main contributions of this chapter can be summarised as follows:

1. We introduce a new type of predicate liftings called the Moss liftings (Definition 5.1.12). Using these we define a new translation of the Moss Logic into the language of predicate liftings (Proposition 5.1.21).

2. We prove two representation theorems that present any coalgebraic modal logic (Definition 3.2.13) as a logic of predicate liftings (Theorems 5.2.2 and 5.2.17).

3. We develop a novel equational coalgebraic logic (Section 5.3) with a sound and complete axiomatization for it.

## 5.1 Presentations of functors

As mentioned before (Section 2.3), equational algebraic theories and algebras for a functor are tightly related. In particular, recall that algebraic signatures and polynomial functors, over Set, are two faces of the same coin. To make this more precise we will use the so called presentations of functors. Roughly speaking, a presentation of a functor makes the algebraic operations and equations of the functor explicit. We begin with the formal definition.

**Definition 5.1.1.** Let $\mathcal{T} : \mathsf{Set} \to \mathsf{Set}$ be a functor. A *finitary presentation* of $\mathcal{T}$ is a polynomial functor $\Sigma : \mathsf{Set} \to \mathsf{Set}$ together with a surjective natural transformation $E : \Sigma \to \mathcal{T}$. More explicitly, for each set $X$ we have a surjective function

$$\Sigma(X) = \coprod_{n<\omega} \Sigma_n \times X^n \xrightarrow{E_X} \mathcal{T}(X), \tag{5.1}$$

natural in $X$, where $\Sigma_n$ is the set of operations of $n$-ary operations of the associated signature. Such a quotient is also called a **presentation $\langle \Sigma, E \rangle$ of $\mathcal{T}$ by operations and equations** and the equations defining $\mathcal{T}$ are the kernel of $E_X$ (for some countably infinite set of 'variables' $X$) (for more on set-functors and their presentations see Adámek and Trnková [9]).

Here is a first example of presentations.

**Example 5.1.2.** The finite powerset functor $\mathsf{Pow}_\omega$ has the *canonical presentation*

$$\coprod_{n<\omega} \mathsf{Pow}_\omega(n) \times X^n \longrightarrow \mathsf{Pow}_\omega(X)$$

and the ***List***-*presentation*

$$\mathsf{List}(X) = \coprod_{n<\omega} X^n \longrightarrow \mathsf{Pow}_\omega(X).$$

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

In the previous example, the term "canonical" is not arbitrary. Using the Yoneda lemma we can show that every finitary functor (Definition 2.1.5) has a finitary presentation.

**Proposition 5.1.3.** *Every finitary functor $\mathcal{T}$ has a* ***canonical presentation*** $(\Sigma_{\mathcal{T}}, E)$ *where*

$$\Sigma_{\mathcal{T}}(X) = \coprod_{n<\omega} \mathcal{T}(n) \times X^n \xrightarrow{\ E_X\ } \mathcal{T}(X).$$

*The function $E_X$ maps a pair $(p, a) \in \mathcal{T}(n) \times X^n$ to $E_X(p, a) = \mathcal{T}(a)(p)$.*

*Proof.* The Yoneda lemma ensures the assignation $E_X(p, a) = \mathcal{T}(a)(p)$ to be natural. In order to show that $(\Sigma_{\mathcal{T}}, E)$ is a presentation of $\mathcal{T}$, it is only left to show that each $E_X$ is surjective. In order to show this we will use the fact that $\mathcal{T}$ is a finitary functor (Definition 2.1.5). More precisely, recall that the action of $\mathcal{T}$ can be described by $\mathcal{T}(X) = \bigcup\{\mathcal{T}(Y) \mid Y \subseteq X, |Y| < \omega\}$, see Definition 2.1.5 and Remark 2.1.7. Using this we can prove surjectivity of $E_X$ as follows: since $\mathcal{T}$ is finitary, each $t \in \mathcal{T}(X)$ belongs to $\mathcal{T}(Y_t)$ for some finite subset of $X$; more precisely there is $t' \in \mathcal{T}(Y_t)$ such that $\mathcal{T}(i)(t') = t$, where $i$ is the inclusion map from $Y_t$ to $X$. Say that the cardinality of $Y_t$ is $n$; let $f : n \longrightarrow Y_t$ be the associated bijection, and write $p$ for $\mathcal{T}(f)^{-1}(t')$. It is now straightforward to check that $E_X(p, if) = t$, indeed

$$\begin{aligned} E_X(p, if) &= \mathcal{T}(if)(p) && \text{(Def. } E\text{)} \\ &= \mathcal{T}(i)\mathcal{T}(f)\mathcal{T}(f)^{-1}(t') && \text{(Def. } p\text{)} \\ &= \mathcal{T}(i)(t') && (\mathcal{T}(f)\mathcal{T}(f)^{-1} = id \text{ because } f \text{ is a bijection}) \\ &= t. && \text{(assumption)} \end{aligned}$$

In other words, each $E_X$ is surjective. This concludes the proof. □

We highlight the previous construction.

**Definition 5.1.4.** Let $\mathcal{T}$ be a finitary functor. The *canonical presentation* of $\mathcal{T}$ is given by $(\Sigma_{\mathcal{T}}, E)$ where $\Sigma_{\mathcal{T}} = \coprod_{n<\omega} \mathcal{T}(n) \times X^n$ and $E_X$ maps a pair $(p, a) \in \mathcal{T}(n) \times X^n$ to $E_X(p, a) = \mathcal{T}(a)(p)$.

As it was shown in the case of the power set functor (Example 5.1.2), the canonical presentation of a functor is not always the most "natural" one. However, the canonical presentation is maximal among presentations in the sense that it contains the operations of any other presentation. In the case of the power set functor this can be seen because we can identify the list $[x_1, \ldots x_n]$ of the **List**-presentation with $(\{1, \ldots n\}, [x_1, \ldots x_n])$ of the canonical presentation. This being said, a more general observation is: Any presentation is a restriction of the canonical presentation. The next lemma makes this precise.

**Lemma 5.1.5.** *Let $\langle \Sigma, E \rangle$ be a presentation for a functor $\mathcal{T}$. There are canonical maps $s_n : \Sigma_n \longrightarrow \mathcal{T}(n)$ such that for all $p \in \Sigma_n$ and all $a : n \longrightarrow X$ we have*

$$E_X(p, a) = \mathcal{T}(a)(s_n(p)).$$

*In other words, the action of $E$ reduces to that of the canonical presentation.*

*Proof.* This is essentially an application of the Yoneda lemma. Notice that each $p \in \Sigma_n$ corresponds to a natural transformation $E(p, -) : (-)^n \longrightarrow \mathcal{T}$ and then by the Yoneda Lemma these natural transformations are in natural bijection with the elements of $\mathcal{T}(n)$. We now make the computations explicit. Consider $E_n : \coprod_{k<\omega} \Sigma_k \times n^k \longrightarrow \mathcal{T}(n)$. For $p \in \Sigma_n$, define $s_n : \Sigma_n \longrightarrow \mathcal{T}(n)$ by $s_n(p) = E_n(p, id_n)$. Since $E_X$ is natural in $X$, for each $a : n \longrightarrow X$ the following diagram

$$\begin{array}{ccc} n^n & \xrightarrow{E_n(p,-)} & \mathcal{T}(n) \\ {\scriptstyle a \circ -}\downarrow & & \downarrow{\scriptstyle \mathcal{T}(a)} \\ X^n & \xrightarrow[E_X(p,-)]{} & \mathcal{T}(X) \end{array}$$

commutes. Hence we have $E_X(p, a) = \mathcal{T}(a)(s_n(p))$, which proves the claim. □

The previous lemma can be rephrased as follows: given any presentation $\langle\Sigma, E\rangle$ of $\mathcal{T}$, we can identify each operation in $\Sigma_n$ with an operation in $\mathcal{T}(n)$, i.e. an operation of the canonical presentation, in a canonical manner. We highlight this property in the next definition

**Definition 5.1.6.** Let $\langle\Sigma, E\rangle$ be a presentation for a functor $\mathcal{T}$. We say that $\langle\Sigma, E\rangle$ is *standard* if for every $n$ we have $\Sigma_n \subseteq \mathcal{T}(n)$ and for each $(p, a) \in \Sigma_n \times X^n$ we have $E_X(p, a) = \mathcal{T}(a)(p)$.

**Assumption:** Because of the previous lemma from now on we assume all presentations to be standard. More explicitly, we always assume the $n$-ary operation of a presentation are given by the image of $s_n : \Sigma_n \longrightarrow \mathcal{T}(n)$.

We fix some terminology and notation before proceeding.

**Definition 5.1.7.** Given a presentation $\langle\Sigma, E\rangle$ we say that $(p, a)$ represents $t \in \mathcal{T}(X)$, or that $(p, a)$ is a *representant* of $t$, if $E(p, a) = t$.

To emphasise the equational axiomatisation given by a presentation of $\mathcal{T}$ we introduce the following notation.

**Notation.** Let $\langle\Sigma, E\rangle$ be a presentation for $\mathcal{T}$ and let $(p, a), (q, b) \in \Sigma(X) = \coprod_{n<\omega} \Sigma_n \times X^n$. We write $p(a)$ for $(p, a)$ and $q(b)$ for $(q, b)$, this to emphasise that $p$ and $q$ can be seen as operators acting on lists.

We write $\approx_{\mathcal{T}}$ for the equivalence relation induced by $E$. More explicitly,

$$p(a) \approx_{\mathcal{T}} q(b) \quad \text{iff} \quad E_X(p, a) = E_X(q, b) \text{ ( i.e. iff } \mathcal{T}(a)(p) = \mathcal{T}(b)(q) \text{ ).}$$

Note that $\approx_{\mathcal{T}}$ depends on the given presentation of $\mathcal{T}$, so in case of danger of confusion we write $\approx_{\langle\Sigma,E\rangle}$ instead.

Here are some examples to get familiar with the terminology.

**Example 5.1.8.**

1. For $\mathcal{T} = 1 + \mathsf{Id}$, the identity

$$\Sigma(X) = 1 + X \xrightarrow{E_X = id_X} 1 + X = \mathcal{T}(X)$$

is itself a presentation of $\mathcal{T}$; in this case $\approx_{\mathsf{Id}}$ is equality.

2. For the functor $\mathcal{T} = 1 + \mathsf{Id}$ the canonical presentation

$$\Sigma_{\mathcal{T}}(X) = \coprod_{n<\omega} (1 + n) \times X^n \xrightarrow{E_X} 1 + X = \mathcal{T}(X)$$

maps a pair $(p, a) \in (1 + n) \times X^n$ to $* \in 1 + X$ in case $p \in 1$ or to $a_p$, the evaluation of $a$ in $p$, in case $p \in n$. The congruence relation is then $p(a) \approx_{\mathcal{T}} q(b)$ iff $p = q = *$ or $a_p = b_q$.

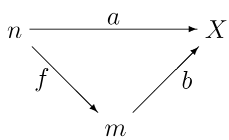
3. In the case of the canonical presentation for $\mathsf{Pow}$ the relation $\approx_{\mathcal{T}}$ can be described as follows: for a pair of elements $(p, a) \in \mathsf{Pow}(n) \times X^n$ and $(p, q) \in \mathsf{Pow}(m) \times X^m$ we have $p(a) \approx_{\mathcal{T}} q(b)$ iff $\{a_i \,|\, i \in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

4. In the case of the $\mathsf{List}$-presentation of $\mathsf{Pow}$, the relation $\approx_{\mathsf{List}}$ has the following characterization: for $a \in X^n$ and $b \in X^m$ we have $a \approx_{\mathsf{List}} b$ iff $\{a_i \mid i \in n\} = \{b_j \mid j \in m\}$, i.e. $a$ and $b$ have the same image.

5. For $\mathcal{T} = \mathcal{B}_{\mathbb{N}}$ the canonical presentation can be described as follows: a pair $(p, a) \in \mathcal{B}_{\mathbb{N}}(n) \times X^n$ is mapped to the bag $b : X \longrightarrow \mathbb{N}$ mapping $x$ to $\sum_{\{i|a_i=x\}} p_i$, where $p_i$ denotes the image of $i$ under $p : \mathbb{N} \longrightarrow X$. The relation $\approx_{\mathcal{T}}$ can be described as follows: $p(a) \approx_{\mathcal{T}} q(b)$ for $p : n \longrightarrow \mathbb{N}$, $q : m \longrightarrow \mathbb{N}$ iff there is a matrix $(r_{ij})_{1\leq i\leq n, 1\leq j\leq m}$ such that $a_i \neq b_j \Rightarrow r_{ij} = 0$ and $\sum_i r_{ij} = q_j$ and $\sum_j r_{ij} = p_i$. For example, $[3, 2](x, y) \approx_{\mathcal{T}} [2, 1, 1, 1](x, y, x, y)$. Compare this with Example 3.3.6. The case of probability distributions is similar.

The following application of Lemma 5.1.5 will be useful later.

**Proposition 5.1.9.** *Let $\langle \Sigma, E\rangle$ be a (standard) presentation of $\mathcal{T}$ and assume the following diagram commutes*




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

*Proof.* The proposition is immediate because presentations are assumed to be standard. Hence $\Sigma_n \subseteq \mathcal{T}(n)$ and for each $(p, a) \in \Sigma_n \times X^n$ we have $E_X(p, a) = \mathcal{T}(a)(p)$. □

### Two uses of presentations

We now illustrate two uses of presentations. The first one concerns relation lifting; the second one concerns the relation between monads and varieties. First, we will use presentations to "simplify" the calculation of relation liftings. This will be a key technical tool in Section 5.3 and it is interesting in its own. Second, we use presentations to clarify the relation between algebras for a functor, and algebraic theories. This is very well know in the world of categorical algebra, see e.g. [8] for a detailed account. We add it here as an illustration, for the sake of completeness, and because the idea will be reused in our representation theorem in Section 5.2.

As said before, we can use presentations to compute relation liftings (Section 2.2); the idea is to "hide" the functor $\mathcal{T}$. This is formalised in the following technical lemma, which is a key stone for the development of our equational coalgebraic logic (Section 5.3).

**Lemma 5.1.10.** *Let $\langle \Sigma, E \rangle$ be a presentation for a finitary endofunctor $\mathcal{T}$ on* Set *and let $R$ be a relation between $X$ and $Y$. For every $t_x \in \mathcal{T}(X)$ and $t_y \in \mathcal{T}(Y)$ the following conditions are equivalent:*

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

More informally, we read the lemma as

$$t_x \, \overline{\mathcal{T}}(R) \, t_y \ \text{ iff } \ t_x \approx_{\mathcal{T}} r(a_1, \ldots a_k) \text{ and } t_y \approx_{\mathcal{T}} r(b_1, \ldots b_k) \text{ and } (\forall i)(a_i R b_i) \tag{5.2}$$

for some $k$-ary operation $r$.

*Proof.* The proof of the lemma follows from the following commutative diagram

$$\begin{array}{ccccc} \Sigma(X) & \xleftarrow{\Sigma(\pi_X)} & \Sigma(R) & \xrightarrow{\Sigma(\pi_Y)} & \Sigma(Y) \\ \Big\downarrow E_X & & \Big\downarrow E_R & & \Big\downarrow E_Y \\ \mathcal{T}(X) & \xleftarrow[\mathcal{T}(\pi_X)]{} & \mathcal{T}(R) & \xrightarrow[\mathcal{T}(\pi_Y)]{} & \mathcal{T}(Y) \end{array} \tag{5.3}$$

taking into account that $E_R$ is surjective. The diagram commutes because $E$ is a natural transformation.

More explicitly, from the definition of relation lifting (Definition 2.2.7) we have that $t_x\overline{\mathcal{T}}(R)t_y$ iff there exists $t \in \mathcal{T}(R)$ such that $\mathcal{T}(\pi_X)(t) = t_x$ and $\mathcal{T}(\pi_Y)(t) = t_y$. Since $E_R$ is surjective, there exists $(r, c) \in \Sigma(R)$ such that $E_R(r, c) = t$. Since $\Sigma$ is a polynomial functor, there exists $k \leq \omega$ such that $(r, c) \in \Sigma_k \times R^k$. These are the $k$ and $r$ required in the statement of the lemma. The functions $a$ and $b$ are obtained by composing $k \xrightarrow{c} R \hookrightarrow X \times Y$ with the respective projections; Notice that $(\forall i \in k)(a_i \, R \, b_i)$ because $c$ has codomain $R$. The commutativity of the diagram says that $E_X(r, a) = t_x$ and $E_Y(r, b) = t_y$. This concludes the proof. □

As we said before, depending on the functor $\mathcal{T}$, relation liftings can be quite complicated, see e.g. the case of distributions in Example 3.3.6. But for polynomial functors relation lifting amounts to take the relation componentwise. The importance of Equation (5.2) is that it presents the relation lifting for $\mathcal{T}$ componentwise, i.e. in the form of a relation lifting for a polynomial functor, modulo the equational theory $\approx_{\mathcal{T}}$. In other words, Equation (5.2) "hides" $\mathcal{T}$ using the equational theory. Polynomial functors can in fact be characterised as those functors which have a presentation such that $\approx_{\langle \Sigma, E\rangle}$ is equality.

We now move on to the second illustration. Presentations can be used to extract equational theories from functors. More explicitly, for every finitary functor $\mathcal{T} : \mathsf{Set} \to \mathsf{Set}$ the category $\mathsf{Alg}(\mathcal{T})$ is a variety for an algebraic signature, hence by Birkhoff's theorem, $\mathsf{Alg}(\mathcal{T})$ is an equational class. An algebraic signature for $\mathsf{Alg}(\mathcal{T})$ is given by the canonical presentation of $\mathcal{T}$. In general every presentation $\langle \Sigma, E\rangle$ of $\mathcal{T}$ provides a signature for $\mathsf{Alg}(\mathcal{T})$, namely the one given by $\Sigma$, see Section 2.3. Now the natural transformation $E$ induces a functor

$$- \circ E : \mathsf{Alg}(\mathcal{T}) \to \mathsf{Alg}(\Sigma).$$

Since $E$ is surjective, $- \circ E$ is an embedding; compare this with the translation functor given by a one-step translation. It follows from the general properties of algebras for a functor, Section 2.3, that $\mathsf{Alg}(\mathcal{T})$ is closed under products, homomorphic images and subalgebras i.e. it is a variety.

This exhibits algebras for a functor and equational theories that can be axiomatized by rank 1 axioms as two sides of the same coin. The general picture is given by monads.

**Theorem 5.1.11.** *Varieties for finitary equational algebraic theories are, up to concrete isomorphism, the categories of Eilenberg-Moore algebras (Definition 2.3.17) for finitary monads on* $\mathsf{Set}$.

Details can be found in [8]; the algebraic signature is obtained as described above.

### 5.1.1 Moss Liftings

We have now arrived at one of the key applications of presentations of functors. We will use presentations to produce predicate liftings which by construction will be translatable into the Moss logic. In Section 5.3, we will use these predicate liftings to develop an equational coalgebraic modal logic. In the next section we will generalize presentations and use the techniques here to show that any coalgebraic modal logic (Definition 3.2.13) can be translated into the language of predicate liftings.

The idea is that given a presentation $\langle \Sigma, E\rangle$ we can use the natural transformation $E$ to define predicate liftings; we call such predicate liftings *Moss liftings*. The next definition makes this precise.

**Definition 5.1.12.** Let $\mathcal{T}$ be a weak-pullback preserving functor and let $\langle \Sigma, E\rangle$ be a (standard) presentation of $\mathcal{T}_\omega$. Each $p \in \Sigma_n$ gives rise to an $n$-ary predicate lifting $\lambda^p$ as shown by the next diagram:

$$\begin{array}{ccc} \mathcal{P}^n & \xrightarrow{E_{\mathcal{P}}(p,-)} & \mathcal{T}_\omega\mathcal{P} \\ & {\scriptstyle \lambda^p}\searrow \quad \swarrow{\scriptstyle \blacktriangledown} & \\ & \mathcal{P}\mathcal{T} & \end{array} \tag{5.4}$$

where $\blacktriangledown$ is as defined in Definition 3.3.4.

We call a predicate lifting arising in this way a $\langle\Sigma, E\rangle$-**Moss lifting**, or simply a Moss lifting. By Lemma 5.1.5 the set of Moss liftings for a presentation $\langle\Sigma, E\rangle$ can be identified with a subset of $\coprod_{n<\omega} \mathcal{T}_\omega(n)$. Notice that by construction $E_{\mathcal{P}}(p, -) : \mathcal{P}^n \longrightarrow \mathcal{T}_\omega\mathcal{P}$ is a translator for $\lambda^p$.

The intuition behind Moss liftings can be described as follows: Recall, Section 3.3.1, that the rough idea of the Moss Logic is to use the functor $\mathcal{T}$ as a modality. A presentation $\langle\Sigma, E\rangle$ of $\mathcal{T}$ decomposes $\mathcal{T}$ using an usual algebraic signature. Hence the operations in $\Sigma$ can also be interpreted as modalities, via $\mathcal{T}$.

We now illustrate Moss liftings with some examples.

**Example 5.1.13.** 1. Let $\mathcal{T} = 1 + \mathsf{Id}$. In the case of the canonical presentation, for each arity $n$ there is a Moss lifting $\lambda^*_n$, which indicates termination; this lifting corresponds to the unique element of 1. All other Moss liftings of arity $n$ correspond to the elements of $n$. For $p \in n$, the Moss lifting $\lambda^p$ maps a sequence $\varphi : n \longrightarrow \mathcal{P}(X)$ to the set $\varphi_p$. Using the identity presentation for $(1 + \mathsf{Id}, id)$, we see that one constant and one unary predicate lifting suffice to describe $\mathcal{T}$-coalgebras.

2. Let $\mathcal{T} = \mathsf{Pow}$. In the case of the canonical presentation, the Moss liftings of arity $n$ are associated with subsets of $n$. Let $p$ be one of those subsets. The Moss lifting $\lambda^p$ maps a sequence $\varphi : n \longrightarrow \mathcal{P}X$ to the set

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

Compare this with the usual definition of $\blacktriangledown$ (Example 3.3.6).

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

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

4. Let $\mathcal{T}$ be the finite multiset functor. A Moss liftings of arity $n$ for the canonical presentation corresponds to a bag $p : n \longrightarrow \mathbb{N}$. The associated predicate lifting maps $\varphi : n \longrightarrow \mathcal{P}X$ to a multiset over $\mathcal{P}(X)$ (Example 5.1.8(5)) followed by an application of $\blacktriangledown$ (Example 3.3.6).

The next remark presents another perspective on Moss liftings.

**Remark 5.1.14.** Using the Yoneda Lemma Moss liftings can also be characterised as follows: Consider the $n$ component of $\blacktriangledown$ and write $2^n$ for $\mathcal{P}(n)$, i.e. consider the function $\blacktriangledown_n : \mathcal{T}(2^n) \longrightarrow 2^{\mathcal{T}(n)}$; this function curries into

$$s : \mathcal{T}(n) \times \mathcal{T}(2^n) \longrightarrow 2.$$

Evaluating at $p \in \mathcal{T}(n)$ gives us the Moss lifting $\lambda^p$. Using properties of adjoints, Lemma A.1.2, we can actually show that $s = ev \circ (id_{\mathcal{T}(n)}, \blacktriangledown_n)$ where $ev : \mathcal{T}(n) \times 2^{\mathcal{T}(n)} \longrightarrow 2$ is the evaluation function.

We now fix some terminology concerning the logic of Moss liftings.

**Definition 5.1.15.** Let $\mathcal{T}$ be a weak-pullback preserving functor and let $\langle \Sigma, E \rangle$ be a presentation of $\mathcal{T}_\omega$. The signature of $\langle \Sigma, E \rangle$-Moss liftings is denoted by $\Sigma^E_{\mathcal{T}}$. The associated coalgebraic modal logic (Definition 3.2.18) is denoted by $(K^{\langle \Sigma, E \rangle}_{\mathcal{T}}, \delta_E)$. The language of the logic (Remark 3.2.15) is denoted by $\mathcal{K}^{\langle \Sigma, E \rangle}_{\mathcal{T}}$. We write $\mathcal{K}_{\mathcal{T}}$ and $(K_{\mathcal{T}}, \delta_E)$ if the presentation is clear from the context.

The next remark recalls different descriptions for the functor $K_{\mathcal{T}}$.

**Remark 5.1.16.** Let $\mathcal{T}$ be a weak-pullback preserving functor and let $\langle \Sigma, E \rangle$ be a presentation of $\mathcal{T}_\omega$. Notice that the signature of Moss liftings is literally $\Sigma$, hence the functor $K_{\mathcal{T}}$ is isomorphic to $F\Sigma U$ (Definition 3.2.18). This functor can also be described by $K_{\mathcal{T}} = F(\coprod_{n<\omega} \coprod_{\Sigma_n} U^n)$.

As said before, and made explicit in the definition, all Moss liftings have translators. Consequently, they can be $\mathsf{BA}$-translated into the Moss logic. However, we can do better: Moss liftings are always translatable.

**Proposition 5.1.17.** *Moss liftings for a (finitary) functor $\mathcal{T}$, which preserves weak-pullbacks, are totally translatable into Moss logic, i.e. for any category of power set algebras $\mathcal{A}$ and every Moss lifting $\lambda$ there exists an $\mathcal{A}$-logical translator for $\lambda$.*

*Proof.* Let $\langle \Sigma, E \rangle$ be a presentation of $\mathcal{T}_\omega$. For $p \in \Sigma_n$ let $\lambda^p$ be the corresponding Moss lifting. Given a category of power set algebras $\mathcal{A}$ we want to find an $\mathcal{A}$-logical translator for $\lambda^p$. By definition the natural transformation $E(p, -) : \mathsf{Id}^n \longrightarrow \mathcal{T}$ is a $\mathsf{Set}$-logical translator for $\lambda^p$, extending $E_{\mathcal{P}}(p, -)$ (Definition 4.3.2). We can also restrict $E(p, -)$ with any functor $U : \mathcal{A} \longrightarrow \mathsf{Set}$ and obtain a natural transformation $E_U(p, -) : U \longrightarrow \mathcal{T}U$; this exhibit an $\mathcal{A}$-logical translator for $\lambda^p$ for any category $\mathcal{A}$ with powerset algebras because $UP = \mathcal{P}$.

The argument also works backwards, i.e. if an $n$-ary predicate lifting $\lambda$ has an $\mathcal{A}$-logical translator for any category of powerset algebras it is a Moss lifting, because it should then have a $\mathsf{Set}$-logical translator; by definition, such logical translator is given by a natural transformation $(-)^n \longrightarrow \mathcal{T}$. By Yoneda lemma this natural transformation correspond to an element of $\mathcal{T}(n)$, say $p$. Therefore $\lambda = \lambda^p$, i.e. it is a Moss lifting for the canonical presentation of $\mathcal{T}$. □

In summary, Moss liftings are the only predicate liftings that can be translated independently of the underlying propositional logic. For this reason they may be called **totally-translatable**.

**Corollary 5.1.18.** *Let $\mathcal{T}$ be a finitary weak-pullback preserving functor. The Moss liftings for $\mathcal{T}$ are precisely those predicate liftings that have a* $\mathsf{Set}$*-logical translator.*

Proposition 4.3.6 implies that there is a one step translation from the logic of Moss Liftings into the Moss logic.

**Corollary 5.1.19.** *Let $\mathcal{A}$ be a category of power set algebras. Let $\langle \Sigma, E\rangle$ be a presentation for $\mathcal{T}$. Let $(K_{\mathcal{T}}, \delta_E)$ be the logic of Moss liftings and let $(M, \nabla)$ be the Moss logic, both over $\mathcal{A}$.*

*There is a one step translation $\bar{\nu} : (K_{\mathcal{T}}, \delta_E) \longrightarrow (M, \nabla)$. More concretely $\bar{\nu} = F(E_U)$, where $U : \mathcal{A} \longrightarrow \mathsf{Set}$ is the forgetful functor and $F$ is its left adjoint.*

An important property of Moss liftings is that they are monotone. The next proposition makes this precise.

**Proposition 5.1.20.** *Every Moss lifting $\lambda^p : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ is monotone. This means that given sequences of sets $\varphi, \psi : n \longrightarrow \mathcal{P}X$, if $(\forall i)(\varphi_i \subseteq \psi_i)$ then $\lambda^p(\varphi) \subseteq \lambda^p(\psi)$.*

*Proof.* Let $E(p, -)$ be the translator of $\lambda^p$. Using Lemma 5.1.10 we see that $(\forall i)(\varphi_i \subseteq \psi_i)$ implies $E_{\mathcal{P}}(p, \varphi)\, \overline{\mathcal{T}}(\subseteq)\, E_{\mathcal{P}}(p, \psi)$. Using one more time Lemma 5.1.10 we can see this "inclusion" is equivalent to say that for every $t \in \mathcal{T}\mathcal{P}(X)$

$$\text{if } t\overline{\mathcal{T}}(\in_X)E_{\mathcal{P}}(p, \varphi) \text{ then } t\overline{\mathcal{T}}(\in_X)E_{\mathcal{P}}(p, \psi).$$

By definition of $\blacktriangledown$ we have $t\overline{\mathcal{T}}(\in_X)E_{\mathcal{P}}(p, \varphi)$ iff $t \in \blacktriangledown E_{\mathcal{P}}(p, \varphi)$. Hence from the line above we conclude: if $t \in \blacktriangledown E_{\mathcal{P}}(p, \varphi)$ then $t \in \blacktriangledown E_{\mathcal{P}}(p, \psi)$ this means $\lambda^p(\varphi) \subseteq \lambda^p(\psi)$ as we wanted to show. □

One more property of the logic $(K_{\mathcal{T}}, \delta_E)$ is that Moss logic can be translated into it. Moreover, there is a translation in the sense of Definition 2.4.1.

**Proposition 5.1.21** ([80]). *For every formula in $\mathcal{M}_{\mathcal{T}}$ there exists an equivalent formula in $\mathcal{K}_{\mathcal{T}}$. More explicitly, for every $\varphi \in \mathcal{M}_{\mathcal{T}}$ there exists $\psi \in \mathcal{K}_{\mathcal{T}}$ such that $[\![\varphi]\!]_\xi^{\mathcal{M}_{\mathcal{T}}} = [\![\psi]\!]_\xi^{\mathcal{K}_{\mathcal{T}}}$ for every coalgebra $\xi$.*

The previous proposition is a particular instance of Theorem 5.2.17.

These two propositions have the following important corollary.

**Corollary 5.1.22.** *For every finitary weak pullback preserving functor $\mathcal{T}$ there exists a set $\Lambda$ of monotone predicate liftings such that the logic $(L_\Lambda, \delta_\Lambda)$ has the Hennessy-Milner property. The set $\Lambda$ is that of Moss liftings for $\mathcal{T}$.*

*Proof.* Since $\mathcal{T}$ preserves weak pullbacks we can define Moss logic. Since $\mathcal{T}$ is finitary we can define Moss liftings for a presentation, e.g. the canonical presentation. Proposition 5.1.21 implies that the language of Moss liftings is as expressive as Moss's language. Since the Moss logic has the Hennessy-Milner property [84] so does $\mathcal{K}_\mathcal{T}$. From Proposition 5.1.20 we know that all Moss liftings are monotone. □

**Remark 5.1.23.** Finding a monotone set of predicate liftings is important in coalgebraic modal logic, as it opens the possibility of adding fixpoint operators. The previous proposition solves this problem in the case of weak-pullback preserving functors. As far as we know, the general problem for non-weak pullback preserving functors is still open.

## 5.2 Concrete logics from abstract logics

In this section we will show how to obtain a logic of predicate liftings from any coalgebraic modal logic (Definition 3.2.13). The idea is to generalise the concept of a presentation (Definition 5.1.1) to functors $L : \mathcal{A} \longrightarrow \mathcal{A}$, over a category of algebras. This idea was developed in [22, 76]. Our contribution in this section is to use these ideas to translate coalgebraic modal logics into the logic of predicate liftings. More concretely, we prove two representation theorems (Theorem 5.2.2 and Theorem 5.2.17). Theorem 5.2.2 shows that every coalgebraic modal logic can be translated into the logic of all predicate liftings. Theorem 5.2.17 elaborates on this by showing that in fact every coalgebraic modal logic is a rank 1 axiomatization of a logic of predicate liftings. To the best of our knowledge these theorems have not appeared explicitly anywhere in the literature, with the sole exception of our work in [74]. The technicalities in this section require fitness in category theory.

### 5.2.1 The first representation Theorem

In this section we prove our first representation theorem which states that every coalgebraic modal logic can be translated into a logic of predicate liftings. We first prove a lemma concerning categories of powerset algebras.

**Lemma 5.2.1.** *Let $\mathcal{A}$ be a category of power set algebras; let $U : \mathcal{A} \longrightarrow \mathsf{Set}$ be the forgetful functor and let $F$ be its left adjoint. Let $S : \mathcal{A} \longrightarrow \mathsf{Set}^{op}$ be the left adjoint of $P : \mathsf{Set}^{op} \longrightarrow \mathcal{A}$ (Proposition 3.2.24), i.e. $S(A) = \mathcal{A}(A, P(1)$.*

*For every finite set $n$ we have $SF(n) = \mathcal{P}(n)$.*

*Proof.* The statement can be proved by a direct computation.

$$\begin{aligned} SF(n) &= \mathcal{A}(F(n), P(1)) && \text{(Definition } S) \\ &= \mathsf{Set}(n, UP(1)) && (F \dashv U) \\ &= \mathsf{Set}(n, \mathcal{P}(1)) && (\mathcal{P} = UP) \\ &= \mathcal{P}(n). && (\mathcal{P}(1) = 2) \end{aligned}$$

□

Before stating and proving the first representation theorem we recall the definition of the logic $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$ (Definition 3.3.13) and make some observations that will be needed in the argument.

The functor $L_{\mathcal{T}} : \mathcal{A} \longrightarrow \mathcal{A}$ is defined on finitely generated free algebras by $L_{\mathcal{T}}(F(n)) = P\mathcal{T}\mathcal{P}(n)$ and extended to arbitrary $A \in \mathcal{A}$ via colimits (Definition 2.3.12). In the light of the previous lemma we have

$$L_{\mathcal{T}}F(n) = P\mathcal{T}SF(n). \tag{5.5}$$

The semantics $(\delta_{\mathcal{T}})_X : L_{\mathcal{T}}P(X) \longrightarrow P\mathcal{T}(X)$ is the unique arrow making the following diagram

$$\begin{array}{ccc} L_{\mathcal{T}}P(X) & \overset{(\delta_{\mathcal{T}})_X}{\dashrightarrow} & P\mathcal{T}(X) \\ {\scriptstyle L_{\mathcal{T}}(c_i)}\uparrow & & \uparrow{\scriptstyle P\mathcal{T}(\widehat{c_i})} \\ L_{\mathcal{T}}F(n_i) & \underset{id}{\longrightarrow} & P\mathcal{T}\mathcal{P}(n_i) \end{array} \tag{5.6}$$

commute for each $i : n_i \longrightarrow UP(X)$; where $\widehat{c_i} : \mathcal{P}(n) \longrightarrow X$ is the $\mathcal{P}$-transpose of $i$ and $c_i : F(n_i) \longrightarrow X$ is the $F$-transpose of $i$. On the light of the previous lemma we have

$$\widehat{c_i} = S(c_i) \circ \varepsilon_X \tag{5.7}$$

where $\varepsilon_X : X \longrightarrow SP(X)$ is the counit of the adjunction. More formally we really have $\widehat{c} = \varepsilon_X \bullet S(c)$ where $\bullet$ is the composition in $\mathsf{Set}^{op}$. Moreover notice that from Lemma A.1.2, item 4, we have $c_i = P(\widehat{c_i}) \circ \eta_{F(n)}$.

We can now state and prove the first representation theorem.

**Theorem 5.2.2.** *For any coalgebraic modal logic $(L, \delta)$ for a functor $\mathcal{T}$, over a category of power set algebras $\mathcal{A}$, there is a one-step translation*

$$\nu : (L, \delta) \longrightarrow (L_{\mathcal{T}}, \delta_{\mathcal{T}})$$

*where $(L_{\mathcal{T}}, \delta_{\mathcal{T}})$ is the logic of all predicate liftings for $\mathcal{T}$ (Definition 3.3.13).*

*In other words, every coalgebraic modal logic can be translated into a logic of predicate liftings.*

*Proof.* We want to define a one-step translation $\nu : (L, \delta) \longrightarrow (L_{\mathcal{T}}, \delta_{\mathcal{T}})$. Since $L$ and $L_{\mathcal{T}}$ are determined by finitely presented algebras (Definition 2.3.12), we first define a natural transformation $\nu' : LF(n) \longrightarrow L_{\mathcal{T}}F(n)$ and then extend $\nu'$ to a natural transformation $\nu : L \longrightarrow L_{\mathcal{T}}$, see Proposition 2.3.14.

We now explain how to define $\nu'$. The underlying idea is to use the adjunction $S \dashv P$ and $\delta$. Consider the following transpose

$$L \xrightarrow{L(\eta)} LPS \xrightarrow{\delta_S} P\mathcal{T}S, \tag{5.8}$$

where $\eta : \mathsf{Id}_{\mathcal{A}} \longrightarrow PS$ is the unit of the adjunction. Precomposing this natural transformation with $F$ and using the observation in Equation (5.5) we obtain $\nu'$ as follows

$$LF(n) \xrightarrow{\nu' = \delta_{SF(n)} \circ L(\eta_{F(n)})} PTSF(n) = L_T F(n)$$

As said before, since both $L$ and $L_T$ are determined by their action on finitely generated free algebras, we can extend $\nu'$ to a natural transformation $\nu : L \longrightarrow L_T$ which is in fact a one-step translation.

We now show that $\nu$ is a one-step translation, i.e we argue $\delta = \nu_P \circ \delta_{\mathcal{T}}$. For this purpose, consider a morphism $c_i : F(n_i) \longrightarrow P(X)$ and the following diagram

$$\begin{array}{ccccc}
LP(X) & \overset{\nu_{P(X)}}{\dashrightarrow} & L_{\mathcal{T}}P(X) & \overset{\delta_{\mathcal{T}}}{\dashrightarrow} & P\mathcal{T}(X) \\
\uparrow {\scriptstyle L(c_i)} & & \uparrow {\scriptstyle L_{\mathcal{T}}(c_i)} & & \uparrow {\scriptstyle P\mathcal{T}(\widehat{c_i})} \\
LF(n_i) & \underset{\nu'}{\longrightarrow} & L_{\mathcal{T}}F(n_i) & \underset{id}{\longrightarrow} & P\mathcal{T}\mathcal{P}(n_i)
\end{array}$$

where $\widehat{c_i} = S(c_i) \circ \varepsilon_X$. By definition, $\nu_{P(X)}$ is the only arrow that makes the rectangle on the left commute for every $c_i : F(n_i) \longrightarrow P(X)$; also by definition $\delta_{\mathcal{T}}$ is the only arrow that makes the rectangle on the right commute. Therefore, by the universal property of colimits, their composition is the only arrow that makes the outer rectangle commute for every $c_i$ (Proposition 2.3.14). Hence to prove $\delta = \delta_{\mathcal{T}} \circ \nu_P$ it is enough to prove that $\delta$ also makes the outer rectangle commute for every $c_i$, i.e. $\delta \circ L(c_i) = P\mathcal{T}(\widehat{c_i}) \circ \nu'$. We now show that this is the case.

$$\begin{aligned}
P\mathcal{T}(\widehat{c_i}) \circ \nu' &= P\mathcal{T}(\widehat{c_i}) \circ \delta_{SF(n)} \circ L(\eta_{F(n)}) && \text{(Def. } \nu'\text{)} \\
&= \delta_X \circ LP(\widehat{c_i}) \circ L(\eta_{F(n)}) && \text{(naturality of } \delta\text{)} \\
&= \delta_X \circ L\big(P(\widehat{c_i}) \circ \eta_{F(n)}\big) && (L \text{ is a functor}) \\
&= \delta_X \circ L(c_i) && \text{(observation after eq. 5.7)}
\end{aligned}$$

From here, by the universal property of coproducts, we conclude $\delta = \delta_{\mathcal{T}} \circ \nu_P$. This means that $\nu$ is a one-step translation. This concludes the proof. □

Here is an illustration using the Moss logic for Kripke frames.

**Example 5.2.3.** Let $\mathcal{T} = \mathsf{Pow}$ and let $(M, \nabla)$ be the Moss logic over $\mathsf{BA}$. By definition, in the particular case of Moss logic the natural transformation in Equation (5.8) is given by the following composite

$$F\mathcal{T}_\omega U \xrightarrow{F\mathcal{T}_\omega U(\eta)} F\mathcal{T}_\omega UPS \xrightarrow{\nabla_S} P\mathcal{T}S.$$

To the describe this natural transformation it is enough to describe its transpose which is given by

$$\mathcal{T}_\omega U \xrightarrow{\mathcal{T}_\omega U(\eta)} \mathcal{T}_\omega UPS \xrightarrow{\blacktriangledown_S} UP\mathcal{T}S,$$

where $\blacktriangledown_S$ is as described in Definition 3.3.4, the transpose of $\nabla$. Computing this last composite in the case $\mathcal{T} = \mathsf{Pow}$, we see that, for a boolean algebra $(A, \alpha)$, an element $\Phi \in \mathsf{Pow}U(A, \alpha)$ is mapped to

$$\{\Psi \subseteq \mathsf{Pow}S(A, \alpha) \mid (\forall a \in \Phi)(\exists u \in \Psi)(a \in u) \text{ and } (\forall u \in \Psi)(\exists a \in \Phi)(a \in u)\}.$$

The natural transformation $\nu' : M_\mathcal{T}F(n) \to L_\mathcal{T}F(n)$ is obtained by evaluating this last equation on finitely genrated free algebras. Since the base category is $\mathsf{BA}$ we have $F(n) = P\mathcal{P}(n)$. Hence we have $\nu'_{F(n)} = \blacktriangledown_{P\mathcal{P}(n)}$. In other words, the one-step translation

$$\nu : (M, \nabla) \to (L_{\mathsf{Pow}}, \delta_{\mathsf{Pow}})$$

from the previous theorem, which is obtained by extending $\nu'_{F(n)} = \blacktriangledown_{P\mathcal{P}(n)}$ via colimits (Proposition 2.3.14), induces the usual translation $tr(\nabla\varphi) = \Box \bigvee \Phi \wedge \bigwedge \Diamond\Phi$.

### 5.2.2 Presentations of functors on categories of algebras

We will now illustrate how to generalize presentations for set functors (Definition 5.1.1) to functors on categories of algebras. We will use this to generalize Moss Liftings and then prove our second representation Theorem 5.2.17, which roughly says that every coalgebraic modal logic is a rank one axiomatization of a logic of predicate liftings.

#### The canonical signature

We begin by showing how the idea of the canonical presentation (Definition 5.1.3) can be generalized to a category of algebras. Recall that for a functor $\mathcal{T} : \mathsf{Set} \to \mathsf{Set}$, in the canonical presentation we use the elements in $\mathcal{T}(n)$ as operations to present the functor $\mathcal{T}$. The first step to present a functor $L : \mathcal{A} \to \mathcal{A}$ is to find a suitable set of operations; the idea is not to take the $n$-ary operations from the set $ULF(n)$. The next proposition makes this idea precise.

**Proposition 5.2.4.** *Let $L$ be an endofunctor over an algebraic category $\mathcal{A}$; we write $U$ for the forgetful functor and $F$ for its left adjoint. For each set $n$, the natural transformations $U^n \longrightarrow UL$ are in natural bijection with the elements of the set $ULF(n)$.*

*Proof.* The proof is a direct computation.

$$\begin{aligned}
\mathsf{Nat}(U^n, UL) &\cong \mathsf{Nat}\Big(\mathsf{Set}(n, U), UL\Big), && (X^n = \mathsf{Set}(n, X)) \\
&\cong \mathsf{Nat}\Big(\mathcal{A}(F(n), -), UL\Big) && (F \dashv U), \\
&\cong ULF(n). && \text{(Yoneda)}.
\end{aligned}$$

More explicitly, given $p \in ULF(n)$ we define a natural transformation $Y(p) : U^n \longrightarrow UL$ as follows: A function $a : n \longrightarrow U(A)$ is mapped to $UL(\widehat{a})(p)$, where $\widehat{a} : F(n) \longrightarrow A$ is the transpose of $a$. □

Note that because $F \dashv U$ the elements of $ULF(n)$ are also in natural bijection with the natural transformations $FU^n \longrightarrow L$.

The previous proposition can also be stated by saying that there is natural transformation

$$E : \coprod_{n<\omega} (ULF(n) \times U^n) \longrightarrow UL. \tag{5.9}$$

Using $E$, we can "decompose" a coalgebraic modal logic $(L, \delta)$ into predicate liftings using the same idea of Moss liftings. More explicitly, each $p \in ULF(n)$ defines a $n$-ary predicate lifting $\lambda^p = U(\delta) \circ E_P(p, -)$ as shown in the following diagram

$$\begin{array}{ccc}
(UP)^n & \xrightarrow{\;E_P(p,-)\;} & ULP \\
& {\scriptstyle\lambda^p} \searrow \quad \swarrow {\scriptstyle U(\delta)} & \\
& UP\mathcal{T} &
\end{array} \tag{5.10}$$

Compare this with Diagram (5.4) for Moss liftings. These predicate liftings induce what we call the canonical signature of the coalgebraic modal logic. The next definition makes this precise.

**Definition 5.2.5.** Let $(L, \delta)$ be a coalgebraic modal logic, over a category of power set algebras $\mathcal{A}$, for a functor $\mathcal{T}$. The *canonical signature over* $\mathcal{A}$ of $(L, \delta)$, written $\Sigma_\delta$, is given by

$$\Sigma_\delta = \coprod_{n<\omega} ULF(n) \times (-)^n.$$

Each $p \in ULF(n)$ corresponds to the predicate lifting $\lambda^p : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ given by $\lambda^p = U(\delta) \circ E_{\mathcal{P}}(p, -)$, where $E$ is as in Equation (5.9).

Notice that the canonical signature will, in principle, add several new operations to a given signature. The following example describes the canonical signature for the Moss logic and for the logic given by a single predicate lifting.

**Example 5.2.6.** Let $\mathcal{T}$ be a set functor, and let $\mathcal{A}$ be a category of power set algebras.

1. Let $(M_{\mathcal{T}}, \nabla)$ be Moss logic for $\mathcal{T}$ over $\mathcal{A}$. The canonical signature, over $\mathcal{A}$, for the Moss logic contains all predicate liftings that have an $\mathcal{A}$-logical translator (Definition 4.3.2). Indeed if $\tau : U^n \longrightarrow \mathcal{T}_\omega U$ is an $\mathcal{A}$ translator for $\lambda$, by applying $F$ to $\tau$ we obtain a natural transformation $F(\tau) : FU^n \longrightarrow M_{\mathcal{T}}$ which by Proposition 5.2.4 corresponds to an element of $UM_{\mathcal{T}}F(n)$. This presents $\lambda$ as a predicate lifting in the canonical signature of $(M_{\mathcal{T}}, \nabla)$.
2. Let $\lambda : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ be a predicate lifting for $\mathcal{T}$. An $m$-ary predicate lifting $\gamma$ in the canonical signature over $\mathcal{A}$ of $(\bar{L}_\lambda, \bar{\delta}_\lambda)$ can be expressed by a formula of the form $\Box_\gamma \varphi = \Box_\lambda tr(\varphi)$, where $tr(\varphi)$ is a formula in the signature of $\mathcal{A}$. Compare this with the definition of translator and logical translator.

### Presentation of functors

This section is devoted to fine-tune the framework for a general result (Theorem 5.2.17) which says that every coalgebraic logic is a rank-1 axiomatization of a logic of predicate liftings, namely the one given by its canonical signature. The technical work on presentations is taken from [76]. Our contribution is to use this to define translations (Theorem, 5.2.17, Corollary 5.2.19). We begin with the definition of presentation by operations and equations.

**Definition 5.2.7** ([22])**.** Let $\mathcal{A}$ be a finitary algebraic category; we write $U$ for the forgetful functor and $F$ for its left adjoint.

A *finitary presentation by operations and equations*, over $\mathcal{A}$, or just presentation for short, is a pair $(\Sigma, E)$ where $\Sigma : \mathsf{Set} \longrightarrow \mathsf{Set}$ is a polynomial functor and $E$ is a family of sets $(E_n)_{n \leq \omega}$, where $E_n \subseteq UF\Sigma UF(n) \times UF\Sigma UF(n)$. The elements of $E_n$ are called equations (on $n$ variables).

We now explain in more detail the equations in Definition 5.2.7 to illustrate the concept of a presentation. Our aim is to explain that the equations above are rank-1 equations. For simplicity let us assume $\mathcal{A}$ is the category of algebras for an algebraic signature. The equations on $n$ variables are given by a subset

$$E_n \subseteq UF\Sigma UF(n) \times UF\Sigma UF(n).$$

Let us see how a generic element of $UF\Sigma UF(n)$ looks like. An element of $UF(n)$ is a term $t(x_1, \ldots, x_n)$ on $n$ variables in the algebraic signature of $\mathcal{A}$, e.g. $x_1 \wedge x_2$ in case $n = 2$. An element in $\Sigma UF(n)$ is a pair given by a modal operator $\Box_j$ and a term on $n$ variables; we can then think of it as $\Box_j t(x_1, \ldots, x_n)$, e.g. $\Box_j(x_1 \wedge x_2)$. Now an element in $UF\Sigma UF(n)$ is a a term, in the signature of $\mathcal{A}$, of those, e.g. $\Box_j(x_1) \wedge \Box_i(x_2)$ or $\Box_j(x_1 \wedge x_2)$. Finally an element of $E_n$ is a pair of those modal terms, i.e. an equation like $\Box_j(x_1) \wedge \Box_i(x_2) = \Box_j(x_1 \wedge x_2)$.

Every presentation presents a functor on $\mathcal{A}$.

**Definition 5.2.8.** Let $\mathcal{A}$ be a finitary algebraic category; we write $U$ for the forgetful functor and $F$ for its left adjoint. Let $(\Sigma, E)$ be a finitary presentation by operations and equations, over $\mathcal{A}$.

For an algebra $(A, \alpha)$, consider the following family:

$$F(E_{n_v}) \underset{\widehat{\pi_2}}{\overset{\widehat{\pi_1}}{\rightrightarrows}} F\Sigma UF(n_v) \xrightarrow{F\Sigma U(v)} F\Sigma U(A, \alpha) \tag{5.11}$$

where $v$ ranges over valuations of variables $v : F(n_v) \longrightarrow (A, \alpha)$.

Write $q_\alpha : F\Sigma U(A, \alpha) \longrightarrow L(A, \alpha)$ for the joint coequalizer of such family. The assignation which maps $(A, \alpha)$ to $L(A, \alpha)$ is called *the functor presented* by $(\Sigma, E)$.

Before showing that $L$ actually defines a functor we explain the construction. By definition the set $E_n$ is a subset of $UF\Sigma UF(n) \times UF\Sigma UF(n)$, i.e. it is a set of rank-1 equations on $n$ varibales. By taking the transpose of the projections we generate a subalgebra of $F\Sigma UF(n) \times F\Sigma UF(n)$. This is equivalent to say that by taking the transpose of the projections we generate a congruence $F(E_n)$ on $F\Sigma UF(n)$. From here, given a valuation $v : F(n) \longrightarrow (A, \alpha)$ what the composition in Equation 5.11 does is to evaluate the equations in the congruence $F(E_n)$ on $(A, \alpha)$; taking a coequalizer forces those (evaluated) equations to hold. The family in previous proposition corresponds to all possible valuations on $(A, \alpha)$ of all equation on $E$. By taking the joint coequalizer we force all of these equations to hold.

**Proposition 5.2.9.** *Let $\mathcal{A}$ be a finitary algebraic category; we write $U$ for the forgetful functor and $F$ for its left adjoint. Let $(\Sigma, E)$ be a finitary presentation by operations and equations, over $\mathcal{A}$.*

*The assignation $L$ in the previous Definition is a functor $L : \mathcal{A} \longrightarrow \mathcal{A}$.*

*Moreover, the arrows $q_\alpha : F\Sigma U(A, \alpha) \longrightarrow L(A, \alpha)$ form a natural transformation.*

*Proof.* The definition of $L : \mathcal{A} \longrightarrow \mathcal{A}$ on objects is clear. We explain its action in arrows.

The functor $L$ maps an arrow $f : (A, \alpha) \longrightarrow (B, \beta)$ to the unique map, given by the universal property of colimits, such that $L(f) \circ q_\alpha = q_\beta \circ F\Sigma U(f)$. In other words, $L(f)$ is the only arrow that makes the following diagram

$$\begin{array}{ccc} F\Sigma U(A,\alpha) & \xrightarrow{q_\alpha} & L(A,\alpha) \\ {\scriptstyle F\Sigma U(f)}\downarrow & & \vdots\, {\scriptstyle L(f)} \\ F\Sigma U(B,\beta) & \xrightarrow[q_\beta]{} & L(B,\beta) \end{array}$$

commute. We now argue that this definition is correct, i.e. that such arrow does indeed exist. By the universal property of coequalizers this will follow once we show that $q_\beta \circ F\Sigma U(f)$ coequalises $F\Sigma U(v) \circ \widehat{\pi_1}$ and $F\Sigma U(v) \circ \widehat{\pi_2}$ for each cardinal $n$ and each $v : F(n) \longrightarrow (A, \alpha)$. In order to see this, notice that for each morphism $f : (A, \alpha) \longrightarrow (B, \beta)$ the composition $f \circ v : F(n) \longrightarrow (B, \beta)$ is itself a valuation of variables into $(B, \beta)$. Therefore the equation

$$q_\beta \circ F\Sigma U(fv) \circ \widehat{\pi_1} = q_\beta \circ F\Sigma U(fv) \circ \widehat{\pi_2}.$$

is part of the definition of $q_\beta$. Hence $q_\beta \circ F\Sigma U(f)$ coequalises $F\Sigma U(v) \circ \widehat{\pi_1}$ and $F\Sigma U(v) \circ \widehat{\pi_2}$ and then by the universal property, of colimits, $L(f) : L(A, \alpha) \longrightarrow L(B, \beta)$, as in the the diagram above, exists. It is routine to check that this assignation preserves identities and composition. By construction the arrows $q_\alpha : F\Sigma U(A, \alpha) \longrightarrow L(A, \alpha)$ form a natural transformation. $\square$

We now fix some terminology with respect to presented functors.

**Definition 5.2.10.** Let $\mathcal{A}$ be a finitary algebraic category. Let $(\Sigma, E)$ be a presentation by operations and equations. Let $L : \mathcal{A} \longrightarrow \mathcal{A}$ be the functor in Proposition 5.2.9.

We say that a functor $L' : \mathcal{A} \longrightarrow \mathcal{A}$ is presented by $(\Sigma, E)$ if $L' \cong L$. In such case we say that $(\Sigma, E)$ is a presentation of $L'$.

The next examples illustrate the construction in previous proposition.

**Example 5.2.11.** Recall examples 3.2.9 and 3.2.17 in Section 3.2.1.

1. In Example 3.2.9 we have $E_2 = \{(\Box(x_1) \wedge \Box(x_2), \Box(x_1 \wedge x_2))\}$, $E_0 = \{(\Box\top, \top)\}$, and $E_n = \emptyset$ for any other $n$.

2. In Example 3.2.17 $E_0 = \{(\Box\top, \top), (\Diamond\bot, \bot)\}$, $E_2 = \{(\Box(x_1) \wedge \Box(x_2), \Box(x_1 \wedge x_2)), (\Diamond(x_1) \vee \Diamond(x_2), \Diamond(x_1 \vee x_2)), (\Box(x_1) \wedge \Diamond(x_2), \Diamond(x_1 \wedge x_2) \wedge \Box(x_1) \wedge \Diamond(x_2)), (\Box(x_1) \vee \Diamond(x_2), \Box(x_1 \vee x_2) \wedge (\Box(x_1) \vee \Diamond(x_2)))\}$, and $E_n = \emptyset$ for any other $n$.

3. In the case of $\mathsf{BA}$, if we add $E_1 = \{(\Box(x_1), \neg\Diamond(\neg x_1))\}$ to the previous item, we obtain the classical presentation of modal logic.

The similarities in the notation of the above definition and that for presentations of functors is not an accident. The next example illustrates this.

**Example 5.2.12.** Every presentation $\langle \Sigma, E \rangle$ for $\mathcal{T}$ induces a presentation for $M_{\mathcal{T}}$, the functor for the Moss logic. Indeed, given a presentation $E : \Sigma \longrightarrow \mathcal{T}$ we already have the functor $\Sigma$. The equations on $n$-variables are then given by the kernel of $F(E_{UF(n)})$), i.e. the action of $E$ on the free algebra generated by $n$.

For our purposes, the key result from [76] sates that Equation (5.9) is a presentation. The next definition and proposition make this precise.

**Definition 5.2.13.** Let $\mathcal{A}$ be a finitary variety. Let $L : \mathcal{A} \longrightarrow \mathcal{A}$ be a (finitary) functor. Let

$$E : \coprod_{n<\omega} (ULF(n) \times U^n) \longrightarrow UL$$

be the natural transformation in Equation (5.9).

The *canonical presentation* $(\Sigma, E')$ of $L$ is defined as follows: The functor $\Sigma$ is given by $\coprod_{n<\omega} ULF(n) \times (-)^n$ and for each $m$ the equations on $m$ variables are given by $E'_m = \ker\left(U(\widehat{E}_{F(m)})\right)$.

We detail the equations of the canonical presentation a bit more. Write $\Sigma = ULF(n) \times (-)^n$. With this notation we have $E_{F(m)} : \Sigma UF(m) \longrightarrow ULF(m)$; its transpose is then given by $\widehat{E}_{F(m)} : F\Sigma UF(m) \longrightarrow LF(m)$. Thus the kernel of $U(\widehat{E}_{F(m)})$ really gives a set of equations as required in Definition 5.2.7.

The key result from [76] can be now states as follows;

**Proposition 5.2.14** ([76]). *Let $\mathcal{A}$ be a finitary variety. Every finitary functor $L : \mathcal{A} \longrightarrow \mathcal{A}$ is presented by its canonical presentation.*

The main characterization theorem in [76] states that a functor on a finitary variety has a finitary presentation iff it preserves sifted colimits. An important result in [22] is that we can obtain a presentation of $\mathsf{Alg}(L)$, over $\mathsf{Set}$ or the base category, by combining a presentation of $L$ and a presentation of $\mathcal{A}$.

We finish by fixing some terminology.

**Definition 5.2.15.** Let $(L, \delta)$ be a coalgebraic modal logic for a functor $\mathcal{T}$, over a (finitary) category of power set algebras $\mathcal{A}$. Let $\langle \Sigma, E \rangle$ be a presentation of $L$.

The *modal signature* induced by $\langle \Sigma, E \rangle$ is given by the following set of predicate liftings: For each $p \in \Sigma_n$ take the predicate lifting given by the following composite

$$\begin{array}{ccc} UP^n & \xrightarrow{\widehat{q}_P(p,-)} & ULP \\ & \searrow \quad \swarrow {\scriptstyle U(\delta)} & \\ & UP\mathcal{T} & \end{array}$$

where $\widehat{q}$ is the transpose of the natural transformation $q : F\Sigma U \longrightarrow L$ in Proposition 5.2.9.

Notice that the coalgebraic modal logic associated with the signature above is given by $(F\Sigma U, \delta \circ q_P)$.

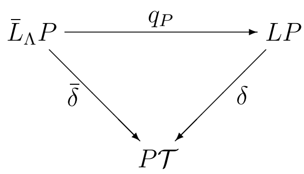
**Soundness and completeness**

We now have all the material to explain soundness and completeness. One of the key advantages of the functorial approach is that phenomena as soundness and completeness correspond to properties of $\delta$. We now elaborate on this.

Let $\Sigma_\Lambda$ be a modal signature for $\mathcal{T}$, let $\mathcal{A}$ be a category of power set algebras. Every set $Ax$ of rank-1 axioms, on the signature $\Sigma_\Lambda + \Sigma_\mathcal{A}$ induces a presentation by operations and equations (Definition 5.2.7).

The presented functor $L : \mathcal{A} \to \mathcal{A}$ (Proposition 5.2.9) maps an algebra $A$ to $F\Sigma_\Lambda U(A)$ quotiented by the axioms in $Ax$. Given formulas $\varphi, \psi \in \mathcal{L}_\Lambda$ we write $Ax \vdash \varphi = \psi$ to indicate that the pair $(\varphi, \psi)$ belongs to the congruence generated by $Ax$.

The existence of a natural transformation $\delta : LP \to P\mathcal{T}$ correspond to soundness of the axioms in $Ax$. This follows from the definition of $L$ because any $\delta$ must make the following diagram




commute, where $q : \bar{L}_\Lambda \to L$ is the quotient map in Definition 5.2.9 and $\bar{\delta}$ is the natural transformation in Equation (3.2), page 41. Hence if $\varphi = \psi$ is an axiom in $Ax$, the formulas $\varphi$ and $\psi$ will be identified in the initial $L$-algebra hence for any coalgebra $(X, \xi)$ we have

$$[\![\varphi]\!]_\xi^{\bar{\delta}} = [\![[\varphi]]\!]_\xi^{\delta} = [\![[\psi]]\!]_\xi^{\delta} = [\![\psi]\!]_\xi^{\bar{\delta}}$$

where $[\varphi]$ and $[\psi]$ denote the equivalence classes modulo $Ax$. More general, for every pair of formulas $\varphi, \psi \in \mathcal{L}_\Lambda$ if $Ax \vdash \varphi = \psi$ the equation above also holds.

It was shown in [68] that completeness correspond to the injectivity of $\delta$.

**Proposition 5.2.16** ([68])**.** *Let $\Lambda$, $L$, and $Ax$ as above. If $\delta : LP \to P\mathcal{T}$ is injective then $Ax$ is sound and complete. More explicitly for every pair of formulas $\varphi, \psi \in \mathcal{L}_\Lambda$ then*

$$Ax \vdash \varphi = \psi \text{ iff for every coalgebra } (X, \xi) \text{ we have } [\![\varphi]\!]_\xi = [\![\psi]\!]_\psi$$

## 5.2.3 The second representation Theorem

Using presentations of functors on varieties we can now show that every coalgebraic modal logic is a rank-1 axiomatization of a logic of predicate liftings.

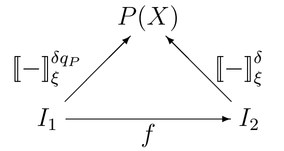
**Theorem 5.2.17.** *Let $(L, \delta)$ be a coalgebraic logic over a category of power set algebras $\mathcal{A}$. Every presentation $(\Sigma, E)$ of $L$ presents $(L, \delta)$ as a rank-1 axiomatization of the modal signature induced by $(\Sigma, E)$ (Definition 5.2.15).*

*Proof.* Let $(\Sigma, E)$ be a presentation of $L$. Without loss of generality, we can assume $L$ to be the joint coequalizer described in Proposition 5.2.9. As mentioned after Definition 5.2.15 the coalgebraic modal logic associated with the presentation $(\Sigma, E)$ is given by $(F\Sigma U, \delta \circ q_P)$, where $q : F\Sigma U \longrightarrow L$ is the quotient (natural transformation) as in Proposition 5.2.9. We want to show that $(L, \delta)$ is a rank-1 axiomatization of this logic.

Clearly $q : (F\Sigma U, \delta \circ q_P) \longrightarrow (L, \delta)$ is a one-step translation. Therefore, as discussed after Definition 4.1.1, see page 67, $q$ induces a functor

$$- \circ q : \mathsf{Alg}(L) \longrightarrow \mathsf{Alg}(F\Sigma U),$$

and this functor induces a function $f : I_1 \longrightarrow I_2$, where $I_1$ is the carrier of the initial $F\Sigma L$ algebra and $I_2$ is the carrier of the initial $L$-algebra, such that the following diagram




(5.12)

commutes for each coalgebra $\xi$.

The functor $- \circ q$ is an embedding. This functor is clearly faithful. It is only left to show that it is injective on objects; this follows because each of the components of $q$ is a coequaliser then each component is an epimorphism.

This presents $\mathsf{Alg}(L)$ as a variety inside $\mathsf{Alg}(F\Sigma U)$. Therefore, the initial $L$-algebra is a quotient of the initial $F\Sigma U$-algebra.

This presents $(L, \delta)$ as an axiomatization of $(F\Sigma U, \delta q_P)$ where the axioms are given by the kernel of $q$. Since the axioms given by the kernel of $q$ are all rank-1, by definition, this presents $(L, \delta)$ as a rank -1 axiomatization of $(F\Sigma U, \delta q_P)$ as we wanted to show. □

Compare the previous argument with the argument before Theorem 5.1.11.

**Corollary 5.2.18.** *Every coalgebraic modal logic $(L, \delta)$ is a rank 1 axiomatization of the logic given by its canonical signature.*

The previous theorem together with Example 5.2.12 shows yet another perspective on the translation of the logic of Moss liftings. For any category of power set algebras we can extend a presentation of $\mathcal{T}$ to a presentation of $M_{\mathcal{T}}$. Then, from the previous result, we can obtain a translation of $(M_{\mathcal{T}}, \nabla)$ into the logic of Moss liftings.

Combining Theorem 5.2.17 with Proposition 5.2.14 we obtain the following important corollary.

**Corollary 5.2.19.** *Every finitary coalgebraic logic for a* Set *endofunctor $\mathcal{T}$ can be translated into the logic of all predicate liftings.*

*Proof.* Write $I_1$ for the carrier of the initial $F\Sigma L$ algebra and $I_2$ for the carrier of the initial $L$-algebra. Let $f : I_1 \longrightarrow I_2$ be the function in Diagram (5.12). As shown in the proof of Theorem 5.2.17 $f$ is onto, hence using the axiom of choice, we can find a function $tr : I_2 \longrightarrow I_1$ such that $f \circ tr = id_{I_2}$; this is the desired translation. Indeed, since Diagram (5.12) commutes, we can check that for each formula $\varphi \in I_1$, we have $[\![\varphi]\!]_\xi^\delta = [\![tr(\varphi)]\!]_\xi^{\delta q_P}$ for each coalgebra $\xi$. $\square$

In the particular case of the Moss logic this was already stated as Proposition 5.1.21.

**Corollary 5.2.20** (Former Proposition 5.1.21)**.** *For every formula in $\mathcal{M}_\mathcal{T}$ there exists an equivalent formula in $\mathcal{K}_\mathcal{T}$. More explicitly, for every $\varphi \in \mathcal{M}_\mathcal{T}$ there exists $\psi \in \mathcal{K}_\mathcal{T}$ such that $[\![\varphi]\!]_\xi^{\mathcal{M}_\mathcal{T}} = [\![\psi]\!]_\xi^{\mathcal{K}_\mathcal{T}}$ for every coalgebra $\xi$.*

The previous results are worth some annotations. Firstly, notice that they generalize Theorem 4.4.9 because the argument works for any finitary category of power set algebras and not just for BA. Secondly, notice that even though we use the axiom of choice to define a translation, the embedding of $L$-algebras into $L_\mathcal{T}$-algebras explicitly shows the "modalities" needed for the translation. The moral of the story is that "nice" translations will come from "nice" presentations of $L$. This was already seen in the case of Moss liftings.

# 5.3 Equational coalgebraic logic

In this section we illustrate the use of presentations of functors further by presenting an equational axiomatization of the Moss logic. The idea is to use the Moss liftings to translate the axiomatic system in [69]. In other words, we use the Moss liftings to give a more standard algebraic face to the complete axiomatizations of the Moss logic; this leads us to an equational coalgebraic modal logic.

## 5.3.1 An equational proof system for $\mathcal{K}_\mathcal{T}$

In this section, we develop a proof systems for $\mathcal{K}_\mathcal{T}$ by translating the system for the Moss logic in [69]. Since this system is sound and complete, our system will be as well.

### Slim redistributions

Before presenting the system in [69] we should introduce the concept of *slim redistribution*. The reader might want to skip this section and refer to it when needed.

Slim redistributions play an important role in the completeness proof in [69]. Consequently, a good description of them using presentations of functors would be a key pinion in our completeness result (Theorem 5.3.30). We begin with the definition of base for an element in $\mathcal{T}(X)$.

**Definition 5.3.1.** Let $\mathcal{T}$ be a finitary $\mathsf{Set}$-endofunctor and let $t \in \mathcal{T}(X)$. The base of $t$ is the smallest $Y$, such that $Y \subseteq X$ and $t \in \mathcal{T}(Y)$. The base of an element $t \in \mathcal{T}(X)$ is denoted by $Base(t)$.

The base of an element exists because the functor is finitary [69].
We now introduce the concept of redistributions.

**Definition 5.3.2.** An element $\Phi \in \mathcal{T}\mathcal{P}(X)$ is a *redistribution* of a set $T \in \mathcal{P}\mathcal{T}_\omega(X)$ if $T \subseteq \blacktriangledown\Phi$. More explicitly, this means that for each $t \in T$ we have $t\,\overline{\mathcal{T}}(\in_X)\,\Phi$.
A redistribution $\Phi$ is *slim* if $\Phi \in \mathcal{T}_\omega\mathsf{Pow}_\omega(\bigcup_{t\in T} Base(t))$. The set of slim redistributions of $T$ is denoted by $SRD(T)$.

**Remark 5.3.3.** In [69] slim redistributions are only defined for finite subsets of $\mathcal{T}\mathcal{P}(X)$. The important property for a completeness proof is that the set of slim redistributions of a finite set is finite, which follows from our definition.

Here is a first illustrative example.

**Example 5.3.4.** In the case of $\mathsf{Pow}$, a set $\Phi \in \mathsf{Pow}\mathcal{P}(X)$ is a slim redistribution of a set $T \subseteq \mathsf{Pow}(X)$ iff $\bigcup T = \bigcup \Phi$ and for every $\varphi \in \Phi$ and $t \in T$ we have $t \cap \varphi \neq \emptyset$.

The previous example illustrates that slim redistributions are in some sense minimal covers. More explicitly, each $t \in T$ is covered by $\Phi$, i.e. $t \subseteq \bigcup \Phi$. Moreover, since $t \cap \varphi \neq \emptyset$ for each $\varphi \in \Phi$, we can say that in fact $\Phi$ is a "minimal" cover of all the elements of $A$. The following example, taken from [17], illustrates this further.

**Example 5.3.5.** Let $\mathcal{M}_{\mathsf{Pow}}$ be the Moss logic for $\mathsf{Pow}$. A slim redistribution for a set $T \in \mathcal{P}\mathsf{Pow}(\mathcal{M}_{\mathsf{Pow}})$ arises semantically as follows. Fix a $\mathsf{Pow}$-coalgebra $(X, \xi)$ and a state $x_0 \in X$. Define, for any successor $x$ of $x_0$, the set $\varphi_x := \{t \in \bigcup T \mid x \in [\![t]\!]_{(X,\xi)}\}$. Then let $\Phi_{x_0}$ be the set $\{\varphi_x \mid x \in \xi(x_0)\}$. It can now be shown that $x_0 \in [\![\bigwedge\{\nabla t \mid t \in T\}]\!]$ iff $\Phi_{x_0} \in SRD(T)$.

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

**Example 5.3.6.** Recall Example 3.3.6. Fix $\Phi \in \mathcal{DP}(X)$ and a set $T \in \mathcal{PD}(X)$. Recall that $\Phi$ can be thought as a sequence $(\varphi_j, q_j)_{1 \le j \le m}$ for $\varphi_j \in \mathcal{P}X, q_j \in [0,1], q_j > 0, m \in \mathbb{N}$. In similar fashion each $t \in T$ can be seen as a sequence $(t_i, p_i)_{1 \le i \le n}$ for some $t_i \in X, p_i \in [0,1], p_i > 0, n \in \mathbb{N}$. Now we can see that $\Phi$ is a redistribution of $T$ if for each $t \in T$ there exists a matrix $(r^t_{ij})_{1 \le i \le n, 1 \le j \le m}$, $r_{ij} \in [0,1]$ such that $t_i \notin \varphi_j \;\Rightarrow\; r^t_{ij} = 0$ and $\sum_i r^t_{ij} = q_j$ and $\sum_j r^t_{ij} = p_i$. The redistribution $\Phi$ would be slim if each $\varphi_i$ is a finite set.

### The system M

We now introduce the axiomatic system in [69] for the Moss Logic .

**Notation.** We recapitulate the notation that we have been/will be using. For an arbitrary set $X$, we use $\varphi, \psi$ for subsets of $X$. The letter $t$ denotes elements in $\mathcal{T}(X)$. We write $T$ for subsets of $\mathcal{T}(X)$. Finally, we use $\Phi$ and $\Psi$ for the entities in $\mathcal{T}\mathsf{Pow}(X)$ or $\mathcal{T}\mathcal{P}(X)$.

**Definition 5.3.7** ([69])**.** Let $(M_{\mathcal{T}}, \nabla)$ be Moss logic, over $\mathsf{BA}$, for a weak-pullback preserving functor $\mathcal{T}$ which preserves finite sets. Let $\le_{\mathcal{M}}$ denote the semantic consequence relation. The system **M** is given by the following axioms and rule on top of a complete axiomatization for classical propositional logic

($\nabla$0) From $t\overline{T}(\le_{\mathcal{M}})t'$ infer $\nabla t \le_{\mathcal{M}} \nabla t'$

($\nabla$1) $\bigwedge\{\nabla t \mid t \in T\} \le_{\mathcal{M}} \bigvee\{\nabla \mathcal{T}(\bigwedge)\Phi \mid \Phi \in SRD(T)\}\,.$

($\nabla$2) $\nabla \mathcal{T}(\bigvee)\Phi \le_{\mathcal{M}} \bigvee\{\nabla t \mid t\,\overline{\mathcal{T}}(\in)\,\Phi\}$

where $t$ and $t'$ denote elements in $\mathcal{T}_\omega(\mathcal{M})$, $T$ is a finite subset of $\mathcal{T}_\omega(\mathcal{M})$, and $\Phi$ is an object in $\mathcal{T}_\omega\mathsf{Pow}_\omega(\mathcal{M})$.

The following remark addresses a discrepancy of our notation with that of [69].

**Remark 5.3.8.** In [69] the axioms and rule above are enumerated starting at ($\nabla$1), i.e. our ($\nabla$0) is ($\nabla$1) there. We justify this divergence because when we move to an equational system the rule ($\nabla$0) will be disappear because it will correspond to congruence rule of algebraic logic which is always assumed in any equational system.

Intuitively, ($\nabla$1) pushes conjunctions down, ($\nabla$2) distributes disjunctions over the $\nabla$ and ($\nabla$0) is a congruence rule stating monotonicity of $\nabla$. Note that these intuitions are not expressed in standard logical concepts, e.g. ($\nabla$1) involves applying $\mathcal{T}$ to the map $\bigwedge : \mathsf{Pow}_\omega(\mathcal{M}) \to \mathcal{M}$ and the congruence rule uses relation lifting instead of simply substituting terms into operation symbols. A more standard presentation can be obtained by moving from $\mathcal{M}_{\mathcal{T}}$ to $\mathcal{K}_{\mathcal{T}}$, as we show in the following. The main result from [69] is the soundness and completeness of the system above. We refer to the original source for details.

**Remark 5.3.9.** The axioms in Definition 5.3.7, i.e. ($\nabla 1$) and ($\nabla 2$), can be replaced by equalities because the inequality from right to left always hold. With this in mind, recall Remark 3.3.8, on page 61, about distributive laws. Notice that Axiom ($\nabla 2$) expresses the commutativity of the rectangle there.

The following definition describes the coalgebraic modal logic associated with **M**, we refer to [69] for more details.

**Definition 5.3.10** ([69])**.** Let $(\bar{M}, \bar{\nabla})$ be the Moss logic as in Definition 3.3.1.

The coalgebraic modal logic $(M, \nabla)$ associated with the system **M** is defined as follows: The functor $M : \mathsf{BA} \to \mathsf{BA}$ maps an algebra $A$ to the the algebra $F\mathcal{T}_\omega U(A)$ quotiented by all evaluations of the axioms ($\nabla 1$) and ($\nabla 2$) in $A$, plus the following version of ($\nabla 0$): All equations $\nabla t = \nabla t \wedge \nabla t'$ for each pair $t, t' \in \mathcal{T}_\omega(A)$ such that $t\overline{\mathcal{T}}(\leq_A)t'$, where $\leq_A$ is the consequence relation of $A$, i.e. all pairs $a, a' \in A$ such that $a \wedge a' = a'$.

The natural transformation $\nabla : MP \to P\mathcal{T}$ is defined by extending $\bar{\nabla} : \bar{M}P \to P\mathcal{T}$ to the quotient.

Now that we have introduced the system **M**, we would like to transfer the axiomatisation to the logical system $\mathcal{K}_\mathcal{T}$. In particular, we would like this axiomatization to be sound and complete. To this end, we will need a more careful analysis of the representations of an element in $\mathcal{T}(X)$. This analysis is done in the next section.

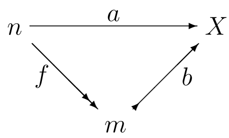
### 5.3.2 Well-based presentations

The base of an elment in $\mathcal{T}(X)$ plays a crucial role in the completeness proof of the axiomatisation of $\mathcal{M}_T$ in [69]. For everything to work we ought to give an account of the base of an element in $\mathcal{T}(X)$ in terms of presentations. Roughly speaking, for any finitary functor $\mathcal{T}$ and any $t \in \mathcal{T}(X)$ there is a smallest *finite* set $n \hookrightarrow X$ such hat $t \in T(n)$. This set $n$ is the base of $t$. The following issue will appear when we try to prove completeness of a translation of the system **M**. In order to replace $\mathcal{M}_T$ by $\mathcal{K}_T$ smoothly, we need that if $\langle \Sigma, E \rangle$ is a presentation of $\mathcal{T}$ and $t \in \mathcal{T}(X)$ has base $n$, then there is an $n$-ary operation symbol $p \in \Sigma_n$, called a basic operation, and an injective $a \in X^n$ such that $(p, a)$ represents $t$. Such presentations will be called well-based and this section studies their properties.

We start with a technical result. The next proposition shows that in fact, given $(p, a) \in \Sigma_n \times X^n$, we can assume $a$ to be injective; this will come in handy to simplify our proofs.

**Proposition 5.3.11.** *Let $\langle \Sigma, E \rangle$ be the canonical presentation for a set functor $\mathcal{T}$. For each $(p, a) \in \mathcal{T}(n) \times X^n$, there exists $(q, b) \in \Sigma(X)$ such that $q(b) \approx_\mathcal{T} p(a)$ and $b$ is injective.*

*Proof.* Factor $a : n \longrightarrow X$ as follows:




where $f$ is onto and $b$ is injective. Let $q = \mathcal{T}(f)(p)$, then by Proposition 5.1.9 we conclude $q(b) \approx_T p(a)$. □

The next example illustrates the construction above.

**Example 5.3.12.** To illustrate the construction in the previous proof, consider the canonical presentation of $\mathsf{List}$. The list $[x, x] \in \mathsf{List}(X)$ has a representative $[0, 1](x, x) \in \mathsf{List}(2) \times X^2$. We can factor it through $2 \longrightarrow 1$ to obtain the representative $[0, 0](x) \in \mathsf{List}(1) \times X^1$.

The previous example illustrates that an element $t \in \mathcal{T}(X)$ might have several representations; some of which are redundant. The next definition will allow us to avoid such redundancies.

**Definition 5.3.13.** Let $\langle \Sigma, E \rangle$ be a (standard) presentation for a functor $\mathcal{T}$. We define the category $\mathsf{IElem}(t)$ of *injective representants* of $t \in \mathcal{T}(X)$ as follows: The objects of $\mathsf{IElem}(t)$ are given by

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

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

We call $(p, a)$ a *basic representant* of $t$ if $(p, a)$ is initial in $\mathsf{IElem}(t)$, that is, $\forall (q, b) \in \mathsf{IElem}(t) \;.\; \exists f : \mathrm{dom}(a) \longrightarrow \mathrm{dom}(b) \;.\; \mathcal{T}(f)(p) = q \;\&\; a = bf$. Notice that $f$ is unique since $b$ is injective.

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

Every well based presentation is injective. The difference resides in that in a well based presentation there is a minimal representant. Here are some illustrations.

**Example 5.3.14.**
1. The standard presentation and the canonical presentation for $\mathsf{Pow}$ are well based.

2. For the functor $\mathsf{List}$, the usual presentation is given by the identity $id : \coprod_{n<\omega} X^n \longrightarrow \mathsf{List}$. This presentation is not well based as, for example, the list $[x, x]$ has no injective representative; in fact the only representant for such list is the function $2 \longrightarrow X$ with constant value $x$.

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

**Proposition 5.3.15.** *Canonical presentations are well based.*

*Proof.* Consider $(p, a : m \longrightarrow X), (q, b : n \longrightarrow X)$ in $\mathsf{IElem}(t)$. Let $(f : k \longrightarrow m, g : k \longrightarrow n)$ be a pullback of $a$ and $b$. Since $\mathcal{T}$ preserves weak pullbacks and it is standard, the following diagram

$$\begin{array}{ccc} \mathcal{T}(k) & \xrightarrow{\mathcal{T}(g)} & \mathcal{T}(n) \\ {\scriptstyle \mathcal{T}(f)}\downarrow & & \downarrow{\scriptstyle \mathcal{T}(b)} \\ \mathcal{T}(m) & \xrightarrow[\mathcal{T}(a)]{} & \mathcal{T}(X) \end{array}$$

is in fact a pullback. Therefore there exists $r \in \mathcal{T}(k)$ such that $\mathcal{T}(f)(r) = p, \mathcal{T}(g)(r) = q$.

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

The next remark highlights a more general result.

**Remark 5.3.16.** We actually proved a stronger statement: A presentation $\langle \Sigma, E \rangle$ is well-based if 1) it is injective, i.e. every $t \in \mathcal{T}(X)$ has a representative $(p, a)$ with $a$ injective and 2) $\langle \Sigma, E \rangle$ is stable under pullbacks. Here we say that $\langle \Sigma, E \rangle$ is **stable under pullbacks** if whenever

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

is a pullback and $p(a) \approx_{\mathcal{T}} q(b)$, and $r \in \mathcal{T}(k)$ is such that $\mathcal{T}(a')(r) = q$ and $T(b')(r) = p$ then $r \in \Sigma_k$.

The next example illustrates basic representants.

**Example 5.3.17.** We consider canonical presentations.

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

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

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

**Proposition 5.3.18.** *Let $\langle\Sigma, E\rangle$ be an injective (standard) presentation of $\mathcal{T}$. For an operation $p \in \Sigma_n$, the following are equivalent.*

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

2. *$(p, id_n)$ is a basic representant.*

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

*Proof.* We will prove (3)⇒(1)⇒(2)⇒(3). The first implication is trivial.

For (1)⇒(2), consider some set $X$ and an injective $a : n \longrightarrow X$ such that $(p, a)$ is basic for $\langle\Sigma, E\rangle$ . We want to show that $(p, id_n)$ is basic. Suppose that $(q, b) \in \mathcal{T}(m) \times n^m$ with injective $b : m \longrightarrow n$ represents the same element as $(p, id_n)$, this element exists because $\langle\Sigma, E\rangle$ is injective. We want to find an arrow $f : n \longrightarrow m$ such that $b \circ f = id_n$ and $T(f)(p) = q$. For this purpose notice that $(q, a \circ b)$ represents the same element as $(p, a)$ because

$$\begin{aligned} E(q, a \circ b) &= \mathcal{T}(a \circ b)(q) && (\langle\Sigma, E\rangle \text{ is standard}) \\ &= \mathcal{T}(a) \circ \mathcal{T}(b)(q) && (\text{functoriality}) \\ &= \mathcal{T}(a) \circ \mathcal{T}(id_n)(p) && (\text{assumption on } (q, b)) \\ &= E(p, a). && (\langle\Sigma, E\rangle \text{ is standard}) \end{aligned}$$

Therefore since $(p, a)$ is basic there is a function $f : n \longrightarrow m$ such that $a \circ b \circ f = a$ and $T(f)(p) = q$. This is the required arrow from $(p, id_n)$ to $(q, b)$; it is unique because $(p, a)$ is basic, and $b \circ f = id_n$ because $a$ is injective.

For (2)⇒(3), assume $(p, id_n)$ is basic. We want to show $(p, a)$ is also basic for any injective $a : n \longrightarrow X$. Suppose $(q, b) \in \mathcal{T}(m) \times X^m$ with injective $b : m \longrightarrow X$ represents the same element as $(p, a)$. Let $b' : k \longrightarrow n$ be the pullback of $b$ along $a$, let $a'$ be the pullback of $a$ along $b$. Since $\mathcal{T}$ preserves weak-pullbacks and $\mathcal{T}$ is standard, the following diagram

$$\begin{array}{ccc} \mathcal{T}(k) & \xrightarrow{\;b'\;} & \mathcal{T}(n) \\ {\scriptstyle a'}\downarrow & & \downarrow{\scriptstyle a} \\ \mathcal{T}(m) & \xrightarrow[\;b\;]{} & \mathcal{T}(X) \end{array}$$

is a pullback. Since $(p, a)$ and $(q, b)$ represent the same element, by definition of pullback, there is an element of $r \in \mathcal{T}(k)$ such that $\mathcal{T}(b')(r) = p$ and $\mathcal{T}(a')(r) = q$. Notice that $(r, b')$ represents the same element as $(p, id_n)$. Since the latter is basic for $\langle \Sigma, E \rangle$ we have $k = n$ and $b'$ iso. It follows that $a' \circ b'^{-1}$ is the arrow from $(p, a)$ to $(q, b)$ required to show that $(p, a)$ is basic. □

Any two basic representations $(p, a : n \longrightarrow X), (q, b : m \longrightarrow X)$ of an element $t \in \mathcal{T}(X)$ are isomorphic in $\mathsf{IElem}(t)$. In particular, we have $n = m$, and the functions $a$ and $b$ define the same subset $\mathsf{Img}(a) = \mathsf{Img}(b)$ of $X$. This gives us a description of *Base* (Definition 5.3.1) in terms of presentations.

**Proposition 5.3.19.** *Let $\mathcal{T}$ be a finitary endofunctor on* Set *and let $t \in \mathcal{T}(X)$. Consider the canonical presentation $\langle \Sigma, E \rangle$ for $\mathcal{T}$ and a basic representant for $t$. Denote the representant by $(p, a)$ and write* $\mathsf{Img}(a)$ *for the image of $a$. Then the image of $a$ is the base of $t$, i.e.* $\mathsf{Img}(a) = Base(t)$.

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

**Remark 5.3.20.** If a presentation $\langle \Sigma, E \rangle$ is well-based then for every $X$ and every $t \in \mathcal{T}(X)$ there exists $(p, a) \in \Sigma_n \times X^n$ such that (i) $(p, a)$ represents $t$ and (ii) $\mathsf{Img}(a) = Base(t)$.

A nice property to have is that when all operations are basic.

**Definition 5.3.21.** A presentation $\langle \Sigma, E \rangle$ is *basic* if all $p \in \Sigma_n$ are basic.

The next proposition shows that basic presentations are not rare.

**Proposition 5.3.22.** *Every well-based presentation $\langle \Sigma, E \rangle$ contains a basic presentation $\langle \Sigma', E' \rangle$ in the sense that for each $n$ we have $\Sigma'_n \subseteq \Sigma_n$.*

*Proof.* We take

$$\Sigma'_n = \{p \in \Sigma_n \mid p \text{ is basic}\}. \tag{5.13}$$

The natural transformation

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

is defined by restricting $E$ to $\Sigma'_n$. To show that this gives a presentation we ought to argue that $E'$ is onto. For this purpose pick $t \in \mathcal{T}(X)$ and let $(q, b : k \longrightarrow X)$ be a representative of $t$. Since $\langle \Sigma, E \rangle$ is well-based we can assume that $q$ is basic and $b$ is injective; but then, by definition, $q \in \Sigma'_k$. $E'$ is well based by construction. □

**Example 5.3.23.** If we apply the procedure described above to the canonical presentation for Pow we obtain the standard presentation (see Example 5.1.2).

### 5.3.3 A completeness proof

Now we can present the axiomatic system for $\mathcal{K}_{\mathcal{T}}$. The idea is to translate the system **M** (Definition 5.3.7) into the logic of Moss liftings. Our first step is to describe redistributions in terms of presentations.

#### Σlim redistributions

Using Lemma 5.1.10 we can present (slim) redistributions in terms of presentations.

**Notation.** Let $\langle \Sigma, E\rangle$ be a presentation of a functor $\mathcal{T}$. We denote subsets of $\Sigma(X)$ with $T$ but the elements of such set will be denoted by pairs $(p, a)$. This is to emphasise that $(p, a)$ is a representant of an element in $\mathcal{T}(X)$.

**Definition 5.3.24.** Let $\langle \Sigma, E\rangle$ be a (standard) presentation of a functor $\mathcal{T}$. A $\Sigma$-*redistribution* of a set $T \subseteq \Sigma(X)$ is an element $(q, \psi) \in \Sigma\mathcal{P}(X)$, say $(q, \psi) \in \Sigma_n \times \mathcal{P}(X)^n$, such that: for each $(p, a) \in T$ there exists $k < \omega$, $r \in \Sigma_k$, $b : k \longrightarrow X$ and $\varphi : k \longrightarrow \mathcal{P}(X)$ such that

$$r(b) \approx_{\mathcal{T}} p(a) \,\wedge\, r(\varphi) \approx_{\mathcal{T}} q(\psi) \,\wedge\, (\forall i)(b_i \in \varphi_i). \tag{5.14}$$

Let $|T| = \{a_i \mid (\exists p)\,((p, a) \in T)\}$, i.e. the set of variables used in $T$. A $\Sigma$-redistribution $(q, \psi)$ is *slim* if (1) $n \leq 2^{|T|}$ and (2) $\bigcup_{i\in n} \psi_i \subseteq |T|$. The set of slim $\Sigma$-redistributions of $T$ is denoted $\Sigma RD(T)$. Slim $\Sigma$-redistributions are also called $\Sigma$*lim redistributions*.

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

The next remark explains the difference between Slim and Σlim redistributions.

**Remark 5.3.25.** To explain the difference between Slim and Σlim consider the following diagram,

$$\begin{array}{ccccc} \Sigma(X) & \xleftarrow{\Sigma(\pi_X)} & \Sigma(\in_X) & \xrightarrow{\Sigma(\pi_{\mathcal{P}(X)})} & \Sigma(\mathcal{P}(X)) \\ {\scriptstyle E_X}\downarrow & & \downarrow{\scriptstyle E_{\in_X}} & & \downarrow{\scriptstyle E_{\mathcal{P}(X)}} \\ \mathcal{T}(X) & \xleftarrow[\mathcal{T}(\pi_X)]{} & \mathcal{T}(\in_X) & \xrightarrow[\mathcal{T}(\pi_{\mathcal{P}(X)})]{} & \mathcal{T}(\mathcal{P}(X)) \end{array} \tag{5.15}$$

This is Diagram (5.3) in Lemma 5.1.10 for the case $R = \in_X$. A $\Sigma RD$ lives in the upper row of this Diagram and has been defined so that it matches the notion of redistribution in Definition 5.3.2 living in the lower row. More explicitly, if $(q, \psi)$ is $\Sigma$-redistribution of $T \subset \Sigma(X)$ then $E_{\mathcal{P}(X)}(q, \psi)$ is a redistribution of $E_X[T]$. Nevertheless, whereas the concept of a $\Sigma$-redistribution is a direct rewriting of the concept of redistribution, the concept of $\Sigma$lim redistribution is not a direct translation of the concept of Slim redistribution as in Definition 5.3.2. $\Sigma$lim redistributions are more restrictive than Slim redistributions in the sense that they impose a *finite bound* in the search space for redistributions; Slim redistributions require a finite set but they do not provide an specific bound. The equivalence of the two notions will be the key step to prove completeness. As expected such equivalence relies on properties of the presentation. We will show that if the presentation is well-based both notions coincide.

The next examples illustrate $\Sigma$-redistributions and $\Sigma$lim redistributions.

**Example 5.3.26.** Recall Example 5.1.8.

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

2. For the $\mathsf{List}$ presentation of $\mathsf{Pow}$ the redistributions of a set $T \subseteq \mathsf{List}(X)$ are all the lists $\Psi = [\psi_1, \ldots, \psi_n]$ of subsets of $\mathcal{P}(X)$ such that for each $[a_1, \ldots, a_m] \in T$ we have $\{a_1, \ldots, a_m\} \subseteq \bigcup \psi_i$. The redistribution $\Psi$ is slim if $\bigcup \psi_i \subseteq \{a_i \mid a \in T\}$ and $n \leq 2^{|\{a_i | a \in T\}|}$.

3. In the case of the canonical presentation of $\mathsf{Pow}$, the redistributions of a set $T \subseteq \Sigma(X)$ are the pairs $(q, \psi)$ such that for each $(p, a) \in T$ we have $\{a_i \mid i \in p\} \subseteq \bigcup\{\psi_j \mid j \in q\}$. The redistribution $\Psi$ is slim if $\bigcup \psi_i \subseteq \{a_i \mid a \in T\}$ and $n \leq 2^{|\{a_i | a \in T\}|}$.

4. In the case of the canonical presentation for $\mathcal{B}_{\mathbb{N}}$, the redistributions of a set $T \subseteq \Sigma(X)$ are the pairs $(p, \psi) \in \mathcal{B}_{\mathbb{N}}(n) \times \mathcal{P}(X)^n$ such that for each

$(p, a) \in T$ there exits a matrix $(r^a_{ij})_{1\leq i\leq n, 1\leq j\leq m}$, such that $a_i \notin \psi_j \Rightarrow r^a_{ij} = 0$ and $\sum_i r^a_{ij} = q_j$ and $\sum_j r^a_{ij} = p_i$.

### The system $\mathsf{K}_\mathcal{T}$

Now we can translate the axioms in Definition 5.3.7.

Using $\Sigma$lim redistributions we can translate ($\nabla$1) as follows

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

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

Axiom ($\Sigma$1) simplifies some, but not all aspects of ($\nabla$1). In particular, it does not replace the notion of a redistribution in the sense of [69] by something fundamentally simpler: Recall that a $\Sigma RD$ lives in the upper row of Diagram (5.15), on page 125, and has the notion of $SRD$ lives in the lower row. One way to understand our axiomatisation in general, and Axiom ($\Sigma$1), and Equation (5.14) in particular, is as an *implementation* of the axiomatisation in [69] using lists. Indeed, given a set $T$ as in ($\nabla$1) or ($\Sigma$1), to apply the axiom we need a join over a sufficiently large set of redistributions of $T$. Equation (5.14) tells us how to compute this set using the equational theory $\approx_\mathcal{T}$. For such computational purposes, one would not work with the canonical representation but rather with a smaller one as e.g. given by $\mathsf{List}$ for the powerset in Example 5.1.2.

To translate ($\nabla$2) we introduce the dual concept of redistributions.

**Definition 5.3.27.** Let $\langle \Sigma, E\rangle$ be a (standard) presentation of a functor $\mathcal{T}$. A *coredistribution* of an element $(q, \psi) \in \Sigma\mathcal{P}(X)$ is an element $(p, a) \in \Sigma(X)$ such that there exists $k < \omega$, $r \in \Sigma_k$, $b : k \to X$ and $\varphi : k \to \mathcal{P}X$ such that

$$r(b) \approx_\mathcal{T} p(a) \;\wedge\; r(\varphi) \approx_\mathcal{T} q(\psi) \;\wedge\; (\forall i)(b_i \in \varphi_i).$$

with $a$ injective. In other words, this means $p(a)\overline{\mathcal{T}}(\in_X)q(\psi)$. The set of coredistributions of $(q, \psi)$ is denoted $CRD(q, \psi)$.

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

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

The next proposition gives sufficient conditions for the set of coredistributions is finite.

**Proposition 5.3.28.** *Let $\mathcal{T}$ be a finitary functor that preserves finite sets. Then the set of coredistributions of an element $(q, \psi) \in \Sigma\mathcal{P}(X)$ is finite.*

*Proof.* In [69] it was proven that if $\mathcal{T}$ is finitary and preserve finite sets there are at most finitely many $t \in \mathcal{T}(X)$ such that $t\overline{\mathcal{T}}(\in_X)q(\psi)$. Since $\mathcal{T}$ preserves finite sets and we assume presentation to be standard, there are only finitely many operations of a given arity because $\Sigma_n \subseteq \mathcal{T}(n)$. Hence, since $\mathcal{T}$ is also finitary, an element $t \in \mathcal{T}(X)$ has finitely many injective representants because the base of $t$ is finite. From all this we conclude that the set of coredistributions of $(q, \psi)$ should be finite. □

One advantage of our equational axiomatisation is that the rule ($\nabla$0) reduces to the standard congruence rule of equational logic.

**Definition 5.3.29.** Let $\langle \Sigma, E\rangle$ be a presentation of $\mathcal{T}$. The derivation system $\mathsf{K}_{\mathcal{T}}$, or just $\mathsf{K}$, is given by the equational logic for $\langle \Sigma, E\rangle$ and the axioms $\Sigma 1$ and $\Sigma 2$ on top of a complete equational axiomatization for classical propositional logic. The semantic consequence relation is denoted by $\leq_{\mathcal{K}}$.

In main result of this section is the soundness and completeness of the system:

**Theorem 5.3.30.** *Let $\langle \Sigma, E\rangle$ be a well-based presentation of $\mathcal{T}_\omega$. The system $\mathsf{K}_{\mathcal{T}}$ is sound and complete for the logic of Moss liftings $(K_{\mathcal{T}}, \delta_E)$.*

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

**Lemma 5.3.31.** *Let $\langle \Sigma, E\rangle$ be a well-based presentation and let $T \subseteq \Sigma(X)$ be finite. For any $\Phi \in \mathcal{T}\mathcal{P}(X)$ the following conditions are equivalent:*

1. *$\Phi \in SRD(E_X[T])$.*

2. *There exists $(q, \psi) \in \Sigma RD(T)$ such that $E_{\mathcal{P}X}(q, \psi) = \Phi$.*

*In other words, slim redistributions and $\Sigma$-slim redistributions coincide for well-based presentations.*

Before proceeding to the proof recall (Remark 5.3.25) that $\Sigma$-redistributions are just the inverse image of the set of redistributions (Definition 5.3.2) under $E$. But $\Sigma$lim redistributions, are in principle, not a direct translation of the concept of Slim redistribution. The lemma above shows that if the presentation is well-based then the two notions coincide.

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

From (1) to (2): Let $\Phi$ be a Slim redistribution of $E_X(T)$. We want to find a a representant of $\Phi$ which is a $\Sigma$lim redistribution of $T$. Since $\langle \Sigma, E\rangle$ is well-based $\Phi$ has a basic representant $(q, \psi) \in \Sigma\mathcal{P}(X)$; this is the representant we are looking for.

We now show that $(q,\psi)$ is a $\Sigma$-redistribution of $T$. Since $\Phi$ is a redistribution of $E_X[T]$ we have $E_X[T] \subseteq \blacktriangledown\Phi$; this means that for each $(p,a) \in T$ we have $E(p,a)\overline{\mathcal{T}}(\in_X)E(q,\psi)$. From this, by Lemma 5.1.10, there exists $k < \omega$, $r \in \Sigma_k$, $b : k \longrightarrow X$ and $\varphi : k \longrightarrow \mathcal{P}(X)$ such that

$$r(b) \approx_{\mathcal{T}} p(a) \;\wedge\; r(\varphi) \approx_{\mathcal{T}} q(\psi) \;\wedge\; (\forall i)(b_i \in \varphi_i). \tag{5.16}$$

This just means that $(q,\psi)$ is a $\Sigma$-redistribution of $T$.

We now show that $(q,\psi)$ is in fact a $\Sigma$lim redistribution. Since $\Phi$ is a slim redistribution of $E_X[T]$ we have

$$\Phi \in \mathcal{T}_\omega \mathsf{Pow}_\omega \left( \bigcup_{(p,a)\in T} Base(E(p,a)) \right).$$

Since $(q,\psi)$ is a representant of $\Phi$, by definition, $(q,\psi) \in \Sigma\mathsf{Pow}_\omega\left(\bigcup_{(p,a)\in T} Base(E(p,a))\right)$. This implies

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

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

$$\bigcup_{i\in dom(\psi)} \psi_i \subseteq \{a_i \mid (p,a) \in T\}$$

as required. It is only left to bound the arity of $\psi$, i.e. $dom(\psi) \leq 2^{|T|}$, but this follows from the previous inclusion and the fact that $\psi$ is injective because $(q,\psi)$ is basic.

From (2) to (1): Let $(q,\psi)$ be a $\Sigma$lim redistribution of $T$. We want to show that $E(q,\psi)$ is a Slim redistribution of $E[T]$. Let $E(q,\psi) = \Phi$. Since $(q,\psi)$ is $\Sigma$-redistribution, by definition, $\Phi$ is a redistribution of $E_X[T]$, see Remark 5.3.25.

Now we show that it is in fact a slim redistribution in the sense of Definition 5.3.2. Since $(q,\psi)$ is a slim $\Sigma$-redistribution we have $\bigcup_{i\in dom(\psi)} \psi_i \subseteq \{a_i \mid (p,a) \in T\}$; each $\psi_i$ is finite because $T$ is finite. Since each $(p,a)$ is a basic representant the right hand side of the inclusion can be replaced by $\bigcup_{(p,a)\in T} Base(E(p,a))$, then we can assume $\psi$ to be a function as follows:

$$\psi : n \longrightarrow \mathsf{Pow}_\omega \left( \bigcup_{(p,a)\in T} Base(E(p,a)) \right)$$

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

In the case of coredistributions, using well based presentations, we have the following result which is immediate from Lemma 5.1.10.

**Lemma 5.3.32.** *Let $\langle \Sigma, E\rangle$ be a well-based presentation and let $(q, \psi) \in \Sigma\mathcal{P}X$. For each $t \in \mathcal{T}(X)$ the following are equivalent:*

1. $t\,\overline{\mathcal{T}}(\in_X)\,E_{\mathcal{P}(X)}(q, \psi)$.

2. *There exists $(p, a) \in CRD(q, \psi)$ such that $t = E(p, a)$.*

*Proof.* Using Lemma 5.1.10 we can choose a representant $(p, a) \in \Sigma(X)$ for $t$ satisfying the condition in Definition 5.3.27, i.e. an element $(p, a) \in \Sigma(X)$ such that $E_X(p, a) = t$ and there exists $k < \omega$, $r \in \Sigma_k$, $b : k \longrightarrow X$ and $\varphi : k \longrightarrow \mathcal{P}X$ such that

$$r(b) \approx_{\mathcal{T}} p(a) \;\wedge\; r(\varphi) \approx_{\mathcal{T}} q(\psi) \;\wedge\; (\forall i)(b_i \in \varphi_i).$$

Since the presentation is well based we can assume this representant to be basic, and in particular we can assume $a$ to be injective, then we have $(p, a) \in CRD(q, \psi)$. □

Now we can prove Theorem 5.3.30.

*Proof Theorem 5.3.30.* We first fix the notation for the logics that will appear in the argument. $(\bar{K}_{\mathcal{T}}, \bar{\delta}_E)$ is the logic of Moss liftings (Definition 5.1.15); $(\bar{M}, \bar{\nabla})$ is the Moss logic (Definition 3.3.4); $(K_{\mathcal{T}}, \delta_E)$ is the logic of Moss liftings quotiented by the system $\mathbf{K}$ (Definition 5.3.29); $(M, \nabla)$ is the Moss logics quotiented by the system $\mathbf{M}$ (Definition 5.3.10). More explicitly, $K_{\mathcal{T}}$ maps an algebra $A$ to the algebra $\bar{K}_{\mathcal{T}}(A)$ quotiented by the congruence generated by the system $\mathsf{K}$, i.e. it is the functor described in Proposition 5.2.9 where the equations are all those in the system $\mathsf{K}$; write $\leq_{\mathsf{K}}$ for the consequence relation of this system. Similarly, the functor $M$ maps and algebra $A$ to the algebra $\bar{M}(A) = FTU(A)$ quotiented by the congruence generated by the system $\mathbf{M}$ relativized to $A$, see Definition 5.3.10; write $\leq_{\mathbf{M}}$ for the consequence relation this of system.

We want to show that the logic $(K_{\mathcal{T}}, \delta_E)$ is complete. We will show this by showing that each of the components of $\delta_E$ is injective; hence, by Proposition 5.2.16, the logic is complete.

It was shown in [69] that the logic $(M, \nabla)$ is complete, hence, by Proposition 5.2.16, each of the components of the natural transformation $\nabla : MP \longrightarrow P\mathcal{T}$ is injective. With this in mind, to prove that $(\delta_E)_X : K_{\mathcal{T}}P(X) \longrightarrow P\mathcal{T}(X)$ is injective it is enough to define a one-step translation $\nu : (K_{\mathcal{T}}, \delta_E) \longrightarrow (M, \nabla)$ for which each of its components is injective. From this, by definition, we will have $\delta_E = \nabla \circ \nu$ which will imply that $(\delta_E)_X$ is injective.

Consider the following diagram of one step translations

$$\begin{array}{ccc}
(\bar{K}_{\mathcal{T}}, \bar{\delta}_E) & \xrightarrow{\;\bar{\nu}\;} & (\bar{M}, \bar{\nabla}) \\
{\scriptstyle q_E}\big\downarrow & & \big\downarrow{\scriptstyle q_M} \\
(K_{\mathcal{T}}, \delta_E) & \overset{}{\underset{\nu}{\dashrightarrow}} & (M, \nabla)
\end{array} \qquad (5.17)$$

The vertical arrows are the respective quotients. The upper horizontal arrow is the one-step translation from Moss liftings into the Moss logic, see Corollary 5.1.19 and Theorem 5.2.17. The dotted arrow below is the one step translations we need to prove completeness. Since $K_{\mathcal{T}}$ and $M$ are quotients of $\bar{K}_{\mathcal{T}}$ and $\bar{M}$; using the systems $\mathsf{K}$ and $\mathbf{M}$, respectively. To define $\nu$ we prove that the derivation relations $\leq_{\mathsf{K}}$ and $\leq_{\mathbf{M}}$ are "equivalent". The next claim makes this precise

**Claim.** If for each pair $\varphi, \psi \in \bar{K}_{\mathcal{T}}(A)$ we have

$$\varphi \leq_{\mathsf{K}} \psi \text{ iff } \bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\psi) \tag{5.18}$$

the there is a one step translation $\nu : (K_{\mathcal{T}}, \delta_E) \longrightarrow (M, \nabla)$, as in Diagram 5.17. Moreover each of the components of $\nu$ is injective.

*Proof.* We use the implication from left to right to define $\nu$. We the implication from right to left to show that each of components is injective.

We now show how to define $\nu : K_{\mathcal{T}} \longrightarrow M$. The $A$-component $\nu_A : K_{\mathcal{T}}(A) \longrightarrow M(A)$ maps an equivalence class $q_E(\varphi) \in K_{\mathcal{T}}(A)$ to equivalence class $q_M(\bar{\nu}(\varphi)) \in M(A)$. We now show that this is well defined. Assume $q_E(\varphi) = q_E(\psi)$, this means $\varphi \leq_{\mathsf{K}} \psi$ and $\psi \leq_{\mathsf{K}} \varphi$. From this, by the implication from left to right in Equation (5.18), we conclude $\bar{\nu}(\varphi) \leq_M \bar{\nu}(\psi)$ and $\bar{\nu}(\psi) \leq_M \bar{\nu}(\varphi)$. This means $q_M(\bar{\nu}(\varphi)) = q_M(\bar{\nu}(\psi))$. In other words, $\nu$ is well defined; clearly the assignation is natural.

By construction $\nu : (K_{\mathcal{T}}, \delta_E) \longrightarrow (M, \nabla)$ is a one step translation.

We now show for each algebra $A$ the component $\nu_A : K_{\mathcal{T}}(A) \longrightarrow M(A)$ is injective. Assume $\nu(q_E(\varphi)) = \nu(q_E(\psi))$. By definition of $\nu$ this equation means $q_M(\bar{\nu}(\varphi)) = q_M(\bar{\nu}(\psi))$. By definition of $q_M$ the previous equation says that $\bar{\nu}(\varphi) \leq_M \bar{\nu}(\psi)$ and $\bar{\nu}(\psi) \leq_M \bar{\nu}(\varphi)$. By the implication from right to left in Equation (5.18) we conclude $\varphi \leq_{\mathsf{K}} \psi$ and $\psi \leq_{\mathsf{K}} \varphi$, this means $q_E(\varphi) = q_E(\psi)$. In other words $\nu_A$ is injective.

This concludes the proof of the claim □

We now proceed to prove Equation (5.18). We split the proof into two claims.

**Claim.** For each pair $\varphi, \psi \in \bar{K}_{\mathcal{T}}(A)$, if $\varphi \leq_{\mathsf{K}} \psi$ then $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\psi)$

*Proof.* The proof goes by induction on the complexity of the derivation $\varphi \leq_{\mathsf{K}} \psi$. A derivation on this system is a chain of inequalities $\leq_{\mathsf{K}}$ where each of those is either a boolean inequality valid on $A$, one of the axioms ($\Sigma1$) or ($\Sigma2$), or it s derived from previous inqualities by substitution or transitivity rules.

Since $\bar{\nu}_A : K_{\mathcal{T}}(A) \longrightarrow M(A)$ is a homomorphism of boolean algebras it is straightforward to show that the implication "if $\varphi \leq_{\mathsf{K}} \psi$ then $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\psi)$" holds for the case of the boolean axioms or the cases of the rules of substitution and transitivity.

It is only left to show the cases where $\varphi \leq_{\mathsf{K}} \psi$ is an instance of the axioms ($\Sigma 1$) and ($\Sigma 2$). The implication will follow if we show that these translate into instances of the axioms ($\nabla 1$) and ($\nabla 2$), respectively.

We now that the axiom ($\Sigma 1$) translates into an instance of ($\nabla 1$). The left hand side of Axiom ($\Sigma 1$) is translated as follows:

$$\begin{aligned}
\bar{\nu}\left(\bigwedge\{\lambda^p(a) \mid (p,a) \in T\}\right) &= \bigwedge\{\bar{\nu}(\lambda^p(a)) \mid (p,a) \in T\} && (\bar{\nu} \text{ is a morphism in } \mathsf{BA}) \\
&= \bigwedge\{\nabla E_A(p,a) \mid (p,a) \in T\} && (\text{Definition } \bar{\nu}) \\
&= \bigwedge\{\nabla t \mid t \in E_A[T]\}. && (\text{Definition } E_A[T])
\end{aligned}$$

The last line has the shape of the left side of axiom ($\nabla 1$). Hence, it is now enough to show that for the right hand side of ($\Sigma 1$) we obtain

$$\bar{\nu}\left(\bigvee\{\lambda^q(\bigwedge\psi) \mid (q,\psi) \in \Sigma RD(T)\}\right) = \bigvee\{\nabla\mathcal{T}(\bigwedge)\Phi \mid \Phi \in SRD(E_A[T])\}$$

This is done as follows

$$\begin{aligned}
&\bar{\nu}\left(\bigvee\ \{\lambda^q(\bigwedge\psi) \mid (q,\psi) \in \Sigma RD(T)\}\right) = \\
&\quad = \bigvee\{\bar{\nu}(\lambda^q(\bigwedge\psi)) \mid (q,\psi) \in \Sigma RD(T)\} && (\bar{\nu} \text{ is a morphism in } \mathsf{BA}) \\
&\quad = \bigvee\{\nabla E_A(q,\bigwedge\psi) \mid (q,\psi) \in \Sigma RD(T)\} && (\text{ Definition of } \bar{\nu}) \\
&\quad = \bigvee\{\nabla E_A\Sigma(\bigwedge)(q,\psi) \mid (q,\psi) \in \Sigma RD(T)\} && (\text{Defininition } \Sigma) \\
&\quad = \bigvee\{\nabla\mathcal{T}(\bigwedge)E_{\mathcal{P}(A)}(q,\psi) \mid (q,\psi) \in \Sigma RD(T)\} && (\text{Naturality of} E) \\
&\quad = \bigvee\{\nabla\mathcal{T}(\bigwedge)\Phi \mid \Phi \in SRD(E_A[T])\} && (\text{Lemma 5.3.31})
\end{aligned}$$

This concludes the case for ($\Sigma 1$) as it shows that it is translated into an instance of axiom ($\nabla 1$).

We now show that the axiom ($\Sigma 2$) translates into an instance of the axiom ($\nabla 2$). The left hand side of ($\Sigma 2$) is translated into

$$\begin{aligned}
\bar{\nu}(\lambda^q(\bigvee\psi)) &= \nabla E_X(q,\bigvee\psi) && (\text{Definition } \bar{\nu}) \\
&= \nabla E_A\Sigma(\bigvee)(q,\psi) && (\text{Definition } \Sigma) \\
&= \nabla\mathcal{T}(\bigvee)E_{\mathcal{P}(A)}(q,\psi). && (\text{Naturality of } E)
\end{aligned}$$

The last line is the left hand side of axiom ($\nabla 2$). Now it is enough to show that the right hand side of axiom ($\Sigma 2$) is

$$\bar{\nu}\left(\bigvee\{\lambda^p(a) \mid (p,a) \in CRD(q,\psi)\}\right) = \bigvee\{\nabla t \mid t\,\overline{\mathcal{T}}(\in_X)\,E_{\mathcal{P}(X)}(q,\psi)\}$$

This is seen as follows:

$$
\begin{aligned}
&\bar{\nu}\Big(\bigvee\{\lambda^p(a) \mid (p,a) \in CRD(q,\psi)\}\Big) = \\
&\qquad = \bigvee\{\bar{\nu}(\lambda^p(a)) \mid (p,a) \in CRD(q,\psi)\} && (\bar{\nu} \text{ is a Boolean morphism}) \\
&\qquad = \bigvee\{\nabla E_A(p,a) \mid (p,a) \in CRD(q,\psi)\} && (\text{Definition of } \bar{\nu}) \\
&\qquad = \bigvee\{\nabla t \mid t\,\overline{\mathcal{T}}(\in_A)\, E_{\mathcal{P}(A)}(q,\psi)\} && (\text{Lemma 5.3.32})
\end{aligned}
$$

This concludes the the proof of the claim. □

We now prove the implication from right to left.

**Claim.** For each pair $\varphi, \psi \in \bar{K}_{\mathcal{T}}(A)$, if $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\psi)$ then $\varphi \leq_{\mathsf{K}} \psi$.

*Proof.* The proof goes by induction on the complexity of the derivation $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\psi)$. A derivation on this system is a chain of inequalities $\leq_{\mathbf{M}}$ where each of those is either an instance of a boolean inequality valid on $A$, an instance of the rule ($\nabla 0$), an instance of the axioms ($\nabla 1$) and ($\nabla 2$), or is derived by previous inqualities by substitution or transitivity rules.

For the axioms ($\nabla 1$) and ($\nabla 2$) we show that they are instances of translations of the axioms ($\Sigma 1$) and ($\Sigma 2$), respectively.

For the axiom ($\nabla 1$), i.e. $\bigwedge\{\nabla t \mid t \in T\} \leq_{\mathcal{M}} \bigvee\{\nabla \mathcal{T}(\bigwedge)\Phi \mid \Phi \in SRD(T)\}$, for each $t \in T$ choose a representant $(p_t, a_t)$ of $t$, let $T_r$ be the set of those representants. From the work done defining $\nu$ it is immediate that the inequality above is obtained from $\bigwedge\{\lambda^{p_t}(a_t) \mid (p_t, a_t) \in T_r\} \leq \bigvee\{\lambda^q(\bigwedge\psi) \mid (q,\psi) \in \Sigma RD(T_r)\}$, an instance of axiom ($\Sigma 1$), by applying $\bar{\nu}$.

For the axiom ($\nabla 2$), i.e. $\nabla \mathcal{T}(\bigvee)\Phi \leq_{\mathcal{M}} \bigvee\{\nabla t \mid t\,\overline{\mathcal{T}}(\in)\,\Phi\}$, choose a representant $(q,\psi)$ of $\Phi$. From the work done in the definition of $\nu$ it is clear that this inequality is obtained by translating $\lambda^q(\bigvee\psi) \leq \bigvee\{\lambda^p(a) \mid (p,a) \in CRD(q,\psi)\}$.

For an instance of rule ($\nabla 0$) we have the following situation. First recall that for an $n$-ary Moss lifting $\lambda^p$ we have $\bar{\nu}(\Box_p(a)) = \nabla E_A(p,a)$, where $\Box_p$ is the modality associated with $\lambda^p$, and $E : \Sigma \longrightarrow \mathcal{T}$ is the natural transformation associated with the presentation.

Since we assume $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\psi)$ is an instance of ($\nabla 0$), we must have $\bar{\nu}(\varphi) = \nabla t$ and $\bar{\nu}(\psi) = \nabla t'$ for some $t, t' \in \mathcal{T}(A)$. Hence, since the component of $E$ are onto, there exists $(p,a),(q,b) \in K_{\mathcal{T}}(A)$ such that $\varphi = \Box_p(a)$ and $\psi = \Box_q(b)$. In particular we observe that $E_A(p,a)\overline{\mathcal{T}}(\leq_A)E_A(q,b)$. From here, by Lemma 5.1.10, there there exists $k < \omega$, $r \in \Sigma_k$, $a' : k \longrightarrow A$, and $b' : k \longrightarrow A$ such that $E_A(r,a') = E_A(p,a)$, $E_A(r,b') = E_A(q,b)$, and $(\forall i \in k)(a'_i \leq_A b'_i)$. The important observation here is that the inequalities $a'_i \leq_A b'_i$ are inequalities in $A$ hence by substitution, in the system $\mathsf{K}$, we obtain $\Box_r(a') \leq_{\mathsf{K}} \Box_r(b')$. From here we conclude

$$\Box_p(a) \cong_{\mathcal{T}} \Box_r(a') \leq_{\mathsf{K}} \Box_r(b') \cong_{\mathcal{T}} \Box_q(b).$$

This finishes the case for ($\nabla 0$).

We now detail the inductive case where $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\psi)$ has been deduced by transitivity from $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \gamma$ and $\gamma \leq_{\mathbf{M}} \bar{\nu}(\psi)$, where $\gamma \in M(A)$. The key observation here is that $\bar{\nu}_A : K_{\mathcal{T}}(A) \to M(A)$ is onto, this holds because $\bar{\nu}_A = F(E_{U(A)})$ and since $F$ is left adjoint it preserves coequalizers (surjections). Thus we can assume $\gamma = \bar{\nu}(\gamma')$. We can then apply inductive hypotheses to $\bar{\nu}(\varphi) \leq_{\mathbf{M}} \bar{\nu}(\gamma')$ and $\bar{\nu}(\gamma') \leq_{\mathbf{M}} \bar{\nu}(\psi)$ and obtain $\varphi \leq_{\mathsf{K}} \gamma'$ and $\gamma' \leq_{\mathsf{K}} \psi$. From here, by transitivity, we conclude $\varphi \leq_{\mathsf{K}} \psi$.

This concludes the proof of the claim. □

This concludes the proof of the theorem □

## 5.4 Conclusions

In this chapter, we have illustrated various uses of presentations of functors. A key development is the introduction of Moss liftings, Section 5.1.1. These predicate liftings are distinguished among all predicate liftings because they are always translatable into the Moss logic.

Using the structural properties of the base category and the presentation of the logic of all predicate liftings, we proved a first representation theorem (Theorem 5.2.2) stating that any coalgebraic modal logic can be translated into the logic of predicate liftings.

Presentations of functors over varieties, lead us to the concept of canonical signature (Definition 5.2.5). Using these, we transmuted such observation into a second representation theorem (Theorem 5.2.17) showing that every coalgebraic modal logic is a rank 1 axiomatization of a a logic of predicate liftings.

We finished by using presentations of the functor of the Moss logic to develop a sound and complete axiomatization of Moss liftings (Theorem 5.3.30). An important technical development here was the introduction of well-based presentation and basic operation Definition 5.3.13.

Presentations of functors as introduced here are related to the so-called analytical functors, introduced by A. Joyal [63]. Roughly speaking an analytic functor is a quotient of a polynomial functor where we also consider permutations of the operations. Perhaps we could use analytical functors to describe axiomatic systems beyond rank-1. More research in the subject is needed.

# Part II

# Coalgebraic Modal Logics at Work

# Chapter 6

# Describing Behavioural Equivalence: three sides of one coin

One of the most important contributions of universal coalgebra is the employment, for different purposes, of final coalgebras. Among those we have: final coalgebras provide a canonical solution to the domain equation $X \cong \mathcal{T}(X)$ [2], final coalgebras can be used to formalise the notions of proof and definition by coinduction [96], and final coalgebras characterise behavioural equivalent states in a canonical manner (Proposition 2.2.5). Elaborating this last point, we could even say that the behaviour of a state in a coalgebra is its image into the final coalgebra. Hence, a very important insight is that we can use final coalgebras to formalise the vague, yet intuitive, notion of behaviour. But this is not the only way to formalise the notion of behaviour. As we saw, we can also use logics for coalgebras to describe the behaviour of a state in a coalgebra. This bipolarity is the moving force of this chapter.

This chapter is about the relationship between final coalgebras and logics for coalgebras. The key conceptual contribution here is to realise that this affair is based on how we represent *behavioural equivalence* inside a category of coalgebras. The key insight of this chapter, is that, in harmony with Part I, in such issues the structural properties of the base category, of the coalgebras in this case, play crucial role.

The main contribution of this chapter is to further develop a systematic study of the relationship between the following three characterisations of behavioural equivalence:

- a structural characterisation using final coalgebras.
- a logical characterisation using coalgebraic languages.
- a structural characterisation using congruences obtained from coalgebraic languages; we call those *logical congruences*

The relationship between the first two items was somehow expected but nobody before Goldblatt [42] had made it explicit in the general case. The relation with the third item is a bit less known; Schröder [99] implicitly uses this third representation for the case of logics of predicate liftings.

We work in a general framework that covers all known logics for set coalgebras and easily generalizes to base categories different from the category Set. Our main theorem (Theorem 6.4.1) can be stated as follows:

> Given a set functor $\mathcal{T}$, a final $\mathcal{T}$-coalgebra exists iff there exists a language for $\mathcal{T}$-coalgebras with the Hennessy-Milner property iff there exists a language for $\mathcal{T}$-coalgebras that has logical congruences.

An important point to note here is that no conditions on the functor, whatsoever, are required. The equivalences entirely depend on the properties of the base category.

We provide relatively simple, and transparent, proofs for these equivalences in order to obtain our main theorem. In particular,

1. we simplify Goldblatt's proof,
2. generalize Schröder's argument, and in addition to that,
3. we use our framework to construct canonical models and characterise simple coalgebras by logical means.

Furthermore we demonstrate that our proofs allow for straightforward generalizations to base categories different from the category Set.

The standard construction of final coalgebras is via the terminal sequence, see e.g. [112, 88]; a concept that requires quite a bit of knowledge of category theory and can become very technical. Our work here can also be seen as an alternative to this construction. As we will later see, this change of view can be used to illustrate several well-known constructions from modal logic and universal coalgebra itself.

**The structure of the chapter is as follows:** In the next section, Section 6.1, we introduce abstract coalgebraic languages; present an elementary construction of final coalgebras. We finish by applying the material to construct canonical models. In Section 6.2 we discuss the connection between coalgebraic congruences and the Hennessy-Milner property; we apply these techniques to characterise simple coalgebras. In Section 6.3 we build up on the previous section and discuss how by using coalgebraic congruences we can see the Hennessy-Milner property of a language as a solution set condition; a concept familiar in category theory. We use this to prove the oldest theorem on the existence of final coalgebras (Theorem

6.3.5). Section 6.4 summarises the results in the previous sections. Finally, in Section 6.5 we explore generalizations of our work to other base categories. For this last section more knowledge of category theory is assumed.

## 6.1 An elementary construction of final coalgebras

We begin our journey by presenting a construction of final coalgebras using the formulas of a language for coalgebras. In the tradition of abstract model theory, languages are regarded just as sets and theories as subsets of those. Using this perspective, Goldblatt [42] explicitly showed how to construct a final coalgebra from a language with the Hennessy-Milner property. His construction relied on several properties of categories of coalgebras over Set e.g. congruences and the axiom of choice[1]. In this section we present the same idea in a much more elementary fashion using just "basic" set theory. We begin by introducing abstract coalgebraic languages.

### Abstract coalgebraic languages

Abstract coalgebraic languages are "languages" where we do not take any algebraic structure into consideration; in principle they are just sets. This generality will allow us to present a very elementary construction of final coalgebras. Unless explicitly stated, we will work with functors on Set.

**Definition 6.1.1.** An *abstract coalgebraic language* for $\mathcal{T}$-coalgebras, or simply a *language for $\mathcal{T}$-coalgebras*, is a set $\mathcal{L}$ together with a function $\Phi_\xi : X \longrightarrow \mathcal{P}(\mathcal{L})$ for each $\mathcal{T}$-coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$. The function $\Phi_\xi$ will be called *the theory map of* $\xi$, elements of $\mathcal{P}(\mathcal{L})$ will be called *$\mathcal{L}$-theories.* An abstract coalgebraic language will be denoted as a pair $(\mathcal{L}, \Phi)$.

Abstract coalgebraic languages were considered by Goldblatt in [42] under the name of "small logic". The next example presents basic modal logic as an abstract coalgebraic language.

**Example 6.1.2.** Let $\mathcal{T} = \mathsf{Pow}$ be the covariant power set functor. Recall that the category of Pow-coalgebras is isomorphic to the category of Kripke frames and bounded morphisms.

Let $\mathcal{L}$ be the set of closed modal formulas of the basic similarity type (see [20] for details). For an arbitrary Pow-coalgebra $\xi : X \longrightarrow \mathsf{Pow}(X)$ we define $\Phi_\xi : X \longrightarrow \mathcal{P}(\mathcal{L})$ to be the "modal theory map", i.e., the function which maps a state $x \in X$ to the set of formulas $\varphi \in \mathcal{L}$ such that $\xi, x \models \varphi$.

[1] The axiom of choice corresponds to the following property of categories. Every epimorphism has a right inverse.

The set $\mathcal{L}$ together with the family $\{\Phi_\xi\}_{\xi\in\mathsf{Coalg(Pow)}}$ is an abstract coalgebraic language for $\mathsf{Pow}$.

Notice that this $\mathcal{L}$ is also an abstract coalgebraic language for $\mathcal{T} = \mathsf{Pow}_\omega$-coalgebras, i.e. finite image Kripke Frames.

More generally, every coalgebraic modal logic, as in Chapter 3, induces an abstract coalgebraic language.

**Example 6.1.3.** Every coalgebraic logic $(L, \delta)$, over a category of power set algebras, for a functor $\mathcal{T}$, as in Definition 3.2.13, induces an abstract coalgebraic language. The language is given by the carrier of the initial $L$-algebra $(I, \iota)$.

In order to see this, first recall that using $\delta : LP \longrightarrow P\mathcal{T}$ we can assign to each $\mathcal{T}$-coalgebra $(X, \xi)$ its complex, or dual, $L$-algebra given by $(P(X), P(\xi) \circ \delta_X)$, see Fact 3.2.4. Now remember that the semantics of formulas on a coalgebra $(X, \xi)$ is given by the initial morphism $[\![-]\!]_\xi : (I, \iota) \longrightarrow (P(X), P(\xi) \circ \delta_X)$. Since in categories of power set algebras $\mathcal{P} = UP$, this morphism is given by a homonymous function $[\![-]\!]_\xi : I \longrightarrow \mathcal{P}(X)$; the $\mathcal{P}$-transpose of which gives the theory map $\Phi_\xi : X \longrightarrow \mathcal{P}(I)$.

Notice that as mentioned in Chapter 3, Remark 3.2.2, the elements of the abstract coalgebraic language in the previous example are not "formulas" in the usual sense but equivalence classes of formulas.

The next remark elaborates the previous example to argue why we would like to seek a more categorical treatment for abstract coalgebraic languages.

**Remark 6.1.4.** The previous example illustrates that the framework of coalgebraic modal logics over categories of power set algebras precisely fits the modal theoretical tradition of considering languages as sets and theories of subsets of those. However, if we would want to go further, e.g. considering coalgebraic modal logics as in Definition 3.2.22, the use of the powersets becomes uncomfortable. For example, from the categorical perspective, once we change the category of algebras $\mathcal{A}$ it is quite natural to also change $\mathsf{Set}$. For example, if we take $\mathcal{A} = \mathsf{DL}$ we would like to consider coalgebras over $\mathsf{Pos}$ instead of over $\mathsf{Set}$. Then somehow, we ought to replace the functor $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ by an appropriate functor $\mathsf{Pos}^{op} \longrightarrow \mathsf{Pos}$. Hence the language of these logics does not conformably fit in the framework of Definition 6.1.1. In Section 6.5 we investigate how we could overcome such difficulties.

As mentioned before, we are interested in describing behavioural equivalence of states. To do this we have two requirements on the language, which together lead to what sometimes is called expressivity:

1) *Adequacy:* the truth of formulas must be invariant under coalgebra morphisms.

2) *Hennessy-Milner property:* the language must distinguish states that are not behaviourally equivalent.

More formally we state the following definition.

**Definition 6.1.5.** An abstract coalgebraic language $\mathcal{L}$ for $\mathcal{T}$-coalgebras is said to be *adequate* if for every pair of pointed $\mathcal{T}$-coalgebras (see Definition 2.1.1) $(\xi_1, x_1)$ and $(\xi_2, x_2)$,

$$x_1 \sim x_2 \text{ implies } \Phi_{\xi_1}(x_1) = \Phi_{\xi_2}(x_2).$$

The language $\mathcal{L}$ is said to have the *Hennessy-Milner property* if for every pair of pointed $\mathcal{T}$-coalgebras $(\xi_1, x_1)$ and $(\xi_2, x_2)$,

$$\Phi_{\xi_1}(x_1) = \Phi_{\xi_2}(x_2) \text{ implies } x_1 \sim x_2.$$

The language $\mathcal{L}$ is said to be *expressive* if it is adequate and has the Hennessy-Milner property.

**Example 6.1.6.** Let $\mathcal{T}$ be a set functor.

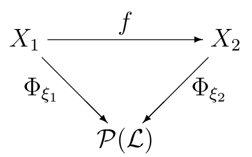
1. The Moss language for $\mathcal{T}$ (Definition 3.3.1) is expressive [84] for any category of power set algebras.

2. Every language of predicate liftings (Definition 3.1.5) is adequate but might not have the Hennessy-Milner property e.g. basic modal logic for all Kripke frames.

3. The language of all finitary predicate liftings for a finitary functor is expressive. In general, for a $\kappa$-accessible functor $\mathcal{T}$ (Definition 2.1.5), the language of of all predicate liftings (Definition 3.1.5), for $\mathcal{T}$, of arity less than $\kappa$ and conjunctions bounded by $\kappa$ is expressive [99].

The reader may worry that the definition of adequacy was not presented as "the truth of formulas are invariant under morphisms". The definition above comes from a tradition in modal logic where bisimilar states should satisfy the same formulas, in the category Set both presentations are equivalent.

**Proposition 6.1.7.** *Let $\mathcal{T}$ be a set endofunctor and $(\mathcal{L}, \Phi)$ an abstract coalgebraic language for $\mathcal{T}$-coalgebras. The language $\mathcal{L}$ is adequate iff the truth of formulas is invariant under coalgebra morphisms, i.e. if $f : \xi_1 \to \xi_2$ is a morphism of coalgebras then the following diagram commutes:*




*Proof.* If the language is adequate then the truth of formulas is invariant under coalgebra morphism because $x \sim f(x)$ for every coalgebra morphism $f$.

Assume now that the truth of formulas is invariant under coalgebra morphisms. We want to show that if states $x_1$ and $x_2$ in coalgebras $\xi_1$ and $\xi_2$, respectively, are behaviourally equivalent states then $\Phi_{\xi_1}(x_1) = \Phi_{\xi_1}(x_1)$; i.e. they satisfy the same formulas.

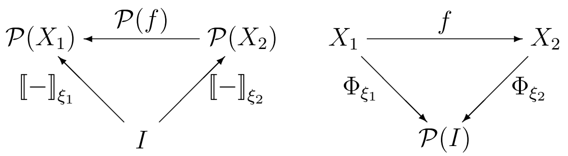
Since the states are behaviourally equivalent, there exists a coalgebra $(Y, \gamma)$ and coalgebra morphisms $f_i : \xi_i \to \gamma$, (i = 1,2), such that $f_1(x_1) = f_2(x_2)$. Since formulas are invariant under coalgebra morphisms we have

$$\Phi_{\xi_1}(x_1) = \Phi_\gamma(f_1(x_1)) = \Phi_\gamma(f_2(x_2)) = \Phi_{\xi_2}(x_2).$$

as we wanted to show. $\square$

Using the previous proposition we can show that the language induced coalgebraic modal logic (Example 6.1.3) is adequate.

**Example 6.1.8.** The abstract coalgebraic language induced by a coalgebraic modal logic $(L, \delta)$, see Example 6.1.3, is adequate. In Proposition 3.2.23 we showed that the interpretation of $(L, \delta)$-formulas is invariant under coalgebra morphisms. This means that for every morphism of coalgebras $f : \xi_1 \to \xi_2$ the diagram on the left




commutes. Let $\Phi_{\xi_i}$ be the transpose of $[\![-]\!]_{\xi_i}$. By properties of adjoints (Lemma A.1.3, item 3) the diagram on the left commutes iff the diagram on the right commutes. This implies that the language $(I, \Phi)$ is adequate.

If the components of the mate of $\delta : LP \to P\mathcal{T}$ are injective then the language also has the Hennessy-Milner property (Proposition 3.2.25).

The next remark elaborates on the previous proposition.

**Remark 6.1.9.** The previous proposition deserves some comments particularly relevant for the reader interested in categorical generalizations. First notice that the previous proposition provides a pointless definition of adequacy, i.e. this definition can be used in any base category $\mathbb{C}$. With this in mind, the categorically minded reader would recognise a natural transformation $\Phi : U \to \Delta_{\mathcal{P}(\mathcal{L})}$, where $U : \mathsf{Coalg}(\mathcal{T}) \to \mathbb{C}$ is the forgetful functor. This is also equivalently to the existence of a functor $\Phi : \mathsf{Coalg}(\mathcal{T}) \to (U \downarrow \mathcal{P}(\mathcal{L}))$, where the codomain of $\Phi$ is the appropriate comma category.

### An elementary construction of final coalgebras

In this section we present an elementary construction of final coalgebras. This section is intended for readers not very familiar with category theory and just basic knowledge of coalgebras. We try to present the construction in detail using very elementary techniques. The reader more familiar with category theory or coalgebra might want to skip to Section 6.2 where we use congruences of coalgebras to construct final coalgebras. Using congruences is particularly interesting because it presents the Hennessy-Milner property as a solution set condition and then the construction of final coalgebras as very simple result from the categorical point of view.

Recall that we said that we can use the states of the final coalgebra to describe the behaviour of a sate in a coalgebra. Since languages are just sets, we can rephrase this by saying that the states of a final coalgebra give an expressive language for coalgebras.

**Theorem 6.1.10** ([42])**.** *For any functor $\mathcal{T}$ : $\mathsf{Set} \to \mathsf{Set}$, if there exists a final coalgebra then there exists an expressive language for $\mathcal{T}$-coalgebras.*

*Proof.* Let $(Z, \zeta)$ be a final coalgebra, and call $f_\xi$ be the final map for a coalgebra $\xi$. Take $\mathcal{L} = Z$ and for each coalgebra $(X, \xi)$ define $\Phi_\xi(x) = \{f_\xi(x)\}$. Since $(Z, \zeta)$ is final this abstract language together with the maps $\Phi_\xi$, defined above, is adequate and has the Hennessy-Milner property. □

At a first glance the language above might seem unnatural. However, this language is essentially the language of coequations in [5, 104]. Here is a concrete example.

**Example 6.1.11.** Let $\mathcal{T} = 1 + (-)$. A final coalgebra for $\mathcal{T}$ is given by the set $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ together with a function $p : \overline{\mathbb{N}} \to 1 + \overline{\mathbb{N}}$ defined as follows $p(0) = *$; $p(n + 1) = n$; $p(\infty) = \infty$, where $*$ is the only element of 1. This presentation of the final coalgebra for $\mathcal{T}$ contains all the information about the observable behaviour of a state in a $\mathcal{T}$-coalgebra as a state can only either lead the machine to stop after $n$ steps or let the machine run forever.

We now proceed to the main theorem of this section. Informally speaking, the main result is the explicit construction of a final coalgebra from an expressive abstract coalgebraic language. The theorem itself was first stated in [42] but we will give a new simpler proof.

**Theorem 6.1.12** ([42])**.** *For any functor $\mathcal{T}$ : $\mathsf{Set} \to \mathsf{Set}$, if there exists an expressive abstract coalgebraic language for $\mathcal{T}$-coalgebras then there exists a final coalgebra.*

We make some observations and describe the construction before presenting the proof. The following are the main features of our construction:

- the first key idea is to notice that if we have a language for $\mathcal{T}$-coalgebras, we can identify a concrete set (object) $Z$ which is a natural candidate for the carrier of a final coalgebra.
- The second observation is that for each coalgebra $(X, \xi)$ there is a natural map $X \longrightarrow \mathcal{T}(Z)$.
- The final observation is: should the language be adequate and have the Hennessy-Milner property then we can combine these functions into a function $\zeta : Z \longrightarrow \mathcal{T}(Z)$ which endows $Z$ with the structure of a final $\mathcal{T}$-coalgebra.

Moreover, using this approach we can show that the function $\zeta$ exists if and only if the language has the Hennessy-Milner property.

When we try to characterise behavioural equivalence, there are two sides to one coin. On the one side, Proposition 2.2.5 tells us that final coalgebras characterise behavioural equivalence. On the other side, if a language is expressive, the satisfiable theories characterise behavioural equivalence. Hence, a natural candidate for the carrier of a final coalgebra is the set of satisfiable theories of the language.

**Definition 6.1.13.** Given a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ and an abstract coalgebraic language $(\mathcal{L}, \Phi)$ for $\mathcal{T}$-coalgebras, the set $Z_{\mathcal{L}}$ of **$\mathcal{L}$-satisfiable theories** is the set

$$Z_{\mathcal{L}} = \{\Psi \subseteq \mathcal{L} \,|\, (\exists \xi)(\exists x \in \xi)(\Phi_{\xi}(x) = \Psi)\},$$

where $x \in \xi$ means that $x$ belongs to the state space of $\xi$. We often drop the subindex $\mathcal{L}$ to simplify our notation.

Clearly the set of satisfiable theories exists for any language. Moreover, notice that if the language would be expressive then the theory $\Phi_{\xi}(x)$ would be the behaviour of the state $x$ in the coalgebra $\xi$. Laying this aside of the intuition on final coalgebras are a mean to characterise the behaviour of coalgebras, we conclude that the set $Z_{\mathcal{L}}$ is (up to isomorphisms) the only natural choice for the carrier of a final coalgebra. As we will later show, the properties of adequacy and Hennessy-Milner ensure this to work.

**Remark 6.1.14.** The reader might worry that in the definition of $Z_{\mathcal{L}}$ we quantify over all coalgebras and all states on them. Hence, we might not be defining a set but a proper class. This is not an issue as we required the language $\mathcal{L}$ to be a set and obviously $Z_{\mathcal{L}} \subseteq \mathcal{P}(\mathcal{L})$. In other words, using the axiom of replacement we make $Z_{\mathcal{L}}$ into a set.

Now we describe the construction needed to prove Theorem 6.1.12.

First notice that for each coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ there is a canonical map $f_\xi : X \longrightarrow Z_\mathcal{L}$ obtained by restricting the codomain of $\Phi_\xi : X \longrightarrow \mathcal{P}(\mathcal{L})$. This restriction is possible as the range of $\Phi_\xi$ is clearly contained in $Z_\mathcal{L}$. In the following, to simplify our notation, we will drop the subindex $\mathcal{L}$, i.e. we write $Z$ instead of $Z_\mathcal{L}$. Using $f_\xi$, we see that for each coalgebra $(X, \xi)$ there is a natural function from $X$ to $\mathcal{T}(Z)$, namely the lower path in the following square

$$\begin{array}{ccc} X & \xrightarrow{f_\xi} & Z \\ {\scriptstyle \xi}\downarrow & & \\ \mathcal{T}(X) & \xrightarrow[\mathcal{T}(f_\xi)]{} & \mathcal{T}(Z) \end{array} \tag{6.1}$$

i.e $\mathcal{T}(f_\xi) \circ \xi$. This suggests the following assignment $\zeta : Z \longrightarrow \mathcal{T}(Z)$:

> a theory $f_\xi(x) = \Phi_\xi(x) \in Z$ is assigned to
>
> $$\zeta(\Phi_\xi(x)) := \mathcal{T}(f_\xi) \circ \xi(x). \tag{6.2}$$

Since in general we may have $\Phi_{\xi_1}(x_1) = \Phi_{\xi_2}(x_2)$ for different pointed coalgebras $(\xi_1, x_1)$ and $(\xi_2, x_2)$ it is not immediate that Equation (6.2) defines a function. We now show that this is indeed the case if the language is adequate and has the Hennessy-Milner property. We prove this in two steps whose illustrate that both conditions are really needed.

First we show that $\zeta$, as defined in Equation (6.2), is well defined if the states are related by a coalgebra morphism. The next lemma states this formally.

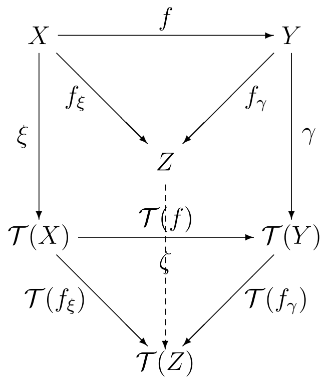
**Lemma 6.1.15.** *Let $(\mathcal{L}, \Phi)$ be an adequate language for $\mathcal{T}$-coalgebras. For any morphism $f : \xi \longrightarrow \gamma$ we have:*

$$\mathcal{T}(f_\xi) \circ \xi = \mathcal{T}(f_\gamma) \circ \gamma \circ f,$$

*where $f_\xi$ and $f_\gamma$ are obtained from the respective theory maps by restricting the codomain, as described above.*

*Proof.* To simplify the notation, we write $Z$ for the set of $\mathcal{L}$-satisfiable theories.

The situation is depicted in the following diagram




The equation in the statement of the lemma says that the pentagon in the back commutes. Since $\mathcal{L}$ is adequate, the upper triangle commutes; therefore, since $\mathcal{T}$ is a functor, the lower triangle commutes, i.e. $\mathcal{T}(f_\gamma) \circ \mathcal{T}(f) = \mathcal{T}(f_\xi)$. Now notice that the back rectangle commutes because $f$ is a morphism of $\mathcal{T}$-coalgebras. Chasing around the diagram we obtain:

$$\begin{aligned} \mathcal{T}(f_\gamma) \circ \gamma \circ f &= \mathcal{T}(f_\gamma) \circ \mathcal{T}(f) \circ \xi && (f \text{ is a coalgebra morph.}) \\ &= \mathcal{T}(f_\xi) \circ \xi. && \text{(lower triangle)} \end{aligned}$$

This concludes the proof. □

We can now show that if in addition to adequacy $\mathcal{L}$ has the Hennessy-Milner property, Equation (6.2) defines a function $\zeta : Z \longrightarrow \mathcal{T}(Z)$. In fact, these two conditions are equivalent.

**Theorem 6.1.16.** *Let $\mathcal{T}$ be a set functor, let $(\mathcal{L}, \Phi)$ be an* adequate *language for $\mathcal{T}$-coalgebras, let $Z_\mathcal{L}$ be the set of $\mathcal{L}$-satisfiable theories, and let $f_\xi : X \longrightarrow Z_\mathcal{L}$ be the function obtained from a theory map $\Phi_\xi$ by restricting the codomain.*

*The following are equivalent:*

1. *The language $(\mathcal{L}, \Phi)$ has the Hennessy-Milner property.*

2. *The assignment $\zeta$ which takes an $\mathcal{L}$-theory $\Psi = \Phi_\xi(x) \in Z_\mathcal{L}$ to $\mathcal{T}(f_\xi)\xi(x)$ does not depend on the choice of $(\xi, x)$, i.e. Equation (6.2) defines a function $\zeta : Z_\mathcal{L} \longrightarrow \mathcal{T}(Z_\mathcal{L})$.*

*Proof.* **From top to bottom:** Assume we have $\Phi_{\xi_1}(x_1) = \Phi_{\xi_2}(x_2)$. Since $\mathcal{L}$ has the Hennessy-Milner property there exists a coalgebra $(Y, \gamma)$ and morphisms $f_1 : \xi_1 \longrightarrow \gamma$ and $f_2 : \xi_2 \longrightarrow \gamma$ such that $f_1(x_1) = f_2(x_2)$. This combined with the adequacy of $\mathcal{L}$ and the previous lemma implies

$$\begin{aligned}
\zeta(\Phi_{\xi_1}(x_1)) &= \mathcal{T}(f_{\xi_1})\xi_1(x_1) \\
&= \mathcal{T}(f_\gamma)\gamma f_1(x_1) && \text{(previous lemma)} \\
&= \mathcal{T}(f_\gamma)\gamma f_2(x_2) && (f_1(x_1) = f_2(x_2)) \\
&= \mathcal{T}(f_{\xi_2})\xi_2(x_2) && \text{(previous lemma)} \\
&= \zeta(\Phi_{\xi_2}(x_2)).
\end{aligned}$$

This precisely states that $\zeta$ does not depend on the choice of $(\xi, x)$; i.e $\zeta$ defines a function.

**From bottom to top:** Assume $\zeta$ does not depend on the representant $(\xi, x)$; i.e. we have a function $\zeta : Z_{\mathcal{L}} \to \mathcal{T}(Z_{\mathcal{L}})$. We have to show that $\mathcal{L}$ has the Hennessy-Milner property. It is immediate by construction of $\zeta$, that for each coalgebra $\xi$ the function $f_\xi$ is a coalgebra morphism from $\xi$ to $\zeta$. Moreover, any two states that are logically equivalent will be identified by the corresponding theory maps therefore also by the respective $f_\xi$ maps. Hence, logically equivalent states are behaviourally equivalent. □

As mentioned in the previous proof, for each coalgebra $\xi$ the function $f_\xi : X \to Z_{\mathcal{L}}$ is a morphism between the coalgebras $\xi$ and $\zeta$. We make this explicit as it will be used it in the proof of Theorem 6.1.12.

**Corollary 6.1.17.** *Let $\mathcal{T}$ be a set functor, let $(\mathcal{L}, \Phi)$ be an expressive language for $\mathcal{T}$-coalgebras, let $Z_{\mathcal{L}}$ be the set of $\mathcal{L}$-satisfiable theories, and let $f_\xi : X \to Z_{\mathcal{L}}$ be the function obtained from a theory map $\Phi_\xi$ by restricting the domain.*

*For any coalgebra $\xi$, the function $f_\xi : \xi \to \zeta$ is a morphism of coalgebras.*

This corollary already implies that a final coalgebra exists; it can be obtained by quotient $\zeta$ modulo bisimilarity (see [96]). However, we can directly show that $(Z, \zeta)$ is a final object without having to go into the details of congruences; i.e. our construction is simpler.

We now prove a technical lemma which we will use in the proof of Theorem 6.1.12 and in our application to canonical models.

**Lemma 6.1.18.** *Let $\mathcal{T} : \mathsf{Set} \to \mathsf{Set}$ be a functor and let $(\mathcal{L}, \Phi)$ be an expressive language for $\mathcal{T}$-coalgebras. Let $Z$ be the set of satisfiable $\mathcal{L}$-theories and let $\zeta : Z \to \mathcal{T}(Z)$ be the coalgebra defined by Equation* (6.2).

*The theory map $\Phi_\zeta : Z \to \mathcal{P}(\mathcal{L})$ is the inclusion.*

*Proof.* The idea is to show that there exists a coalgebra $(Y, \gamma)$ such that the function $f_\gamma : Y \to Z$ is onto. From this, since $\mathcal{L}$ is adequate we conclude $i_Z f_\gamma = \Phi_\gamma = \Phi_\zeta f_\gamma$ and then, because $f_\gamma$ is onto, $i_Z = \Phi_\zeta$.

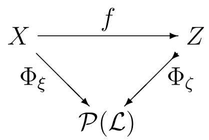
We now show how to construct $(Y, \gamma)$. For each element $\Psi \in Z$ choose a $\mathcal{T}$-coalgebra $(X_\Psi, \xi_\Psi)$ and a state $x \in X_\Psi$ such that $\Phi_{\xi_\Psi}(x) = \Psi$. The coalgebra $(Y, \gamma)$

is given by the coproduct of all these coalgebras, i.e. $(Y, \gamma) = \coprod_{\Psi \in Z}(X_\Psi, \xi_\Psi)$ in $\mathsf{Coalg}(\mathcal{T})$. Since $\mathcal{L}$ is adequate and each of the coproduct inclusions is a morphism of coalgebras, we conclude that the image of $\Phi_\gamma : Y \longrightarrow \mathcal{P}(\mathcal{L})$ is $Z$. $\square$

This finishes the description of the construction. Now we have all the material to prove Theorem 6.1.12.

*Proof of Theorem 6.1.12.* Let $Z$ be the set of $\mathcal{L}$-satisfiable theories, and let $\Phi_\xi : X \longrightarrow \mathcal{P}(\mathcal{L})$ be the theory map for a coalgebra $(X, \xi)$. Let $f_\xi : X \longrightarrow Z$ be the function obtained by restricting the codomain of $\Phi_\xi$. Theorem 6.1.16 implies that the assignment $\zeta$ which takes a theory $\Phi_\xi(x) \in Z$ to $\mathcal{T}(f_\xi)\xi(x)$ does not depend on the choice of $(\xi, x)$, i.e., it is a function $\zeta : Z \longrightarrow \mathcal{T}(Z)$. Corollary 6.1.17 implies that for each coalgebra $\xi$ the function $f_\xi : \xi \longrightarrow \zeta$ is a morphism of coalgebras.

It is only left to show that $f_\xi : \xi \longrightarrow \zeta$ is the only morphism of coalgebras. Since the language is adequate, this will follow because any morphism of coalgebras $f : \xi \longrightarrow \zeta$ makes the following diagram




commute; in particular $f_\xi$ makes the diagram commute. Therefore for every coalgebra morphism $f : \xi \longrightarrow \zeta$ we have $\Phi_\zeta \circ f = \Phi_\xi = \Phi_\zeta \circ f_\xi$. Hence since $\Phi_\zeta$ is injective, Lemma 6.1.18, we conclude $f = f_\xi$. $\square$

Gathering Theorem 6.1.10 and Theorem 6.1.12 we have:

**Corollary 6.1.19.** *For any functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ the following are equivalent:*

1. *There exists a final $\mathcal{T}$-coalgebra.*
2. *There exists an expressive language for $\mathcal{T}$-coalgebras.*

As an illustration, consider the contrapositive of previous result. If a functor $\mathcal{T}$ fails to have a final coalgebra there is no way to completely describe the behaviour of $\mathcal{T}$-coalgebras using any language for which the collection of formulas forms a set. Also notice that the proof of Theorem 6.1.12 tells us a bit more about the relation of final coalgebras and abstract coalgebraic languages; we can fine tune Theorem 6.1.16 as follows.

**Theorem 6.1.20.** *Let $(\mathcal{L}, \Phi)$ be a language for $\mathcal{T}$-coalgebras, let $Z$ be the set of $\mathcal{L}$-satisfiable theories. Let $f_\xi$ be the function obtained from the theory map $\Phi_\xi$ by restricting the codomain.*

*The following are equivalent:*

1. *The language $\mathcal{L}$ is adequate and has the Hennessy-Milner property.*

2. *There exists a function $\zeta : Z \longrightarrow \mathcal{T}(Z)$ which furnishes $Z$ with a coalgebra structure in such a way that $(Z, \zeta)$ is final and for each coalgebra $(X, \xi)$ the function $f_\xi : X \longrightarrow Z$ is the final map.*

*Proof.* From bottom to top, the Hennessy-Milner property follows because for each coalgebra $\xi$ the map $f_\xi$ is a morphism of coalgebras. Adequacy follows because $f_\xi$ is the only possible coalgebra map from $\xi$ to $\zeta$.

The implication from top to bottom follows from the proof of Theorem 6.1.12. □

**An application: Canonical Models**

Until now we have illustrated that for set endofunctors, there exists a final $\mathcal{T}$-coalgebra iff there exists an expressive language for $\mathcal{T}$-coalgebras.

As mentioned before, the language given by a final coalgebra are the states of the final coalgebra itself. Example 6.1.11 and the work on coequations [5, 104] illustrate how this language is non trivial and actually worth of study.

In this section we illustrate the assembly of final coalgebras from languages by showing how this technique is in fact the construction of canonical models from modal logic, see [20].

In Lemma 6.1.18 we showed that the theory map of $(Z, \zeta)$ is the inclusion. Since the states of $(Z, \zeta)$ are the satisfiable theories of $\mathcal{L}$ we can rewrite this as the well known Truth Lemma of modal logic.

**Lemma 6.1.21** (Truth Lemma)**.** *Let $(\mathcal{L}, \Phi)$ be an expressive language for $\mathcal{T}$-coalgebras. Let $Z$ be the set of $\mathcal{L}$-satisfiable theories (Definition 6.1.13) and let $\zeta : Z \longrightarrow \mathcal{T}(Z)$ be the function defined as in Equation* (6.2)*.*

*For any $\Psi \in Z$ and any $\varphi \in \mathcal{L}$ we have*

$$\Psi \Vdash_\zeta \varphi \textit{ iff } \varphi \in \Psi,$$

*where $\Psi \Vdash_\zeta \varphi$ means $\varphi \in \Phi_\zeta(\Psi)$.*

The previous lemma illustrates that our construction is similar to the canonical model construction from modal logic (see [20]). The difference relies on that we use the set of satisfiable theories to build the final coalgebra whereas the usual canonical modal construction uses the set of consistent theories to shape the canonical model. Assuming that the language $\mathcal{L}$ has some notion of consistency we can ask: are maximally consistent sets satisfiable? In other words, we could investigate completeness. We do not peruse such issue in this chapter, but notice the following result:

**Proposition 6.1.22.** *Let $(\mathcal{L}, \Phi)$ be an adequate language for $\mathcal{T}$-coalgebras, and let $Z$ be the set of $\mathcal{L}$-satisfiable theories. The set $Z$ is the largest subset of $\mathcal{P}(\mathcal{L})$ for which we can define a $\mathcal{T}$-coalgebra structure $\zeta : Z \longrightarrow \mathcal{T}(Z)$ such that*

1. *the Truth Lemma is satisfied, i.e. the theory map is the inclusion, and*

2. *for each coalgebra the codomain restrictions of the theory maps are morphisms of $\mathcal{T}$-coalgebras.*

*Proof.* Let $Z' \subseteq \mathcal{P}(\mathcal{L})$ be a set for which conditions 1) and 2) are satisfied, and let $\zeta'$ be the mentioned coalgebraic structure. We want to show $Z' \subseteq Z$. By Theorem 6.1.16, the second item implies that $\mathcal{L}$ has the Hennessy-Milner property. From this, using Theorem 6.1.20, we conclude that there is a coalgebra $\zeta : Z \longrightarrow \mathcal{T}(Z)$ which is final. By construction, the final map $f_{\zeta'} : \zeta' \longrightarrow \zeta$ is obtained by restricting the codomain of the theory map $\Phi_{\zeta'}$. This together with condition 1) implies that $f_{\zeta'}$ is an inclusion map because for every $x \in Z'$ we have

$$\begin{aligned} x &= \Phi_{\zeta'}(x) && \text{(Condition 1)} \\ &= \Phi_{\zeta} f_{\zeta'}(x) && \text{(Adequacy)} \\ &= f_{\zeta'}(x). && \text{(Lemma 6.1.18)} \end{aligned}$$

In other words, $Z' \subseteq Z$ as we wanted to show. □

We finish this section with two illustrations concerning the canonical model $M$ for the logic **K**, see [20].

In the first place, notice that $M$ is not a final Kripke frame. If this would be the case, we would then conclude that modal logic has the Hennessy-Milner property with respect to all frames; this is well known to be false.

In the second place, notice that $M$ is neither final for finite image Kripke frames, because the set of consistent **K**-theories is strictly larger than the set of theories satisfiable in a finite image Kripke frame, see [20].

As a consequence, we conclude that there should exists a (finite image) Kripke frame $(X, \xi)$ for which the theory map $\Phi_\xi : X \longrightarrow M$ is not a bounded morphism.

## 6.2 Behaviour & Congruences

In this section we introduce congruences of coalgebras and illustrate their relation with abstract coalgebraic languages.

The Hennessy-Milner property states that if two states are logically equivalent then they are identified in some coalgebra. However, this coalgebra is not made explicit. The work in the previous section provides a canonical coalgebra where logically equivalent states are identified, namely the final coalgebra. In this section we investigate another construction to identify logically equivalent states; we use so-called *logical congruences*. We now recall the notion of coalgebraic congruence and its equivalent characterisations.

**Definition 6.2.1.** Let $(X, \xi)$ be a $\mathcal{T}$-coalgebra for a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$. An equivalence relation $\theta$ on the set $X$ is a *congruence of* $\mathcal{T}$*-coalgebras* iff there exists a coalgebraic structure $\xi_\theta : X/\theta \longrightarrow \mathcal{T}(X/\theta)$ such that the following diagram

$$\begin{array}{ccc} X & \xrightarrow{\quad e \quad} & X/\theta \\ {\scriptstyle \xi}\downarrow & & \downarrow{\scriptstyle \xi_\theta} \\ \mathcal{T}(X) & \xrightarrow[\mathcal{T}(e)]{} & \mathcal{T}(X/\theta) \end{array}$$

commutes. Here $e$ is the canonical quotient map.

In the category $\mathsf{Set}$, it can be shown that the notion of a congruence for coalgebra can be characterised as the kernel of coalgebra morphisms. In other words, it behaves like the notion of a congruence in universal algebra [96]. Congruences for coalgebras where first introduced by Aczel and Mendler in [3].

**Fact 6.2.2** ([44])**.** *Let $(X, \xi)$ be a $\mathcal{T}$-coalgebra for a functor $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$, For an equivalence relation $\theta$, on $X$, the following conditions are equivalent:*

1. *$\theta$ is a congruence of coalgebras.*
2. *$\theta \subseteq \ker(\mathcal{T}(e) \circ \xi)$.*
3. *$\theta$ is the kernel of some morphism of $\mathcal{T}$-coalgebras with domain $\xi$.*

The next example relates bisimulations and congruences.

**Example 6.2.3.** If $\mathcal{T}$ is the covariant power set functor, two states in a coalgebra $(X, \xi)$ are related by a congruence iff they are related by some bisimulation.

First recall that, by definition (Remark 2.2.3, page 19), a bisimulation $R$ on $(X, \xi)$ induces coalgebra $\rho : R \longrightarrow \mathsf{Pow}(R)$ and a pair of coalgebra morphisms $(R, \rho) \xrightarrow{f_i} (X, \xi)$, $i \in \{1, 2\}$. The argument goes as follows.

From right to left. Let $q : (X, \xi) \longrightarrow (Y, \gamma)$ be the coequalizer, in $\mathsf{Coalg}(\mathcal{T})$, of $f_1$ and $f_2$. From the previous fact, we know that the kernel of $q$ is a congruence; by construction of coequalizer in $\mathsf{Set}$ we also have $R \subseteq \ker(q)$; more precisely, $\ker(q)$ is the equivalence relation generated by $R$.

From left to right. Notice that in the case of $\mathsf{Pow}$ the kernel of a morphism of coalgebras, i.e. a congruence, is a bisimulation. In fact, this is the case for any functor that weakly preserves kernels (see [96] for details)

**Remark 6.2.4.** Fact 6.2.2 depends on the fact that set functors preserve monomorphisms with non-empty domain, as will become clear in Proposition 6.5.5. This is not true in all categories.

### Simple Coalgebras

Before presenting the relationship between behavioural equivalence and congruences, we discuss the notion of a *simple coalgebra*. These coalgebras will be of peculiar interest to us because using the Hennessy-Milner property of a coalgebraic language we can show that they from a solution set.

As in algebra, the set of all congruences on a coalgebra (X, $\xi$) is a complete lattice under the partial ordering of set inclusion. In particular, there is a smallest congruence $\Delta_X$ (the identity relation on $X$) and a largest congruence. However, unlike in the universal algebra case where every algebra $A$ has at least two trivial congruences; one is the identity $\Delta_A$ and the other is the universal relation $A \times A$. The largest congruence in coalgebra may be smaller than the universal relation, the next example shows that the universal relation may not be a congruence of coalgebras.

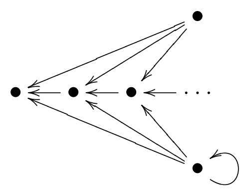
**Example 6.2.5.** Consider a constant functor $\mathsf{K}_C$, where $C$ has more than two elements. Consider a $\mathsf{K}_C$-coalgebra $\xi : X \longrightarrow C$ for which the structural map is not constant. The universal relation $X \times X$ is not a congruence; the largest congruence is given by $\ker(\xi)$. Hence the largest congruence is equal to the universal relation only if $\xi$ is constant.

An algebra is said to be *simple* if it has no non-trivial congruences. Following the same spirit, simple coalgebras are defined as coalgebras with only one congruence.

**Definition 6.2.6.** A coalgebra $\xi : X \longrightarrow \mathcal{T}(X)$ is *simple* if its largest (and hence only) congruence is the identity relation $\Delta_X$.

Here is an example of a simple coalgebra.

**Example 6.2.7.** The following Kripke frame is a simple $\mathsf{Pow}$-coalgebra.



From Example 6.2.3 we know that two states in the coalgebra above will be related by a congruence iff they are related by a bisimulation. But a straightforward argument will show that non of the points in the Kripke frame above is bisimilar to any other point in the frame; hence the coalgebra is simple.

Using coalgebraic languages we can give a more concrete characterisation of simple coalgebras; a first step is given by the following result.

**Proposition 6.2.8.** *Let $\mathcal{T}$ be a set functor and let $(\mathcal{L}, \Phi)$ be an adequate language for $\mathcal{T}$-coalgebras. A $\mathcal{T}$-coalgebra $\xi$ is simple if the theory map $\Phi_\xi$ is injective.*

*Proof.* Assume $\Phi_\xi$ to be injective. Since the language is adequate, for any morphism $f : \xi \longrightarrow \gamma$ we have $\Phi_\xi = \Phi_\gamma f$, which implies that $f$ is injective. In other words, every coalgebra morphism with domain $\xi$ is injective. Since congruences on $\xi$ are the kernel of coalgebra morphism with domain $\xi$, Fact 6.2.2, we conclude that the only congruence on $\xi$ is the identity, hence $\xi$ is simple. □

The converse of the previous proposition is not true in general; the Kripke frame in Example 6.2.7 is a counterexample. As we will later see, if the language $\mathcal{L}$ also has the Hennessy-Milner property then we obtain an equivalence. Moreover, since we assume languages are sets the Hennessy-Milner property implies that, up to isomorphisms, there is a set of simple coalgebras. Hence by Freyd's (Adjoint Functor) Theorem, see Theorem 6.3.1 here, a final coalgebra exists.

We now detail the fact that using the Hennessy-Milner property we can bound the class of simple coalgebras. The trick is to characterise simple coalgebras using a "logical" representation of each coalgebra; we do this using logical congruences, a concept that we now introduce.

### Logical congruences

Logical congruences are congruences obtained using logical equivalence of states.

**Definition 6.2.9.** Given an abstract coalgebraic language $(\mathcal{L}, \Phi)$, we say that two pointed coalgebras $(\xi_i, x_i)$, (i = 1,2), are *logically equivalent*, written $(\xi_1, x_1) \leftrightsquigarrow_{\mathcal{L}} (\xi_2, x_2)$, iff $\Phi_{\xi_1}(x_1) = \Phi_{\xi_2}(x_2)$. We call $\leftrightsquigarrow_{\mathcal{L}}$ the *logical equivalence relation of states*. Given a coalgebra $\xi$, we write $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ for the relation $\leftrightsquigarrow_{\mathcal{L}}$ restricted to the states of $\xi$.

Our interest in these equivalence relations has two main reasons. The first one is to make Proposition 6.2.8 into an equivalence and then obtain a concrete characterisation of simple coalgebras. The second and most important motivation is to generalise Proposition 6.1.19 and Theorem 6.1.20 to arbitrary categories. To our surprise logical congruences proved to be remarkably useful to simplify our constructions. As we will see, logical congruences provide a well-balanced categorical description of the Hennessy-Milner property. In fact, in the presence of the Hennessy-Milner property logical equivalence in a coalgebra coincides, by definition, with behavioural equivalence of states. Hence logical equivalence is a congruence. We now detail how to make this quotient. One reason for this is that it will reveal an abstract (pointless) face of the Hennessy-Milner property which is ideal for generalizations beyond $\mathsf{Set}$.

**Definition 6.2.10.** Let $(\mathcal{L}, \Phi)$ be an abstract coalgebraic language for $\mathcal{T}$-coalgebras. For each $\mathcal{T}$-coalgebra $(X, \xi)$, we write $Z_\xi$ for the quotient $X/\leftrightsquigarrow^{\xi}_{\mathcal{L}}$. We identify $Z_\xi$ with the *set of satisfiable theories in* $\xi$, i.e.

$$Z_\xi := \{\Psi \subseteq \mathcal{L} \mid (\exists x \in X)(\Phi_\xi(x) = \Psi)\}.$$

We use $e_\xi : X \longrightarrow Z_\xi$ for the canonical (quotient) map.

If our language happens to be adequate and have the Hennessy-Milner property we can show that logical equivalence of states is a congruence of coalgebras.

**Lemma 6.2.11.** *Let $(\mathcal{L}, \Phi)$ be a language for $\mathcal{T}$-coalgebras. If $\mathcal{L}$ is expressive, then for each coalgebra $(X, \xi)$ the relation $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ is a congruence of coalgebras. Moreover, $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ is the largest congruence on $(X, \xi)$.*

*Proof.* The idea is to follow the construction in Theorem 6.1.12, on page 143, relativized to the set $Z_\xi$. Explicitly this is: we define a function $\zeta_\xi : Z_\xi \longrightarrow \mathcal{T}(Z_\xi)$ such that the canonical map $e_\xi$ is a morphism of coalgebras.

The function $\zeta_\xi$ is defined as follows: an element $\Psi \in Z_\xi$ such that $\Psi = \Phi_\xi(x)$ is mapped to $\zeta_\xi(\Psi) = T(e_\xi)(\xi(x))$. Following the argument in the proof of Theorem 6.1.12 it is not difficult to see that $\zeta_\xi$ is well-defined because $\mathcal{L}$ is adequate and has the Hennessy-Milner property. It is a direct consequence of the definition of $\zeta_\xi$ that $e_\xi$ is a coalgebra morphism from $\xi$ to $\zeta_\xi$. Hence $\ker(e_\xi)$, which is equal to $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$, is a congruence on $\xi$.

In order to see that $\ker(e_\xi)$ is the largest congruence, one has to observe that for any $x, x' \in \xi$ with $e_\xi(x) \neq e_\xi(x')$ we have $\Phi_\xi(x) \neq \Phi_\xi(x')$ and thus, by adequacy of $\mathcal{L}$, there can be no coalgebra morphism $f$ with $f(x) = f(x')$. □

Mind how adequacy is crucial to prove that $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ is the largest congruence.

Now we can easily make Proposition 6.2.8 into an equivalence.

**Theorem 6.2.12.** *Let $\mathcal{T}$ be a set endofunctor, and let $(\mathcal{L}, \Phi)$ be an expressive language for $\mathcal{T}$-coalgebras. A $\mathcal{T}$-coalgebra $\xi$ is simple iff the theory map $\Phi_\xi$ is injective.*

*Proof.* The implication from right to left is Proposition 6.2.8. For the implication from left to right, notice that the previous Lemma tells us that $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ is a congruence of coalgebras, hence since $\xi$ is simple we obtain $\leftrightsquigarrow^{\xi}_{\mathcal{L}} = \Delta_X$, the identity. Therefore, $\ker(\Phi_\xi) = \ker(f_\xi) = \leftrightsquigarrow^{\xi}_{\mathcal{L}} = \Delta_X$, and this concludes the proof. □

Before proceeding an useful corollary.

**Corollary 6.2.13.** *If $(\mathcal{L}, \Phi)$ is an expressive language for $\mathcal{T}$-coalgebras then every coalgebra morphism $f : \xi \longrightarrow \gamma$ restricts to a coalgebra morphism $\hat{f} : \zeta_\xi \longrightarrow \zeta_\gamma$ such that $e_\gamma f = \hat{f} e_\xi$.*

*Proof.* Let $\pi_1$ and $\pi_2$ be the projections from $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$. To define $\hat{f}$ we will use the universal property of $e_\xi$. In order to do this we ought to show $e_\gamma f \pi_1 = e_\gamma f \pi_2$, i.e. that $e_\gamma f$ coequalizes the projections. In other words, we want to show that if two states $x$ and $y$ in $\xi$ are logically then so is their image under $f$. This follows because the language is expressive. Indeed, since $f$ is a coalgebra morphism then we have $f(x) \sim x$ which by adequacy implies $f(x) \leftrightsquigarrow_{\mathcal{L}} x$; the same holds for $y$. Since we assume $x \leftrightsquigarrow_{\mathcal{L}} y$ and $\leftrightsquigarrow_{\mathcal{L}}$ is transitive we conclude $f(x) \leftrightsquigarrow_{\mathcal{L}} f(y)$. Hence by the universal property of coequalizers we obtain a morphism $\hat{f} : \zeta_\xi \longrightarrow \zeta_\gamma$ such that $e_\gamma f = \hat{f} e_\xi$. This concludes the proof.

The Hennessy-Milner property was not mentioned explicit in the argument but is needed to guarantee that the coalgebras $\zeta_\xi$ and $\zeta_\gamma$ are defined. □

Notice that the construction used in the proof of Lemma 6.2.11 generalises the construction of final coalgebras of the previous section; before we used the fact that all satisfiable theories can be satisfied in a single coalgebra (cf. proof of Lemma 6.1.18). These observations leads us to the following definition.

**Definition 6.2.14.** An abstract coalgebraic language $(\mathcal{L}, \Phi)$ for $\mathcal{T}$-coalgebras is said *to have logical congruences* iff for each coalgebra $\xi$ the equivalence relation $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$, is a congruence of $\mathcal{T}$-coalgebras. The quotient of $\xi$ using $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ is called *the logical quotient of* $\xi$. We write $(Z_\xi, \zeta_\xi)$ for this coalgebra.

In [99] it was noticed that languages of predicate liftings that have logical congruences have the Hennessy-Milner property. We turn his observation into a general theorem for abstract coalgebraic languages.

**Theorem 6.2.15.** *If an abstract language $(\mathcal{L}, \Phi)$ for $\mathcal{T}$-coalgebras is adequate, the following are equivalent:*

1. *$\mathcal{L}$ has the Hennessy-Milner property.*
2. *$\mathcal{L}$ has logical congruences.*

*Proof.* The implication from (1) to (2) is Lemma 6.2.11.

Conversely suppose that $\mathcal{L}$ has logical congruences and let $\xi_1$ and $\xi_2$ be $\mathcal{T}$-coalgebras with logically equivalent states $x_1$ and $x_2$ respectively. Let $\xi_1 + \xi_2$ be the coproduct of $\xi_1$ and $\xi_2$ in $\mathsf{Coalg}(\mathcal{T})$ and let $\kappa_1(x_1), \kappa_2(x_2)$ be the image of $x_1$ and $x_2$ under the canonical coproduct embeddings. By adequacy of $\mathcal{L}$ the states $\kappa_1(x_1)$ and $\kappa_2(x_2)$ are logically equivalent. Since $\mathcal{L}$ has logical congruences, by assumption, we can make the quotient, in $\mathsf{Coalg}(\mathcal{T})$, using $\leftrightsquigarrow^{\xi_1+\xi_2}_{\mathcal{L}}$; the canonical quotient map $e$ will identify $\kappa_1(x_1)$ and $\kappa_2(x_2)$. Therefore, the states $x_1$ and $x_2$ are identified by the morphisms $e \circ \kappa_1$ and $e \circ \kappa_2$ and are thus behaviourally equivalent. This shows that $\mathcal{L}$ has the Hennessy-Milner property as required. □

The condition of adequacy is necessary as the following example shows.

**Example 6.2.16.** Let $\mathcal{T} = \mathsf{Id}$ be the identity functor on $\mathsf{Set}$. We define an abstract coalgebraic language for $\mathcal{T}$ by putting $\mathcal{L} := \{\top, \bot\}$ and by defining the theory map $\Phi_\xi$ for a $\mathcal{T}$-coalgebra $\xi : X \longrightarrow X$ as follows: $\Phi_\xi(x) = \{\top\}$ for all $x \in X$ if $\xi$ is the identity on $X$; and $\Phi_\xi(x) = \{\bot\}$ for all $x \in X$ if $\xi$ is not the identity on $X$. This language is clearly not adequate as any two given $\mathcal{T}$-coalgebra states are behaviourally equivalent. Also notice that this language has logical congruences because in any case all states in a single coalgebra are logically equivalent. However, it is easy to see that the coalgebra structure $\zeta$ on the set $Z = \{\{\top\}, \{\bot\}\}$ given by Equation (6.2) is well-defined and is equal to the identity on $Z$. But this is not the final $\mathcal{T}$-coalgebra hence the language does not have the Hennessy-Milner property.

### An application: A Concrete Characterization of Simple Coalgebras

As mentioned before, in [99] we have a non-trivial use of logical congruences to establish the Hennessy-Milner property for languages of predicate liftings. Theorem 6.2.15 tells us that in fact the two properties are equivalent. In this section, we illustrate the construction of logical congruences (cf. proof of Lemma 6.2.11) giving a concrete characterization of simple coalgebras. We first show how Lemma 6.1.18 also applies to the theory maps of logical quotients (Definition 6.2.14).

**Proposition 6.2.17.** *Let $(\mathcal{L}, \Phi)$ be an adequate language for $\mathcal{T}$-coalgebras that has logical congruences and let $\xi : X \longrightarrow \mathcal{T}(X)$ be a $\mathcal{T}$-coalgebra. The theory map $\Phi_{\zeta_\xi} : Z_\xi \longrightarrow \mathcal{P}(\mathcal{L})$ of the logical quotient of $\xi$ is equal to the inclusion.*

*Proof.* Let $e : X \longrightarrow Z_\xi$ be the quotient map. By adequacy of $\mathcal{L}$, we have $\Phi_\xi = \Phi_{\zeta_\xi} e$; and by definition of $e$ we have $e(x) = \Phi_\xi(x)$ for all $x \in X$. Therefore $i_{Z_\xi} e = \Phi_\xi = \Phi_{\zeta_\xi} e$. Since $e$ is onto, $\Phi_{\zeta_\xi}$ has to be the inclusion map.

Compare this argument with the one given for Lemma 6.1.18. □

Now we can use logical congruences in order to characterise simple coalgebras as logical quotients, i.e. quotients using the relations $\leftrightsquigarrow^\xi_\mathcal{L}$.

**Theorem 6.2.18.** *Let $(\mathcal{L}, \Phi)$ be an adequate language for $\mathcal{T}$-coalgebras that has logical congruences. Any logical quotient $(Z_\xi, \zeta_\xi)$ is simple and any simple $\mathcal{T}$-coalgebra is isomorphic to the logical quotient of some coalgebra $\xi$.*

*Proof.* By definition, if $\mathcal{L}$ has logical congruences, the logical quotient $(Z_\xi, \zeta_\xi)$ of any $\mathcal{T}$-coalgebra $\xi$ exists. By Proposition 6.2.17 the theory map is injective. Therefore Theorem 6.2.12 implies that each of these coalgebras is a simple coalgebra.

Now we show that every simple coalgebra is isomorphic to a logical quotient. Let $\xi : X \longrightarrow \mathcal{T}(X)$ be a simple coalgebra. Since $(X, \xi)$ is simple and $\leftrightsquigarrow^\xi_\mathcal{L}$ is a congruence, we conclude $\leftrightsquigarrow^\xi_\mathcal{L} = \Delta_X$. Therefore $(X, \xi) \cong (Z_\xi, \zeta_\xi)$. □

Using this characterization of simple coalgebras we can easily prove that truth-preserving functions with simple codomain must be coalgebra morphisms. This was a key result used by Goldblatt in [42] to construct final coalgebras.

**Corollary 6.2.19.** *Let* $(\mathcal{L}, \Phi)$ *be an expressive language for* $\mathcal{T}$*-coalgebras. Let* $f : X \longrightarrow Y$ *be a truth invariant function between coalgebras* $(X, \xi)$*, and* $(Y, \gamma)$ *i.e.* $\Phi_\xi(x) = \Phi_\gamma(f(x))$ *for all* $x \in X$*. If* $\gamma$ *is simple, then* $f : \xi \longrightarrow \gamma$ *is a coalgebra morphism.*

*Proof.* Let $f$ be a truth invariant morphism whose codomain is simple. The previous theorem implies that we can assume the codomain of $f$ to be the logical quotient $(Z_\gamma, \zeta_\gamma)$; say $f : X \longrightarrow Z_\gamma$. Since $f$ is truth invariant we have that $\ker(f) \subseteq \leftrightsquigarrow^{\xi}_{\mathcal{L}}$, this implies $Z_\xi \subseteq Z_\gamma$. Now using the fact that $\mathcal{L}$ is adequate and has the Hennessy-Milner property, one can prove either directly or using the construction of Lemma 6.2.11 that the inclusion map $i : Z_\xi \longrightarrow Z_\gamma$ is a morphism of coalgebras. This exhibits $f$ as the composition of the quotient map $e_\xi$ and the inclusion $i$; since both maps are coalgebra morphism we conclude that so is $f$. □

## 6.3 Logical Congruences & Weak Finality

In [42] a final coalgebra is constructed by using the argument in Lemma 6.1.18 and then quotienting by the largest congruence, which coincides with logical equivalence. This means that with the material that we have until here, we can already use logical congruences to construct final coalgebras. In despite of this, we will give a direct proof of this fact. One reason for this is that we can then provide a categorical argument which can be reused in several other contexts. For example, we will use this argument to prove Barr-Aczel-Mendler-Adámek Theorem on the existence of "large" final coalgebras for set functors.

The key insight from the previous section and this section is that the Hennessy-Milner property provides a solution set condition.

Our main categorical tool to produce final coalgebras is Freyd's Theorem on the existence of a final object [81]:

**Theorem 6.3.1.** *A cocomplete category* $\mathbb{C}$ *has a final object iff it has a set of objects* $S$ *which is weakly final, i.e. for every object* $c \in \mathbb{C}$ *there exists an object* $s \in S$ *and arrow* $c \longrightarrow s$*; a final object is obtained as a colimit of the diagram (the full subcategory) generated by* $S$*.*

The set $S$, mentioned in the previous theorem, is called a **solution set**. More explicitly, a set of objects $S$, in a category $\mathbb{C}$, satisfies the **solution set condition** iff for every object $c$ in $\mathbb{C}$ there exists an object $s \in S$ and arrow $c \longrightarrow s$. Freyd's Theorem is strongly related to the Adjoint Functor Theorem, see [81, 6] for details.

Recall that in the category Set every object only has a set of subobjects (subsets). In Proposition 6.2.8 we proved that if a language $\mathcal{L}$ is adequate and has logical congruences, each coalgebra $(X, \xi)$ can be mapped to its logical quotient $(Z_\xi, \zeta_\xi)$. Since each $Z_\xi$ is a subset of $\mathcal{P}(\mathcal{L})$, the coalgebras based on subsets of $\mathcal{P}(\mathcal{L})$ form a solution set, which by Freyd's theorem implies the existence of a final object. Therefore the following holds true.

**Proposition 6.3.2.** *Let $\mathcal{T}$ be a set functor. If a language $(\mathcal{L}, \Phi)$ for $\mathcal{T}$-coalgebras is adequate and has logical congruences then there exists a final $\mathcal{T}$-coalgebra. This coalgebra is obtained as a colimit of the diagram induced by the logical quotients $(Z_\xi, \zeta_\xi)$, i.e. the simple coalgebras.*

This proposition supplies us with another description of the final coalgebra. Moreover, following the path:

$$\text{Hennessy-Milner} \Rightarrow \text{logical congruences} \Rightarrow \text{final coalgebras},$$

we have another proof of Goldblatt's Theorem. This alternative proof is not as simple as the construction presented in Theorem 6.1.12 but illustrates the importance of adequacy and why the language should be a set. This can be restated by saying that the Hennessy-Milner property is a solution set condition to obtain final coalgebras; the solution set is given by the logical quotients $(Z_\xi, \zeta_\xi)$. However, if the language is not adequate we will not be able to combine all of those coalgebras into a single structure.

### An application: "large" final coalgebras

Very early in the development of coalgebra it was noticed that every set endofunctor has a final coalgebra [3]. However, this final coalgebra might have a proper class as carrier set. Michael Barr [14] noticed that looking at functors on the category of (possibly proper) classes could be replaced by considering sets up to some regular cardinal.

We will illustrate this construction using logical congruences and languages of predicate liftings. We believe our construction is more accessible to readers not familiar with category theory. The following definition is needed for the formulation of the theorem.

**Definition 6.3.3.** A cardinal number $\kappa$ is said to be *weakly inaccessible* if $\kappa$ is an uncountable, regular cardinal such that for all cardinals $\lambda$ we have $\lambda < \kappa$ implies $2^\lambda \leq \kappa$.

The existence of weakly inaccessible cardinals is independent of the system ZFC. The important fact to know is that the class of sets up to a weakly inaccessible cardinal is a model of ZFC, see e.g. [105] for details.

The following lemma, due to M. Barr, will be of key importance. It states that categories of coalgebras for $\kappa$-accessible functors (Definition 2.1.5) are presentable using coalgebras of cardinality less that $\kappa$.

**Lemma 6.3.4** ([14]). *Let $\mathcal{T}$ be a $\kappa$-accessible functor where $\kappa$ is weakly inaccessible. Every $\mathcal{T}$-coalgebra can be obtained as a colimit of coalgebras whose carrier sets have cardinality less than $\kappa$.*

Now we can prove the main theorem.

**Theorem 6.3.5.** *Let $\kappa$ be a weakly inaccessible cardinal and let $\mathcal{T} : \mathsf{Set} \longrightarrow \mathsf{Set}$ be a $\kappa$-accessible functor. Suppose, in addition, that for all sets $X$ we have $|X| < \kappa$ implies $|\mathcal{T}(X)| < \kappa$. Then $\mathsf{Coalg}(\mathcal{T})$ has a final coalgebra of cardinality no larger than $\kappa$.*

*Proof.* Let $\mathcal{L}^\kappa$ be the language of predicate liftings (Definition 3.1.5), for $\mathcal{T}$, of arity less than $\kappa$ and conjunctions bounded by $\kappa$. As we have mentioned before, this language is adequate and has the Hennessy-Milner property (Example 6.1.6). Theorem 6.1.12 implies that there exists a final $\mathcal{T}$-coalgebra $(Z, \zeta)$. Notice that $|\mathcal{L}^\kappa| = \kappa$, therefore $|Z| \leq 2^\kappa$. We will use logical congruences to show $|Z| \leq \kappa$.

Proposition 6.3.2 states that we can obtain a final coalgebra as a colimit of the coalgebras $(Z_\xi, \zeta_\xi)$. Since the functor $\mathcal{T}$ is $\kappa$-accessible, by the previous lemma, it is enough to consider coalgebras with a carrier set such that $|X| < \kappa$; because, in $\mathsf{Set}$, the quotient of a colimit is the colimit of the quotients. This implies that $|Z_\xi| < \kappa$ for each coalgebra $\xi$.

Since by assumption $|X| < \kappa$ implies $|\mathcal{T}(X)| < \kappa$ and $\kappa$ is weakly inaccessible, we can conclude that there are, up to isomorphisms, $\kappa$-many coalgebras with carrier set $|X| < \kappa$ because $|\mathcal{T}(X)^X| \leq \kappa^\kappa \leq \kappa$. Therefore there are, up to isomorphisms, at most $\kappa$-many coalgebras of the form $(Z_\xi, \zeta_\xi)$.

Gathering the previous two paragraphs we conclude that a final $\mathcal{T}$-coalgebra can be obtained as a colimit of at most $\kappa$-many sets of cardinality less that $\kappa$. Since $\kappa$ is weakly inaccessible we conclude $|Z| \leq \kappa$. This concludes the proof. □

## 6.4 Different faces of the Hennessy-Milner property

There is another face of the Hennessy-Milner property that we have not discussed yet. Namely, a direct translation of the statement in Definition 6.1.5 into the language of categories. This can be achieved using pullbacks. Recall that in the category $\mathsf{Set}$, the pullback of two functions $f_1 : X_1 \longrightarrow Y$ and $f_2 : X_2 \longrightarrow Y$ can be canonically characterised as the set $P = \{(x_1, x_2) \in X_1 \times X_2 \mid f_1(x_1) = f_2(x_2)\}$ together with respective projections.

We can now present the Henessy-Milner property as follows: If for $\mathcal{T}$-coalgebras $(X_1, \xi_1)$ and $(X_2, \xi_2)$ the diagram on the left is a pullback (in $\mathsf{Set}$), then there exists a coalgebra $(Y, \gamma)$ and morphisms $f_1 : \xi_1 \longrightarrow \gamma; f_2 : \xi_2 \longrightarrow \gamma$ such that the diagram on the right

$$
\begin{array}{ccc}
P & \xrightarrow{p_1} & X_1 \\
{\scriptstyle p_2}\downarrow & & \downarrow{\scriptstyle \Phi_{\xi_1}} \\
X_2 & \xrightarrow[\Phi_{\xi_2}]{} & \mathcal{P}(\mathcal{L})
\end{array}
\qquad\qquad
\begin{array}{ccc}
P & \xrightarrow{p_1} & X_1 \\
{\scriptstyle p_2}\downarrow & & \downarrow{\scriptstyle f_1} \\
X_2 & \xrightarrow[f_2]{} & Y
\end{array}
$$

commutes (in $\mathsf{Set}$).

Notice that the previous version of the Hennessy-Milner property is a bit different to the one in Definition 6.1.5. The divergence is that in Definition 6.1.5 for each pair of logically equivalence states $x_1 \in X_1$ and $x_2 \in X_2$, we find a coalgebra to identify them, whereas in the version given here there is a single coalgebra, namely $(Y, \gamma)$ which identifies all the logically equivalent states in $X_1$ and $X_2$. At first glance it may seem strange to use one single coalgebra to identify all behaviourally equivalent states. This is not an issue if the language is adequate because in such case we can always use the coproduct of $\xi_1$ and $\xi_2$ modulo behavioural equivalence.

Now we present a summary of our findings.

**Theorem 6.4.1.** *Let $(\mathcal{L}, \Phi)$ be an adequate language for $\mathcal{T}$-coalgebras and $Z_{\mathcal{L}}$ the set of satisfiable theories of $\mathcal{L}$. The following conditions are equivalent:*

1. *$\mathcal{L}$ has the Hennessy-Milner property.*

2. *The function $\zeta : Z_{\mathcal{L}} \longrightarrow \mathcal{T}(Z_{\mathcal{L}})$ from Equation (6.2), page 145, on the set of satisfiable $\mathcal{L}$-theories is well-defined and endows $Z_{\mathcal{L}}$ with the structure of a final coalgebra.*

3. *The set $Z_{\mathcal{L}}$ admits a coalgebraic structure, for $\mathcal{T}$, such that for each coalgebra $\xi$ the function $f_\xi : X \longrightarrow Z_{\mathcal{L}}$, i.e. the restriction of the codomain of the theory map $\Phi_\xi$, is a morphism of coalgebras.*

4. *For each coalgebra $\xi$ the relation $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ is a congruence of coalgebras.*

5. *For each coalgebra $\xi$ the set of satisfiable theories in $\xi$ admits a coalgebraic structure $\zeta_\xi : Z_\xi \longrightarrow \mathcal{T}(Z_\xi)$, such that the function $e_\xi : X \longrightarrow Z_\xi$ mapping a state $x \in \xi$ to its $\mathcal{L}$-theory is a morphism of coalgebras.*

6. *Let $(X_1, \xi_1)$ and $(X_2, \xi_2)$ be $\mathcal{T}$-coalgebras. If the diagram on the left is a pullback (in $\mathsf{Set}$), there exists a coalgebra $(Y, \gamma)$ and morphisms $f_1 : \xi_1 \longrightarrow \gamma$*

*and $f_2 : \xi_2 \to \gamma$ such that*

$$
\begin{array}{ccc}
P & \xrightarrow{p_1} & X_1 \\
{\scriptstyle p_2}\downarrow & & \downarrow{\scriptstyle \Phi_{\xi_1}} \\
X_2 & \xrightarrow[\Phi_{\xi_2}]{} & \mathcal{P}(\mathcal{L})
\end{array}
\qquad\qquad
\begin{array}{ccc}
P & \xrightarrow{p_1} & X_1 \\
{\scriptstyle p_2}\downarrow & & \downarrow{\scriptstyle f_1} \\
X_2 & \xrightarrow[f_2]{} & Y
\end{array}
$$

*the diagram on the right commutes (in* Set*).*

*Proof.* The equivalence between 1) and 2) is the content of Theorem 6.1.16.

The implication from 2) to 3) is obvious and the converse direction is a consequence of the definition of $\zeta$; any map $\zeta' : Z_{\mathcal{L}} \to \mathcal{T}(Z_{\mathcal{L}})$ that turns the theory maps $f_\xi$ into coalgebra morphisms must be equal to $\zeta$.

The equivalence between 1) and 4) follows from Theorem 6.2.15.

Item 4) is equivalent to item 5) by definition of congruence. More explicitly, since $\leftrightsquigarrow^{\xi}_{\mathcal{L}} = \ker(\Phi_\xi) = \ker(f_\xi)$ then $\leftrightsquigarrow^{\xi}_{\mathcal{L}}$ is a congruence iff the statement on item 5) holds.

Finally for the equivalence with 6), from the canonical characterisation of pullbacks in Set it is clear that 6) implies 1). We now show that 4) implies 6). For this first take the coproduct $\xi_1 + \xi_2$, and let $\kappa_1$ and $\kappa_2$ be the respective inclusions. By 4), we have that $\leftrightsquigarrow^{\xi_1+\xi_2}_{\mathcal{L}}$ is a congruence; hence, since the language is adequate, the coalgebra $(Z_{\xi_1+\xi_2}, \zeta_{\xi_1+\xi_2})$ together with $f_1 = e_{\xi_1+\xi_2}\kappa_1$ and $f_1 = e_{\xi_1+\xi_2}\kappa_2$ does the job. More explicitely, we will show $e_{\xi_1+\xi_2}\kappa_2 p_2 = e_{\xi_1+\xi_2}\kappa_1 p_1$. This equality will follow because the theory map of $(Z_{\xi_1+\xi_2}, \zeta_{\xi_1+\xi_2})$ is injective. Indeed.

$$
\begin{aligned}
\Phi_{\xi_1+\xi_2} e_{\xi_1+\xi_2}\kappa_2 p_2 &= \Phi_{\xi_2} p_2 && \text{(Adequacy)} \\
&= \Phi_{\xi_1} p_1 && \text{(Assumption on item 6))} \\
&= \Phi_{\xi_1+\xi_2} e_{\xi_1+\xi_2}\kappa_1 p_1 && \text{(Adequacy)}
\end{aligned}
$$

Since $\Phi_{\xi_1+\xi_2}$ is injective we conclude $e_{\xi_1+\xi_2}\kappa_2 p_2 = e_{\xi_1+\xi_2}\kappa_1 p_1$ as we wanted to show. □

## 6.5 Beyond sets

In this section we generalize our results for coalgebras on Set to coalgebras over other base categories. The first part of the section discusses how to generalise the notion of a language for coalgebras. We aim at a notion that works for functors $\mathcal{T} : \mathbb{C} \to \mathbb{C}$ on an arbitrary category $\mathbb{C}$. After that we focus on a special class of categories, those that are regularly algebraic over Set, and show that the results from the previous section generalise smoothly to these categories. This section requires more knowledge of category theory. We refer the reader to a standard text like [6] for details.

**Abstract coalgebraic languages abstractly**

When generalising the notion of an abstract coalgebraic language to categories, other than $\mathsf{Set}$, we face the problem that we do not know much about the structure of the given base category $\mathbb{C}$. In particular, unlike in the case $\mathbb{C} = \mathsf{Set}$, we do not know how to move freely from an object $\mathcal{L}$ representing the formulas to an object $\mathcal{P}(\mathcal{L})$ that represents the theories of a given language. Based on the work in the previous section we point out that the set of formulas itself is not very relevant. In our constructions we only used the set $\mathcal{P}(\mathcal{L})$. Even more explicit, the relevant object for our constructions was the set of satisfiable theories. This leads us to the following definition of an adequate object for $\mathcal{T}$-coalgebras; intuitively speaking the points of an adequate object are theories.

**Definition 6.5.1.** Let $\mathcal{T}$ be a functor $\mathcal{T} : \mathbb{C} \longrightarrow \mathbb{C}$. An object $\pounds$, in $\mathbb{C}$ is said to be an *adequate object* for $\mathcal{T}$-coalgebras if there exists a natural transformation $\Phi : U \longrightarrow \Delta_{\pounds}$, where $U : \mathsf{Coalg}(\mathcal{T}) \longrightarrow \mathbb{C}$ is the forgetful functor and $\Delta_{\pounds} : \mathsf{Coalg}(\mathcal{T}) \longrightarrow \mathbb{C}$ is the constant functor with value $\pounds$. We call the components of $\Phi$ *theory morphisms*.

Of course, every adequate abstract language for coalgebras induces and adequate object.

**Example 6.5.2.** In the case $\mathbb{C} = \mathsf{Set}$, every *adequate abstract coalgebraic language* (Definition 6.1.1) induces an adequate object. If $\mathcal{L}$ is an adequate abstract language for $\mathcal{T}$-coalgebras with theory maps $\{\Phi_\xi\}_{\xi \in \mathsf{Coalg}(\mathcal{T})}$, then $\mathcal{P}(\mathcal{L})$ together with $\{\Phi_\xi\}_{\xi \in \mathsf{Coalg}(\mathcal{T})}$ is an adequate object for $\mathcal{T}$-coalgebras.

We now argue that our definition of an adequate object for $\mathcal{T}$-coalgebras is a good generalisation of the notion of abstract coalgebraic language for $\mathcal{T}$-coalgebras. The whole point of the definition above is that given a coalgebraic logic $(L, \delta)$ we want the initial $L$-algebra to induce an adequate object (language). In Example 6.1.3 we showed how this is done in the case the we are considering a coalgebraic modal logic over a category of power set algebras. However, in case we consider a more general coalgebraic logic like in Definitions 3.2.21 or 3.2.22 it is not natural, neither desirable, to relate the initial $L$-algebra to its power set, see Remark 6.1.4. We would ratter use the set of "ultrafilters" given by the underlying adjunction.

**Example 6.5.3.** Let $P : \mathbb{C}^{op} \longrightarrow \mathcal{A}$ be a functor with a left adjoint $S$. Let $\mathcal{T} : \mathbb{C} \longrightarrow \mathbb{C}$ be a functor.

Every coalgebraic logic $(L, \delta)$ for $\mathcal{T}$-coalgebras (Definition 3.2.22) induces an adequate object for $\mathcal{T}$-coalgebras. More precisely, the adequate object is given by the image of initial $L$-algebra under $S$.

Let $(I, \iota)$ be the initial $L$-algebra and $[\![-]\!]_\xi : I \longrightarrow P(X)$ be the underlying morphism of initial map from $(I, \iota)$ to the complex algebra $(P(X), P(\xi) \circ \delta_X)$. The image of $I$ under $S$ together with the transposes of $[\![-]\!]_\xi$, relative to $S$; i.e

$\Phi_\xi : X \to S(I)$, is an adequate object. The fact that this is in fact adequate object, i.e. "formulas" are invariant under morphisms, uses properties of adjoints as it was shown in Example 6.1.8. More concretely,

- In the case of (Boolean) coalgebraic modal logics. An adequate object is given by the set of ultrafilters of the initial $L$-algebra. In other words, and adequate object is given by the set of maximally consistent $L$-theories.

- In the case that $P$ : $\mathsf{Stone} \to \mathsf{BA}$ and $S$ are the usual Stone duality adjunction. An adequate object is given by the set of ultra filters of boolean algebra.

### The Hennessy-Milner property abstractly

We will now investigate how we can construct final coalgebras only using the properties of the base category of coalgebras. In this section we discuss how to express the Henessy-Milner property in a general category. Theorem 6.4.1 shows that there are at least three ways to obtain a generalisation. Informally these are:

1) The set of theories of the language contains the carrier of a final coalgebra.

2) If states are identified by theory morphisms then they are identified by some coalgebra morphisms.

3) The image of a theory morphism carries a coalgebraic structure, equivalently, logical equivalence of states is a congruence.

The last item is related to $(RegEpi, Mono)$-factorisations and more generally to factorisation structures. The intuition behind $(RegEpi, Mono)$-structured categories is that the image of a morphism is an object in the category, examples of such factorisations are the isomorphism theorems of algebra.

**Definition 6.5.4.** Let $\mathcal{T} : \mathbb{C} \to \mathbb{C}$ be a functor, and let $\pounds$ be an adequate object for $\mathcal{T}$-coalgebras. We say that

1. $\pounds$ is *almost final* if $\pounds$ has a subobject $m : Z \to \pounds$ that can be uniquely lifted to a final $\mathcal{T}$-coalgebra $(Z, \zeta)$ such that $m = \Phi_\zeta$.

2. If the base category $\mathbb{C}$ has pullbacks, we say $\pounds$ has the *Hennessy-Milner property* if every pullback $(P, p_1, p_2)$ (in $\mathbb{C}$) of theory morphisms $\Phi_{\xi_1}$ and $\Phi_{\xi_2}$ can be factored (in $\mathbb{C}$) using a pair of coalgebra morphisms; see last item in Theorem 6.4.1.

3. Finally, if the base category $\mathbb{C}$ is $(RegEpi, Mono)$-structured, we say $\pounds$ has *logical congruences* if for each theory morphism $\Phi_\xi$ and each $(RegEpi, Mono)$-factorisation $(e, Z_\xi, m)$ of $\Phi_\xi$, there exists a coalgebraic structure $\zeta_\xi : Z_\xi \to \mathcal{T}(Z_\xi)$ such that $e$ is a coalgebra morphism from $\xi$ to $\zeta_\xi$.

As we proved in Theorem 6.4.1, all of the previous notions are equivalent if our base category is Set. It is important to recall that no requirements on the functor $\mathcal{T}$ were needed to prove this. A natural question is, how do they relate in other categories? The following proposition provides a partial answer

**Proposition 6.5.5.** *Let $\mathbb{C}$ be a cocomplete and* (RegEpi, Mono)*-structured category with pullbacks. Let $\mathcal{T}$ be an endofunctor on $\mathbb{C}$ and let $\pounds$ be an adequate object for $\mathcal{T}$-coalgebras which is well-powered.*

*For the the following properties*

1. *$\pounds$ has logical congruences.*

2. *$\pounds$ is almost final.*

3. *$\pounds$ has the Hennessy-Milner property.*

*we have* 1) $\Rightarrow$ 2) $\Rightarrow$ 3)*. Furthermore, if $\mathcal{T}$ preserves monomorphisms, the converse implications are true as well.*

*Proof.* Let $\mathbb{C}$ be a category that satisfies the conditions of the proposition. Let $\mathcal{T} : \mathbb{C} \longrightarrow \mathbb{C}$ be a functor and let $\pounds$ be an adequate object for $\mathcal{T}$-coalgebras.

1) $\Rightarrow$ 2). Suppose that $\pounds$ has logical congruences, we want to show that it is almost final. Let $\xi : X \longrightarrow \mathcal{T}(X)$ be an arbitrary $\mathcal{T}$-coalgebra and let $(e_\xi, Z_\xi, m_\xi)$ be a $(RegEpi, Mono)$-factorisation of $\Phi_\xi$; explicitly, $e_\xi : X \longrightarrow Z_\xi$ is a regular epi, $m_\xi : Z_\xi \longrightarrow \pounds$ is a monomorphism and $\Phi_\xi = m_\xi e_\xi$. By our assumption that $\pounds$ has logical congruences, for each $\mathcal{T}$-coalgebra $\xi$ there exists a morphism $\zeta_\xi : Z_\xi \longrightarrow \mathcal{T}(Z_\xi)$ such that $e_\xi : \xi \longrightarrow \zeta_\xi$ is a coalgebra morphism. Since $\pounds$ is well-powered, the collection of subobjects $Z_\xi$, and hence also the collection $S$ of $\mathcal{T}$-coalgebras based on those forms a set. In other words, the collection of the coalgebras $(Z_\xi, \zeta_\xi)$, as described above, is a solution set, in the sense of Theorem 6.3.1, Hence by Freyd's theorem, there exist a final $\mathcal{T}$-coalgebra $(Z, \zeta)$. Using the properties of of factorisation structures, we can show that $\Phi_\zeta : Z \longrightarrow \pounds$ is a monomorphism. The key point is that when we factor $\Phi_\zeta$ via $Z_\zeta$, this object is part of the diagram used to produce $Z$. This finishes that proof of the fact that $\pounds$ is almost final.

2) $\Rightarrow$ 3). Let us now assume that $\pounds$ is almost final. We want to show that $\pounds$ has the Hennessy-Milner property as described. To this aim consider two $\mathcal{T}$-coalgebras $(X_1, \xi_1)$ and $(X_2, \xi_2)$ and their respective theory maps $\Phi_{\xi_1}$ and $\Phi_{\xi_2}$. Furthermore, let $(P, p_1, p_2)$ be the pullback of $\Phi_{\xi_1}$ and $\Phi_{\xi_2}$; in particular we have $\Phi_{\xi_1} p_1 = \Phi_{\xi_2} p_2$. As $\pounds$ is almost final, there exists some subobject $m : Z \longrightarrow \pounds$ and some $\mathcal{T}$-coalgebra structure $\zeta : Z \longrightarrow \mathcal{T}(Z)$ such that $(Z, \zeta)$ is a final $\mathcal{T}$-coalgebra and such that $m = \Phi_\zeta$. Let $f_{\xi_1} : \xi_1 \longrightarrow \zeta$ and $f_{\xi_2} : \xi_2 \longrightarrow \zeta$ be the unique coalgebra

morphisms from $\xi_1$ and $\xi_2$ into $\zeta$. We have

$$\begin{aligned} m \circ f_{\xi_1} \circ p_1 &= \Phi_\zeta \circ f_{\xi_1} \circ p_1 && (m = \Phi_\zeta) \\ &= \Phi_{\xi_1} \circ p_1 && \text{(Adequacy)} \\ &= \Phi_{\xi_2} \circ p_2 && \text{(Def. pullback)} \\ &= \Phi_\zeta \circ f_{\xi_2} \circ p_2 && \text{(Adequacy)} \\ &= m \circ f_{\xi_2} \circ p_2. && (m = \Phi_\zeta) \end{aligned}$$

Since $m$ is a monomorphism this implies $f_{\xi_1} \circ p_1 = f_{\xi_2} \circ p_2$. More explicitly, the pullback $(P, p_1, p_2)$ "factors" through $f_{\xi_1}$ and $f_{\xi_2}$ as required. This demonstrates that $\mathcal{L}$ has the Hennessy-Milner property.

Consider now some endofunctor $\mathcal{T}$ that preserves monomorphisms and assume $\mathcal{L}$ is an adequate object for $\mathcal{T}$-coalgebras that has the Hennessy-Milner property. We want to prove that $\mathcal{L}$ has logical congruences.

Let $(X, \xi)$ be a $\mathcal{T}$-coalgebra and let $(e_\xi, Z_\xi, m_\xi)$ be a $(RegEpi, Mono)$-factorisation of $\Phi_\xi$. Recall that $\ker(\Phi_\xi)$ is a pullback of $\Phi_\xi$ with itself call the projections $p_1$ and $p_2$. The key idea is to realise that $e_\xi : X \longrightarrow Z_\xi$ coequalises the projections $p_1$ and $p_2$. To define the structural map on $Z_\xi$ we will use the universal property of $e_\xi$ by showing that the composite $X \xrightarrow{\xi} \mathcal{T}(X) \xrightarrow{\mathcal{T}(e_\xi)} \mathcal{T}(Z_\xi)$ coequalizes the projections $p_1$ and $p_2$. In other words, we will show

$$\mathcal{T}(e_\xi) \circ \xi \circ p_1 = \mathcal{T}(e_\xi) \circ \xi \circ p_2.$$

In order to show the equality above we use Hennessy-Milner property of $\mathcal{L}$. Since $\mathcal{L}$ has the Hennessy-Milner property there is a $\mathcal{T}$-coalgebra $(Y, \gamma)$ and a coalgebra morphisms $f_i : \xi \longrightarrow \gamma$, $(i = 1, 2)$, such that $f_1 p_1 = f_2 p_2$. Since $\mathcal{L}$ is adequate we have $m_\xi e_\xi = \Phi_\xi = \Phi_\gamma f_1 = \Phi_\gamma f_2$.

The computation now goes as follows:

$$\begin{aligned} \mathcal{T}(m_\xi)\mathcal{T}(e_\xi)\xi p_1 &= \mathcal{T}(\Phi_\gamma)\mathcal{T}(f_1)\xi p_1 && \text{(adequacy)} \\ &= \mathcal{T}(\Phi_\gamma)\gamma f_1 p_1 && (f_1 \text{ is a coalgebra morphism}) \\ &= \mathcal{T}(\Phi_\gamma)\gamma f_2 p_2 && \text{(HM-property)} \\ &\vdots \quad \text{(use argument backwards)} \\ &= \mathcal{T}(m_\xi)\mathcal{T}(e_\xi)\xi p_2. \end{aligned}$$

Notice that $\mathcal{T}(m_\xi)$ is a monomorphism because $m_\xi$ was a monomorphism and $\mathcal{T}$ preserves those by assumption. Therefore we obtain $\mathcal{T}(e_\xi)\xi q_1 = \mathcal{T}(e_\xi)\xi q_2$, as we wanted.

This implies that there exists a unique morphism $\zeta_\xi : Z_\xi \longrightarrow \mathcal{T}(Z_\xi)$ such that $\zeta_\xi \circ e_\xi = \mathcal{T}(e_\xi) \circ \xi$; in other words, such that $e_\xi$ is a coalgebra morphism from

$\xi$ to $\zeta_\xi$. Moreover adequacy of $\pounds$ implies that $m_\xi \circ e_\xi = \Phi_\xi = \Phi_{\zeta_\xi} \circ e_\xi$. Since $e_\xi$ is an epimorphism this implies that $m_\xi = \Phi_{\zeta_\xi}$. This shows that $\zeta_\xi$ is the logical quotient of $\xi$ as required. □

In particular, the previous proposition demonstrates that under mild assumptions on our base category we can establish the existence of a final coalgebra for a functor $\mathcal{T}$ by proving that there exists some adequate object for $\mathcal{T}$-coalgebras that has logical congruences.

We would now like to see whether the construction in Section 6.1 can be carried out in other base categories. In particular, we would like to proof the existence of a final coalgebra using only properties of the base category without any assumptions on the functor. Our main result here is to show that the construction in Section 6.1 still works on so-called regularly algebraic categories.

Roughly speaking, regularly algebraic categories, over Set correspond to classes of algebras for which we have free constructions, and $(RegEpi, mono)$-factorisations. In particular every variety of algebras is a regularly algebraic. Hence our results hold for coalgebras on the category BA of Boolean algebras and the category DL of distributive lattices. But also to categories like the category Stone of Stone spaces which regularly algebraic but not monadic, see [6].

**Definition 6.5.6.** A concrete category $(\mathcal{A}, U)$, over Set is regularly algebraic if the following conditions are satisfied.

1. $\mathcal{A}$ has coequalizers.
2. $U$ is a right adjoint.
3. $U$ is uniquely transportable.
4. $U$ preserves and reflects extremal epimorphisms.

The important property that we will need is that regularly algebraic categories can be embedded into categories of Eilenberg-Moore algebras.

**Proposition 6.5.7.** *Let $(\mathcal{A}, U)$ be a regularly algebraic category. Let $F$ be the left adjoint of $U$ and write $\mathbb{M}$ for the associated monad. Then the comparison functor $K : \mathcal{A} \longrightarrow \mathsf{Set}^{\mathbb{M}}$ is an embedding whose image is a regular epireflective subcategory of $\mathsf{Set}^{\mathbb{M}}$.*

We use previous proposition to prove the following theorem where no condition on the functor $\mathcal{T}$ are required.

**Theorem 6.5.8.** *Let $\mathcal{A}$ be a category that is regularly algebraic over* Set *with forgetful functor $V : \mathcal{A} \longrightarrow \mathsf{Set}$ and let $\mathcal{T} : \mathcal{A} \longrightarrow \mathcal{A}$ be a functor. The functor $\mathcal{T}$ has a final coalgebra iff there exists an adequate object $\pounds$ for $\mathcal{T}$-coalgebras that has the Hennessy-Milner property.*

*Proof.* The implication from left to right follows from the previous proposition.

For the implication from right to left, we will show that $\pounds$ has logical congruences. The core of the proof is to show this without any assumptions on the functor $\mathcal{T}$. The proof strategy is to lift the construction in $\mathsf{Set}$. To show that this works we use Proposition 6.5.7. More explicitly, from now on we work with the representation of $\mathcal{A}$ as a subcategory of Eilenberg-Moore algebras; we write $\mathbb{M} = (M, \eta, \mu)$ for the associated monad. A key gain of this representation is that now we can just talk about functions.

Before moving into the technicalities we clarify some issues concerning the new framework. First notice that carriers for coalgebras are functions $x : X \longrightarrow M(X)$ that satisfy the axioms of $\mathbb{M}$. To simplify our notation, we write $x_\mathcal{T} : X_\mathcal{T} \longrightarrow M(X_\mathcal{T})$ for the action of $\mathcal{T}$ on an $\mathbb{M}$-algebra $(X, x)$. In particular notice that a $\mathcal{T}$-coalgebra on $X$ is given by a function $\xi : X \longrightarrow X_\mathcal{T}$ that makes the usual diagram commute, i.e. $M(\xi) \circ x = x_\mathcal{T} \circ \xi$. In the same way, the theory map $\Phi_\xi : (X, x) \longrightarrow (\pounds, l)$ is given by a function $\Phi_\xi : X \longrightarrow \pounds$ such that $\Phi_\xi \circ l = M(\Phi_\xi) \circ x$.

We now show that $(\pounds, l)$ has logical congruences. Let $(X, \xi)$ be a $\mathcal{T}$-coalgebra. Consider the canonical factorisation $(e_\xi, Z_\xi, m_\xi)$ of the function $\Phi_\xi : X \longrightarrow \pounds$ via the set

$$Z_\xi = \{\Psi \mid (\exists x \in X)(\Psi = \Phi_\xi(x))\}.$$

Since $\mathcal{A}$ is regularly algebraic over $\mathsf{Set}$, this factorisation can be lifted to $\mathcal{A}$. This means that there exists an algebraic structure $z_\xi : Z_\xi \longrightarrow M(Z_\xi)$ such that $e_\xi$ and $m_\xi$ are morphism of algebras.

Since we are now working on a category of Eilenberg-Moore algebras pullbacks are computed as in $\mathsf{Set}$. In particular, since $(\pounds, l)$ has the Hennessy-Milner property, we can follow same argument of Theorem 6.1.12 to define a function $\zeta_\xi : Z_\xi \longrightarrow (Z_\xi)_\mathcal{T}$ such that the following diagram

$$\begin{array}{ccc} X & \xrightarrow{\;e_\xi\;} & Z_\xi \\ {\scriptstyle \xi}\downarrow & & \downarrow{\scriptstyle \zeta_\xi} \\ X_\mathcal{T} & \xrightarrow[\mathcal{T}(e_\xi)]{} & (Z_\xi)_\mathcal{T} \end{array} \tag{6.3}$$

commutes in $\mathsf{Set}$. One more time, we stress that here the use of the Eilenberg-Moore category is essential for this to work.

It is only left to proof that $\zeta_\xi$ is a morphism of $\mathbb{M}$-algebras. In order to show this, notice that since $e_\xi$ is an onto function it is a coequalizer and in particular an extremal epimorphism. Since forgetful functors from regular algebraic categories reflect coequalizers we have that $e_\xi$ is a coequalizer in $\mathcal{A}$. Say that $e_\xi$ coequalizes $\mathbb{M}$-morphism $p$ and $q$. We will show that $\zeta_\xi$ is a morphism of $\mathbb{M}$-algebras by showing that $\mathcal{T}(e_\xi)\xi$ coequalizes $p$ and $q$ in $\mathcal{A}$.

We now show $\mathcal{T}(e_\xi)\xi p = \mathcal{T}(e_\xi)\xi q$. Since $V$ is faithful this will follow once we show $V(\mathcal{T}(e_\xi)\xi p) = V(\mathcal{T}(e_\xi)\xi q)$. We prove this now.

$$\begin{aligned} V(\mathcal{T}(e_\xi)\xi p) &= V(\mathcal{T}(e_\xi))V(\xi)V(p) \\ &= \zeta_\xi V(e_\xi)V(p) && \text{(Diagram 6.3)} \\ &= \zeta_\xi V(e_\xi)V(q) && (e_\xi p = e_\xi q) \\ &= V(\mathcal{T}(e_\xi))V(\xi)V(q) && \text{(Diagram 6.3)} \\ &= V(\mathcal{T}(e_\xi)\xi q) \end{aligned}$$

Therefore $\mathcal{T}(e_\xi)\xi p = \mathcal{T}(e_\xi)\xi q$. From this, by the universal property of coequalizers, we conclude that there is a homomorphism of algebras $(Z_\xi, z_\xi) \longrightarrow ((Z_\xi)_\mathcal{T}, (z_\xi)_\mathcal{T})$ which makes the square above commute in $\mathbb{A}$. Using on more time the faithfulness of $V$ we conclude that this morphism must have $\zeta_\xi$ as underlying function.

Therefore $(\pounds, l)$ has logical congruences. Since every object in a regularly algebraic category is well-powered, we conclude that there exists a final $\mathcal{T}$-coalgebra which can be obtained as the colimit of the coalgebras $\zeta_\xi : (Z_\xi, z_\xi) \longrightarrow ((Z_\xi)_\mathcal{T}, (z_\xi)_\mathcal{T})$. This concludes the proof. □

## 6.6 Conclusions

In this chapter, we have studied three ways to express behavioural equivalence of coalgebra states:

1. using final coalgebras,
2. using coalgebraic languages that have the Hennessy-Milner property,
3. using coalgebraic languages that have logical congruences.

We provided a simple proof for the fact that these three different methods are equivalent when used to express behavioural equivalence between set coalgebras. An important point here is that no conditions on the signature functor are required. This shows that these results only depend on the structure of the state spaces i.e. the base category. As by-products of our study, we obtained a straightforward construction of canonical models of coalgebraic logics; a concrete characterisation of simple coalgebras as logical quotients, and a proof of the Aczel-Mendler-Adámek theorem on final coalgebras.

A very important insight from this chapter is to explicitely show that the well known Hennessy-Milner property from Modal logic corresponds to a solution set condition. Such solution set can then be used to obtain a final coalgebra.

A main topic for further research is that of abstract coalgebraic languages for functors on categories different from Set. Section 6.5 illustrates how abstract coalgebraic languages can be generalised to arbitrary categories. The main result of that section states that for any functor on a regularly algebraic category over Set an adequate object with the Hennessy-Milner property exists iff there exists a final coalgebra (Theorem 6.5.8). The use of logical congruences is a crucial ingredient in the proof. We hope to be able to extend the scope of Theorem 6.5.8 to categories that are topological over Set such as Pos or the category Meas of measurable spaces; the key problem here is that those categories are not uniquely transportable.

Another gain of using logical congruences is that they reveal how the Hennessy-Milner property, as we all know it, is related to the description of a particular factorisation structure (cf. Definition 6.5.4, Proposition 6.5.5). Here we only considered $(RegEpi, Mono)$-structured categories, but it is quite natural to generalise the results here to other factorisation structures. We believe that a study of these factorisations will lead to a coalgebraic understanding of non standard bisimulations.

Another question that could be interesting to investigate is the following: What conditions over the functor or category are needed to keep the language countable. In the case of the category Set, it is easy to see that the functor should at least be finitary and preserve finite sets.

Note that the theorems presented here can be restricted to covarieties of coalgebras; in fact to classes closed under coproducts, quotients and isomorphisms. It is not clear to us if having a axiomatization describing such covarieties will help to obtain a neater presentation of final coalgebras.

Another interesting issue would be to extract axiomatizations from our presentation of final coalgebras, a bit on the flavour used in coequational logic. More broadly, the relation between our construction and coequational logic should be investigate further. In general, the relation between coalgebraic modal logics and coequations still needs to be clarified.

# Chapter 7

# Dynamic Modalities and Coalgebraic Logics

Applications of modal logic are abundant in computer science and related disciplines. An important application of modal logics is in reasoning about programs. In [47] two main approaches to modal logics of programs are described. The first one is called an *exogenous* approach and is exemplified by Propositional Dynamic Logic (Section 1.2). The second approach is called *endogenous* and is exemplified by *temporal logic*. The key characteristic of exogenous logics is that programs are explicitly mentioned in the language. In an endogenous logic, programs are fixed and considered part of the structure over which the logic is interpreted. Exogenous logics give an outer, compositional perspective on programs. In contrast, endogenous logics take an inner perspective where programs are viewed as a single, monolithic structure.

In this chapter we investigate how to account for exogenous logics coalgebraically. Our primal example is PDL. A similar logic, but perhaps less known, is Game Logic (GL) [87], see page 33 here, which is a non-normal modal logic for reasoning about strategic ability in determined two-player games. In GL, modalities are indexed by games and their semantics is given in terms of monotone neighborhood functions. As said before, common to PDL and GL is that programs and games are explicit in the language. Moreover, programs can be combined to create more complex program. In general, programs in those logics are terms over an algebraic signature.

In this chapter, we concentrate on a coalgebraic framework for exogenous modal logics which encompasses (test free) PDL and GL. Our key idea is to consider extra structure on the functor $\mathcal{T}$. We exploit this structure to give the desired outer perspective on programs/games. For example, for sequential composition we assume $\mathcal{T}$ to be a monad (Section 7.3.1); this is in accordance with [83] where monads are used to describe notions of computation.

The main contributions of this chapter can be summarized as follows. We provide a general notion of dynamic structure which describes the algebraic structure on programs, and their interpretation as $\mathcal{T}$-coalgebras. Once this view is in place, *labelled modalities* arise in a natural way by a generic process of labelling (Definition 7.1.1). We then proceed to investigate the nature of PDL and GL axioms such as $[a;b]\varphi \iff [a][b]\varphi$ and $[a \cup b]\varphi \iff [a]\varphi \wedge [b]\varphi$ in our general setting. We show that such axioms hold if the underlying $\mathcal{T}$-modality preserves the extra structure on $\mathcal{T}$ in a manner that we make precise in Theorem 7.3.7 (sequential composition) and Theorem 7.3.18 (pointwise operations).

## 7.1 Labelling predicate liftings

In this section we study how, given a set of labels $L$ and $a \in L$, we can label modalities. We begin by discussing the relation between a modal formula $\Box\varphi$, *"$\varphi$ is necessary"*, and the PDL formula $[a]\varphi$, *"after every execution of $a$, $\varphi$ holds"*. The first modality is interpreted over Kripke frames, i.e. $\mathsf{Pow}$-coalgebras, and its labelling is interpreted over labelled transition systems, i.e. $\mathsf{Pow}^L$-coalgebras. As we discussed in Section 1.2, the modality $\Box$ leads to basic modal logic whereas the modalities $[a]$ lead to propositional dynamic logic. As we will later see, there is a similar relation between the non-normal modality $\Box$ for $\mathsf{Mon}$-coalgebras and the Game Logic modalities $[a]$ which are interpreted over $\mathsf{Mon}^L$ coalgebras. The next table presents the predicate liftings for $\Box$ in both situations. For a set $X$ and a subset $\varphi \subseteq X$ we have

| | $\Box_X(\varphi)$ | $[a]_X(\varphi)$ |
|---|---|---|
| $\mathsf{Pow}$ | $\{\psi \subseteq X \mid \psi \subseteq \varphi\}$ | $\{\delta \in \mathsf{Pow}(X)^L \mid \delta(a) \subseteq \varphi\}$ |
| $\mathsf{Mon}$ | $\{N \in \mathsf{Mon}(X) \mid \varphi \in N\}$ | $\{\delta \in \mathsf{Mon}(X)^L \mid \varphi \in \delta(a)\}$ |

As we can see in both cases the labelled modality $[a]$ can be described in terms of $\Box$ as follows: For $\varphi \subseteq X$ we have

$$[a]_X(\varphi) = \{\delta \mid \delta(a) \in \Box_X(\varphi)\}$$

What the last equation indicates is that the labelling of modalities can be seen as a generic process independent of the functor and any structures on the labels. The next definition makes this precise.

**Definition 7.1.1.** Let $L$ be a set of labels, $a \in L$, and $\lambda : \mathcal{P}^n \to \mathcal{P}\mathcal{T}$ a predicate lifting. We define the *$a$-labelling* of $\lambda$ as the natural transformation $\lambda^a : \mathcal{P}^n \to \mathcal{P}(\mathcal{T}^L)$ which maps a set $\varphi \subseteq X$ to

$$\lambda^a_X(\varphi) = \{\delta \in \mathcal{T}^L(X) \mid \delta(a) \in \lambda_X(\varphi)\}.$$

Given a coalgebra $\xi : X \longrightarrow \mathcal{T}(X)^L$, we write $\llbracket [a]_\lambda - \rrbracket_{(X,\xi)}$, instead of $\llbracket \Box_{\lambda^a} - \rrbracket_{(X,\xi)}$, for the associated predicate transformer i.e. the composite $\xi^{-1}\lambda^a$.

The next technical lemma relates the predicate transformers for $\lambda$ and $\lambda^a$.

**Lemma 7.1.2.** *Let $L$ be a set of labels. For every coalgebra $\xi : X \longrightarrow \mathcal{T}(X)^L$, every $\mathcal{T}$-predicate lifting $\lambda$, and every $a \in L$, we have*

$$\llbracket [a]_\lambda - \rrbracket_{(X,\xi)} = \llbracket \Box_\lambda - \rrbracket_{(X,\widehat{\xi}(a))},$$

*where $\widehat{\xi} : L \longrightarrow \mathcal{T}(X)^X$ is the exponential transpose of $\xi$.*

*Proof.* Recall that $\widehat{\xi}(a) : X \longrightarrow \mathcal{T}(X)$ maps an element $x \in X$ to $\xi(x)(a)$. Using this we will show that for every $\varphi \subseteq X$ we have $\llbracket [a]_\lambda - \rrbracket_{(X,\xi)}(\varphi) = \llbracket \Box_\lambda - \rrbracket_{(X,\widehat{\xi}(a))}(\varphi)$. For every $x \in X$ we have

$$\begin{aligned} x \in \llbracket [a]_\lambda - \rrbracket_{(X,\xi)}(\varphi) &\iff \xi(x) \in \lambda^a(\varphi) && (\text{Def. } \llbracket [a]_\lambda - \rrbracket_{(X,\xi)}) \\ &\iff \xi(x)(a) \in \lambda(\varphi) && (\text{Def. } \lambda^a) \\ &\iff \widehat{\xi}(a)(x) \in \lambda(\varphi) && (\text{Def. } \widehat{\xi}) \\ &\iff x \in \llbracket \Box_\lambda - \rrbracket_{(X,\widehat{\xi}(a))}(\varphi). && (\text{Def. } \llbracket \Box_\lambda - \rrbracket_{(X,\widehat{\xi}(a))}) \end{aligned}$$

This concludes the proof. □

**Remark 7.1.3.** During this chapter we ought to be more precise with our notation for the contravariant power set functor. We write $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ for the right adjoint and $\mathcal{P}^{op} : \mathsf{Set} \longrightarrow \mathsf{Set}^{op}$ for the left adjoint. In the convention of categories of power set algebras (Definition 3.2.12), we are considering $\mathcal{A} = \mathsf{Set}$. Recall that from Proposition 3.2.24 we know that the predicate functor $P : \mathsf{Set}\mathcal{A}$ has a left adjoint $S$; hence $\mathcal{P}$ is the functor $P$ and $\mathcal{P}^{op}$ is left adjoin $S$.

## 7.2 More on Monads

One of the most important operations between programs is sequential composition. It was shown in [83] that sequential composition of programs could be formalised using monads. This section is a technical intermezzo where to familiarise with the relevant preliminaries. More explicit, in this section we introduce the concepts of Kleisli category and morphism of monads. The reader familiar with these concept can skip this section and refer back to it when needed.

As we mentioned in Section 2.3, page 28, every monad comes from an adjunction. One way to obtain the adjunction is to use the category of Eilenberg-Moore algebras for the monad. This is the largest canonical solution. There is another canonical solution which is minimal; it is called the Kleisli category of the monad. The next definition introduces it.

**Definition 7.2.1** (Kleisli Categories)**.** Let $\mathbb{M} = (M, \eta, \mu)$ be a monad on a category $\mathbb{C}$. The Kleisli category of $\mathbb{M}$, written $\mathcal{K}l(\mathbb{M})$, has as objects the same objects of $\mathbb{C}$. A morphism $f : X \nrightarrow Y$ in $\mathcal{K}l(\mathbb{M})$ is a morphism $f : X \longrightarrow M(Y)$ in $\mathbb{C}$. We use $\bullet_M$ for morphism composition in $\mathcal{K}l(\mathbb{M})$. Given two morphisms $f : X \nrightarrow Y$ and $g : Y \nrightarrow Z$, in $\mathcal{K}l(\mathbb{M})$, we define $g \bullet_{\mathbb{M}} f : X \nrightarrow Z$ to be the following composite

$$\begin{array}{ccc} M(Y) & \xrightarrow{M(g)} & MM(Z) \\ {\scriptstyle f}\uparrow & & \downarrow{\scriptstyle \mu_Z} \\ X & \xrightarrow[g \bullet_{\mathbb{M}} f]{} & M(Z) \end{array}$$

in $\mathbb{C}$; the identity morphism of $\mathcal{K}l(\mathbb{M})$ are given by the unit of the monad, i.e. $id_X : X \nrightarrow X$ is given by $\eta_X : X \longrightarrow M(X)$.

If there is no risk of confusion we write $\bullet$ instead of $\bullet_{\mathbb{M}}$. The composition $\bullet_{\mathbb{M}}$ is also called the *Kleisli composition of the monad* of the monad $\mathbb{M}$.

Each Kleisli category is equipped with a functor $J_{\mathbb{M}} : \mathbb{C} \longrightarrow \mathcal{K}l(\mathbb{M})$, called the *Kleisli inclusion*, defined as follows: An object $X$ is mapped to $X$; a morphism $f : X \longrightarrow Y$ is mapped to $M(f) \circ \eta_X : X \longrightarrow MY$.

The next example illustrates how the composition of partial functions and relations can be presented using Kleisli categories.

**Example 7.2.2.** The following examples illustrate arrows and their composition in Kleisli categories:

1. Let $\mathbb{M} = 1 + (-)$. An arrow $f : X \nrightarrow Y$ in $\mathcal{K}l(1 + (-))$ is a partial function from $X$ to $Y$. Indeed, given a partial function from $X$ to $Y$ we make it into a total function $X \longrightarrow 1 + Y$ by mapping all the elements for which the function is not defined into $* \in 1$. Conversely given an arrow $f : X \nrightarrow Y$ we obtain a partial function by considering all the the elements that are not mapped to $* \in 1$. It is straightforward to see that the composition of partial functions corresponds to the Kleisli composition of the monad $1 + (-)$. In other words, the category $\mathcal{K}l(1 + (-))$ is the category of sets and partial functions.

2. As mentioned in Section 1.2 binary relations from $X$ to $Y$ correspond to arrows $X \longrightarrow \mathsf{Pow}(Y)$. More explicitly given $\xi : X \longrightarrow \mathsf{Pow}(Y)$ we define a binary relation $R_\xi$ by $xR_\xi y$ iff $y \in \xi(y)$.

   We now show that the Kleisli composition of the monad $\mathsf{Pow}$ corresponds to the usual composition of the associated relations. Given $\xi : X \nrightarrow Y$ and

$\gamma : Y \nrightarrow Z$, and an element $x \in X$ we have

$$\begin{aligned}
\gamma \bullet \xi(x) &= \mu_Z \circ \mathsf{Pow}(\gamma) \circ \xi(x) && \text{(Def. } \bullet) \\
&= \mu_Z \circ \mathsf{Pow}(\gamma)\,[\{y \mid xR_\xi y\}] && \text{(Def. } R_\xi) \\
&= \bigcup \{\gamma(y) \mid xR_\xi y\} && \text{(Def. } \mathsf{Pow}(\gamma), \mu_Z) \\
&= \{z \mid (\exists y)(z \in \gamma(y) \wedge xR_\xi y\} && \text{(Def. } \bigcup) \\
&= \{z \mid (\exists y)(yR_\gamma z \wedge xR_\xi y\}. && \text{(Def. } R_\gamma)
\end{aligned}$$

All this can by summarised by saying that $\mathcal{K}l(\mathsf{Pow})$ is the category of sets and relations.

3. Recall the distribution monad $(\mathcal{D}, \eta, \mu)$ (Example 2.3.16); in particular recall that the multiplication $\mu : \mathcal{D}\mathcal{D} \longrightarrow \mathcal{D}$ is given by multiplying the probabilities along the way i.e. given $D \in \mathcal{D}\mathcal{D}(X)$ and $x \in X$, we have $\mu_X(D)(x) = \sum_{d \in \mathcal{D}(X)} D(d).d(x)$. Recall (Example 2.2.1) that a coalgebra $\xi : X \longrightarrow \mathcal{D}(X)$ is a transition system where transitions are labelled by probabilities. The Kleisli composition of $\mathcal{D}$ literally compose the probabilities along the way.

The next remark describes the right adjoint to the Kleisli inclusion functor

**Remark 7.2.3.** The Kleisli inclusion $J_{\mathbb{M}} : \mathbb{C} \longrightarrow \mathcal{K}l(\mathbb{M})$ together with the functor $K_{\mathbb{M}} : \mathcal{K}l(\mathbb{M}) \longrightarrow \mathbb{C}$ which maps $f : X \nrightarrow Y$, in $\mathcal{K}l(M)$, to $M(X) \xrightarrow{\mu_Y f} M(Y)$, in $\mathbb{C}$, form an adjoint pair which recovers the monad $\mathbb{M}$, see e.g [8] for details.

The next lemma shows that the Kleisli composition for the neighbourhood monad $\mathcal{P}\mathcal{P}^{op}$ (cf. Example 2.3.16) corresponds to function composition of predicate transformers.

**Lemma 7.2.4.** *Let $\bullet$ denote Kleisli-composition for the monad $(\mathcal{P}\mathcal{P}^{op}, \eta, \mu)$. For all sets $X$, all $U \in \mathcal{P}(X)$ and all $E, F \colon X \longrightarrow \mathcal{P}\mathcal{P}^{op}(X)$, we have*

$$U \in (F \bullet E)(x) \text{ iff } x \in \widehat{E}(\widehat{F}(U)).$$

*where $\widehat{E}$ and $\widehat{F}$ are the $\mathcal{P}$-transposes of $E$ and $F$.*

*Proof.* Notice that the transpose of a map $f : X \longrightarrow \mathcal{P}\mathcal{P}^{op}(X)$ is given by the function $\widehat{f} : \mathcal{P}^{op}(X) \longrightarrow \mathcal{P}^{op}(X)$ which maps $U \in \mathcal{P}^{op}(X)$ to the set

$$\widehat{f}(U) = \{y \in X \mid U \in f(y)\}$$

With this in mind we can now calculate.

Let $X$ be a set, $U \in \mathcal{P}(X)$ and $E, F\colon X \to \mathcal{P}\mathcal{P}^{op}(X)$:

$$
\begin{aligned}
U \in &(F \bullet E)(x) \iff \\
&\iff U \in \mu_X \circ \mathcal{P}\mathcal{P}^{op}(F) \circ E(x) && \text{(Def. } \bullet\text{)}\\
&\iff \{H \in \mathcal{P}\mathcal{P}^{op}(X) \mid U \in H\} \in \mathcal{P}\mathcal{P}^{op}(F) \circ E(x) && \text{(Def. } \mu_X \text{ Ex. 2.3.16)}\\
&\iff F^{-1}(\{H \in \mathcal{P}\mathcal{P}^{op}(X) \mid U \in H\}) \in E(x) && \text{(Def. } \mathcal{P}\mathcal{P}^{op}(F)\text{)}\\
&\iff \{y \in X \mid U \in F(y)\} \in E(x) && \text{(Def. } F^{-1}\text{)}\\
&\iff \widehat{F}(U) \in E(x) && \text{(Def. } \widehat{F}\text{)}\\
&\iff x \in \widehat{E}(\widehat{F}(U)), && \text{(Def. } \widehat{E}\text{)}
\end{aligned}
$$

as we wanted to show. □

The next remark describes a more general situation from which the previous lemma is a particular case.

**Remark 7.2.5.** The previous lemma is an instance of a more general phenomenon. For any adjoint situation $(F, U, \varphi, \eta, \varepsilon)$, and any pair of arrows $f : F(A) \to F(B)$ and $g : F(B) \to F(C)$ the following holds

$$\varphi(g \circ f) = \varphi(g) \bullet \varphi(f),$$

where $\bullet$ is the Kleisli composition associated with $UF$.

We will need the notion of monad morphism to tame sequential composition.

**Definition 7.2.6.** Let $\mathbb{M} = (M, \eta, \mu)$ and $\mathbb{M}' = (M', \eta', \mu')$ be monads on a category $\mathbb{C}$. A *monad morphism* from $\mathbb{M}$ to $\mathbb{M}'$ is a natural transformation $\rho : M \to M'$ such that the diagrams

$$
\begin{array}{ccc}
 & id_{\mathbb{C}} & \\
{\scriptstyle\eta}\swarrow & & \searrow{\scriptstyle\eta'} \\
M & \xrightarrow[\rho]{} & M'
\end{array}
\qquad
\begin{array}{ccccc}
MM & \xrightarrow{M(\rho)} & MM' & \xrightarrow{\rho_{M'}} & M'M' \\
{\scriptstyle\mu}\downarrow & & & & \downarrow{\scriptstyle\mu'} \\
M & & \xrightarrow[\rho]{} & & M'
\end{array}
$$

commute. In case $\rho$ is a monad morphism we write $\rho : \mathbb{M} \to \mathbb{M}'$

The following remark provides another choice for the upper arrow in the rectangle above.

**Remark 7.2.7.** Since $\rho : M \to M'$ is natural the following diagram

$$
\begin{array}{ccc}
MM & \xrightarrow{\rho_M} & M'M \\
{\scriptstyle M(\rho)}\downarrow & & \downarrow{\scriptstyle M'(\rho)} \\
MM' & \xrightarrow[\rho_{M'}]{} & M'M'
\end{array}
$$

commutes. Hence for upper edge of the rectangle in the previous definition we could have as have taken $(M'\rho)(\rho M)$.

The next lemma presents a connection between monads morphisms and functors between Kleisli categories. Recall that $J_{\mathbb{M}}$ denotes the Kleisli inclusion functor (Definition 7.2.1).

**Lemma 7.2.8.** *Given monads $\mathbb{M}$ and $\mathbb{M}'$ on a category $\mathbb{C}$, there is a bijective correspondence between monads morphisms $\rho : \mathbb{M} \longrightarrow \mathbb{M}'$ and functors $F : \mathcal{K}l(\mathbb{M}) \longrightarrow \mathcal{K}l(\mathbb{M}')$ for which the triangle*

$$\begin{array}{ccc} & \mathbb{C} & \\ {}^{J_{\mathbb{M}}}\swarrow & & \searrow^{J_{\mathbb{M}'}} \\ \mathcal{K}l(\mathbb{M}) & \xrightarrow[F]{} & \mathcal{K}l(\mathbb{M}') \end{array}$$

*commutes.*

*Proof.* We just mention how to define the functor from a morphism and viceversa; for details we refer the reader to the literature, e.g. [8].

Given a monad morphism $\rho : \mathbb{M} \longrightarrow \mathbb{M}'$ the assignation $\rho\circ - : \mathcal{K}l(\mathbb{M}) \longrightarrow \mathcal{K}l(\mathbb{M}')$ is the mentioned functor. More explicitly $f : X \twoheadrightarrow Y$, in $\mathcal{K}l(\mathbb{M})$, is mapped to $\rho_Y \circ f : X \twoheadrightarrow Y$ in $\mathcal{K}l(\mathbb{M}')$, i.e. the following composite $X \xrightarrow{\;f\;} M(Y) \xrightarrow{\;\rho_Y\;} M'(Y)$.

Given a functor $F : \mathcal{K}l(\mathbb{M}) \longrightarrow \mathcal{K}l(\mathbb{M}')$ as above, notice that since $F$ commutes with the Kleisli inclusions we have $F(X) = X$ for every object $X$. Hence, we define $\rho_X : M(X) \longrightarrow M'(X)$ to be the image of $id_{M(X)} : M(X) \twoheadrightarrow X$ under the functor $F$. □

## 7.3 Algebraic structures over labels

As we saw in Section 7.1 the process of labelling modalities can be described independently of any extra structure on the labels. However, one of the key features of exogenous logics for programming languages is that they can actually account for combinations between the programs. Examples of this are axioms like

$$[a;b]\varphi \iff [a][b]\varphi \text{ or } [a\cup b]\varphi \iff [a]\varphi \vee [b]\varphi.$$

In this section we discuss how to account for such extra algebraic structure within labels. More precisely, we give an algebraic-coalgebraic framework for exogenous coalgebraic modal logic, and we show that (test free) PDL and GL are particular instances of it.

Formally speaking, the interpretation of labels is a map $L \longrightarrow \mathcal{T}(X)^X$ which describes how actions change the global system state. The algebraic structure on $L$ describes how one can construct complex actions from simpler ones. We would like this map to be an algebra homomorphism; should this be the case, then we obtain a compositional semantics of actions.

For this to work, it is, of course, necessary that $\mathcal{T}(X)^X$ carries an algebraic structure of the same type as $L$. It is not entirely clear to us how to define an algebraic structure on $\mathcal{T}(X)^X$ in general. In this chapter we discuss a general framework for sequential composition (Section 7.3.1) and operations which are obtained by pointwise extensions (Section 7.3.2).

The key idea behind the framework here is illustrated by the following dual situation. By considering the exponential adjoint of and interpretation of labels we obtain $X \to \mathcal{T}(X)^L$, this is a behavioural description of the system in the form of a $\mathcal{T}^L$-coalgebra. These two (equivalent) views of a dynamic system form the basis of our modelling. In short,

| $L \to \mathcal{T}(X)^X$ | $X \to \mathcal{T}(X)^L$ |
|---|---|
| (algebraic view: structure, compositionality) | (coalgebraic view: behaviour, modalities) |

A similar observation was made in [57] in the context of Java semantics. The next definition summarises and makes the ideas above precise.

**Definition 7.3.1** (Dynamic structure)**.** Let $\Sigma$ be an algebraic signature (Section 2.3), let $L_0$ be a set (of atomic labels), and let $L = T_\Sigma(L_0)$ be the term algebra over $L_0$. Moreover, let $\mathcal{T}$ be a $\mathsf{Set}$-functor such that for each set $X$ the set $\mathcal{T}(X)^X$ carries a $\Sigma$-algebra structure $\theta\colon \Sigma(\mathcal{T}(X)^X) \to \mathcal{T}(X)^X$.

A coalgebra $\xi : X \to \mathcal{T}(X)^L$ is said to be *standard* if its transpose $\widehat{\xi} : L \to \mathcal{T}(X)^X$ is a morphism of $\Sigma$-algebras; in other words if the following diagram

$$\begin{array}{ccc} \Sigma(L) & \xrightarrow{\Sigma(\widehat{\xi})} & \Sigma(\mathcal{T}(X)^X) \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\theta} \\ L & \xrightarrow[\widehat{\xi}]{} & \mathcal{T}(X)^X \end{array}$$

commutes, where $\alpha$ is the structural map of the free algebra. In this case, we call the map $\widehat{\xi}$ a *$\mathcal{T}$-dynamic $\Sigma$-structure (on $X$)*.

If $\Sigma$ is irrelevant or clear from the context, we will sometimes leave it out, and simply speak of $\mathcal{T}$-dynamic structures; if also $\mathcal{T}$ is clear, we will simply talk about dynamic structures.

The next remark discusses how the previous definition can be generalized to monads.

**Remark 7.3.2.** Notice that in the previous definition we only need free algebras. Hence it can be generalized to monads. More precisely, given a monad $\mathbb{M}$ let $(L, \alpha)$ be the free $\mathbb{M}$-algebra over $L_0$. Assume for each set $X$ the set $\mathcal{T}(X)^X$ carries an

$\mathbb{M}$-algebra structure $\theta : M(\mathcal{T}(X)^X) \to \mathcal{T}(X)^X$. Then we say that a coalgebra $\xi : X \to \mathcal{T}(X)^L$ is *standard* if the following diagram

$$\begin{array}{ccc} M(L) & \xrightarrow{M(\widehat{\xi})} & M(\mathcal{T}(X)^X) \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\theta} \\ L & \xrightarrow[\widehat{\xi}]{} & \mathcal{T}(X)^X \end{array}$$

commutes. As in the previous definition The map $\widehat{\xi}$ is called a $\mathcal{T}$-dynamic $\mathbb{M}$-structure on $X$. An advantage of this approach via monads is that we can also account for equations between programs.

In summary, a dynamic structure $\widehat{\xi} : L \to \mathcal{T}(X)^X$ provides a compositional $\mathcal{T}$-coalgebra semantics of complex labels. The following examples show that the semantics of PDL and GL are dynamic structures.

**Example 7.3.3.** Let $\Sigma_{\mathrm{PDL}} = \{\cup, ;, ^*\}$ be the signature of test-free PDL and $\Pi$ the set of test-free PDL programs over a set $\Pi_0$ of atomic program labels. Using the isomorphism between functions $X \to \mathsf{Pow}(X)$ and relations on $X$, we see that a PDL model $(X, \{R_a | \, a \in \Pi\})$ is a $\mathsf{Pow}$-dynamic $\Sigma_{\mathrm{PDL}}$-structure $\xi \colon \Pi \to \mathsf{Pow}(X)^X$. Indeed, for any set $X$, the set $\mathsf{Pow}(X)^X$ carries a $\Sigma_{\mathrm{PDL}}$-algebra defined for all $\xi, \zeta \in \mathsf{Pow}(X)^X$ as follows. For all $x \in X$,

$$(\xi \cup \zeta)(x) := \xi(x) \cup \zeta(x), \quad (\xi;\zeta)(x) := (\zeta \bullet \xi)(x), \quad \xi^*(x) := \bigcup_{n<\omega} \xi^n(x),$$

where $\xi^0 := \eta_X$ and $\xi^{n+1} = (\xi^n;\xi)$; and $\bullet$ denotes the Kleisli composition of the monad $\mathsf{Pow}$. It can easily be confirmed that these operations on $\mathsf{Pow}(X)^X$ correspond to the standard PDL operations (cf. [47])

**Example 7.3.4.** Let $\Sigma_{\mathrm{GL}} = \{\cup, ;, ^*, ^d\}$ be the signature of test-free Game Logic and $\Gamma$ the set of test-free GL games over a set $\Gamma_0$ of atomic game labels. A GL model $(X, \{E_\gamma | \, \gamma \in \Gamma\})$ is a $\mathsf{Mon}$-dynamic $\Sigma_{\mathrm{GL}}$-structure. For any set $X$, the set $\mathsf{Mon}(X)^X$ carries a $\Sigma_{\mathrm{GL}}$-algebra defined for all $E, F \in \mathsf{Mon}(X)^X$ as follows. For all $x \in X$,

$$\begin{array}{lclllcl} (E \cup F)(x) & := & E(x) \cup F(x), & (E;F)(x) & := & (F \bullet E)(x), \\ E^*(x) & := & \bigcup_\kappa E^\kappa(x) & E^d(x) & := & \{U \subseteq X \mid X \setminus U \notin E(x)\}, \end{array}$$

where $\bullet$ corresponds to the Kleisli composition of the monad $\mathsf{Mon}$, and $E^\kappa$ is defined by transfinite induction over all ordinals; more explicitly, $E^0 := \eta_X$, for successors ordinals $E^{\kappa+1} = E^\kappa; E$, and for limit ordinals $E^\kappa = \bigcup_{\alpha<\kappa} E^\alpha(x)$.

The above definition corresponds to the semantics of complex games in Equation (2.2), page 34. We illustrate the case for the choice operator. Given programs $\gamma$ and $\chi$ we have:

$$\begin{array}{rcl} x \in \widehat{E}_{\gamma\cup\chi}(U) & \iff & x \in \widehat{E}_{\gamma}(U) \cup \widehat{E}_{\chi}(U) \\ & \iff & U \in E_{\gamma}(x) \cup E_{\chi}(x) \\ & \iff & U \in (E_{\gamma} \cup E_{\chi})(x). \end{array}$$

The identity $\widehat{E}_{\gamma;\chi} = \widehat{E_{\gamma}; E_{\chi}}$ for the sequential composition follows from Lemma 7.2.4. The identity for iteration holds by the Knaster-Tarski fixpoint theorem since $\mathsf{Mon}(X)$ is join-complete with top element $\mathcal{P}\mathcal{P}^{op}(X)$. The case for dual follows by straight forward calculations.

### 7.3.1 Sequential composition

The Examples 7.3.3 and 7.3.4 show that the sequential composition of relations and that of the neighbourhood functions can be described using the Kleisli composition of the monads. Assuming that $\mathcal{T}$ is a monad, the Kleisli composition for $\mathcal{T}$ yields a sequential composition operation on $\mathcal{T}(X)^X$, which generalises those of PDL and GL. We will now investigate the axiom

$$[a;b]\varphi \iff [a][b]\varphi \tag{7.1}$$

in the general setting of $\mathcal{T}$-dynamic structures where $\mathcal{T}$ is an arbitrary monad on $\mathsf{Set}$ (we leave unit and multiplication implicit). Roughly speaking, our main result in this section (Theorem 7.3.7) states that the axiom above is valid if the predicate lifting corresponds to a monad morphism.

From now on we assume $\Sigma$ is a signature containing the binary operation symbol ;, $L = T_{\Sigma}(L_0)$ the set of $\Sigma$-terms over a set $L_0$ of atomic labels, and most importantly that in any $\mathcal{T}$-dynamic $\Sigma$-structure $\widehat{\xi} : L \to \mathcal{T}(X)^X$, the operation ; is interpreted on $\mathcal{T}(X)^X$ as Kleisli-composition for $\mathcal{T}$.

We use predicate transformers to describe Axiom 7.1 semantically and formalise its validity for any predicate lifting.

**Definition 7.3.5.** A predicate lifting $\lambda$ for $\mathcal{T}$ is said to *support sequential composition* if for every standard $\mathcal{T}$-dynamic $\Sigma$-structure $\widehat{\xi} : L \to \mathcal{T}(X)^X$ and every $a, b \in L$,

$$[\![ [a;b]_{\lambda} - ]\!]_{\xi} = [\![ [a]_{\lambda} - ]\!]_{\xi} \circ [\![ [b]_{\lambda} - ]\!]_{\xi}.$$

where $[\![ [\alpha]_{\lambda} - ]\!]_{\xi} : \mathcal{P}(X) \to \mathcal{P}(X)$, $\alpha \in \{a;b, a, b\}$, is the predicate transformer associated with $\lambda^{\alpha}$.

We note that this definition is not vacuous, since not all predicate liftings $\lambda$ will make the axiom above valid.

**Example 7.3.6.** Consider the predicate lifting $\lambda\colon \mathcal{P} \longrightarrow \mathcal{P}\mathsf{Pow}$ defined for all sets $X$ and $U \subseteq X$ by $\lambda_X(U) = \{\emptyset\}$. For a label $a \in L$, the $a$-labelling of $\lambda$ is then given by $\lambda^a_X(U) = \{\delta \in \mathsf{Pow}(X)^L \mid \delta(a) = \emptyset\}$. In other words, for any $L$-labelled transition system $\xi\colon X \longrightarrow \mathsf{Pow}(X)^L$, $x \in [\![[a]_\lambda -]\!]^\xi(U)$ iff there is no $a$-transition from $x$, note that $U$ is irrelevant.

The axiom for sequential composition for $\lambda$ would then say: *for any $x \in X$, $x$ has no $a;b$-successor iff $x$ has no $a$-successor.* This property clearly is not valid, for example, suppose $\xi(x)(a) = \{x'\}$ and $\xi(x')(b) = \emptyset$.

We now arrive to the main result of this section. We will show that a predicate lifting $\lambda : \mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ supports sequential composition if it respects the monad structure given in terms of Kleisli categories. The next theorem makes this precise.

**Theorem 7.3.7.** *A predicate lifting $\lambda\colon \mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ supports sequential composition if its transpose $\widehat{\lambda} : \mathcal{T} \longrightarrow \mathcal{P}\mathcal{P}^{op}$ is a monad morphism.*

*Proof.* We want to show that for every standard coalgebra $\xi : X \longrightarrow \mathcal{T}(X)^L$ and every $a, b \in L$, we have

$$[\![[a;b]_\lambda -]\!]_\xi = [\![[a]_\lambda -]\!]_\xi \circ [\![[b]_\lambda -]\!]_\xi.$$

Since we assume $\widehat{\lambda} : \mathcal{T} \longrightarrow \mathcal{P}\mathcal{P}^{op}$ to be a monad morphism by Lemma 7.2.8 we have a functor $\widehat{\lambda} \circ - : \mathcal{K}l(\mathcal{T}) \longrightarrow \mathcal{K}l(\mathcal{P}\mathcal{P}^{op})$. In particular, this implies

$$\widehat{\lambda} \circ (f \bullet_{\mathcal{T}} g) = (\widehat{\lambda} \circ f) \bullet_{\mathcal{P}\mathcal{P}^{op}} (\widehat{\lambda} \circ g). \tag{7.2}$$

From the definition of $\widehat{\lambda}$ and the fact that $[\![\Box_\lambda -]\!]_\xi = \xi^{-1} \circ \lambda = \mathcal{P}(\xi) \circ \lambda$ we have: for all $x \in X$ and $U \subseteq X$,

$$x \in [\![\Box_\lambda -]\!]_\xi(U) \iff \xi(x) \in \lambda_X(U) \iff U \in \widehat{\lambda}(\xi(x)). \tag{7.3}$$

The theorem now follows from the following equivalences:

$$\begin{aligned}
x \in [\![[a;b]_\lambda -]\!]_\xi(U) &\iff x \in [\![\Box_\lambda -]\!]_{\widehat{\xi}(a;b)} && \text{(Lemma 7.1.2)}\\
&\iff x \in [\![\Box_\lambda -]\!]_{\widehat{\xi}(a) \bullet_{\mathcal{T}} \widehat{\xi}(b)}(U) && \text{($\xi$ is standard)}\\
&\iff U \in (\widehat{\lambda} \circ (\widehat{\xi}(a) \bullet_{\mathcal{T}} \widehat{\xi}(b)))(x) && \text{(Eq. 7.3)}\\
&\iff U \in ((\widehat{\lambda} \circ \widehat{\xi}(a)) \bullet_{\mathcal{P}\mathcal{P}^{op}} (\widehat{\lambda} \circ \widehat{\xi}(b)))(x) && \text{(Eq. 7.2)}\\
&\iff x \in [\![\Box_\lambda -]\!]_{\widehat{\xi}(a)} \circ [\![\Box_\lambda -]\!]_{\widehat{\xi}(b)}(U) && \text{(Lemma 7.2.4)}\\
&\iff x \in [\![[a]_\lambda -]\!]_\xi \circ [\![[b]_\lambda -]\!]_\xi(U) && \text{(Lemma 7.1.2)}
\end{aligned}$$

This concludes the proof. □

The next example illustrates the case of PDL and GL.

**Example 7.3.8.** The following predicate liftings support sequential composition.

1. The universal modality $\Box$ for $\mathsf{Pow}$ supports sequential composition. The transpose $\widehat{\Box} : \mathsf{Pow} \to \mathcal{P}\mathcal{P}^{op}$ maps a set $U \subseteq X$ by $\widehat{\Box}_X(U) = \{V \subseteq X \mid U \subseteq V\}$. Recall that the unit and multiplication of $\mathsf{Pow}$ are the singleton map and union, respectively (cf. Example 2.3.16). Here we use $\eta$ and $\mu$ to denote the unit and multiplication associated with $\mathcal{P}\mathcal{P}^{op}$, respectively, (cf. Example 2.3.16). Now we show that $\widehat{\Box}$ is a monad morphism.

   For the units we want to show $\widehat{\Box} \circ \{-\} = \eta$. This follows because for all $x \in X$,

   $$\widehat{\Box}_X(\{x\}) = \{V \subseteq X \mid \{x\} \subseteq V\} = \{V \subseteq X \mid x \in V\} = \eta(x).$$

   For the multiplication we want to show that $\widehat{\Box}_X \circ \bigcup = \mu \circ \widehat{\Box}_{\mathcal{P}\mathcal{P}^{op}(X)} \circ \mathsf{Pow}(\widehat{\Box}_X)$. For this, let $N \in \mathsf{Pow}(\mathcal{P}(X))$ and $V \subseteq X$. We now have:

   $$\begin{aligned}
   & V \in \mu_X \circ \widehat{\Box}_{\mathcal{P}\mathcal{P}^{op}(X)} \circ \mathsf{Pow}(\widehat{\Box}_X)(N) \iff \\
   \iff\ & \{H \in \mathcal{P}\mathcal{P}^{op}(X) \mid V \in H\} \in \widehat{\Box}_{\mathcal{P}\mathcal{P}^{op}(X)} \circ \mathsf{Pow}(\widehat{\Box}_X)(N) && (\text{Def. } \mu_X) \\
   \iff\ & \mathsf{Pow}(\widehat{\Box}_X)(N) \subseteq \{H \subseteq \mathcal{P}X \mid V \in H\} && (\text{Def. } \Box_{\mathcal{P}\mathcal{P}^{op}(X)}) \\
   \iff\ & \{\widehat{\Box}_X(U) \mid U \in N\} \subseteq \{H \subseteq \mathcal{P}X \mid V \in H\} && (\text{Def. } \mathsf{Pow}(\widehat{\Box}_X) \\
   \iff\ & \forall U \in N : V \in \widehat{\Box}_X(U) && (\text{Def. } \subseteq) \\
   \iff\ & \forall U \in N : U \subseteq V && (\text{Def. } \widehat{\Box}_X) \\
   \iff\ & \bigcup N \subseteq V && (\text{Def. } \bigcup) \\
   \iff\ & V \in \widehat{\Box}_X(\bigcup N) && (\text{Def. } \widehat{\Box}_X)
   \end{aligned}$$

2. The non-normal modality $\Box : \mathcal{P} \to \mathcal{P}(\mathcal{P}\mathcal{P}^{op})$ supports sequential composition. Recall (Example 3.1.3) that $\Box(\varphi) = \{N \in \mathcal{P}\mathcal{P}^{op}(X) \mid \varphi \in N\}$. Hence its transpose $\widehat{\Box} : \mathcal{P}\mathcal{P}^{op} \to \mathcal{P}^{op}\mathcal{P}$ maps $N \in \mathcal{P}\mathcal{P}^{op}(X)$ to

   $$\begin{aligned}
   \widehat{\Box}(N) &= \{\varphi \in \mathcal{P}(X) \mid N \in \Box(\varphi)\} && (\text{Def. } \widehat{\Box}) \\
   &= \{\varphi \in \mathcal{P}(X) \mid \varphi \in N\} = N. && (\text{Def. } \Box)
   \end{aligned}$$

   In other words, $\widehat{\Box}$ is the identity which is clearly a monad morphism.

3. The monotone non-normal modality $\Box : \mathcal{P} \to \mathcal{P}\mathsf{Mon}$ supports sequential composition. Following the computation for the non-normal modality $\Box$ we can see that $\widehat{\Box} : \mathsf{Mon} \to \mathcal{P}^{op}\mathcal{P}$ is the natural inclusion which is also a monad morphism.

Using the Yoneda lemma we can show that the transpose of a predicate lifting $\lambda$ is a monad morphism iff the corresponding function $\mathcal{T}(2) \longrightarrow 2$, via Proposition 3.3.9, is an Eilenberg-Moore algebra on 2; this was indeed how [83] suggested to define modalities to study notions of computation.

**Lemma 7.3.9.** *Let $\mathcal{T}$ be a monad, on* Set. *The transpose of a predicate lifting $\lambda : \mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ is a monad morphism iff the associated function $p : \mathcal{T}(2) \longrightarrow 2$, via Proposition 3.3.9, is an Eilenberg-Moore $\mathcal{T}$-algebra (Definition 2.3.17).*

*Proof.* The proof is a long straightforward computation hence we omit it here. □

This last observation can be used to elegantly show that the modality $\Diamond$, for Kripke frames, supports sequential composition.

**Example 7.3.10.** In the case of Pow-coalgebras, $\Diamond$ supports sequential composition. Indeed, $\Diamond$ corresponds to the free Pow-algebra over 1.

More concretely, recall Example 3.1.3 on page 39. Taking $2 = \{\bot, \top\}$ we have that $\Diamond$ corresponds to the set $\{\{\top\}, \{\bot, \top\}\}$, we call $\chi_\Diamond : \mathcal{P}(2) \longrightarrow 2$ the characteristic function of this set.

We want to show that $\chi_\Diamond$ is the function $\bigcup_1 : \mathsf{PowPow}(1) \longrightarrow \mathsf{Pow}(1)$. For this Now take $\mathsf{Pow}(1) = \{\emptyset, \{*\}\}$ and identify $\emptyset$ with $\bot$ and $\{*\}$ with $\top$; we then have $\mathsf{PowPow}(1) = \big\{\emptyset, \{\emptyset\}, \{\{*\}\}, \{\emptyset, \{*\}\}\big\}$.

First notice that with his notation $\Diamond$ corresponds to the set $\big\{\{\{*\}\}, \{\emptyset, \{*\}\}\big\}$. Second notice that $\bigcup_1 \{\{*\}\} = \bigcup \{\emptyset, \{*\}\} = \{*\}$ and $\bigcup_1 x = \emptyset$ in any other case. This means $\chi_\Diamond = \bigcup_1$, i.e the free Pow algebra over 1, as we wanted to show.

It is well known that in PDL and GL, the sequential composition axiom holds not only for the labelled $\Box$-modalities, but also for the labelled $\Diamond$-modalities. This is a more general phenomenon, since we can show that a $\mathcal{T}$-predicate lifting $\lambda$ supports sequential composition iff its Boolean dual does.

**Proposition 7.3.11.** *Let $\mathcal{T}$ be a monad. A predicate lifting $\lambda\colon \mathcal{P} \longrightarrow \mathcal{P}\mathcal{T}$ supports sequential composition iff its Boolean dual $\neg\lambda\neg$ does.*

*Proof.* We will use the previous lemma. Notice that if a predicate lifting corresponds to a function $\lambda : \mathcal{T}(2) \longrightarrow 2$ then its Boolean dual is associated with the function $\neg \circ \lambda \circ \mathcal{T}(\neg)$, i.e. the following composite $\mathcal{T}(2) \xrightarrow{\mathcal{T}(\neg)} \mathcal{T}(2) \xrightarrow{\lambda} 2 \xrightarrow{\neg} 2$, where $\neg : 2 \longrightarrow 2$ is the usual negation.

Since $\neg\neg = id_2$, the Boolean dual of the Boolean dual of $\lambda$ is $\lambda$. Then to prove the proposition, by Lemma 7.3.9, it is enough to show that if $\lambda$ is an Eilenberg-Moore algebra (Definition 2.3.17) so is $\neg \circ \lambda \circ \mathcal{T}(\neg)$.

For the unit we want to show $\neg \circ \lambda \circ \mathcal{T}(\neg) \circ \eta_2 = id_2$. This follows because

$$\begin{aligned} \neg \circ \lambda \circ \mathcal{T}(\neg) \circ \eta_2 &= \neg \circ \lambda \circ \eta_2 \circ \neg && (\eta \text{ is natural}) \\ &= \neg \circ id_2 \circ \neg && (\lambda \text{ is an EM-algebra}) \\ &= id_2 \end{aligned}$$

For the multiplication we want to show $\neg \circ \lambda \circ \mathcal{T}(\neg) \circ \mathcal{T}(\neg \circ \lambda \circ \mathcal{T}(\neg)) = \neg \circ \lambda \circ \mathcal{T}(\neg) \circ \mu$. This follows because

$$\begin{aligned}
\neg \circ \lambda \circ \mathcal{T}(\neg) \circ \mu_2 &= \neg \circ \lambda \circ \mu_2 \circ \mathcal{T}\mathcal{T}(\neg) && (\mu \text{ is natural})\\
&= \neg \circ \lambda \circ \mathcal{T}(\lambda) \circ \mathcal{T}\mathcal{T}(\neg) && (\lambda \text{ is an EM-algebra})\\
&= \neg \circ \lambda \circ \mathcal{T}(id_2) \circ \mathcal{T}(\lambda) \circ \mathcal{T}\mathcal{T}(\neg) && (\mathcal{T}(id_2) = id_{\mathcal{T}(2)})\\
&= \neg \circ \lambda \circ \mathcal{T}(\neg\neg) \circ \mathcal{T}(\lambda) \circ \mathcal{T}\mathcal{T}(\neg) && (\neg\neg = id_2)\\
&= \neg \circ \lambda \circ \mathcal{T}(\neg) \circ \mathcal{T}(\neg) \circ \mathcal{T}(\lambda) \circ \mathcal{T}\mathcal{T}(\neg)\\
&= \neg \circ \lambda \circ \mathcal{T}(\neg) \circ \mathcal{T}(\neg \circ \lambda \circ \mathcal{T}(\neg))
\end{aligned}$$

This concludes the proof. □

### 7.3.2 Pointwise extensions

In this section we will investigate the framework underlying axioms of the form

$$[a \cup b]\varphi \Longleftrightarrow [a]\varphi \vee [b]\varphi.$$

In terms of predicate transformers the axiom is valid if and only if the following equation holds

$$[\![ [a \cup b]_\lambda - ]\!]_\xi = [\![ [a]_\lambda - ]\!]_\xi \cup [\![ [b]_\lambda - ]\!]_\xi. \tag{7.4}$$

The aim of this section is twofold. First we want to formalise what it means for a predicate lifting to support such structure. Second, we want to isolate properties of the predicate lifting $\lambda$ so that the equation above is valid.

Our main result in this section (Theorem 7.3.18) roughly states that the axiom above is valid if the transpose of $\lambda$ preserves the structure. In particular, Corollary 7.3.21 states that in the case the structure on programs is given by an algebraic signature $\Sigma$ then for any term $p(\alpha_1, \ldots \alpha_2)$ in $\Sigma$ the axiom

$$\big[p(\alpha_1, \ldots, \alpha_n)\big]\varphi \iff p([\alpha_1]\varphi, \ldots, [\alpha_n]\varphi)$$

is valid if the transpose of $\lambda$ preserves the structure.

Instead of working just with a signature of programs we use a monad. There are two reasons for this more abstract approach. A first reason is that the technical work is not more complicated, although it might be conceptually more involved for the reader not used to see algebraic theories as monads. A second reason is that using monads we can also account for equations between programs, e.g $a \cup b = b \cup a$ or $a^* = \epsilon \cup a; a^*$, see Examples 7.3.17 and 7.3.22.

The next remark concerns the two different perspectives of monads used in this chapter.

**Remark 7.3.12.** We point out that we are using two essentially different perspectives on monads. The first perspective is that of a monad as a categorical representation of an algebraic theory (Section 2.3); this is the perspective that we will use in this section. The second perspective, described in the previous section, presents monads as a notion of computation [83] and used them to describe sequential composition.

In general, the situation will be as follows: let $\mathbb{M}$ be a monad and assume that for each $X$ there are $\mathbb{M}$-algebras $\theta : M\left(\mathcal{T}(X)^X\right) \to \mathcal{T}(X)^X$ and $\chi : M\left(\mathcal{P}(X)^{\mathcal{P}(X)}\right) \to \mathcal{P}(X)^{\mathcal{P}(X)}$. A predicate lifting $\lambda$ (for $\mathcal{T}$) *supports the algebraic theory of* $\mathbb{M}$ if $[\![\Box_\lambda -]\!]_{(-)} : \mathcal{T}(X)^X \to \mathcal{P}(X)^{\mathcal{P}(X)}$ is a homomorphism of $\mathbb{M}$-algebras. Before explaining the general case we will concentrate on Equation (7.4) where the structures $\theta$ and $\chi$ are obtained by *pointwise extension*.

We now explain pointwise extensions. First consider the case of an algebra $A$ with a binary operation $+ : A \times A \to A$. We can extend this operation pointwise to $A^X$ as the operation $+^X : A^X \times A^X \to A^X$ which maps a pair of functions $f$ and $g$ to the function given by $(f +^X g)(x) = f(x) + g(x)$. This can be generalised to any algebraic structure given by a monad using a natural transformation

$$st^X \colon M \circ (-)^X \to (-)^X \circ M$$

called *strength*. In the case of $\mathsf{Set}$, $(st^X)_A \colon M(A^X) \to M(A)^X$ is defined by

$$(st^X)_A(t)(x) = M(ev(-, x))(t),$$

where $ev \colon A^X \times X \to A$ is the obvious evaluation map. If there is no risk of confusion we simply write $st$.

**Definition 7.3.13.** Let $X$ be a set. The $X$ *pointwise extension of an* $\mathbb{M}$*-algebra* $\alpha \colon M(A) \to A$, written $\alpha^X : M(A^X) \to A^X$, is given by $\alpha^X = (\alpha \circ -) \circ (st^X)_A$, i.e the following composite $M(A^X) \xrightarrow{(st^X)_A} M(A)^X \xrightarrow{\alpha\circ -} A^X$.

In order to show that $\alpha^X$ is, in fact, an $\mathbb{M}$-algebra we need that $st$ is a distributive law, i.e. $\eta^X = st \circ \eta_{(-)^X}$ and $st \circ \mu_{(-)^X} = \mu^X \circ st_M \circ M(st)$ (see Remark 3.3.8) . Since we are working on $\mathsf{Set}$ this will always be the case, see e.g. [54] for details.

As we will see later, Equation (7.4) concerns pointwise operations. By using a monad $\mathbb{M}$ instead of just a signature, the equations incorporated by $\mathbb{M}$ will also hold for the pointwise extensions. For example, if $+$ is a commutative operation on $A$, then $+^X$ is commutative as well.

We tame equation (7.4) using pointwise extensions that are “natural” on $A$. We formalise this with the concept of *natural* $\mathbb{M}$*-algebras*.

**Definition 7.3.14.** Let $\mathbb{M}$ be a monad on $\mathsf{Set}$, and let $H : \mathbb{C} \longrightarrow \mathsf{Set}$ be a functor. A *natural* $\mathbb{M}$*-algebra on* $H$ is a natural transformation $\alpha\colon MH \longrightarrow H$ such that each of its components is an $\mathbb{M}$-algebra.

**Example 7.3.15.** 1. All Boolean operations, like union, complement, etc., are natural on $\mathcal{P}$. Since the inverse image map of a function preserves all of those, we have a natural transformation $\Sigma_{\mathsf{BA}}\mathcal{P} \longrightarrow \mathcal{P}$, where $\Sigma_{\mathsf{BA}}$ is the boolean signature; in other words we have a natural $\Sigma_{\mathsf{BA}}$-algebra on $\mathcal{P}$. Moreover, since these operations make $\mathcal{P}(X)$ into a boolean algebra we have a natural transformation $\mathbb{M}_{\mathsf{BA}}\mathcal{P} \longrightarrow \mathcal{P}$, where $\mathbb{M}_{\mathsf{BA}}$ is the Boolean algebra monad, i.e. $\mathbb{M}_{\mathsf{BA}} = U_{\mathsf{BA}}F_{\mathsf{BA}}$. In other words we have a natural $\mathbb{M}_{\mathsf{BA}}$-algebra on $\mathcal{P}$.

2. For similar reasons, all Boolean operations are natural on $\mathcal{P}\mathcal{P}^{op}$ and $\mathsf{Mon}$. Moreover, the dual operation of Game Logic is also natural for those functors.

3. On the contrary, the only Boolean operation that is natural on $\mathsf{Pow}$ is union, because the direct image of a function preserves unions, but not intersections or complements.

4. Let $\mathbb{M}$ be the commutative monoid monad. The binary union gives a natural $\mathbb{M}$-algebra on $\mathsf{Pow}$.

Now we can formally define what it means for a predicate lifting to support the pointwise structure given by a monad.

**Definition 7.3.16.** Let $\mathbb{M}$ be a monad on $\mathsf{Set}$, let $\mathcal{T}$ be an endofunctor on $\mathsf{Set}$ and let $\lambda$ be a predicate lifting for $\mathcal{T}$. Assume there are natural $\mathbb{M}$-algebras $\theta\colon M\mathcal{T} \longrightarrow \mathcal{T}$ and $\chi\colon M\mathcal{P} \longrightarrow \mathcal{P}$. Let $(\theta_X)^X : M(\mathcal{T}(X)^X) \longrightarrow \mathcal{T}(X)$ and $(\chi_X)^{\mathcal{P}(X)} : M(\mathcal{P}(X)^{\mathcal{P}(X)} \longrightarrow \mathcal{P}(X)^{\mathcal{P}(X)}$ be the pointwise extensions of $\theta$ and $\chi$, respectively.

Let $L_0$ be set (of atomic labels), and $L = M(L_0)$ be the free $\mathbb{M}$-algebra over $L_0$, write $\alpha$ for the structural map. We say that $\lambda$ *supports the* $\mathbb{M}$*-structure relative to* $\theta$ *and* $\chi$ if the following diagram

$$\begin{array}{ccccc}
M(L) & \xrightarrow{M(\widehat{\xi})} & M\left(\mathcal{T}(X)^X\right) & \xrightarrow{M\left([\![\Box_\lambda -]\!]_{(-)}\right)} & M\left(\mathcal{P}(X)^{\mathcal{P}(X)}\right) \\
{\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle(\theta_X)^X} & & \downarrow{\scriptstyle(\chi_X)^{\mathcal{P}(X)}} \\
L & \xrightarrow[\widehat{\xi}]{} & \mathcal{T}(X)^X & \xrightarrow[{[\![\Box_\lambda -]\!]_{(-)}}]{} & \mathcal{P}(X)^{\mathcal{P}(X)}
\end{array} \tag{7.5}$$

commutes for every standard coalgebra $\xi : X \longrightarrow \mathcal{T}(X)^L$ (Definition 7.3.1).

In the case of a signature $\Sigma$, the previous definition says the following: A predicate lifting $\lambda$ supports the $\Sigma$ structure if for each term $p(\alpha_1, \ldots, \alpha_n) \in T_\Sigma(L_0)$ and each standard coalgebra $\xi : X \longrightarrow \mathcal{T}(X)^{T_\Sigma(L_0)}$ the following holds

$$[\![ [p(\alpha_1, \ldots, \alpha_n)]_\lambda - ]\!]_\xi = p^\chi([\![ [\alpha_1]_\lambda - ]\!]_\xi, \ldots, [\![ [\alpha_n]_\lambda - ]\!]_\xi)$$

The next example illustrates this equation, and the previous definition, in more detail for the particular case of a binary operation $+$.

**Example 7.3.17.** Consider the functor $\Sigma(X) = X \times X$ instead of the free monad. Let $L = T_\Sigma(L_0)$. Write $+ : L \times L \to L$ for the operation between labels. Assume we have natural transformations $\oplus : \Sigma\mathcal{T} = \mathcal{T} \times \mathcal{T} \to \mathcal{T}$ and $\boxplus : \Sigma\mathcal{P} = \mathcal{P} \times \mathcal{P} \to \mathcal{P}$. Let $\lambda$ be a unary predicate lifting for $\mathcal{T}$ and write $\Box_\lambda$ for the associated modality.

We will now show

$$\left[\!\left[ [t + t']_\lambda - \right]\!\right]_\xi = \left[\!\left[ [t]_\lambda - \right]\!\right]_\xi \boxplus_X \left[\!\left[ [t']_\lambda - \right]\!\right]_\xi .$$

for every standard coalgebra $\xi$. The operation $\oplus$ is needed to obtain a dynamic structure on $L$ by freeness. Using the commutativity of Diagram 7.5 we can show that for every standard coalgebra $\xi : X \to \mathcal{T}(X)^{T_\Sigma(L_0)}$ (Definition 7.3.1), every pair of terms $t, t' \in L = T_\Sigma(L_0)$ and every $U \in \mathcal{P}(X)$ we have:

$$\begin{aligned}
\left[\!\left[ [t + t']_\lambda - \right]\!\right]_\xi (U) &= [\![\Box_\lambda - ]\!]_{\widehat{\xi}(t+t')}(U) && \text{(Lemma 7.1.2)}\\
&= (\boxplus_X)^{\mathcal{P}(X)} \Sigma \left( [\![\Box_\lambda - ]\!]_{(-)} \circ \widehat{\xi} \right) (t, t')(U) && \text{(Diagram 7.5)}\\
&= \left( [\![\Box_\lambda - ]\!]_{\widehat{\xi}(t)} (\boxplus_X)^{\mathcal{P}(X)} [\![\Box_\lambda - ]\!]_{\widehat{\xi}(t')} \right)(U) && \text{(Def. } \Sigma)\\
&= [\![\Box_\lambda - ]\!]_{\widehat{\xi}(t)}(U) \boxplus_X [\![\Box_\lambda - ]\!]_{\widehat{\xi}(t')}(U) && \text{(Def. } (\boxplus_X)^{\mathcal{P}(X)})\\
&= \left[\!\left[ [t]_\lambda - \right]\!\right]_\xi (U) \boxplus_X \left[\!\left[ [t']_\lambda - \right]\!\right]_\xi (U). && \text{(Lemma 7.1.2)}
\end{aligned}$$

In the case that $t$ and $t'$ are PDL programs and $\boxplus = \cup$, we obtain Equation (7.4).

The use of monads allows to incorporate the axioms. For example if $+$ is commutative, then to have a dynamic structure $\oplus$ and $\boxplus$ should also be commutative, and then the following holds:

$$\left[\!\left[ [t + t']_\lambda - \right]\!\right]_\xi = \left[\!\left[ [t' + t]_\lambda - \right]\!\right]_\xi .$$

We can now state the main theorem of this section analogous to Theorem 7.3.7.

**Theorem 7.3.18.** *Let $\mathbb{M}$ be a monad on* $\mathsf{Set}$*, let $\mathcal{T}$ be an endofunctor on* $\mathsf{Set}$ *and let $\lambda$ be a predicate lifting for $\mathcal{T}$. Assume there are natural $\mathbb{M}$-algebras $\theta \colon M\mathcal{T} \to \mathcal{T}$ and $\chi \colon M\mathcal{P} \to \mathcal{P}$. A predicate lifting $\lambda \colon \mathcal{P} \to \mathcal{P}\mathcal{T}$ supports the $\mathbb{M}$-structure relative to $\theta$ and $\chi$ if the following diagram*

$$\begin{array}{ccc}
M\mathcal{T} & \xrightarrow{M(\widehat{\lambda})} & M\mathcal{P}\mathcal{P}^{op} \\
\downarrow{\scriptstyle \theta} & & \downarrow{\scriptstyle \chi_{\mathcal{P}^{op}}} \\
\mathcal{T} & \xrightarrow[\widehat{\lambda}]{} & \mathcal{P}\mathcal{P}^{op}
\end{array} \tag{7.6}$$

*commutes, i.e. $\widehat{\lambda}$ is a morphism of natural $\mathbb{M}$-algebras.*

We will need the following lemma in the proof of the theorem

**Lemma 7.3.19.** *Let* $\mathbb{M}$ *be a monad on* $\mathsf{Set}$*. For any natural* $\mathbb{M}$*-algebra* $\chi\colon M\mathcal{P} \longrightarrow \mathcal{P}$ *and every pair of sets* $A, B$ *the following diagram commutes:*

$$\begin{array}{ccccc}
M(\mathcal{P}(A)^B) & \xrightarrow{st^B_{\mathcal{P}(A)}} & (M\mathcal{P}(A))^B & \xrightarrow{\chi_A \circ -} & \mathcal{P}(A)^B \\
{\scriptstyle M(\psi)}\downarrow & & & & \downarrow{\scriptstyle \psi} \\
M(\mathcal{P}(B)^A) & \xrightarrow[st^A_{\mathcal{P}(B)}]{} & (M\mathcal{P}(B))^A & \xrightarrow[\chi_B \circ -]{} & \mathcal{P}(B)^A
\end{array}$$

*where* $\psi\colon \mathsf{Set}[-, \mathcal{P}] \longrightarrow \mathsf{Set}^{op}[\mathcal{P}^{op}, -]$ *is the natural isomorphism associated with the adjunction* $\mathcal{P}^{op} \dashv \mathcal{P}$*.*

*Proof.* The idea of the proof is to present $M$ by operations and equations, see Chapter 5; the intention of this is to decompose $\chi : M\mathcal{P} \longrightarrow \mathcal{P}$ into several $n$-ary operations where $n$ ranges over the cardinals. We can then use the Yoneda Lemma and the fact that the connectives $\bigvee, \neg$ are a functionally complete set, i.e. every function $2^n \longrightarrow 2$ can be expressed as a term of these, see [35] for details.

Without loss of generality, we can assume there exist a signature functor $\Sigma$ and a surjective natural transformation $E : \Sigma \longrightarrow M$, where $\Sigma = \coprod \Sigma_n \times (-)^n$, for example $\Sigma$ could be the canonical signature (Example 5.1.3).

Notice that the horizontal edges in the diagram above are the pointwise extensions $(\chi_A)^B$ and $(\chi_B)^A$, respectively. Since $E$ is surjective, and all the morphisms involved are natural, it suffices to show that

$$\psi \circ \chi_A^B \circ E_{\mathcal{P}(A)^B} = \chi_B^A \circ M(\psi) \circ E_{\mathcal{P}(A)^B}.$$

More concretely, we want to show that for each $n$-ary operation $p \in \Sigma_n$ and every sequence $(f_1, \ldots, f_n) \in \big(\mathcal{P}(A)^B\big)$ the following holds: for each $a \in A$

$$\chi_B\Big(p, \widehat{f_1}(a), \ldots, \widehat{f_n}(a)\Big) = \Big\{b \mid a \in \chi_A\big(p, f_1(b), \ldots f_n(b)\big)\Big\}. \tag{7.7}$$

We first show that Equation (7.7) holds for the operations $\bigvee, \neg$. Keep in mind that, in principle, $\bigvee, \neg$ need not be in the signature induced by $\Sigma$.

**Claim.** Let $n$ be an arbitrary but fixed cardinal number. Each of the operations $\bigvee : \mathcal{P}^n \longrightarrow \mathcal{P}$, $\neg : \mathcal{P} \longrightarrow \mathcal{P}$ make Equation (7.7) true. Moreover, any term of these also makes Equation (7.7) true.

*Proof of the claim.* In the case of negation, Equation (7.7) reduces to prove that for each $f : B \longrightarrow \mathcal{P}(A)$ and each $a \in A$, we have $\neg\widehat{f}(a) = \{b \mid a \in \neg f(b)\}$. This follows directly from the definition of $\widehat{f}$.

The case of $n$-ary disjunctions reduces Equation (7.7) to the following: For each $n$-tuple $(f_1, \dots, f_n) \in \big(\mathcal{P}(A)^B\big)$ and each $a \in A$ we have $\bigvee_{i\leq n} \widehat{f_i}(a) = \{b \mid a \in \bigvee_{i\leq n} f_i(b)\}$. This is also immediate from the definition of exponential transposes.

The case of an arbitrary term will now follow from the case of disjunctions and negations because every boolean term can be expressed using only disjunctions ($\bigvee$) and negations ($\neg$). This finishes the proof of the claim. $\square$

It remains to show that the equation still holds for every possible operation $p$ in any other signature. Here is where the Yoneda Lemma and functional completeness come into the picture.

Recall that each operation $p \in \Sigma_n$ induces a natural transformation $\chi \circ E(-,p) : \mathcal{P}^n \longrightarrow \mathcal{P}$ and that $\chi \circ E : \Sigma\mathcal{P} \longrightarrow \mathcal{P}$ is totally determined by those (Section 5.1). The Yoneda Lemma tells us that for each operation $p \in \Sigma_n$, the natural transformation $\chi \circ E(-,p)$ is totally described by an arrow $Y(p) : 2^n \longrightarrow 2$. By the functional completeness of $\bigvee, \neg$ the arrow $Y(p) : 2^n \longrightarrow 2$ can be expressed as a term over $\bigvee, \neg$. From this, using the claim above, we conclude that Equation (7.7) holds. This concludes the proof of the lemma. $\square$

We can now prove Theorem 7.3.18.

*Proof of Theorem 7.3.18.* We want to prove that Diagram (7.5) commutes for every standard coalgebra $\xi : X \longrightarrow \mathcal{T}(X)^L$. First notice that the rectangle on the left of Diagram (7.5) commutes for every standard $\xi$, by definition of standard coalgebra. Hence to prove the theorem, it is enough to show that the rectangle on the right of Diagram (7.5) commutes. In other words, we want to show that $[\![\Box_\lambda -]\!]_{(-)}$ is a $\mathbb{M}$-morphism between the pointwise extensions of $\theta$ and $\chi_{\mathcal{P}}$ (Definition 7.3.16). More specifically, for each $X$, we want to show

$$(\chi_X)^{\mathcal{P}(X)} \circ M\Big([\![\Box_\lambda -]\!]_{(-)}\Big) = [\![\Box_\lambda -]\!]_{(-)} \circ (\theta_X)^{\mathcal{P}(X)}. \tag{7.8}$$

The proof of this, and then of the theorem, will follow from the following diagram:

$$\begin{array}{ccccc}
M(\mathcal{T}(X)^X) & \xrightarrow{M(\widehat{\lambda}\circ -)} & M(\mathcal{P}\mathcal{P}(X)^X) & \xrightarrow{M(\psi)} & M(\mathcal{P}(X)^{\mathcal{P}(X)}) \\
\downarrow{\scriptstyle st^X_{\mathcal{T}(X)}} & & \downarrow{\scriptstyle st^X_{\mathcal{P}\mathcal{P}X}} & & \downarrow{\scriptstyle st^{\mathcal{P}(X)}_{\mathcal{P}(X)}} \\
\Big(M\mathcal{T}(X)\Big)^X & \xrightarrow{M(\widehat{\lambda})\circ -} & \Big(M\mathcal{P}\mathcal{P}(X)\Big)^X & & \Big(M\mathcal{P}(X)\Big)^{\mathcal{P}(X)} \\
\downarrow{\scriptstyle \theta_X \circ -} & & \downarrow{\scriptstyle \chi_{\mathcal{P}(X)} \circ -} & & \downarrow{\scriptstyle \chi_X \circ -} \\
\mathcal{T}(X)^X & \xrightarrow[\widehat{\lambda}\circ -]{} & \mathcal{P}\mathcal{P}(X)^X & \xrightarrow[\psi]{} & \mathcal{P}(X)^{\mathcal{P}(X)}
\end{array}$$

where $\psi$ is as in Lemma 7.3.19.

We now argue that the outer rectangle is the rectangle on the right of Diagram (7.5), i.e. its commutativity is Equation 7.8. Firstly, observe that by definition the vertical outer edges are the pointwise extensions $(\theta_X)^{\mathcal{P}(X)}$ and $(\chi_X)^{\mathcal{P}(X)}$, respectively. Secondly, by properties of adjoints, see equations after Lemma A.1.2, notice that the horizontal edges of the outer rectangle are $M(\llbracket \Box_\lambda - \rrbracket_{(-)})$ and $\llbracket \Box_\lambda - \rrbracket_{(-)}$ respectively.

Now we argue that the above diagram commutes. The upper left rectangle commutes because $st$ is natural. The lower left rectangle commutes because we assume $\widehat{\lambda}$ to be a morphism of natural $\mathbb{M}$-algebras and $(-)^X$ is a functor. Finally, the rectangle on the right commutes because of Lemma 7.3.19. Since all the inner rectangles commute so does the outer rectangle. This concludes the proof. □

The next examples illustrate Theorem 7.3.18.

**Example 7.3.20.** Using Theorem 7.3.18 we can see that

1. the PDL axiom $[\alpha \cup \beta]\varphi \iff [\alpha]\varphi \wedge [\beta]\varphi$ is valid because the adjoint, $\widehat{\lambda}\colon \mathsf{Pow} \to \mathcal{P}^{op}\mathcal{P}$, of the predicate lifting associated with $\Box$ transforms unions into intersections. To see this first recall that $\widehat{\Box}_X(U) = \{V \subseteq X \mid U \subseteq V\}$. Then for $U_1, U_2 \in \mathsf{Pow}(X)$ we have:
$$\begin{aligned}\widehat{\Box}_X(U_1 \cup U_2) &= \{V \subseteq X \mid U_1 \cup U_2 \subseteq V\}\\ &= \{V \subseteq X \mid U_1 \subseteq V \text{ and } U_2 \subseteq V\}\\ &= \{V \subseteq X \mid U_1 \subseteq V\} \cap \{V \subseteq X \mid U_2 \subseteq V\}\\ &= \widehat{\Box}_X(U_1) \cap \widehat{\Box}_X(U_2).\end{aligned}$$

2. On the contrary, in Game Logic the axiom $[\alpha \cup \beta]\varphi \iff [\alpha] \vee \varphi[\beta]\varphi$ is valid. Because, as we saw in Example 7.3.8, the transposes of the non-normal modalities $\Box : \mathcal{P} \to \mathcal{P}\mathsf{Mon}$ and $\Box : \mathcal{P} \to \mathcal{P}\mathcal{P}\mathcal{P}^{op}$ are the inclusion and the identity, respectively. Both of them maps joins to joins hence they yield the axiom above.

We now rephrase Theorem 7.3.18 in term of algebraic signatures.

**Corollary 7.3.21.** *Under the assumptions of Theorem 7.3.18. If* $\widehat{\lambda}\colon (\mathcal{T}, \theta) \to (\mathcal{P}\mathcal{P}^{op}, \chi_{\mathcal{P}})$ *is a morphism of* $\Sigma$*-algebras, then for each term* $p(\alpha_1, \dots, \alpha_n) \in T_\Sigma(L_0)$ *and each* $\mathcal{T}$*-dynamic* $\Sigma$*-structure* $\sigma\colon T_\Sigma(L_0) \to \mathcal{T}(X)^X$ *the following holds*

$$\left\llbracket [p(\alpha_1, \dots, \alpha_n)]_\lambda - \right\rrbracket_{\widehat{\xi}} = p^\chi\left(\left\llbracket [\alpha_1]_\lambda - \right\rrbracket_{\widehat{\xi}}, \dots, \left\llbracket [\alpha_n]_\lambda - \right\rrbracket_{\widehat{\xi}}\right)$$

*where* $p^\chi$ *is the interpretation of the operation* $p$ *in* $(\mathcal{P}\mathcal{P}^{op}, \chi_{\mathcal{P}})$.

Phrasing the previous corollary with axioms we have that if $\widehat{\lambda}$ is a morphism of $\Sigma$ algebras then the axiom

$$\big[p(\alpha_1, \dots, \alpha_n)\big]\varphi \iff p([\alpha_1]\varphi, \dots, [\alpha_n]\varphi)$$

is valid.

The next example uses Theorem 7.3.7 and Theorem 7.3.18 to deduce the usual axiom for iteration.

**Example 7.3.22.** It is well known that iteration satisfies $a^* = \epsilon \cup a; a^*$. We will illustrate how to obtain the usual axiom for iteration for the $\Box$ modality of Pow, i.e. i.e. $[a^*]\varphi \iff \varphi \wedge [a][a^*]\varphi$ and $\mathcal{P}\mathcal{P}^{op}$ (or Mon), i.e. $[a^*]\varphi \iff \varphi \vee [a][a^*]\varphi$, in our framework. In both cases we interpret the atomic label $\epsilon$ as the unit of the monad, i.e. given a standard coalgebra $\xi : X \longrightarrow \mathcal{T}(X)^L$, $\widehat{\xi}(\epsilon) = \eta$. This implies that in both cases for the $\Box$ modality we have

$$[\![ [\epsilon] - ]\!]_\xi(\varphi) = \varphi. \tag{7.9}$$

1. The universal modality $\Box : \mathcal{P} \longrightarrow \mathcal{P}\mathsf{Pow}$ supports sequential composition (Example 7.3.8) and "transforms" $\cup$ into $\wedge$ (Example 7.3.20) then we have

$$\begin{aligned}
[\![ [a^*]_\lambda - ]\!]_\xi(\varphi) &= [\![ [\epsilon \cup a; a^*]_\lambda - ]\!]_\xi(\varphi) && (\xi \text{ is standard})\\
&= [\![ [\epsilon] - ]\!]_\xi(\varphi) \cap [\![ [a; a^*]_\lambda - ]\!]_\xi(\varphi) && (\text{Ex. } 7.3.20)\\
&= [\![ [\epsilon] - ]\!]_\xi(\varphi) \cap [\![ [a]_\lambda - ]\!]_\xi \circ [\![ [a^*]_\lambda - ]\!]_\xi(\varphi) && (\text{Ex. } 7.3.8)\\
&= \varphi \cap [\![ [a] - ]\!]_\xi \circ [\![ [a^*]_\lambda - ]\!]_\xi(\varphi). && (\text{Eq. } 7.9)
\end{aligned}$$

2. Following the same computations in the previous item for the case of the the non-normal modalities we obtain

$$[\![ [a^*]_\lambda - ]\!]_\xi(\varphi) = \varphi \cup [\![ [a]_\lambda - ]\!]_\xi \circ [\![ [a^*]_\lambda - ]\!]_\xi(\varphi)$$

## 7.4 Conclusions

In this chapter we developed a coalgebraic framework which covers Dynamic logics like (test-free) PDL and GL. We illustrated how the process of labelling modalities can be described by a generic process independent of any structure on the labels. We have shown with Theorems 7.3.7 and 7.3.18 that the usual axioms for PDL and GL present a property of the associated predicate liftings not of labelled modalities itself.

A key technical tool in our work was Lemma 7.3.19; this result also shows a limitation of the functorial approach to develop dynamic logics further. More concretely, Lemma 7.3.19 shows that we can only account for boolean terms at the pointwise level. In other words, we can only use boolean operations for axioms that preserve the rank of the modal formulas.

There are several topics that could be developed further. For example.

**Sequential composition of input-output systems:** In this chapter we studied sequential composition for coalgebras $\xi : X \to \mathcal{T}(X)$ where $\mathcal{T}$ is a monad and $X$ is the state space. Although, this view covers PDL and GL, it leaves out examples where coalgebras are seen as functional programs [83] or components [48] that transform input to output. For example, Mealy machines with input in $A$ and output in $B$ are coalgebras of type $X \to (B \times X)^A$ (cf. [96]), but the functor $(B \times -)^A$ is not a monad (unless $A = B$). However, the transpose of a Mealy machine is a map of type $A \to (B \times X)^X$ and the functor $\mathsf{S}(B) = (B \times X)^X$ is a monad in $B$ (called the monad of side effects in [83]). We can therefore define the sequential composition of two Mealy machines $\xi_1 : X \to (B \times X)^A$ and $\xi_2 : X \to (C \times X)^B$ as the transpose of the composition in $\mathcal{K}l(\mathsf{S})$ of $\widehat{\xi_1} : A \to \mathsf{S}(B)$ and $\widehat{\xi_2} : B \to \mathsf{S}(C)$. This functional perspective on programs is essentially that of [83], and it was also used in [57] to describe sequential composition in a coalgebraic semantics of Java programs.

Our notion of dynamic structure (Definition 7.3.1) does not support this functional view. However, there seems to be a way out of this. To illustrate, we can still define labelled modalities by labelling predicate liftings for the Mealy functor $(B \times -)^A$. For example, let $\xi_1 : X \to (B \times X)^A$ and $\xi_2 : X \to (C \times X)^B$ be two Mealy machines. For $a \in A$ consider the predicate lifting $[a]^A : \mathcal{P}(X) \to \mathcal{P}\left((B \times X)^A\right)$ which maps a set $U \subseteq X$ to $[a]^A(U) = \{\delta \in (B \times X)^A \mid \pi_2(\delta(a)) \in U\}$. Using the sequential composition described above we can show that

$$[\![ [a]^A - ]\!]_{\xi_2;\xi_1} = \bigcup_{b \in B} Rs^{\xi_2}(a)(b) \cap [\![ [a]^A - ]\!]_{\xi_1} \circ [\![ [b]^B - ]\!]_{\xi_2},$$

where $Rs^{\xi_2}(a)(b) = \{x \in X \mid \pi_1(\xi_2(x)(a)) = b\}$ (in words: "with input $a$ the output is $b$"). The same equation appeared in [57] for describing the sequential composition of normal termination for Java. This suggests that there is a general framework for exogenous modal logics for input-output systems yet to be developed.

**Other descriptions of sequential composition:** In [55], a perspective on predicate liftings using indexed categories (fibrations) is developed using the concept of *monad with a predicate lifting*. In this framework, predicate transformers are functors on a category of predicates; compositionality then follows from functoriality. Our characterisation in Theorem 7.3.7 is equivalent to the notion of a monad with a predicate lifting, in [55], when the base logic is given by the functor $\mathcal{P}\colon \mathsf{Set}^{op} \to \mathsf{Set}$. Our restriction of the base logic functor to $\mathcal{P}$, instead of any functor with codomain $\mathsf{Cat}$, allows us to define the concept of natural algebras for a monad (Definition 7.3.14) using a single monad. Also, the restriction of an algebraic structure on $\mathcal{P}$ to a structure on $\mathcal{P}\mathcal{P}^{op}$ does not seem natural in the framework of [55], hence our Theorem 7.3.18 does not seem to transfer. Although the approach in [55] is more general in some ways, it does not seem to include the non-monotonic predicate transformers arising from neighbourhood modalities [26], our framework does include those (Example 7.3.8).

**Bialgebra:** Bialgebras specify interaction between algebraic and coalgebraic structure by means of a distributive law. Distributive laws of the type $\delta : M\mathcal{T} \to \mathcal{T}M$, where $M$ is a monad specifying syntax and $\mathcal{T}$ is a coalgebra functor specifying behaviour, are specification formats in an abstract form of structural operational semantics, see e.g. [65]. Given that exogenous modal logics involve algebraic structure on programs and program behaviour is formalised as coalgebras, it might seem that dynamic modalities and their axioms should belong to the realm of bialgebra. However, at this point it is not clear that such a bialgebraic modelling exists, at least not in the expected manner.

**Meta-theory:** Finally, the general meta-theoretic properties such as completeness of the exogenous coalgebraic modal logics that fall under the scope of our framework still need to be clarified. Given that the completeness of Game Logic is still open, we expect that such results are difficult to obtain.

# Chapter 8

# Fixed Points Coalgebraically

In the functorial approach to coalgebraic modal logics is that the nesting of modalities in a formulas has finite *depth*. This is a key feature of the framework because it implies that the category of $\mathsf{Alg}(L)$, for some coalgebraic modal logic $(L, \delta)$, is finitary over the base category. However, this severely restricts the expressive power of these logics; as we mentioned before (Remark 3.2.6), coalgebraic modal logics only describe the one-step behavior of a state in a coalgebra. Moreover, as we proved in Theorem 5.2.17 every coalgebraic modal logic is a rank one axiomatization of a logic of predicate liftings. These characteristics of the the coalgebraic logics introduced in Part I make difficult to specifying the *ongoing* behavior of a state in a coalgebra by only finitary means. To account for the ongoing behaviour of states, one can extend the language with *fixpoint operators*, generalizing the modal $\mu$-calculus [66]. A coalgebraic fixpoint language on the basis of the Moss logic was introduced in [108]. Recently, [27] introduced the *coalgebraic $\mu$-calculus* $\mu\mathrm{ML}_\Lambda$ parametrized by a set $\Lambda$ of predicate liftings.

Given the success of automata-theoretic approaches towards fixpoint logics, one may expect a rich and elegant *universal automata theory* that generalizes the theory of specific devices for streams, trees or graphs, by dealing with automata that operate on coalgebras. A first step in this direction was the introduction of so-called *coalgebra automata* in [108, 67] where many results in automata theory, such as closure properties of recognizable languages, were generalized to this class of automata. However, coalgebra automata are related to fixpoint languages based on the Moss modality $\nabla$, and do not correspond directly to coalgebraic modal languages associated with predicate liftings (such as the graded modal $\mu$-calculus). In addition, the theory of coalgebra automata needs the *type* of the coalgebras, i.e. the functor, to preserves weak pullbacks, and hence cannot be applied as generally as possible.

This Chapter introduces automata for an *arbitrary* type of coalgebras (Definition 8.2.1). More precisely, given a set $\Lambda$ of monotone predicate liftings, we introduce $\Lambda$-automata as devices that accept or reject pointed $\mathcal{T}$-coalgebras on the basis of so-called *acceptance games*. With this, $\Lambda$-automata provide the game theoretical counterpart to the coalgebraic $\mu$-calculus, for $\Lambda$, as in [108]. In particular, there is a construction transforming a $\mu\mathrm{ML}_\Lambda$-formula into an equivalent $\Lambda$-automaton (of size quadratic in the length of the formula). Hence we may use the theory of $\Lambda$-automata in order to obtain results about coalgebraic modal fixpoint logic.

The main technical contribution of this chapter concerns a *small model property* for $\Lambda$-automata (Theorem 8.3.4). We show that any $\Lambda$-automaton $\mathbb{A}$ with a non-empty language recognizes a pointed coalgebra $(\xi, x)$ that can be obtained from $\mathbb{A}$ via some uniform construction involving a satisfiability game (Definition 8.3.2) that we associate with $\mathbb{A}$. The size of $(X, \xi)$ is exponential in the size of $\mathbb{A}$.

We also provide some ideas of how coalgebra automata could be treated within the functorial approach to modal logics.

In this chapter we do not go into all the technical game theoretical details as they would distract from our main story of drawing the boundaries of functorial framework to coalgebraic modal logics. We refer the reader to G. Fontaine PhD dissertation [40] where all the game theoretical details are addressed in an elegant and detailed manner.

## 8.1 Preliminaries

### Game theory & Automata

We assume familiarity with the basic notions of the theory of automata and infinite games [43]. Here we fix some notation and terminology.

**Definition 8.1.1.** 1. Given a set $A$, we write $A^*$ and $A^\omega$, respectively, for the sets of *words* (finite sequences) and *streams* (infinite sequences) over $A$, respectively. Given $\pi \in A^* + A^\omega$ we write $\mathit{Inf}(\pi)$ for the set of elements in $A$ that appear infinitely often in $\pi$.

2. A *stream automata* (also called $\omega$-automata) is a tuple $\mathbb{A} = (X, x_I, \xi, \Omega)$ where $X$ is a finite set, $x_I \in X$, $\xi$ is a function $\xi : X \longrightarrow \mathsf{Pow}(X)^A$, and $\Omega$ is a function $\Omega : X \longrightarrow \mathbb{N}$. An automata is said to be *deterministic* if for every $a \in A$ and every $x \in X$, the set $\xi(x)(a)$ is a singleton.

   A *run of an automata* on a word $a_0 a_1 \ldots \in A^\omega$ is a sequence $x_0 x_1 \ldots \in X^\omega$ such that $x_0 = x_I$ and for each $i \in \mathbb{N}$, $x_{i+1} \in \xi(x_i)(a_i)$.

   A word $a_0 a_1 \ldots A^\omega$ is *accepted* by the automata $\mathbb{A}$ if there is a run $x_0 x_1 \ldots$ automaton on $a_0 a_1 \ldots$ such that the maximum of the set $\{\Omega(x) \mid x \in \mathit{Inf}(x_0 x_1 \ldots)\}$ is even.

3. A subset of $A^\omega$ is called a language. A language is *$\omega$-regular* if there is a stream automata $\mathbb{A}$ such that $\mathcal{L}$ is the set of words accepted by $\mathbb{A}$.

4. A *graph game* is a tuple $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ where $G_\exists$ and $G_\forall$ are disjoint sets, and (with $G := G_\exists + G_\forall$) we have $E \subseteq G^2$, and $Win \subseteq G^\omega$. The set $G$ is called the *board* of the game. A game $\mathbb{G}$ is said to be a parity game if $\mathit{Win}$ is given by a parity function $\Omega : G \longrightarrow \mathbb{N}$ as follows: for all sequences $\pi \in G^\omega$
$$\pi \in \mathit{Win} \iff \max\{\Omega(z) \mid z \in \mathit{Inf}(\pi)\} \text{ is even.}$$
In such case we write $\mathbb{G} = (G_\exists, G_\forall, E, \Omega)$ .

5. A *match* in a game $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ is a sequence $v_0 \dots v_k \in G^* \cup G^\omega$ such that for all $k$ $(v_k, v_{k+1}) \in E$. A match is *full* if either $k = \omega$, or $k$ is finite and there is no $v \in G$ such that $(v_k, v) \in E$. In the latter case, if $v_k$ belongs to a player $J$ in $\mathbb{G}$, we say that $J$ got stuck.

   The winner of a finite full match is the player who does not get stuck. In an infinite (full) match $\exists$ wins if $v_0 \dots v_k \in Win$, otherwise player $\forall$ wins.

6. A *strategy* in a game $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ is a map $\alpha : G^* \longrightarrow G$. A $\mathbb{G}$-match $\pi = v_0 v_1 \dots$ is *$\alpha$-conform*, for a player $J$, if $v_{i+1} = \alpha(v_0 \dots v_i)$ for all $i \geq 0$ such that $v_i \in G_J$. A strategy $\alpha$ is *winning* for a player $J$ if all $\alpha$-conform matches, for $J$, are winning for $J$.

7. A strategy $\alpha$ in a game $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ is a *finite memory strategy* if there is a finite set $M$, called the *memory set*, an element $m_I \in M$ and a map $(\alpha_1, \alpha_2) : G \times M \longrightarrow G \times M$ such that for all matches $v_0 \dots v_k$, in $\mathbb{G}$, there exists and $m_0 \dots m_k \in M^*$ such that $m_0 = m_I$, $m_{i+1} = \alpha_2(v_i, m_i)$ (for all $i < k$), and $\alpha(v_0 \dots v_k) = \alpha_1(v_k, m_k)$.

8. A strategy $\alpha$ is said to be *positional* if there exists a map $\alpha_p : G^* \longrightarrow G$ such that for all $v_0 \dots v_k \in G^*$ we have $\alpha(v_0 \dots v_k) = \alpha_p(v_k)$. We often identify $\alpha$ with $\alpha_p$.

9. A game $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ is called *regular* if there exists an $\omega$-regular language $\mathcal{L}$ over a finite alphabet $C$, and a map $col : G \longrightarrow C$, such that $\mathit{Win} = \{v_0 v_1 \dots \in G^\omega : col(v_0)col(v_1) \dots \in \mathcal{L}\}$.

Regular games always have finite memory strategies.

**Proposition 8.1.2** ([23]). *For each regular game $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ there exists finite memory strategies $\alpha_\exists$ and $\alpha_\forall$, for $\exists$ and $\forall$ respectively, such that for all positions $z$ on the board of $\mathbb{G}$, either $z$ is winning with respect to $\alpha_\exists$ or with respect to $\alpha_\forall$.*

*Moreover, the size of the memory set is bounded by the size of the smallest deterministic parity automaton recognising an $\omega$-regular language associated with the regular game.*

Parity games always have positional strategies.

**Proposition 8.1.3** ([34, 85]). *Let $\mathbb{G} = (G_\exists, G_\forall, E, Win)$ be a parity game. There exists positional strategies $\alpha_\exists$ and $\alpha_\forall$, for $\exists$ and $\forall$ respectively, such that for all positions $z$ on the board of $\mathbb{G}$, either $z$ is winning with respect to $\alpha_\exists$ or with respect to $\alpha_\forall$.*

### Coalgebraic modal fixpoint logic

We can now introduce coalgebraic modal fixpoint logic, also called the coalgebraic $\mu$-calculus. We first introduce some terminology concerning predicate liftings.

**Definition 8.1.4.** Let $\lambda : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$ be a predicate lifting for a functor $\mathcal{T}$.

1. $\lambda$ is said to be monotone if for every pair of sequences of sets $\varphi, \psi : n \longrightarrow \mathcal{P}X$ we have that $(\forall i)(\varphi_i \subseteq \psi_i)$ implies $\lambda(\varphi) \subseteq \lambda(\psi)$.

2. The boolean dual of $\lambda$, denoted by $\lambda^d : \mathcal{P}^n \longrightarrow \mathcal{P}\mathcal{T}$, is the predicate lifting which maps a sequence $\varphi : n \longrightarrow \mathcal{P}$ to $\neg\lambda(\neg\varphi_1, \ldots, \neg\varphi_n)$.

We now define the language of fixpoint modal formulas.

Fix a set $Q$ of variables, a set of monotone predicate liftings $\Lambda$ which is closed under boolean duals.

The set $\mu\mathrm{ML}_\Lambda$ of fixpoint formulas by the following gramar:

$$\varphi ::= q \in Q \mid \bot \mid \top \mid \varphi_0 \wedge \varphi_1 \mid \varphi_0 \vee \varphi_1 \mid \Box_\lambda(\varphi_0, \ldots, \varphi_{ar(\lambda)}) \mid \mu q.\varphi \mid \nu q.\varphi$$

where $\lambda \in \Lambda$. Syntactic notions pertaining to formulas, such as alternation depth, are defined as usual. The size of a formula is defined as its length

The semantics of this language is completely standard. Let $(X, \xi)$ be a $\mathcal{T}$-coalgebra. Given a valuation $V : X \longrightarrow \mathcal{P}(S)$, we define the *meaning* $[\![\varphi]\!]_{(X,\xi),V}$ of a formula $\varphi$ by a standard induction which includes the following clauses:

$$[\![q]\!]_{\xi,V} := V(q), \quad [\![\mu q.\varphi]\!]_{\xi,V} := \mathsf{LFP}.\varphi_q^{\xi,V}, \quad [\![\nu q.\varphi]\!]_{\xi,V} := \mathsf{GFP}.\varphi_q^{\xi,V}.$$

Here $\mathsf{LFP}.\varphi_q^{\xi,V}$ and $\mathsf{GFP}.\varphi_q^{\xi,V}$ are the least and greatest fixpoint, respectively, of the monotone map $\varphi_q^{\xi,V} : \mathcal{P}(X) \longrightarrow \mathcal{P}(X)$ given by $\varphi_q^{\xi,V}(A) := [\![\varphi]\!]_{\xi,V[q\mapsto A]}$ (with $V[q \mapsto A](q) = A$ and $V[q \mapsto A](q') = V(q')$ for $q' \neq q$). For *sentences*, that is, formulas without free variables, the valuation does not matter; we write $\xi, q \Vdash \varphi$ iff $q \in [\![\varphi]\!]_{\xi,V}$ for some/any valuation $V$.

The next remark present another definition of the semantics for the fixpoint operators.

**Remark 8.1.5.** In the definition above, since all operations are monotone the function $\varphi_q^{\xi,V} : \mathcal{P}(X) \to \mathcal{P}(X)$ is monotone. Then by Knaster-Tarski Theorem it has a a least and greatst fixpoint.

However, notice that we can define the by defining the semantics of the fixpoint operators via

$$\llbracket \mu q.\varphi \rrbracket_{(X,\xi),V} = \bigcap \left\{ U \subseteq X \mid U \subseteq \llbracket \varphi \rrbracket_{\xi,V[q\mapsto U]} \right\} \text{ and}$$
$$\llbracket \nu q.\varphi \rrbracket_{(X,\xi),V} = \bigcup \left\{ U \subseteq X \mid \llbracket \varphi \rrbracket_{\xi,V[q\mapsto U]} \subseteq U \right\}.$$

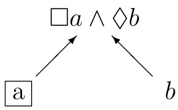
## 8.2 Automata via Predicate liftings

We are now ready for the definition of the key structures of this chapter, $\Lambda$-automata, and their semantics.

**Definition 8.2.1** ($\Lambda$-automata)**.** Let $\Lambda$ be a set of predicate liftings for a functor $\mathcal{T}$ and let $\mathcal{A}$ be a category of power set algebras. Let $\mathbb{L}_\Lambda : \mathsf{Set} \to \mathsf{Set}$ be the functor $UF\Sigma_\Lambda$, where $U : \mathcal{A} \to \mathsf{Set}$ is the forgetful functor, $F$ is its left adjoint, and $\Sigma_\Lambda : \mathsf{Set} \to \mathsf{Set}$ is the functor associated with the modal signature of $\Lambda$.

A $\Lambda$-*automaton* $\mathbb{A}$ is a quadruple $\mathbb{A} = (A, a_I, \rho, \Omega)$, where $A$ is a finite set of states, $a_I \in A$ is the initial state, $\rho : A \to \mathbb{L}_\Lambda(A)$ is the *transition map*, and $\Omega : A \to \mathbb{N}$ is a parity map. The size of $\mathbb{A}$ is defined as the cardinality of $A$, and its index as the size of the range of $\Omega$.

**Example 8.2.2.** Let $A = \{a, b\}$ and let $\Lambda = \{\Box, \Diamond\}$. The following is an example of a $\Lambda$-automata.




where $a_I = a$, an the parity map is given by $\Omega(a) = 1$ and $\Omega(b) = 0$.

In general the transition map of $\Lambda$-automata maps an element $a \in A$ to a rank-1 formula where the formulas inside the modalities are propositional variable in $A$.

The next remark mentions how could we generalize the previous definition to any coalgebraic modal logic.

**Remark 8.2.3.** An idea, due to Y. Venema, to generalize the previous definition to an arbitrary coalgebraic modal logic $(L, \delta)$, is the following: Require the transition map to be a function $A \to ULF(A)$. Since $\mathbb{L}_\Lambda = UF\Sigma_\Lambda$, every $\Lambda$-automaton $\mathbb{A} = (A, a_I, \rho, \Omega)$ induces one of these new automata via the following transition map $\mathbb{L}_\Lambda(\eta_A) \circ \rho : A \to UL_\Lambda F(A) = UF\Sigma_L UF(A)$; recall that $L_\Lambda = F\Sigma_\Lambda U$. We do not peruse this path in this manuscript and stick to $\Lambda$-automata.

**Convention 8.2.4.** For the rest of this chapter we work within standard automata theory. This means we assume the base category of our logic to be DL and a fixed set of monotone predicate liftings for $\mathcal{T}$.

### The acceptance game

To introduce the acceptance of $\Lambda$-automata we will use the one-step semantics, Section 3.2.3. We now recall those definitions in the particular case of a logic of predicate lifitings.

Let $Q$ be a set of propositional variables, $\Lambda$ be a set of predicate liftings, and $V : Q \longrightarrow UP(X)$ a valuation. The *one-step semantics* of depth-one modal formulas over $Q$ (Definition 3.2.19), relative to $V$, written $[\![-]\!]^1_V$, is inductively defined by

$$
\begin{aligned}
[\![\top]\!]^1_V &= \mathcal{T}(X),\\
[\![\bot]\!]^1_V &= \emptyset,\\
[\![\Box_\lambda(q_1, \dots q_{ar(\lambda)})]\!]^1_V &= \lambda_X(V(q_1), \dots, V(q_{ar(\lambda)}))\\
[\![\varphi \wedge \psi]\!]^1_V &= [\![\varphi]\!]^1_V \cap [\![\psi]\!]^1_V,\\
[\![\varphi \vee \psi]\!]^1_V &= [\![\varphi]\!]^1_V \cup [\![\psi]\!]^1_V,\\
[\![\Box_\lambda(\varphi_1, \dots \varphi_{ar(\lambda)})]\!]^1_V &= \lambda_X([\![\varphi_1]\!]^1_V, \dots, [\![\varphi_{ar(\lambda)}]\!]^1_V)
\end{aligned}
$$

Given $t \in \mathcal{T}(X)$ and a depth-one formula $\varphi$, over $Q$, we write $\mathcal{T}(X), t \Vdash^1_V \varphi$ to indicate $t \in [\![\varphi]\!]^1_V$. An important point to remember here is to no coalgebra is needed to define this semantics.

The acceptance game of $\Lambda$-automata is a two players game, noted $\exists$ (Eloise) and $\forall$ (Abelard), which proceeds in *rounds* moving from one basic position in $A \times X$ to another. In each round, at position $(a, x)$ first $\exists$ picks a valuation $V$ that makes the depth-one formula $\delta(a)$ true at $\xi(x)$. Looking at this $V : A \longrightarrow \mathcal{P}(X)$ as a binary relation $\{(a', x') \mid x' \in V(a')\}$ between $A$ and $X$, $\forall$ closes the round by picking an element of this relation.

**Definition 8.2.5** (Acceptance game)**.** Let $(X, \xi)$ be a $\mathcal{T}$-coalgebra and let $\mathbb{A} = (A, a_I, \rho, \Omega)$ be a $\Lambda$-automaton. The associated acceptance game $Acc(\mathbb{A}, \xi)$ is the parity game given by the table below.

| Position | Player | Admissible moves | Priority |
|---|---|---|---|
| $(a, x) \in A \times X$ | $\exists$ | $\{V : A \longrightarrow \mathcal{P}(X) \mid \mathcal{T}(X), \xi(x) \Vdash^1_V \delta(a)\}$ | $\Omega(a)$ |
| $V \in \mathcal{P}(X)^A$ | $\forall$ | $\{(a', x') \mid x' \in V(a')\}$ | 0 |

where $\Vdash^1_V$ is the one-step semantics (Definition 3.2.20). A pointed coalgebra $(\xi, x_0)$ is *accepted* by the automaton $\mathbb{A}$ if the pair $(a_I, x_0)$ is a winning position for player

$\exists$, where the wining condition is given as in Definition 8.1.1 for a parity game. The class of coalgebras accepted by $\mathbb{A}$ is denoted by $Acc(\mathbb{A})$.

As expected, this generalizes the automata-theoretic perspective on the modal $\mu$-calculus [43]. $\Lambda$-automata are the counterpart of the coalgebraic $\mu$-calculus associated with $\Lambda$.

We say that a $\Lambda$-automaton $\mathbb{A}$ is *equivalent* to a sentence $\varphi \in \mu\mathrm{ML}_\Lambda$ if any pointed $\mathcal{T}$-coalgebra $(\xi, x)$ is accepted by $\mathbb{A}$ iff $\xi, x \Vdash \varphi$.

**Proposition 8.2.6.** *There is an effective procedure transforming a sentence $\varphi$ in $\mu\mathrm{ML}_\Lambda$ into an equivalent $\Lambda$-automaton $\mathbb{A}_\varphi$ of size $d.n$, and index $d$; where $n$ is the size and $d$ is the alternation depth of $\varphi$.*

The previous proposition requires our set of predicate liftings to be monotone and closed under boolean duals, see [43].

## 8.3 Bounded model Property

In this section we show that $\mu\mathrm{ML}_\Lambda$ has the small model property. The key tool in our proof is a satisfiability game that characterizes whether the class of pointed coalgebras accepted by a given $\Lambda$-automaton, is empty or not.

**Definition 8.3.1.** Let $A$ be a finite set and $\Omega$ a map from $A$ to $\mathbb{N}$. Given a sequence $R_0 \dots R_k$ in $(\mathcal{P}(A \times A))^*$ the set of *traces through* $R_0 \dots R_k$ is defined as $Tr(R_0 \dots R_k) := \{a_0 \dots a_{k+1} \in A^* \mid (a_i, a_{i+1}) \in R_i \text{ for all } i \leq k.\}$. Similarly $Tr(R_0 R_1 \dots) \subseteq A^\omega$ denotes the set of (infinite) traces through $R_0 R_1 \dots$. With $NBT(A, \Omega)$ we denote the set of $R_0 R_1 \dots \in (\mathcal{P}(A \times A))^\omega$ that contain no *bad trace*, that is, no trace $a_0 a_1 \dots$ such that $\max\{\Omega(a) \mid a \in \mathit{Inf}(a_0 a_1 \dots)\}$ is odd.

We fix some notation before presenting the satisfiabilty game.

**Notation.** Let $\mathbb{A} = (A, a_I, \rho, \Omega)$ be an automaton. Given an element $a \in A$, we write $\varsigma^a$ for the function $\varsigma^a : A \to A \times A$ which maps $b \in A$ to $(a, b)$; given a formula $\varphi \in \mathbb{L}_\Lambda(A)$ we write $\varsigma^a \varphi$ for the formula obtained by applying the substitution $\varsigma^a$ to $\varphi$, more explicitly this is the formula $\varsigma^a \varphi = \mathbb{L}_\Lambda(\varsigma^a)(\varphi)$.

The range of a relation is denoted by $Ran(R)$.

We now illustrate the substitution above with an example. By definition $\mathbb{L}_\Lambda(A) = UF\Sigma_\Lambda(A)$ hence a formula in $\mathbb{L}_\Lambda(A)$ is a rank one formula where the formulas under the scope of a modality are propositional variables in $A$. For example $\Box_\lambda a \vee \Box_{\lambda'} b$. Applying the substitution $\varsigma^a$ to this formula gives $\Box_\lambda (a, a) \vee \Box_{\lambda'} (a, b)$ which is a formula in $\mathbb{L}_\Lambda(A \times A)$.

We now introduce the satisfiability game.

**Definition 8.3.2** (Satisfiability game)**.** Let $U_{\mathcal{R}} : A \times A \longrightarrow \mathcal{P}(\mathcal{R})$ denote the valuation given by $U_{\mathcal{R}}(a,b) = \{R \in \mathcal{R} \mid (a,b) \in R\}$.

The satisfiability game $Sat(\mathbb{A})$ associated with a $\Lambda$-automaton $\mathbb{A} = (A, a_I, \rho, \Omega)$ is the graph game given by the rules of the tableau below.

| Position | Player | Admissible moves |
|---:|:---:|---|
| $R \subseteq A \times A$ | $\exists$ | $\{\mathcal{R} \in \mathcal{PP}(A \times A) \mid [\![\bigwedge\{\varsigma^a\rho(a) \mid a \in Ran(R)\}]\!]^1_{U_{\mathcal{R}}} \neq \emptyset\}$ |
| $\mathcal{R} \in \mathcal{PP}(A \times A)$ | $\forall$ | $\{R \mid R \in \mathcal{R}\}$ |

Unless specified otherwise, we assume $\{(a_I, a_I)\}$ to be the starting position of $Sat(\mathbb{A})$. An infinite match $R_0\mathcal{R}_0R_1\dots$ is winning for $\exists$ if $R_0R_1\dots \in NBT(A,\Omega)$.

We now show that $Sat(\mathbb{A})$ is a regular game. In fact we will show that its winning condition is an $\omega$-regular language $\mathcal{L}$ of which the complement is recognized by a nondeterministic parity stream automaton of size $|A|$ and index $|Ran(\Omega)|$. So by Proposition 8.1.2, $\mathcal{L}$ is recognized by a deterministic parity stream automaton of size exponential in $|A|$ and index polynomial in $|A|$.

**Proposition 8.3.3** ([40])**.** *Let $\mathbb{A} = (A, a_I, \rho, \Omega)$ be a $\Lambda$-automata. The game $Sat(\mathbb{A})$ is a regular game.*

*Proof.* By definition the board of the game is given by $G = \mathcal{P}(A \times A) + \mathcal{PP}(A \times A)$. We want to find a finite alphabet $C$, a colouring $col : G \longrightarrow C$, and a $\omega$-regular language $\mathcal{L}$, over $C$; such that for all infinite sequences $v_0v_1 \dots \in G^\omega$ the sequence $col(v_0)col(v_1)\dots$ belongs to $\mathcal{L}$ iff for all $i < \omega$ we have $v_{2i} \in \mathcal{P}(A \times A)$, $v_{2i+1} \in \mathcal{PP}(A \times A)$, and $v_0v_2v_4\dots$ does not contain a bad trace (Definition 8.3.1).

The alphabet $C$ is given by the set $\mathcal{P}(A \times A) + 1$. Let $*$ be the only element of 1. The colouring $col : G \longrightarrow C$ is defined as follows: for $R \in \mathcal{P}(A \times A)$ we define $col(R) = R$ and $col(x) = *$ for any other $x \in G$.

The language $\mathcal{L}$ is given by the set $\{R_0 * R_1 * \dots \mid R_0R_1\dots \in NBT(A,\Omega)\}$. Clearly for all infinite sequences $v_0v_1 \dots \in G^\omega$ the sequence $col(v_0)col(v_1)\dots$ belongs to $\mathcal{L}$ iff for all $i < \omega$ we have $v_{2i} \in \mathcal{P}(A \times A)$, $v_{2i+1} \in \mathcal{PP}(A \times A)$, and $v_0v_2v_4\dots$ does not contain a bad trace.

It is only left to prove that $\mathcal{L}$ is $\omega$-regular. It is well known [43] that a language is $\omega$-regular iff its complement is $\omega$-regular. We now show the latter. We define a stream automata $\mathbb{B} = (A_B, q_B, \rho_B, \Omega_B)$ as follows: The set $A_B$ is given by $A+A+1$; we write the elements of the second factor by $a_*$, and the only element of 1 by $a_\perp$. The initial state $a_B$ is $a_I$, i.e. the initial state of $\mathbb{A}$. The transition map $\rho_B : A_B \longrightarrow \mathsf{Pow}(A_B)^C$ is given by

$$\begin{aligned} \rho_B(a,R) &= \{a'_* \mid (a,a') \in R\} \\ \rho_B(a_*,*) &= \{a\} \\ \rho_B(x) &= \{a_\perp\} \text{ for any other case.} \end{aligned}$$

The parity map $\Omega_B : A_B \to \mathbb{N}$ is defined by $\Omega_B(a_\perp) = 0$ and $\Omega_B(a_*) = \Omega_B(a) := \Omega(a) + 1$, where $\Omega$ is the parity map of $\mathbb{A}$.

For the previous automata, on any word, over $C$, which is not of the form $R_0 * R_1 * \ldots$ the automata moves to $a_\perp$ and the word is accepted. If the words is of form $R_0 * R_1 * \ldots$ in mach of the parity game corresponding to $\mathbb{B}$, player $\forall$ construct a trace $R_0 R_1 \ldots$. Since we switched the parity of $\mathbb{A}$ by 1, the match is wining for $\exists$ if the trace is bad. In other words, the word is accepted if $R_0 R_1 \ldots$ contains a bad trace, i.e it is in the complement of $\mathcal{L}$. This concludes the proof. □

### The main theorem

**Theorem 8.3.4.** *Given a $\Lambda$-automaton $\mathbb{A} = (A, a_I, \rho, \Omega)$, the following are equivalent.*

1. *There is a coalgebra accepted by $\mathbb{A}$, i.e. $Acc(\mathbb{A})$ is not empty.*
2. *Player $\exists$ has a winning strategy in the game $Sat(\mathbb{A})$.*
3. *$Acc(\mathbb{A})$ contains a finite coalgebra of size exponential in the size of $\mathbb{A}$.*

### The proof of the main theorem (Theorem 8.3.4)

We prove $3 \Rightarrow 1 \Rightarrow 2 \Rightarrow 3$. The implication ($3 \Rightarrow 1$) is immediate from the definitions.

We now prove the implication ($1 \Rightarrow 2$). Let $(\xi, x_0)$ be a pointed coalgebra accepted by $\mathbb{A}$. We will show that $\exists$ has a winning strategy in the non-emptiness game $Sat(\mathbb{A})$.

Before we go into the details we need some terminology and notation. By assumption, player $\exists$ has a winning strategy $\alpha$ in the acceptance game for the automaton $\mathbb{A}$ with starting position $(a_I, x_0)$. Since the acceptance game is a parity game, we may assume this strategy to be positional (Proposition 8.1.3). Given two finite sequences $\vec{x} = x_0 \ldots x_k \in X^*$ and $\vec{a} = a_0 \ldots a_k \in A^*$, we say that $\vec{a}$ *$\alpha$-corresponds to* $\vec{x}$ if there is an $\alpha$-conform match which has basic positions $(a_0, x_0) \ldots (a_k, x_k)$. The set of all sequences in $A^*$ that $\alpha$-correspond to $\vec{x}$ is denoted as $Corr_\alpha(\vec{x})$. Intuitively, this set represents the collection of all $\alpha$-conform matches passing through $\vec{x}$.

The definition of the winning strategy for $\exists$ in the non-emptiness game $Sat(\mathbb{A})$ will be given by induction on the length of partial matches. Simultaneously we will select, through the coalgebra $(X, \xi)$, a path $x_1 x_2 \ldots$, which is related to the $Sat(\mathbb{A})$-match $\pi$ as follows: At each finite stage $R_0 \mathcal{R}_0 R_1 \ldots R_k$ of $\pi$, $R_0 = R_I$,

$$Tr(R_1 \ldots R_k) \subseteq Corr_\alpha(x_0 \ldots x_k)$$
$$\text{and each } a \in Ran(R_k) \text{ occurs in some trace through } R_0 \ldots R_k \qquad (*)$$

This implies in particular that for each element $a \in Ran(R_k)$, the pair $(a, x_k)$ is a winning position for $\exists$ in the acceptance game.

First, we check that when the satisfiability game starts, condition (*) is satisfied. In this case, we have $R_0 = \{(a_I, a_I)\}$ and $\vec{x}$ is the one element sequence $x_0$. It is routine to verify that (*) holds.

For the induction step, assume that in the satisfiability game, the partial match $\vec{R} = R_0\mathcal{R}_0 R_1 \dots R_k$ has been played. First we will provide $\exists$ with an appropriate response $\mathcal{R} \subseteq \mathcal{P}(A \times A)$.

By inductive hypotheses, we have selected a sequence $\vec{x} = x_0 x_1 \dots x_k$ satisfying condition (*). Since $\alpha$ is by assumption a *winning* strategy for $\exists$ in the acceptance game, the pair $(a, x_k)$ is a winning position for $\exists$ for each $a \in Ran(R_k)$. This means that $\exists$'s strategy $\alpha$ will provide her with a collection of valuations $\{V_a : A \to \mathcal{P}(S) \mid a \in ran(R_k)\}$ such that

$$\mathcal{T}(X), \xi(x_k) \Vdash^1_{V_a} \rho(a). \tag{8.1}$$

for all $a \in Ran(R_k)$. The collection $\{V_a \mid a \in Ran(R_k)\}$ induces a map $f_V : X \to \mathcal{P}(A \times A)$ given by

$$f_V(x) := \{(a, b) \in A \times A \mid a \in Ran(R_k) \text{ and } x \in V_a(b)\}.$$

Define $\mathcal{R}_k$ as the image of $X$ under $f_V$, that is,

$$\mathcal{R}_k := f_V[X].$$

Thus we may and will see $f_V$ as a surjective map from $X$ to $\mathcal{R}_k$. We now show that this is a legitimate position.

**Claim.** $\mathcal{R}_k$ is a legitimate move for $\exists$ in $Sat(\mathbb{A})$ at position $R_k$.

*Proof.* To see this, first observe that we can see $f_V$ as a map with codomain $\mathcal{R}_k$ and consequently we have

$$\mathcal{T}(f_V) : \mathcal{T}(X) \to \mathcal{T}(\mathcal{R}_k),$$

and so the object $\mathcal{T}(f_V)\xi(x_k)$ is indeed a member of the set $\mathcal{T}(\mathcal{R}_k)$.

Now, in order to prove that $\exists$ may legitimately play $\mathcal{R}_k$ at $R_k$, it suffices to prove that, for all $a \in Ran(R_k)$:

$$\mathcal{T}(\mathcal{R}_k), \mathcal{T}(h_V)\xi(x_k) \Vdash^1_{U_{\mathcal{R}_k}} \varsigma_a \rho(a). \tag{8.2}$$

Fix $a \in Ran(R_k)$, and abbreviate $U := U_{\mathcal{R}_k}$, where $U_{\mathcal{R}_k}$ is defined as in Definition 8.3.2. By Equation (8.1), it clearly suffices to prove that

$$\mathcal{T}(\mathcal{R}_k), \mathcal{T}(f_V)\xi(x_k) \Vdash^1_U \varsigma_a \varphi \text{ iff } \mathcal{T}(X), \xi(x_k) \Vdash^1_{V_a} \varphi \tag{8.3}$$

for all formulas $\varphi$ in $\mathbb{L}_\Lambda(A)$. We will prove Equation (8.3) by induction on the complexity of $\varphi$.

In the base case we are dealing with a formula $\varphi = \Box_\lambda(b_1, \ldots, b_n)$. For simplicity however we confine ourselves to the (representative) special case where $n = 1$, and write $b = b_1$. In this setting, Equation (8.3) follows from the following chain of equivalences:

$$\begin{aligned}
\mathcal{T}(\mathcal{R}_k), \mathcal{T}(f_V)\xi(x_k) \Vdash^1_U \varsigma_a\varphi, &\iff \mathcal{T}(\mathcal{R}_k), \mathcal{T}(f_V)\xi(x_k) \Vdash^1_U \Box_\lambda(a,b) \\
&\qquad\qquad \text{(definition of } \varsigma_a \text{ and } \varphi) \\
&\iff \mathcal{T}(f_V)\xi(x_k) \in \lambda_{\mathcal{R}_k}[\![(a,b)]\!]_U \quad \text{(definition of } \Vdash^1) \\
&\iff \xi(x_k) \in \mathcal{T}(f_V)^{-1}(\lambda_{\mathcal{R}_k}[\![(a,b)]\!]_U) \\
&\qquad\qquad \text{(definition of } (-)^{-1}) \\
&\iff \xi(x_k) \in \lambda_X f_V^{-1}([\![(a,b)]\!]_U) \quad \text{(naturality of } \lambda) \\
&\iff \xi(x_k) \in \lambda_X([\![b]\!]_{V_a}) \quad (\ddagger) \\
&\iff \mathcal{T}(X), \xi(x_k) \Vdash^1_{V_a} \Box_\lambda b \quad \text{(definition of } \Vdash^1)
\end{aligned}$$

The step marked (‡) follows from the identity $[\![b]\!]_{V_a} = f_V^{-1}([\![(a,b)]\!]_U)$, which follows from the following chain of equivalences, all applying to an arbitrary $x \in X$:

$$\begin{aligned}
x \in [\![b]\!]_{V_a} &\iff x \in V_a(b) && \text{(definition of } [\![-]\!]) \\
&\iff (a,b) \in f_V(x) && \text{(definition of } f_V) \\
&\iff b \in U(f_V(x)) && \text{(definition of } U = U_{\mathcal{R}_k}) \\
&\iff f_V(x) \in [\![(a,b)]\!]_U && \text{(definition of } [\![-]\!]) \\
&\iff x \in f_V^{-1}([\![(a,b)]\!]_U)
\end{aligned}$$

The other inductive steps in the proof of Equation (8.3) are routine, and therefore omitted. This finishes the proof of Equation (8.3), and thus also the proof of the claim. □

Given the legitimacy of $\mathcal{R}_k$ as a move for $\exists$ at position $R_k$, we may propose it as her move in the satisfiability game. This yields the definition of the desired strategy.

Notice that playing this strategy enables $\exists$ to maintain the inductive condition (*). Indeed, by definition of $\mathcal{R}_k$, for every $R \in \mathcal{R}_k$ there is an $x_R \in X$ such that $R = f_V(x_R)$. Hence if $\forall$ picks such a relation $R$, that is putting $R_{k+1} := R$, $\exists$ adds state $x_R$ to her sequence $\vec{x}$, putting $x_{k+1} := x_R$.

To verify that the sequences $R_0 \ldots R_{k+1}$ and $x_0 \ldots x_{k+1}$ satisfy (*), let $a_0 \ldots a_{k+1}$ be a trace through $R_0 \ldots R_{k+1}$. Since $R_0 \ldots R_k$ and $x_0 \ldots x_k$ satisfy (*), there is an $\alpha$-conform match of the form $(a_0, x_0) \ldots (a_k, x_k)$. In this match, when the position $(a_k, x_k)$ is reached, $\exists$ choose a valuation $V_{a_k} : A \longrightarrow \mathcal{P}(X)$ such that $\xi, x_k \Vdash^1_{V_{a_k}} \rho(a_k)$. Then, $\forall$ picks a pair $(x_{k+1}, a_{k+1})$ such that $x_{k+1} \in V_{a_k}(a_{k+1})$. So in order to show that there is a partial $\alpha$-conform match of the form $(a_0, x_0) \ldots (a_{k+1}, x_{k+1})$, it suffices to prove that $x_{k+1} \in V_{a_k}(a_{k+1})$. Recall that $(a_k, a_{k+1}) \in R_{k+1}$. Since $R_{k+1} = f_V(x_{k+1})$, $a_{k+1}$ belongs to $Ran(R_k)$ and $a_{k+1} \in V_a(x_{k+1})$, which finishes the proof that the first part of (*) holds for $R_0 \dots R_{k+1}$ and $x_0 \dots x_{k+1}$.

It remains to show that for all $a \in Ran(R_{k+1})$, $a$ occurs in a trace through $R_0 \dots R_{k+1}$. Fix $a \in ran(R_{k+1})$. So there exists $a_k \in A$ such that $(a_k, a)$ belongs to $R_{k+1}$. Since $R_{k+1} = f_V(x_{k+1})$, $a_k$ belongs to $Ran(R_k)$. Moreover, it follows from the induction hypothesis that if $a \in Ran(R_k)$, there is a sequence $a_{-1} \dots a_{k-1}$ such that $a_{-1}R_0a_0 \dots R_ka_k$. Putting this together with $(a_k, a) \in R_{k+1}$, this finishes to prove that $a$ occurs in a trace through $R_0 \dots R_{k+1}$.

Finally we show why this strategy is winning for her in the game $Sat(\mathbb{A})$, initiated at $\{(a_I, a_I)\}$. Consider an arbitrary match of this game, where $\exists$ plays the strategy as defined above. First, suppose that this match is finite. From our definition of $\exists$'s strategy in $Sat(\mathbb{A})$ she never gets stuck. So if the match is finite, means that $\forall$ could not play and $\exists$ wins.

In case the match is infinite, $\exists$ has constructed an infinite sequence $\vec{x} = x_0x_1x_2 \dots$ corresponding to the infinite sequence $\vec{R} = R_0R_1R_2 \dots$ induced by the $Sat(\mathbb{A})$-match. Since the relation (*) holds at each finite level, for every infinite trace $a_0a_1a_2 \dots$ through $\vec{R}$ there is an $\alpha$-conform infinite match of the acceptance game on $\mathbb{S}$ with basic positions $(a_0, x_0)(a_1, x_1) \dots$ Since $\alpha$ was assumed to be a winning strategy, none of these traces is bad. In other words, the sequence $\vec{R}$ satisfies the winning condition of $Sat(\mathbb{A})$ for $\exists$, and thus she is declared to be the winner of the $Sat(\mathbb{A})$-match. Since we considered an arbitrary match in which she is playing the given strategy, this shows that this strategy is winning, and thus finishes the proof of the implication $(1 \Rightarrow 2)$.

We now focus on the implication $(2 \Rightarrow 3)$.

Suppose that $\exists$ has a winning strategy in the game $Sat(\mathbb{A}) = (G_\exists, G_\forall, E, Win)$. Since $Sat(\mathbb{A})$ is a regular game, by Proposition 8.1.2, we may assume this strategy to only use finite memory. More concretely, this means that there is a finite set $M$, $m_I \in M$, and maps $\alpha_1 : G_\exists \times M \longrightarrow G$ and $\alpha_2 : G_\exists \times M \longrightarrow M$ which satisfy the conditions of Definition 8.1.1(3). Moreover, the size of $M$ is at most exponential in the size of $\mathbb{A}$. Without loss of generality, we may assume that for all $(R, m) \in G_\exists \times M$, $\alpha_2(R, m) = m$.

We denote by $W_\exists$ the set of pairs $(R, m) \in G_\exists \times M$ satisfying the following: For all $Sat(\mathbb{A})$-matches $R_0\mathcal{R}_0R_1\mathcal{R}_1 \dots$ for which there exists a sequence $m_0m_1 \dots$ with $R_0 = R, m_0 = m$ and for all $i \in \mathbb{N}$, $\mathcal{R}_i = \alpha_1(R_i, m_i)$, $m_{i+1} = \alpha_2(R_i, m_i)$, we have that $R_0\mathcal{R}_0R_1\mathcal{R}_1 \dots$ is won by $\exists$.

The finite coalgebra in $L(\mathbb{A})$ that we are looking for will be given by a map

$$\xi : G_\exists \times M \longrightarrow \mathcal{T}(G_\exists \times M).$$

We base this construction on two observations.

First, let $(R, m)$ be an element of $W_\exists$, and write $\mathcal{R} := \alpha_1(R, m)$; then by the rules of the satisfiability game, there is an element $g(R, m) \in \mathcal{T}(\mathcal{R})$ such that for every $a \in Ran(R)$, the formula $\varsigma^a \rho(a)$ is true at $g(R, m)$ under the valuation $U_\mathcal{R}$. Note that $\mathcal{R} \subseteq G_\exists$, and thus we may think of the above as defining a function $g : W_\exists \longrightarrow \mathcal{T}(G_\exists)$. Choosing some dummy values for elements $(R, m) \in (G_\exists \times M) \setminus W_\exists$, the domain of this function can be extended to the full set $G_\exists \times M$. To simplify our notation we will also let $g$ denote the resulting map, with domain $G_\exists \times M$ and codomain $\mathcal{T}(G_\exists)$.

Second, consider the map $add_m : G_\exists \longrightarrow G_\exists \times M$, given by $add_m(R) = (R, m)$. Based on this map we define the function $h : \mathcal{T}(G_\exists) \times M \longrightarrow \mathcal{T}(G_\exists \times M)$ such that $h(\tau, m) = \mathcal{T}(add_m)(\tau)$. Notice that this is the exponential transpose of the strength defined on Chapter 7, page 185.

We let the coalgebra be $\xi : G_\exists \times M \longrightarrow \mathcal{T}(G_\exists \times M)$ where $\xi$ is the map $\xi := h \circ (g, \alpha_2)$. Observe that the size of $(G_\exists \times E, \xi)$ is at most exponential in the size of $\mathbb{A}$, since $G_\exists$ is the set $\mathcal{P}(A \times A)$ and $M$ is at most exponential in the size of $A$. As the designated point of $\xi$ we take the pair $(R_I, m_I)$, where $R_I := \{(a_I, a_I)\}$.

It is left to prove that the pointed coalgebra $(\xi, (R_I, m_I))$ is accepted by $\mathbb{A}$. That is, using $\exists$'s winning strategy $\alpha$ in the satisfiability game we need to find a winning strategy for $\exists$ in the acceptance game for the automaton $\mathbb{A}$ with starting position $(a_I, (R_I, m_I))$. We will define this strategy by induction on the length of a partial match, simultaneously setting up a shadow match of the satisfiability game. Inductively we maintain the following relation between the two matches:

(*) If $(a_0, (R_0, m_0)), \ldots, (a_k, (R_k, m_k))$ is a partial match of the acceptance game (during which $\exists$ plays the inductively defined strategy), then $a_I a_0 \ldots a_k$ is a trace through $R_0 \ldots R_k$ (and so in particular, $a_k$ belongs to $Ran(R_k)$),

(**) and for all $i \in \{0, \ldots, k-1\}$, $R_{i+1} \in \alpha_1(R_i, m_i)$ and $m_{i+1} = \alpha_2(R_i, m_i)$.

Setting up the induction, it is clear that condition (*) is met at the start $(a_0, (R_0, m_0)) = (a_I, (R_I, m_I))$ of the acceptance match; indeed, $a_I a_I$ is the (unique) trace through the one element sequence $R_I$. Condition (**) holds vacuously.

Inductively assume that, with $\exists$ playing as prescribed, the play of the acceptance game has reached position $(a_k, (R_k, m_k))$. By the induction hypothesis, we have $a_k \in Ran(R_k)$ and the position $(R_k, m_k)$ is a winning position for $\exists$ in the acceptance game. Abbreviate $\mathcal{R} := \alpha_1(R_k, m_k)$ and $n := \alpha_2(R_k, m_k)$. As the next move for $\exists$ we propose the valuation $V : A \longrightarrow \mathcal{P}(G_\exists \times M)$ such that $V(a) := \{(R, n) \mid (a_k, a) \in R \text{ and } R \in \mathcal{R}\}$. We now show that this gives a legitimate move.

**Claim.** $V$ is a legitimate move at position $(a_k, (R_k, m_k))$.

*Proof.* We need to show that $\mathcal{T}(G_\exists \times M), \xi(R_k, m_k) \Vdash^1_V \rho(a_k)$. Recall that $(R_k, m_k)$ belongs to $W_\exists$. Because of this, the element $\gamma := \xi(R_k, m_k)$ of $\mathcal{T}(\mathcal{R})$ satisfies the formula $\varsigma^{a_k}\rho(a_k)$ under the valuation $U := U_\mathcal{R}$ (where $U_\mathcal{R}$ is defined as in Definition 8.3.2). That is $\mathcal{T}(\mathcal{R}), \gamma \Vdash^1_{U_\mathcal{R}} \varsigma^{a_k}\delta(a_k)$. Gathering all these observations, in order to prove the claim it suffices to show that

$$\mathcal{T}(G_\exists \times M), \xi(R_k, m_k) \Vdash^1_V \varphi \text{ iff } \mathcal{T}(\mathcal{R}), \gamma \Vdash^1_U \varsigma^{a_k}\varphi \tag{8.4}$$

for all formulas $\varphi$ in $\mathbb{L}_\Lambda(A)$. The proof of Equation (8.4) proceeds by induction on the complexity of $\varphi$. We only consider a simplified version of the base step, where $\varphi$ is of the form $\Box_\lambda a$. We can now prove Equation (8.4) as follows (recall that $n = \alpha_2(R_k, m_k)$):

$$\begin{aligned}
\mathcal{T}(G_\exists \times M), \xi(R_k, m_k) \Vdash^1_V \Box_\lambda b &\iff \xi(R_k, m_k) \in \lambda_{G_\exists \times M}([\![b]\!]_V). \\
&\qquad\qquad\qquad\qquad\qquad \text{(definition of } \Vdash^1) \\
&\iff \mathcal{T}(add_n)(\gamma) \in \lambda_{G_\exists \times M}([\![b]\!]_V). \\
&\qquad\qquad\qquad\qquad\qquad \text{(definition of } \xi) \\
&\iff \gamma \in \mathcal{T}(add_n)^{-1}(\lambda_{G_\exists \times M}[\![b]\!]_V) \\
&\qquad\qquad\qquad\qquad\qquad \text{(definition of } (\cdot)^{-1}) \\
&\iff \gamma \in \lambda_{G_\exists}(add_n^{-1}([\![b]\!]_V)) \quad \text{(naturality of } \lambda) \\
&\iff \gamma \in \lambda_\mathcal{R}([\![(a_k, b)]\!]_U) \qquad (\ddagger) \\
&\iff \mathcal{T}(\mathcal{R}), \gamma \Vdash^1_U \Box_\lambda(a_k, b) \quad \text{(definition of } \Vdash^1) \\
&\iff \mathcal{T}(\mathcal{R}), \gamma \Vdash^1_U \varsigma^{a_k}\Box_\lambda b \quad \text{(definition of } \varsigma^{a_k})
\end{aligned}$$

For (‡), consider the following valuation $U' : A \times A \to \mathcal{P}(G_\exists)$ such that

$$U'(a', b') := U(a', b') \cap \mathcal{R}.$$

It follows from $\mathcal{R} \subseteq G_\exists$ and standardness that $\lambda_\mathcal{R}[\![a]\!]_U = \lambda_{G_\exists}[\![a]\!]_{U'}$. But then (‡) follows because $add_n^{-1}([\![b]\!]_V) = [\![(a, b)]\!]_{U'}$, which holds by a relatively routine proof. This finishes the proof of the Claim. □

It is now straightforward to show that with this definition of a strategy for ∃, the inductive hypothesis (including the relation (*) between the two matches) remains true. In particular this shows that ∃ will never get stuck. Hence in order to verify that the strategy is winning for ∃, we may confine our attention to infinite matches of $Acc(\mathbb{A}, \xi)$. Let $\pi = (a_0, (R_0, m_0))(a_1, (R_1, m_1)) \ldots$ be such a match, then it follows from (*) that $a_I a_0 a_1 \ldots$ is a trace through $R_0 R_1 \ldots$, and so we may infer from the assumption that $(\alpha_1, \alpha_2)$ is a winning strategy for ∃ in $Sat(\mathbb{A})$, that $a_I a_0 a_1 \ldots$ is not bad. This means that the match $\pi$ is won by ∃. This concludes the proof of the Theorem.

Putting this theorem together with Proposition 8.2.6 and Proposition 8.3.3, we obtain a small model property for the coalgebraic $\mu$-calculus, for *every* set of predicate liftings.

**Corollary 8.3.5.** *If $\varphi \in \mu\mathrm{ML}_{\mathcal{L}}$ is satisfiable in a $\mathcal{T}$-coalgebra, it is satisfiable in a $\mathcal{T}$-coalgebra of size exponential in the size of $\varphi$.*

## 8.4 Conclusions

In this chapter we introduced $\Lambda$-automata which are automata using predicate liftings (Definition 8.2.1). We generalize [108] in that our presentation works for any type of coalgebra i.e. no restriction on the functor.

We introduced an acceptance game (Definition 8.2.5) for $\Lambda$-automata, and established a finite model property (Theorem 8.3.4) using a satisfiability game (Definition 8.3.2) for $\Lambda$-automata. The result here and in [27] give evidence that there should be a general framework to define acceptance and satisfiability games. Notice that the game here is a direct translation of the game for the Moss logic in [108] into the language of predicate liftings. More research in this subject is needed.

## Chapter 9

# Beyond the Stone Age

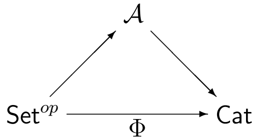
In this thesis we have studied modal logic from a coalgebraic perspective.

In Part I we used a functorial framework, also called the Stone duality approach, to investigate modal logics for coalgebras.

In Chapter 3 we introduced the functorial framework for modal logics as a natural generalisation of the usual modal similarity types, or predicate liftings. We illustrated how the functorial framework for modal logics can be developed by simply considering a "predicate" functor $P : \mathsf{Set}^{op} \longrightarrow \mathcal{A}$, where $\mathcal{A}$ is a category of power set algebras (Definition 3.2.12). In the usual Stone Duality approach, $\mathcal{A}$ is assumed to be the category of Boolean algebras, or even the category of complete atomic Boolean algebras. We showed that using $\mathsf{BA}$ is not essential to develop modal logics for coalgebras. It is our claim that replacing the category $\mathsf{BA}$ by another category of algebras is desirable, and needed, to account for different base logics which lead to interesting variations of basic modal logic, e.g. monotone modal logic (Example 3.2.17). In summary, in this chapter we have illustrated that, up to large extent, the functorial modal logics does not depend on a given "Stone like" adjunction. Some of the points where having and adjunction matters are sketched in Section 3.2.3. One point where the adjunction is essential is in the first representation theorem (Theorem 5.2.2), in Chapter 5, which says that every coalgebraic modal logic can be translated into the language of predicate liftings.

From the categorical perspective, once we change the category $\mathsf{BA}$ as a base for algebras it is quite natural to also change the category $\mathsf{Set}$ as abase for coalgebras. For example, if we take $\mathcal{A} = \mathsf{DL}$ we would like to consider coalgebras over $\mathsf{Pos}$ instead of over $\mathsf{Set}$. This approach will also require a slight variation of the concept of predicate lifting. More precisely, we ought then to replace the functor $\mathcal{P} : \mathsf{Set}^{op} \longrightarrow \mathsf{Set}$ by an appropriate functor, e.g. a functor $\mathsf{Pos}^{op} \longrightarrow \mathsf{Pos}$ in the previous case. It is not totally clear to us how to chose such functor in general and up to what extent such abstraction step is worth for modelling systems; clearly this is an interesting theoretical development to investigate.

The approach to coalgebraic modal logics using just a functor $P : \mathsf{Set}^{op} \longrightarrow \mathcal{A}$ resembles recent insights in [55] where modalities are studies using so-called fibrations. With this perspective the essential element is a functor $\Phi : \mathsf{Set}^{op} \longrightarrow \mathsf{Cat}$, where $\mathsf{Cat}$ denotes the category of all (locally small) categories. Modalities are then obtained as liftings of functors to the Grothendieck construction of $\Phi$. We do not go into the details here but we remark that this approach also, in some sense, captures the intuition *a modality transforms properties of states into properties of successors*. The relationship of this approach with our perspective here should be developed further. Our conjecture is that the functorial framework corresponds to fibrations $\Phi : \mathsf{Set}^{op} \longrightarrow \mathsf{Cat}$ which can be factored via




where $\mathcal{A}$ is an algebraic category, over $\mathsf{Set}$. More research on the subject is needed.

One of the most important conceptual contributions of this thesis is the use of the structural properties of the base category, of the logic, to study coalgebraic modal logics. The most illustrative example can be found in Section 4.4 where we show that every translator can be extended into a $\mathsf{BA}$-translator. In other words, over $\mathsf{BA}$, "every" predicate liftings can be translated into the Moss logic. We showed this using the finite presentability of $\mathsf{BA}$ and the fact that every finitely generated algebra is a power set algebra. Another illustration can be found in Section 3.3.2 where we use the properties of the base category to present the logic of all predicate liftings. Yet another evidence of the power of this technique can be seen in the representation theorems in Chapter 5. A similar technique can be found in Chapter 6 where we use the structural properties of the base category, of coalgebras, to study various formalisation of behavioural equivalence.

In Chapter 4 we introduced the basic translation techniques to compare coalgebraic modal logics. More precisely, we introduced the notion of *one-step translation* (Definition 4.1.1). In order to translate predicate liftings into the Moss logic we developed the notion of *translator* (Definition 4.2.1). The intuition behind a translator is a simple semantic translation of the form $\Box_\lambda \varphi = \nabla tr(\varphi)$. As we showed not all predicate liftings have such simple transaltions (Example 4.2.4), in fact some predicate liftings can simply not be translated into the Moss Logic by finitary means (Example 4.4.6). In order to overcome these difficulties, we introduced the so-called *singleton lifting* (Definition 4.2.5). Singleton liftings are the simplest kind of predicate liftings. Among their properties we highlight 1) they generate all other predicate liftings and 2) every singleton lifting has a translator. To be able to define translations we developed the notion of *logical* *translator* (Definition 4.3.2). Using the structural properties of the category BA we showed that every translator induces a one-step translation. We also showed how using the structural properties of the base category we can translate the Moss logic (Theorem 4.4.8). Moreover, we presented a Lindström like theorem for coalgebraic logics (Theorem 4.4.9). We gave conditions on the functor for a translation between Moss logic and logics of predicate liftings to exist.

In Chapter 5 we used presentations of functors to define predicate liftings. Using presentations of Set functors we introduced the Moss liftings, Section 5.1.1. These predicate liftings are distinguished among all predicate liftings because they are always translatable into Moss logic. We Illustrated this use of presentations by 1) introducing an equational coalgebraic modal logic (Section 5.3) 2) introducing the canonical signature of a coalgebraic modal logic and 3) showed two representation theorems (Theorems 5.2.2 and 5.2.17) which show that every coalgebraic modal logic is a logic of predicate liftings.

The representation theorems in Chapter 5 draw a boundary for the functorial framework and the use of duality. Namely, they show that such framework can not do more than rank 1 axiomatizations of logics predicate liftings. This is not per-se a shortcoming. However, some modal logics will fall out of the scope of the functorial framework e.g logics with fix points. These limitative results also suggest that *Dynamic Epistemic Logic* can not be subsumed within the functorial framework. One reason for this is that the semantics of public announcement, or product updates in general, involves more that one model (coalgebra). It is not clear to us how predicate liftings of functors over Set can account for this.

In Part II we investigated the uses of coalgebraic modal logics further.

In Chapter 6 we studied three ways to express behavioural equivalence of coalgebra states:

1. using final coalgebras,
2. using coalgebraic languages that have the Hennessy-Milner property,
3. using coalgebraic languages that have logical congruences.

We provided a simple proof for the fact that these three different methods are equivalent when used to express behavioural equivalence between set coalgebras. An important point here is that no conditions on the signature functor are required. Our argument relies only on the structural properties of the base category, this is in harmony with the work on Part I where we used the structural properties of the base category of coalgebraic logics to define and compare them. We used this to present six versions of the Hennessy-Milner property (Theorem 6.4.1). An important finding from our work is that it show the Hennessy-Milner property as a solution set condition to construct final coalgebras. We used these characterisations to investigate the situation beyond the category Set. Our main result here is that the constructions on Set lift to regularly algebraic categories.

An important insight from this chapter is to show that the use of categories of power set algebras for coalgebraic modal logics is not ad-hoc. We showed (Example 6.1.3) how categories of power set algebras are precisely the right level of generality to, 1), accommodate the language given by a coalgebraic modal logic in the tradition of abstract model theory where languages are sets and theories belong to the powerset of the language, and 2) still give enough freedom to develop various modal logics.

In Chapter 7 we developed a coalgebraic framework which covers Dynamic logics like (test free) PDL and GL. We illustrated how the process of labelling modalities can be described by a generic process independent of any structure on the labels. We have shown with Theorems 7.3.7 and 7.3.18 that the usual axioms for PDL and GL present a property of the associated predicate liftings not of labelled modalities itself.

Contrary to Chapter 6, Chapter 7 shows the limitations of the framework using categories of power set algebras. More concretely, Lemma 7.3.19 shows that using categories of power set algebras we can only use boolean operations for axioms that preserve the rank of the modal formulas. This shows a limitation of the functorial approach to develop dynamic logics further. As shown in [55] this could perhaps be overcome by switching to an approach using fibrations.

In Chapter 8 we introduced $\Lambda$-automata which are automata using predicate liftings (Definition 8.2.1). This generalizes the work for the Moss logic in [108]. In particular, our presentation works for any type of coalgebra i.e. no restriction on the functor.

We introduced an acceptance game (Definition 8.2.5) for $\Lambda$-automata, and established a bounded model property (Theorem 8.3.4) using a satisfiability game (Definition 8.3.2) for $\Lambda$-automata.

# Appendix A

# Some definitions from Category Theory

We assume the reader to be familiar with the basics of category theory as are: category, functor, natural transformation and adjunction. In this chapter we provide some definitions and results that are used in the manuscript.

**Definition A.0.1.** Consider a functor $\mathcal{T} : \mathbb{C} \longrightarrow \mathbb{D}$ and the following diagrams in $\mathbb{C}$ and $\mathbb{D}$, respectively:

$$\begin{array}{ccc} A & \xrightarrow{\ p\ } & B \\ {\scriptstyle q}\downarrow & & \downarrow{\scriptstyle f} \\ C & \xrightarrow[\ g\ ]{} & D \end{array} \qquad\qquad \begin{array}{ccc} \mathcal{T}(A) & \xrightarrow{\ \mathcal{T}(p)\ } & \mathcal{T}(B) \\ {\scriptstyle \mathcal{T}(q)}\downarrow & & \downarrow{\scriptstyle \mathcal{T}(f)} \\ \mathcal{T}(C) & \xrightarrow[\ \mathcal{T}(g)\ ]{} & \mathcal{T}(D) \end{array}$$

The functor $\mathcal{T}$ is said to preserve weak-pullbacks if the diagram on the right is a weak-pullback, in $\mathbb{D}$, whenever the diagram on the left is a weak-pullback, in $\mathbb{C}$.

**Definition A.0.2.** Let $\mathbb{C}$ be a category.

1. A *concrete category* over $\mathbb{C}$ is a pair $(\mathcal{A}, U)$, where $\mathcal{A}$ is a category and $U : \mathcal{A} \longrightarrow \mathbb{C}$ is a faithful functor. The functor $U$ is called the *forgetful functor* of the concrete category.

2. Let $(\mathcal{A}, U)$ and $(\mathcal{B}, V)$ be concrete categories over $\mathbb{C}$. A *concrete functor* from $(\mathcal{A}, U)$ to $(\mathcal{B}, V)$ is a functor $F : \mathcal{A} \longrightarrow \mathcal{B}$ such that $U = V \circ F$. We denote such a functor by $F : (\mathcal{A}, U) \longrightarrow (\mathcal{B}, V)$.

3. A *concrete isomorphism* is a concrete functor which is an isomorphism. A *concrete equivalence* is a concrete functor which is an equivalence.

**Definition A.0.3.** Let $\mathcal{A}$ be a full subcategory of a category $\mathbb{C}$ and let $X$ be an object of $\mathbb{C}$. The *comma category* over $X$, written $\mathcal{A} \downarrow X$, is the category of all

$\mathbb{C}$-arrows $f : A \longrightarrow X$, where $A$ is an object in $\mathcal{A}$. The $\mathcal{A} \downarrow X$ morphisms from $f : A \longrightarrow X$ to $f' : A' \longrightarrow X$ are the $\mathbb{C}$-morphism $h : A \longrightarrow A'$ such that $f'h = f$. The composition of morphism is that of $\mathbb{C}$.

**Definition A.0.4.** By a *diagram* on a category $\mathbb{C}$ is meant a functor $D : \mathcal{D} \longrightarrow \mathbb{C}$ from a (small) category $\mathcal{D}$. The category $\mathcal{D}$ is called the schema of the diagram.

**Definition A.0.5.** A *directed diagram*, on $\mathbb{C}$, is a functor $D : (I, \leq) \longrightarrow \mathbb{C}$, where $(I, \leq)$ is a directed poset (considered as a category). A *directed colimit* is a colimit of a directed diagram.

In the category $\mathsf{Set}$ directed colimits can be concretely computed as follows:

**Proposition A.0.6.** *Let $D : (I, \leq) \longrightarrow \mathsf{Set}$ be a directed diagram. A colimit for $D$ is given by quotienting the coproduct $\left(\coprod_{i \in I} D(i)\right)$ with the following equivalence relation: $x \in D(i)$ and $y \in D(j)$ are equivalent if there exists $k \in I$ and morphisms $f_i : i \longrightarrow k$ and $f_j : j \longrightarrow k$ such that $D(f_i)(x) = D(f_j)(y)$.*

**Definition A.0.7.** Let $\mathbb{C}$ be a category. Let $\{X_i \rightrightarrows Y \mid i \in I\}$ be a family of pairs of parallel arrows in $\mathbb{C}$. A *join coequalizer* of the family is an arrow $q : Y \longrightarrow E$, in $\mathbb{C}$, which is an coequaliser for each pair in the family.

## A.1 Adjunctions

We first introduce the notion of adjoint functor.

**Definition A.1.1.** A functor $F : \mathbb{C} \longrightarrow \mathbb{D}$ is said to be a *left adjoint* to a functor $U : \mathbb{D} \longrightarrow \mathbb{C}$, written $F \dashv U$, if for every pair of objects $C$ and $D$, in $\mathbb{C}$ and $\mathbb{D}$ respectively, there is a bijection

$$\varphi_{(C,D)} : \mathbb{D}\big(F(C), D\big) \rightarrow \mathbb{C}\big(C, U(D)\big).$$

which is natural in $C$ and $D$. The functor $U$ is called the right adjoint.

This is esquematically presented as follows:

$$\frac{F(C) \longrightarrow D}{C \longrightarrow U(D)}.$$

This means that there is an arrow, in $\mathbb{D}$, with the functor $F$ on the left iff there is an arrow, in $\mathbb{C}$, with the functor $U$ on the right. We can also describe adjunctions using, so called, universal properties and triangular equalities. The next lemma presents the equivalences formally.

**Lemma A.1.2.** *For a pair of functors $F : \mathbb{C} \longrightarrow \mathbb{D}$ and $U : \mathbb{D} \longrightarrow \mathbb{C}$ the following conditions are equivalent.*

1. *There is a natural bijection* $\varphi : \mathbb{D}(F, -) \longrightarrow \mathbb{C}(-, U)$, *i.e.* $F \dashv U$.

2. *There exists a natural transformation* $\eta : \mathsf{Id}_{\mathbb{C}} \longrightarrow UF$ *satisfying the following universal property:* For each arrow $f : C \longrightarrow U(D)$, in $\mathbb{C}$, there exists a unique arrow $\widehat{f} : F(C) \longrightarrow D$, in $\mathbb{D}$, such that $U(\widehat{f}) \circ \eta_C = f$.

3. *There exists a natural transformation* $\varepsilon : FU \longrightarrow \mathsf{Id}_{\mathbb{D}}$ *satisfying the following universal property:* For each arrow $g : F(C) \longrightarrow D$, in $\mathbb{D}$, there exists a unique arrow $\widehat{g} : C \longrightarrow U(D)$, in $\mathbb{C}$, such that $\varepsilon_D \circ F(\widehat{g}) = g$.

4. *There exists natural transformations* $\eta : \mathsf{Id}_{\mathbb{C}} \longrightarrow UF$ *and* $\varepsilon : FU \longrightarrow \mathsf{Id}_{\mathbb{D}}$ *satisfying the following triangular equalities* $\varepsilon_F \circ F(\eta) = id_F$ *and* $\eta_U \circ U(\varepsilon) = id_U$.

We refer the reader to a standard book on Category Theory, e.g. [6], for the details. The important part to remember is that the isomorphism $\varphi$, and the natural transformations $\eta$ and $\varepsilon$ are interdefinable; we now sketch how this is done:

For every pair of arrows $g : F(C) \longrightarrow D$ and $f : C \longrightarrow U(D)$ we can describe the arrows $\widehat{f}$ and $\widehat{g}$, in the previous lemma, as follows.

$$\varphi(g) = \widehat{g} = U(g) \circ \eta_C \text{ and } \varphi^{-1}(f) = \widehat{f} = \varepsilon_D \circ F(f).$$

The natural transformations $\eta$ and $\varepsilon$ can be obtained from the isomorphism $\varphi$ as follows

$$\eta_C = \varphi^{-1}(id_{F(C)}) \text{ and } \varepsilon_D = \varphi(id_{U(D)}).$$

We now introduce some notation an terminology concerning adjunctions.

**Notation.** Let $F : \mathbb{C} \longrightarrow \mathbb{D}$ and $U : \mathbb{D} \longrightarrow \mathbb{C}$ be functors. Let $\varphi : \mathbb{D}(F, -) \longrightarrow \mathbb{C}(-, U)$, $\eta : \mathsf{Id}_{\mathbb{C}} \longrightarrow UF$, and $\varepsilon : FU \longrightarrow \mathsf{Id}_{\mathbb{D}}$, be natural transformations, inter defined, as in the previous lemma. We follow the following conventions:

- We call the tuple $(F, U, \varphi, \eta, \varepsilon)$ an *adjoint situation*. The first component is always the left adjoint.

- The natural transformation $\varphi : \mathbb{D}(F, -) \longrightarrow \mathbb{C}(-, U)$ is called the *isomorphism associated with the adjunction*.

- The natural transformation $\eta : id_{\mathbb{C}} \longrightarrow UF$ is called the *unit of the adjunction*.

- The natural transformation $\varepsilon : FU \longrightarrow id_{\mathbb{D}}$ is called the *counit of the adjunction*.

- The image of an arrow $g : F(C) \longrightarrow D$ under $\varphi$, i.e. $\widehat{g}$, is called the *transpose*, relative to $(F, U, \varphi, \eta, \varepsilon)$, of $g$. The same terminology applies for an arrow $f : C \longrightarrow U(D)$, i.e. $\widehat{f}$ is called the transpose, relative to $(F, U, \varphi, \eta, \varepsilon)$, of $f$.

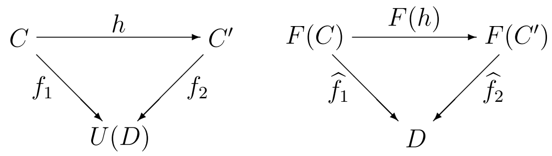
The following properties of adjoints will be used often.

**Lemma A.1.3.** *For any adjoint situation* $(F, U, \varphi, \eta, \varepsilon)$ *the following holds*

1. *The right adjoint preserves limits and the left adjoint preserves colimits.*

2. *The right adjoint* $U$ *is faithful iff each of the components of the counit* $\varepsilon$ *is an epimorphism.*

3. *Assume* $F$ *has domain and codomain as follows* $F : \mathbb{C} \longrightarrow \mathbb{D}$*. For any,* $\mathbb{C}$*-arrow,* $h : C \longrightarrow C'$ *the diagram on the left commutes iff the diagram on the right commutes.*




We refer the reader to a standard book on Category Theory, like [6], for detailed proofs.

# Bibliography

[1] Samson Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51, 1991. Cited on pp. 8, 52.

[2] Peter Aczel. *Non-Well-Founded Sets.* Center for the Study of Language and Information, Stanford, 1988. Cited on pp. 3, 4, 18, 19, 20, 31, 137.

[3] Peter Aczel and Nax Paul Mendler. A final coalgebra theorem. In *Category Theory and Computer Science*, volume 389 of *Lecture Notes In Computer Science*. Springer, 1989. Cited on pp. 3, 151, 158.

[4] Jirí Adámek. Observability and nerode equivalence in concrete c5ategories. In Ferenc Gécseg, editor, *Fundamentals of Computation Theory, FCT'81, Proceedings of the 1981 International FCT-Conference, Szeged, Hungary, August 24-28, 1981*, volume 117 of *Lecture Notes in Computer Science*. Springer, 1981. Cited on p. 3.

[5] Jiri Adámek. A logic of coequations. In C.-H. Luke Ong, editor, *CSL*, volume 3634 of *Lecture Notes in Computer Science*. Springer, 2005. Cited on pp. 143, 149.

[6] Jiří Adámek, Horst Herrlich, and George E. Strecker. *Abstract and Concrete Categories: The Joy of Cats.* John Wiley & Sons, 1990. Cited on pp. 157, 161, 166, 217, 218.

[7] Jiří Adámek and Jiří Rosický. *Locally Presentable and Accessible Categories*, volume 189 of *London Mathematical Society Lecture Notes Series*. Cambridge University Press, 1994. Cited on pp. 18, 25, 27.

[8] Jiří Adámek, Jiří Rosický, and E. M. Vitale. *Algebraic Theories: a Categorical Introduction to General Algebra*, volume 184 of *Cambridge Tracts in Mathematics*. Cambridge University Press, 2011. Cited on pp. 24, 25, 26, 99, 101, 175, 177.

[9] Jiří Adámek and Vera Trnková. *Automata and Algebras in Categories.* Kluwer Academic Publishers, 1990. Cited on pp. 17, 96.

[10] Michael A. Arbib and Ernest G. Manes. Machines in a category. *Journal of Pure and Applied Algebra*, 19, 1980. Cited on p. 3.

[11] Michael A. Arbib and Ernest G. Manes. Parametrized data types do not need highly constrained parameters. *Information and Control*, 52(2):139 – 158, 1982. Cited on p. 3.

[12] Alexandru Baltag. A logic for coalgebraic simulation. *Electronic Notes Theoretical Computer Science*, 33, 2000. Cited on p. 6.

[13] Michael Barr. Relational algebras. In *Reports of the Midwest Category Theory Seminar IV*, volume 137 of *Lecture Notes in Mathematics.* Springer Berlin / Heidelberg, 1970. Cited on p. 22.

[14] Michael Barr. Terminal Coalgebras in Well-Founded Set Theory. *Theoretical Computer Science*, 114(2), 1993. Cited on pp. 158, 159.

[15] Falk Bartels, Ana Sokolova, and Erik de Vink. A hierarchy of probabilistic system types. *Theoretical Computer Science*, 327, 2004. Cited on pp. 18, 31.

[16] Jon Barwise and Lawrence Moss. *Vicious Circles: On the Mathematics of Non-Wellfounded Phenomena.* Center for the Study of Language and Information, Stanford, 1996. Cited on p. 5.

[17] Marta Bílková, Alessandra Palmigiano, and Yde Venema. Proof systems for the coalgebraic cover modality. In Carlos Areces and Rob Goldblatt, editors, *Advances in Modal Logic*, volume 7 of *King's College Publications*, 2008. Cited on pp. 6, 117.

[18] Marta Bílková, Jiri Velebil, and Yde Venema. On monotone modalities and adjointness. *Mathematical Structures in Computer Science*, 21(2), 2011. Cited on p. 6.

[19] Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter. *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning.* Elsevier Science Inc., 2006. Cited on pp. 1, 4, 227.

[20] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2002. Cited on pp. 4, 5, 6, 7, 8, 31, 40, 42, 46, 47, 139, 149, 150.

[21] Marcello Bonsangue and Alexander Kurz. Modal logics from coalgebras over topological spaces. *talk at The International Conference on Algebraic and Topological Methods in Non-Classical Logics.* Tbilisi, Georgia 7 - 11 July 2003. Cited on p. 8.

[22] Marcello Bonsangue and Alexander Kurz. Presenting functors by operations and equations. In Luca Aceto and Anna Inglfsdttir, editors, *Foundations of Software Science and Computation Structures*, volume 3921 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006. Cited on pp. 27, 44, 47, 53, 105, 110, 113.

[23] Julius R. Büchi and L H. Landweber. Solving sequential conditions by finite state strategies. *Transactions of the American Mathematical Society*, 138, 1969. Cited on p. 197.

[24] Stanley Burris and H. P. Sankappanavar. *A Course in Universal Algebra.* Number 78 in Graduate Texts in Mathematics. Springer-Verlag, 1981. Cited on pp. 5, 24, 25.

[25] Georgel Calin, Robert Myers, Dirk Pattinson, and Lutz Schröder. ColoSS: The coalgebraic logic satisfiability solver. *Electronic Notes Theoretical Computer Science*, 231:41–54, 2009. Proc. Methods for Modalities 5 (2007). Cited on p. 7.

[26] Brian Chellas. *Modal logic: an introduction.* Cambridge University press, 1980. Cited on pp. 33, 34, 192.

[27] Corina Cîrstea, Clemens Kupke, and Dirk Pattinson. EXPTIME tableaux for the coalgebraic $\mu$-calculus. In E. Grädel and R. Kahle, editors, *Computer Science Logic 2009*, volume 5771 of *Lecture Notes in Computer Science.* Springer, 2009. Cited on pp. 7, 195, 209.

[28] Corina Cîrstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, and Yde Venema. Modal logics are coalgebraic. *The Computer Journal*, 2009. Cited on pp. 4, 7.

[29] Corina Cîrstea and Dirk Pattinson. Modular construction of modal logics. In *CONCUR*, 2004. Cited on p. 6.

[30] Giovanna D'Agostino and Giacomo Lenzi. On modal $\mu$-calculus with explicit interpolants. *Journal of Applied Logic*, 338, 2006. Cited on p. 6.

[31] Erik de Vink and Jan Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. In *Proceedings of ICALP'97*, volume 1256 of *Lecture Notes in Computer Science.* Springer, 1997. Cited on pp. 19, 21.

[32] Ernst-Erich Doberkat. *Stochastic Relations*. Chapman & Hall/CRC, 2007. Cited on p. 3.

[33] J. Michael Dunn. Positive modal logic. *Studia Logica*, 55(2), 1995. Cited on p. 52.

[34] Ernest A. Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy. In *Proceedings of the 32nd annual symposium on Foundations of computer science*, SFCS '91, 1991. Cited on p. 198.

[35] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Harcourt, Academic Press, 2001. Cited on pp. 72, 188.

[36] Gianluigi Ferrari, Ugo Montanari, and Marco Pistore. Minimizing transition systems for name passing calculi: A co-algebraic formulation. In *FoSSaCS'02*, volume 2303 of *Lecture Notes in Computer Science*, 2002. Cited on p. 3.

[37] Kit Fine. In so many possible worlds. *Notre Dame Journal of formal logic*, (13), 1972. Cited on p. 31.

[38] Marcelo P. Fiore and Daniele Turi. Semantics of name and value passing. In *Proceedings 16th Logic in Computer Science (LICS)*. IEEE Computer Society Press, 2001. Cited on p. 3.

[39] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18, 1979. Cited on p. 32.

[40] Gaëlle Fointaine. *Modal fixpoint logic: some model-theoretic questions*. PhD thesis, Institute for Logic Language and Computation - Universiteit van Amsterdam, 2010. Cited on pp. 196, 202.

[41] Gaëlle Fontaine, Raul Leal, and Yde Venema. Automata for coalgebras: an approach using predicate liftings. In *Proceedings of the 37th international colloquium conference on Automata, languages and programming: Part II*, ICALP'10. Springer-Verlag, 2010. Cited on p. 11.

[42] Robert Goldblatt. Final coalgebras and the Hennessy-Milner property. *Annals of Pure and Applied Logic*, 138(1-3), 2006. Cited on pp. 138, 139, 143, 157.

[43] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logic, and Infinite Games*, volume 2500 of *LNCS*. Springer, 2002. Cited on pp. 196, 201, 202.

[44] H. Peter Gumm. Elements of the general theory of coalgebras. 1999. Cited on p. 151.

[45] Helle Hansen and Clemens Kupke. A coalgebraic perspective on monotone modal logic. volume 106 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2004. Cited on pp. 3, 19, 33.

[46] Helle Hansen and Raul Leal. Dynamic coalgebraic modalities. Cited on p. 11.

[47] David. Harel, Kozen Dexter., and Jerzy. Tiuryn. *Dynamic Logic*. The MIT Press, 2000. Cited on pp. 32, 171, 179.

[48] Ichiro Hasuo, Chris Heunen, Bart Jacobs, and Ana Sokolova. Coalgebraic components in a many-sorted microcosm. In *Proceedings of CALCO*, volume 5728 of *Lecture Notes In Computer Science*, pages 64–80, 2009. Cited on p. 192.

[49] Aviad Heifetz and Philippe Mongin. Probabilistic logic for type spaces. *Games and Economic Behavior*, 35, 2001. Cited on pp. 31, 40, 60.

[50] Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Information and Computation*, 145, 1997. Cited on p. 6.

[51] Bart Jacobs. Towards a duality result in coalgebraic modal logic. *Electr. Notes Theor. Comput. Sci.*, 33, 2000. Cited on p. 8.

[52] Bart Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theoretical Informatics and Applications*, 35(1), 2001. Cited on pp. 6, 8.

[53] Bart Jacobs. Exercises in coalgebraic specification. Springer-Verlag New York, Inc., 2002. Cited on p. 21.

[54] Bart Jacobs. Distributive laws for the coinductive solution of recursive equations. *Information and Computation*, 204, April 2006. Cited on p. 185.

[55] Bart Jacobs. Predicate logic for functors and monads. Preprint, 2010. Cited on pp. 192, 212, 214.

[56] Bart Jacobs. Trace semantics for coalgebras. *Electronic Notes Theoretical Computer Science*, 106, 204. Cited on p. 61.

[57] Bart Jacobs and Erik Poll. Coalgebras and monads in the semantics of Java. *Theoretical Computer Science*, 291, 2003. Cited on pp. 178, 192.

[58] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62, 1997. Cited on pp. 2, 20.

[59] Bart Jacobs and Ana Sokolova. Exemplaric expressivity of modal logics. *Journal of Logic and Computation*, 20(5), 2010. Cited on pp. 56, 57.

[60] David Janin and Igor Walukiewicz. Automata for the modal mu-calculus and related results. In *MFCS'95 Proceedings*. Springer-Verlag, 1995. Cited on p. 6.

[61] Peter Johnstone. Vietoris locales and localic semilattices. In *Continuous Lattices and their Applications*, volume 101 of *Lecture Notes in Pure and Applied Mathematics*. Marcel Dekker, 1985. Cited on p. 52.

[62] Peter Johnstone. The art of pointless thinking: a student's guide to the category of locales. In H. Herrlich and H.-E. Porst, editors, *Category Theory at Work*. Heldermann Verlag, 1991. Cited on p. 30.

[63] Andre Joyal. Foncteurs analytiques et esp'eces de structures. *Lecture Notes in Mathematics*, page 126159, 1986. Cited on p. 133.

[64] Bartek Klin. The least fibred lifting and the expressivity of coalgebraic modal logic. volume 3629 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005. Cited on p. 6.

[65] Bartek Klin. Bialgebraic methods and modal logic in structural operational semantics. *Information and Computation*, 207(2), 2009. Cited on p. 193.

[66] Dexter Kozen. Results on the propositional $\mu$-calculus. *Theoretical Computer Science*, 27(3). Cited on p. 195.

[67] C. Kupke and Y. Venema. Coalgebraic automata theory: basic results. *Logical Methods in Computer Science*, 4, 2008. Cited on p. 195.

[68] Clemens Kupke, Alexander Kurz, and Dirk Pattinson. Algebraic semantics for coalgebraic logics. *Electronics Notes Theoretical Computer Science*, 106, 2004. Cited on pp. 8, 43, 44, 54, 114.

[69] Clemens Kupke, Alexander Kurz, and Yde Venema. Completeness for the coalgebraic cover modality. Manuscript available on request from the authors. An earlier version appeared in the proceedings of AiML2008. Cited on pp. 6, 61, 116, 117, 118, 119, 126, 127, 129.

[70] Clemens Kupke and Raul Leal. Characterising behavioural equivalence: Three sides of one coin. In Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki, editors, *Algebra and Coalgebra in Computer Science*, volume 5728 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009. Cited on p. 11.

[71] Clemens Kupke and Yde Venema. Coalgebraic automata theory: basic results. *Logical Methods in Computer Science*, 4, 2008. Cited on p. 6.

[72] Alexander Kurz. Specifying coalgebras with modal logic. volume 11 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1998. Cited on p. 6.

[73] Alexander Kurz and Raul Leal. Equational coalgebraic logic. *Electronic Notes Theortical Computer Science*, 249, 2009. Cited on p. 11.

[74] Alexander Kurz and Raul Leal. Modalities in the Stone age. *submitted to Theoretical Computer Science*, 249, 2010. Cited on pp. 11, 105.

[75] Alexander Kurz and Jiří Rosický. The goldblatt-thomason theorem for coalgebras. In Till Mossakowski, Ugo Montanari, and Magne Haveraaen, editors, *Algebra and Coalgebra in Computer Science*, volume 4624 of *Lecture Notes in Computer Science*. Springer, 2007. Cited on p. 64.

[76] Alexander Kurz and Jiří Rosický. Strongly complete logics for coalgebras. *Logical Methods in Computer Science*, 2010. Submitted. Cited on pp. 27, 44, 46, 53, 64, 65, 105, 110, 113.

[77] Alexander Kurz and Yde Venema. Coalgebraic Lindström theorems. In L. Beklemishev et alii, editor, *Advances in Modal Logic*, volume 8 of *King's College Publications*, 2010. Cited on p. 93.

[78] William Lawvere. Adjointness in foundations. *Dialectica*, 23, 1969. Cited on p. 8.

[79] Raul Andres Leal. Expressivity of coalgebraic modal language. Master of Logic dissertation series. ILLC-Universiteit van Amsterdam, 2007. Cited on pp. 9, 10.

[80] Raul Andres Leal. Predicate liftings versus nabla modalities. *Electronic Notes in Theoretical Computer Science*, 203(5), 2008. Cited on pp. 9, 11, 77, 78, 79, 104.

[81] Saunders MacLane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5. Cited on p. 157.

[82] Ernest G. Manes. *Algebraic Theories*. Springer, Berlin, 1976. Cited on p. 3.

[83] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1), 1991. Cited on pp. 171, 173, 183, 185, 192.

[84] Lawrence S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96(1-3), 1999. Erratum published Ann.P.Appl.Log 99:241-259, 1999. Cited on pp. 5, 6, 44, 57, 58, 60, 105, 141.

[85] Andrzej Mostowski. Games with forbidden positions. Technical Report 78, University of Gdansk, 1991. Cited on p. 198.

[86] Robert Myers, Dirk Pattinson, and Lutz Schröder. Coalgebraic hybrid logic. In L. de Alfaro, editor, *Proc. FOSSACS 2009*, volume 5504 of *Lecture Notes in Computer Science*. Springer, 2009. Cited on p. 7.

[87] Rohit Parikh. The logic of games and its applications. In *Topics in the Theory of Computation*, number 14 in Annals of Discrete Mathematics. Elsevier, 1985. Cited on pp. 33, 34, 171.

[88] Dirk Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309(1–3), 2003. Cited on pp. 6, 38, 40, 53, 54, 57, 138.

[89] Dirk Pattinson and Lutz Schröder. Eliminability of cut in coalgebraic logics. *Electr. Notes in Theoret. Comp. Sci.*, 203(5), 2008. Cited on p. 7.

[90] Dirk Pattinson and Lutz Schröder. Generic modal cut elimination applied to conditional logics. *Logical Methods in Computer Science*, 7(1), 2011. Cited on p. 7.

[91] M. Pauly. *Logic for Social Software*. PhD thesis, Institute for Logic Language and Computation - Universiteit van Amsterdam, 2001. Cited on pp. 33, 34.

[92] Vaughan Pratt. Semantical considerations on floyd-hoare logic. In *Proceedings of the 17th IEEE Symposium on Foundations of Computer Science*, 1976. Cited on p. 32.

[93] H. Reichel. Behavioural equivalencea unifying concept for initial and final specification methods. In *Proceedings 3rd Hungarian Computer Science Conference*, 1981. Cited on p. 3.

[94] Martin Rößiger. Coalgebras and modal logic. volume 33 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2000. Cited on pp. 6, 15.

[95] Jan Rutten. Automata and coinduction - an exercise in coalgebra. In D. Sangiorigi and R. de Simone, editors, *CONCUR'98*, volume 1466 of *Lecture Notes in Computer Science*. Springer, 1998. Cited on pp. 3, 19.

[96] Jan Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249, 2000. Cited on pp. 2, 3, 7, 19, 20, 137, 147, 151, 192.

[97] Luigi Santocanale and Yde Venema. Completeness for flat modal fixpoint logics. In Nachum Dershowitz and Andrei Voronkov, editors, *LPAR*, volume 4790 of *Lecture Notes in Artificial Inteligence*. Springer, 2007. Cited on p. 6.

[98] Lutz Schröder. A finite model construction for coalgebraic modal logic. *J. Log. Algebr. Program.*, 73(1-2):97–110, 2007. Cited on p. 7.

[99] Lutz Schröder. Expressivity of Coalgebraic Modal Logic: The Limits and Beyond. *Theoretical Computer Science*, 390, 2008. Cited on pp. 6, 62, 138, 141, 155, 156.

[100] Lutz Schröder and Dirk Pattinson. Rank-1 logics are coalgebraic. *Journal of Logic and Computation*. in press. Cited on p. 7.

[101] Lutz Schröder and Dirk Pattinson. How many toes do I have? parthood and number restrictions in description logics. In *Proc. Knowledge Representation 2008*. AAAI Press, 2008. Cited on p. 7.

[102] Lutz Schröder and Dirk Pattinson. Pspace bounds for rank-1 modal logics. *ACM Transaction in Computational Logic*, 10, 2009. Cited on pp. 7, 53.

[103] Lutz Schröder and Dirk Pattinson. Coalgebraic correspondence theory. In *FOSSACS*, 2010. Cited on p. 7.

[104] Daniel Schwencke. Coequational Logic for Finitary Functors. *Electronic Notes in Theoretical Computer Science*, 203(5), 2008. Cited on pp. 143, 149.

[105] Jech Thomas. *Set theory*. Springer, 2000. Third Millennium edition revised and expanded. Cited on p. 158.

[106] Yde Venema. Automata and fixed point logics: a coalgebraic perspective. *Electronic Notes in Theoretical Computer Science*, 204, 2004. Cited on p. 6.

[107] Yde Venema. *Algebras and Coalgebras*. Volume 3 of *Studies in Logic and Practical Reasoning* [19], 2006. Cited on p. 8.

[108] Yde Venema. Automata and fixed point logic: a coalgebraic perspective. *Information and Computation*, 204, 2006. Cited on pp. 6, 58, 195, 196, 209, 214.

[109] Steven J. Vickers. *Topology Via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989. Cited on p. 52.

[110] Albert Visser and Giovanna D'Agostino. Finality regained: A coalgebraic study of Scott-sets and multisets. *Archive for Mathematical Logic*, 41, 2002. Cited on pp. 19, 31.

[111] Mitchell Wand. Final algebra semantics and data type extensions. *Journal of Computer and System Sciences*, 19(1):27 – 44, 1979. Cited on p. 3.

[112] James Worrell. Terminal sequences for accessible endofunctors. *Electronic Notes Theoretical Computer Science*, 19, 1999. Cited on p. 138.

# Index

## Symbols



## A

## B



## C

# Samenvatting

Dit proefschrift gaat over coalgebra's en modale logica's.

Simpel gezegd zijn coalgebra's machines gezien vanuit het perspectief van de gebruiker. Iets formeler gesproken kunnen we stellen dat coalgebra's de basis vormen voor een wiskundige theorie van computersystemen. Met coalgebra's kunnen we systemen bestuderen waartoe we maar beperkt toegang hebben, of waarvan de toestanden niet volledig bekend zijn. Dit heet het black-box perspectief. Een concreet voorbeeld van dit perspectief is onze interactie met een koffiezetapparaat. De meeste mensen weten niet hoe het binnenste van een koffiezetapparaat eruitziet, laat staan dat ze weten hoe dat binnenste werkt. Toch weten de meeste mensen hoe ze koffiezetapparaat moeten bedienen. De gebruiker ervaart het binnenste van de machine als een black-box. Het koffiezetapparaat is een coalgebra.

Modale logica's bieden een interne, lokale kijk op relationele structuren. De oorsprong ervan ligt in de wijsbegeerte. Modale logica's is begonnen als de studie van logische eigenschappen van modaliteiten zoals "het moet dat...", "het is mogelijk dat...", "ergens in de toekomost...". Maar dankzij de relationele semantiek hebben deze modale logica's hun weg gevonden in de taalkunde, de kunstmatige intelligentie, en de theoretische informatica. Deze flexibiliteit is de belangrijkste reden om de talen waarin deze modale logica's gesteld zijn, te kiezen als uitdrukkingsmiddel om de eigenschappen van coalgebra's te beschrijven. Tegenwoordig kunnen we ook zeggen dat modale logica's coalgebraïsch zijn.

Het proefschrift heeft twee delen: *Modalities in the Stone age* en *Coalgebraic modal logics at work*.

In het eerste deel ontwikkelen we coalgebraïsche modale logica's. Dit soort van logica's vormen heden ten dage een van het belangrijkste studie-object in het vakgebied van logica's voor coalgebra's. Coalgebraïsche modale logica's brengen de veelheid van modale logica's die een rol spelen in de Theoretische Informatica onder één noemer. Vanuit dit perspectief kunnen we al die systemen met dezelfde wiskundige technieken bestuderen.

Meer specifiek ziet het eerste deel van dit proefschrift er als volgt uit: allereerst introduceren we coalgebraïsche modale logica's als een veralgemenisering van de 'oer'-modale logica's. De basis ingredienten hier zijn de zogenoemde *predicate liftings* of *concrete modalities*. We gebruiken deze concrete modaliteiten om het zogenoemde *functorial framework* voor coalgebraïsche modale logica's te ontwikkelen. Zo belanden we bij een algebrasche semantiek voor modale logica. We gebruiken dit perspectief om verschillende coalgebraïsche modale logica's te vergelijken via vertalingen. We richten onze aandacht daarbij in het bijzonder op de de zogenoemde *Moss logica*. We sluiten dit deel af met een representatie stelling waarin we bewijzen dat alle modale logica's binnen het *functorial framework* axiomatische systemen zijn van logica's waarin de modaliteiten *predicate liftings* zijn.

In het tweede deel van dit proefschrift onderzoeken we de grenzen van coalgebrašche modale logica's verder. Het bestaat uit drie case studies. In de eerste daarvan gebruiken we logica's voor coalgebra's om coalgebra's te bouwen. We passen dit toe om inzicht te krijgen in het verband tussen eindige coalgebra's en de Hennessy-Milner eigenschap. In de tweede case studie kijken we naar dynamische logica's als coalgebraïsche modale logica's; dynamische logica's worden dikwijls gebruikt om over computerprogramma's te redeneren. Als derde voorbeeld bestuderen we coalgebraïsche modale logica's als een formalisme om het (ongoing)-gedrag van een toestand in een toestandssysteem te beschrijven. We richten onze aandacht daarbij op dekpuntlogica's met predicate liftings als modaliteiten. We ontwikkelen een speltheoretische semantiek voor deze logica's en bepalen een bovengrens voor het vervulbaarheidsprobleem.

# Abstract


This thesis hovers over the interaction of coalgebras and modal logics.

Intuitively, coalgebras are machines from the point of view of the user. They arise from computer science as a promising mathematical foundation for computer systems. Colagebras study different state-based systems, where the set of states can be understood as a black box to which one has limited access. For an intuitive illustration of this, think of a coffee vending machine. Most people do not really know what the inner mechanism of the machine is, or even haven ever seen such mechanism. Nevertheless, they can use the machine efficiently. Here we have an interaction with a system in the black box perspective. The coffee machine is a coalgebra.

Modal languages provide an internal, local perspective on relational structures. They origin in philosophy as the informal study of modalities like "it is necessary that...", "it is possible that...", "at some point in the future...". However, thanks to the so-called relational semantics, modal logics have found their way to areas such as linguistics, artificial intelligence, and computer science. This versatility has helped to place modal languages as the appropriate choice of languages to describe coalgebras. Moreover, nowadays, it is also fair to say that *modal logics are coalgebraic.*

This thesis has two parts: *Modalities in de Stone age* and *Coalgebraic modal logics at work.*

In the first part of this manuscript we investigate *coalgebraic modal logics.* These logics have become one of the main currents of modal logics for coalgebras. Coalgebraic modal logics bring uniformity to the rising wave of modal logics in computer science and provide generality to the interactions of coalgebras and modal logics. More concretely, the structure of the first part is as follows: we first introduce coalgebraic modal logics as a generalisation of basic modal logic using so-called *predicate liftings* or *concrete modalities.* We develop these modalities to introduce the so-called *functorial framework* for coalgebraic modal logics.

This accounts to give an algebraic semantics of modal logics. We then use this perspective to compare different coalgebraic modal logics by means of translations. We devote special attention to the so-called *Moss logic*. We finish this part with a representation theorem which states that each coalgebraic modal logic within the functorial framework is an axiomatization of a logic of predicate liftings.

In the second part of this manuscript we investigate how coalgebraic modal logics can be used to study coalgebras. We work three case studies. In the first case we investigate the use of logics for coalgebras to build coalgebras. More concretely we study the relation between final coalgebras and the Hennessy-Milener property. In the second case we investigate how coalgebraic modal logics align with so-called dynamic logics, dynamic logics are often used to reason about programs. In the third place we study coalgebraic modal logics to describe the ongoing behaviour of a state in a coalgebra. We focus on logics of predicate liftings with fixpoint operators. We give a game semantics and prove a bounded modal property for these logics.

ILLC DS-2008-03: **Olivier Roy**
*Thinking before Acting: Intentions, Logic, Rational Choice*

ILLC DS-2008-04: **Patrick Girard**
*Modal Logic for Belief and Preference Change*

ILLC DS-2008-05: **Erik Rietveld**
*Unreflective Action: A Philosophical Contribution to Integrative Neuroscience*

ILLC DS-2008-06: **Falk Unger**
*Noise in Quantum and Classical Computation and Non-locality*

ILLC DS-2008-07: **Steven de Rooij**
*Minimum Description Length Model Selection: Problems and Extensions*

ILLC DS-2008-08: **Fabrice Nauze**
*Modality in Typological Perspective*

ILLC DS-2008-09: **Floris Roelofsen**
*Anaphora Resolved*

ILLC DS-2008-10: **Marian Counihan**
*Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning*

ILLC DS-2009-01: **Jakub Szymanik**
*Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language*

ILLC DS-2009-02: **Hartmut Fitz**
*Neural Syntax*

ILLC DS-2009-03: **Brian Thomas Semmes**
*A Game for the Borel Functions*

ILLC DS-2009-04: **Sara L. Uckelman**
*Modalities in Medieval Logic*

ILLC DS-2009-05: **Andreas Witzel**
*Knowledge and Games: Theory and Implementation*

ILLC DS-2009-06: **Chantal Bax**
*Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.*

ILLC DS-2009-07: **Kata Balogh**
*Theme with Variations. A Context-based Analysis of Focus*

ILLC DS-2009-08: **Tomohiro Hoshi**
*Epistemic Dynamics and Protocol Information*

ILLC DS-2009-09: **Olivia Ladinig**
*Temporal expectations and their violations*

ILLC DS-2009-10: **Tikitu de Jager**
*"Now that you mention it, I wonder...": Awareness, Attention, Assumption*

ILLC DS-2009-11: **Michael Franke**
*Signal to Act: Game Theory in Pragmatics*

ILLC DS-2009-12: **Joel Uckelman**
*More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains*

ILLC DS-2009-13: **Stefan Bold**
*Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.*

ILLC DS-2010-01: **Reut Tsarfaty**
*Relational-Realizational Parsing*

ILLC DS-2010-02: **Jonathan Zvesper**
*Playing with Information*

ILLC DS-2010-03: **Cédric Dégremont**
*The Temporal Mind. Observations on the logic of belief change in interactive systems*

ILLC DS-2010-04: **Daisuke Ikegami**
*Games in Set Theory and Logic*

ILLC DS-2010-05: **Jarmo Kontinen**
*Coherence and Complexity in Fragments of Dependence Logic*

ILLC DS-2010-06: **Yanjing Wang**
*Epistemic Modelling and Protocol Dynamics*

ILLC DS-2010-07: **Marc Staudacher**
*Use theories of meaning between conventions and social norms*

ILLC DS-2010-08: **Amélie Gheerbrant**
*Fixed-Point Logics on Trees*

ILLC DS-2010-09: **Gaëlle Fontaine**
*Modal Fixpoint Logic: Some Model Theoretic Questions*

ILLC DS-2010-10: **Jacob Vosmaer**
*Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.*

ILLC DS-2010-11: **Nina Gierasimczuk**
*Knowing One's Limits. Logical Analysis of Inductive Inference*

ILLC DS-2011-01: **Wouter M. Koolen**
*Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice*

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**
*Small steps in dynamics of information*

ILLC DS-2011-03: **Marijn Koolen**
*The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

ILLC DS-2011-04: **Junte Zhang**
*System Evaluation of Archival Description and Access*

ILLC DS-2011-05: **Lauri Keskinen**
*Characterizing All Models in Infinite Cardinalities*

ILLC DS-2011-06: **Rianne Kaptein**
*Effective Focused Retrieval by Exploiting Query Context and Document Structure*

ILLC DS-2011-07: **Jop Briët**
*Grothendieck Inequalities, Nonlocal Games and Optimization*

ILLC DS-2011-08: **Stefan Minica**
*Dynamic Logic of Questions*

ILLC DS-2011-09: **Raul Andres Leal**
*Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications*

ILLC DS-2011-10: **Lena Kurzen**
*Complexity in Interaction*