Logics for Coalgebras and Applications to Computer Science

Dissertation

zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften an der Fakultät für Mathematik und Informatik der Ludwig-Maximilians-Universität München

von

Alexander Kurz

Tag der Einreichung:	30. Mai 2000
Tag der mündlichen Prüfung:	19. Juli 2000
Berichterstatter:	Prof. Dr. Fred Kröger
	Prof. Dr. Horst Reichel

Summary

In the last years coalgebras and their applications to computer science have received much attention. This thesis studies logics to specify coalgebras and, in particular, coalgebras as models for (generalised) modal logics.

Chapter 1 contains some contributions to universal coalgebra. Concerning logics and coalgebras, section 1.7 shows which categories of coalgebras are (isomorphic to) categories of algebras and thus give naturally rise to models for (first-order) equational logic.

Chapter 2 shows that modal logic for coalgebras is dual to equational logic for algebras. The expressive power of modal logics for coalgebras is characterised by dualising Birkhoff's variety theorem and similar results.

Chapter 3 is based on the idea that—having enough information on the signature functor—coalgebras can be viewed as transition systems, i.e. as Kripke models for a suitable modal logic. This approach is used here to present a modal logic for specifying those coalgebras that arise from classes in the sense of object-oriented programming.

Chapter 4 combines algebraic and coalgebraic specification techniques. For the special signatures arising from behavioural algebraic specifications a modular, sound, and complete proof system for specifications using first-order logic is given.

Contents

Chapter 0

Introduction

At the heart of this thesis are the investigations on the duality of equational and modal logic in chapter 2. The interested reader may start immediately with chapter 2 and will be referred to chapter 1 and appendix A only where needed.

The work reported here started with the idea that modal logic may be a good specification language for coalgebras, an intuition that is due to coalgebras being generalised transition systems. The work on modal logic and coalgebras started with Barwise and Moss [13]. Then Moss [87] (a paper which circulated in the community at least since 1997) developed 'coalgebraic logic' which can be understood as a generalisation of modal logic to a large class of coalgebras over **Set**. Moss' ideas were one starting point of this thesis, the other being the work by Reichel [93] and Jacobs [62] on coalgebras and final semantics for objects and classes in the sense of object-oriented programming. One natural question then was how a modal logic for the kind of coalgebras used by Reichel and Jacobs would look like. This is reported in chapter 3. Similar ideas were developed independently and simultaneously by Martin Rößiger [102, 105] and developed further by Rößiger [103, 104] and Jacobs [66].

After having studied modal logic and coalgebras for the case of applications to object-oriented programming, the next question was whether it might be that modal logic is a 'natural' logic for coalgebras in much the same way as equational logic is a natural logic for algebras. Since Birkhoff's variety theorem is the classical result on the relationship of equational logic and algebras the quest for a co-variety theorem became the next step to take. To the author's knowledge, the first covariety theorems were presented at the CMCS workshop 1998 in Lisbon. Roşu [98] presented a result on equational logic and coalgebras for a restricted class of functors (called 'algebraic' in section 1.7). Gumm showed co-Birkhoff theorems[1] for a large class of functors,[2] the proof being based on the idea of dualising the algebraic proof of Birkhoff's variety theorem. These contributions, however, did not touch on the following questions: the relation of modal logic to coalgebras and, second, to what extend co-Birkhoff theorems can be obtained as *formal* duals to Birkhoff theorems. In Kurz [75, 74] it was observed that the propositional variables in modal formulas play a dual role to variables in equational logic. Based on this observation and the fact that co-Birkhoff theorems for

[1]Gumm's co(quasi)variety theorems are not in the CMCS'98 proceedings [41] but in [45].

[2]Bounded and weak-pullback preserving functors over the base category of sets.

modal logic could be obtained, the slogan that

modal logic is dual to equational logic

was introduced in [75]. In these papers, however, the duality of equational and modal logic is used only as a heuristic means, not as a formal principle. Based on the work of Banaschewski and Herrlich [11] on equational logic and algebras, chapter 2 now gives a formal account of this duality. The results of [75, 74] are obtained as corollaries to the duals of the Birkhoff theorems proved in [11].

Chapters 4 and 1 are following two different turn-offs from the main road sketched above. Chapter 4, which is joint work with Rolf Hennicker, extends coalgebraic specifications in the sense of Reichel [93] and Jacobs [62] with ideas from behavioural algebraic specifications in order to provide more flexibility (constants, 'binary methods', modularity). Chapter 1 adds some new results to universal coalgebra as developed by Rutten [109], mainly by exploiting the technique of factorisation systems which proved to be central to make explicit the duality of modal and equational logic in chapter 2.

Overview of the Chapters

The chapters have been written in a way such that they can be read almost independently. Their order being somewhat arbitrary we separated the theoretical investigations from the studies motivated by applications to computer science. Appendices recalling basic notions of category theory, modal logic, and coalgebraic logic are provided.

Categorical Universal Coalgebra

Rutten's Universal Coalgebra [109] is developed, like classical universal algebra, for coalgebras *over sets*. But since the theory of coalgebras needs categorical notions right from the beginning, it seems reasonable to develop the theory completely in a categorical style, i.e., not using any special properties of the category of sets.

The motivation for considering arbitrary base categories in this work came from our work on the duality between modal and equational logic, see chapter 2. This duality can be made precise only if we do not stick to the category of sets.

Concerning applications of this categorical approach it should be pointed out that coalgebras over algebras are an interesting topic in concurrency theory (see e.g. Corradini et. al. [28], Turi and Plotkin [117]). Coalgebras over metric spaces were investigated in Rutten [109], section 18, in Turi and Rutten [118, 106], in Monteiro [85, 86], and particularly in Worrell [128, 126]. Moreover, coalgebras over domains will be a topic of future research (there has already some work be done in this direction, see e.g. Rutten [107]).

The main contributions of this chapter are:

1. In order for bisimulations to work nicely additional assumptions like signatures preserving weak pullbacks are needed. These assumptions can be avoided using *cocongruences* or *behavioural equivalences* instead.

2. For an axiomatic and categorical development of universal coalgebra we propose to make systematic use of *factorisation systems.*

3. We generalise the notion of a bounded functor (depending on **Set**) to that of a *bounded category* (not depending on **Set**).[3]

4. We use factorisation systems to show how to construct (under suitable conditions) *limits in categories of coalgebras.*

5. Suppose two functors Σ, Ξ are adjoint, $\Sigma \dashv \Xi$. *Then the categories of* Σ*-algebras and* Ξ*-coalgebras are isomorphic.* It is shown that this observation explains the special format of the functors typically used in behavioural algebraic and hidden algebra specifications.

Points 4 and 5 appeared in Kurz and Pattinson [71].

Modal Logic and Coalgebra

This chapter is about (the semantics of) modal logics for coalgebras and the duality of modal and equational logic. The basic idea is to interpret formulas of modal logic as subcoalgebras of cofree coalgebras. This has the following consequences:

1. Coalgebraic semantics for modal logics appears as dual (in the categorical sense) to algebraic semantics for equational logic.

2. Modal rules appear as dual to implications.

3. Proofs for general co-Birkhoff theorems are obtained as formal duals to proofs of Birkhoff theorems.

4. In particular these proofs do not depend on the signature being bounded or preserving weak pullbacks or on the base category being the category of sets.

These results can be applied to well-known modal logics:

5. The coalgebraic logic of Moss [87] is extended with propositional variables and corresponding co-Birkhoff theorems are shown.

6. We characterise the expressive power of infinitary modal logics on Kripke frames.

The covariety theorem for infinitary modal logic has been published in [75], the coquasivariety theorem was presented at the 11th Conference on Logic, Methodology and Philosophy of Science, Krakow, 1999 (see [74]), and an extended abstract of chapter 2 is available as [76].

[3] Despite the name bounded 'functor' the condition is really on the category of coalgebras, not on the signature. The name bounded category seems therefore appropriate anyway.

Modal Logic and Coalgebras: A Case Study

The basic idea of this chapter is that—when we have enough information on the signature functor—coalgebras can be viewed as Kripke models for a suitable modal logic. This approach is used here to present a modal logic for specifying those coalgebras that arise from classes in the sense of object-oriented programming, see the work of Reichel [93] and Jacobs [62] on final semantics for classes. We develop the idea that (finitary) modal logic is suitable to specify classes in connection with the 'final paradigm'. Special features of this approach are:

1. States (which are supposed to be not observable) do not appear in formulas.
2. Specifications are invariant under bisimulations (i.e., they are behavioural).
3. Modally defined classes of models have a final coalgebra.
4. Modal logic provides us with a complete axiomatisation.

This chapter will be published as Kurz [77] (an earlier version appeared in the proceedings of CMCS'98 [57]).

Algebraic and Coalgebraic Specifications

The second application concerns (behavioural) algebraic and coalgebraic specifications. Technically speaking, algebraic and coalgebraic specification techniques are combined. One can see this work as a transformation of the approach of Hennicker and Bidoit [50], called observational logic, into the framework of coalgebras or as an algebraic extension of coalgebraic specifications in the sense of Reichel [93] and Jacobs [62] (see the previous chapter). The main points of our approach are:

1. We propose to start writing a specification by fixing the observer operations (in order to formally specify the intended notion of observational equivalence). Only then further operations are added (which are required not to contribute to observations). The observers are interpreted coalgebraically, the operations algebraically.
2. The requirement that algebraic operations must not contribute to observations is expressed axiomatically.
3. This requirement has as a consequence that first order logic is sound.[4] It is also shown that it is complete if one admits an infinitary rule.
4. From a coalgebraic perspective, the separation of operations and observers increases the expressivity of coalgebraic specifications in the following sense: One can now handle constants and n-ary methods. Moreover, a compositional style of writing specifications and verifying properties becomes possible.

[4]This is shown for a special kind of coalgebraic signatures $\Xi : \mathbf{Set} \to \mathbf{Set}$, namely the ones that are 'algebraic' in the sense of section 1.7.

5. Technically, we introduce the notion of a *behaviour monad* B mapping a given model M to its behaviour BM. We then show how, given an institution for a 'standard' logic on 'behaviours', one also obtains institutions for 'behavioural' logics on 'standard' models.[5]

In Bidoit et al. [16], the approach of this chapter has been dualised to give an account of reachability in algebraic specifications. This led to a new notion of constructor-based specifications as well as to the insight that observability and reachability in (co)algebraic specifications are dual concepts, a phenomenon which was discovered earlier in the context of automata theory, see Arbib and Manes [7].

The work in this chapter has been done jointly with Rolf Hennicker. It will be published as Kurz and Hennicker [72] (an earlier version has appeared in Hennicker and Kurz [51]).

[5] A logic is called behavioural (or observational) if the formulas can not distinguish between a model and its behaviour.

Acknowledgements

I would like to thank

Fred Kröger, my supervisor, for his continuous support during the whole work;

Martin Wirsing for integrating me in his research group where there always has been a very pleasant working atmosphere;

Helmut Schwichtenberg, Felix Joachimski, and the Graduiertenkolleg 'Logik in der Informatik' for generous organisational and financial support.

Very special thanks are due to

Horst Reichel for his continuous interest in my work, for inviting me two times to Dresden and for refereeing this dissertation;

Larry Moss for his brilliant ESSLLI'97 lecture on 'vicious circles' and coalgebras introducing me to this exciting field;

Bart Jacobs, Larry Moss, Horst Reichel, Jan Rutten for numerous discussions and, especially, for organising the CMCS workshops and giving me the possibility to continuously present my results and to meet other researchers in this area.

A great influence on my research and learning have always been my colleagues in Munich. In particular, I want to thank:

Pietro Cenciarelli for his thrilling improvised lectures on (the beauty of) category theory;

Rolf Hennicker for his continuous critique of coalgebraic specifications which resulted in the joint work reported in chapter 4;

Alexander Knapp, my office mate, for a great time and years of discussion on computer science, mathematics, and much more;

Stephan Merz for guidance in the beginning when my research was on temporal logic;

Dirk Pattinson with whom I had the pleasure to discover category theory during the last two years;

Bernhard Reus for his encouragement in exploring questions not driven by applications;

Holger Sturm whose masterful expertise in classical and modal logic has been an inspiration.

For discussions on logic and coalgebra which contributed to this work I am indebted also to Alexandru Baltag, H. Peter Gumm, Ataru Nakagawa, Martin Rößiger, Grigore Roşu, Tobias Schröder, Hendrik Tews, Yde Venema. I also want to thank Jesse Hughes, Dirk Pattinson, and Tobias Schröder for their suggestions helping to improve the final version.

Part I

Theory of Coalgebras

Chapter 1

Categorical Universal Coalgebra

Universal coalgebra is the study of coalgebras along the lines of universal algebra as initiated by Rutten [109]. There, coalgebras are investigated whose carriers are sets. Here, the term Categorical Universal Coalgebra is intended to denote a development of universal coalgebra in purely categorical terms, that is, independent of the category of sets.

The aims of this chapter are to review standard notions and make the connection to modal logic (section 1), to introduce the notions of cocongruence and behavioural equivalence as a generalisation of bisimulations (section 2), to investigate factorisation systems for coalgebras (section 3), to develop universal coalgebra by axiomatising the forgetful functors (section 4), to introduce a generalisation of the notion of boundedness which does not depend on the category of sets (section 5), to show how to construct limits using factorisation systems (section 6), to analyse signatures for coalgebras giving rise to categories of algebras (section 7). We conclude with comments on duality and future research.

Most of the sections can be read independently with the exception of section 1.4 which depends on all of the previous ones.

For introductions to coalgebras see Jacobs and Rutten [58], Rutten [109] (or the earlier electronically available [108]), Gumm [46].

1.1 Coalgebras, Bisimulations, and Modal Logic

This section reviews the basic definitions (in **Set**-independent presentation) and introduces notation. In continuing examples we develop the connection to modal logic. Also, discussing bisimulations at length we want to make the point that some of the peculiarities of bisimulations are due to the fact that the structure of a (largest) bisimulation is in general *not* uniquely determined. This will prepare the introduction of cocongruences and behavioural equivalences (where the structure is uniquely determined) which will replace bisimulations in our axiomatic development of universal coalgebra in section 1.4.

1.1.1 Coalgebras for a Functor

A **coalgebra** is given w.r.t. a **base category** $\mathcal{X}$ and an endofunctor (also called **signature**) $\Omega : \mathcal{X} \to \mathcal{X}$: An Ω-coalgebra $A = (UA, \alpha)$ consists of an object $UA \in \mathcal{X}$ and an arrow $\alpha : UA \to \Omega UA$. Occasionally, we will refer to a coalgebra A using the structure α. Ω-coalgebras form a category $\mathcal{X}_\Omega$ where a **coalgebra morphism** $f : (UA, \alpha) \to (UB, \beta)$ is an arrow $f : UA \to UB \in \mathcal{X}$ such that $\Omega f \circ \alpha = \beta \circ f$:

$$\begin{tikzcd} UA \arrow[r, "\alpha"] \arrow[d, "f"'] & \Omega UA \arrow[d, "\Omega f"] \\ UB \arrow[r, "\beta"'] & \Omega UB \end{tikzcd}$$

The **forgetful functor** $U : \mathcal{X}_\Omega \to \mathcal{X}$ maps a coalgebra $A = (UA, \alpha)$ to UA and a morphism $f : (UA, \alpha) \to (UB, \beta)$ to the arrow $f : UA \to UB$ in $\mathcal{X}$. Note that according to this definition $Uf = f$ which allows to simplify notation in some cases. Also, since many of the interesting properties of a category of coalgebras $\mathcal{X}_\Omega$ depend on the forgetful functor $U : \mathcal{X}_\Omega \to \mathcal{X}$ we often say that the functor $U : \mathcal{X}_\Omega \to \mathcal{X}$ (or the pair $(\mathcal{X}_\Omega, U)$ is a category of coalgebras. This is the usual terminology concerning 'concrete' categories (see section A.1).

Example (Modal Logic) 1.1.1. We show how that the standard models of modal logic, Kripke models and Kripke frames, are coalgebras. For a brief review of modal logic and references see the appendix. Let $\mathcal{X} = \mathbf{Set}$ and consider the functor $\Omega X = \mathcal{P}P \times \mathcal{P}X$ where P is a set of propositional variables and $\mathcal{P}$ denotes powerset.[1] Then the Ω-coalgebras are the usual *Kripke models* of modal logic: given a coalgebra (UA, α) and a world $x \in UA$, we have $\alpha(x) = (Q, Y)$ where $Q \subset P$ and $Y \subset UA$. This determines a Kripke model as follows: Q is the set of propositional variables true in the world x and Y is the set of worlds accessible from x. We write $x \models p$ iff $p \in Q$ and $x \to y$ iff $y \in Y$. The case $P = \{\}$ yields as Ω-coalgebras the *Kripke frames*. Also, coalgebra morphisms give the standard notion of morphism for Kripke models/frames, which are called in the modal literature p-morphisms, zig-zag-morphisms, bounded morphism or functional bisimulations.

[1] This only defines Ω on sets. On functions Ω is defined in the standard way, $\mathcal{P}$ being the covariant powerset functor: Given $f : X \to Y$, $\Omega f = \mathrm{id}_{\mathcal{P}P} \times \lambda A \in \mathcal{P}X.\{f(a) : a \in A\}$.

The following definition and proposition may be skipped but we want to mention nevertheless how the relationship between frames and models can be expressed categorically (for the definition of $U{\downarrow}\mathcal{X}$ see the section A.1).

Definition 1.1.2. Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ be a category of coalgebras. Considering $\mathcal{X}_\Omega$ as a category of Kripke frames, $U{\downarrow}\mathcal{X}$ is the corresponding category of models.

Remark. Spelling out the details we see that $U{\downarrow}\mathcal{X}$ has objects $(A, UA \xrightarrow{\gamma} C)$ for $A \in \mathcal{X}_\Omega$, $C \in \mathcal{X}$ and morphisms $(f,g) : (A,\gamma) \to (B,\delta)$ with $f \in \mathcal{X}_\Omega$, $g \in \mathcal{X}$ such that

$$\begin{tikzcd} UA \arrow[r, "\gamma"] \arrow[d, "Uf"'] & C \arrow[d, "g"] \\ UB \arrow[r, "\delta"'] & D \end{tikzcd}$$

commutes.

In case that U has a right adjoint F we can describe $U{\downarrow}\mathcal{X}$ also as $\mathcal{X}_\Omega{\downarrow}F$:

Proposition 1.1.3. *Let $U : \mathcal{C} \to \mathcal{X}$ and $F : \mathcal{X} \to \mathcal{C}$ be functors. Then $U \dashv F$ iff $U{\downarrow}\mathcal{X}$ is isomorphic to $\mathcal{X}_\Omega{\downarrow}F$.*

Proof. "only if": Let $(-)^\# : \mathcal{X}(UA, C) \to \mathcal{C}(A, FC)$ be the natural iso given by the adjunction. Then $\varphi : U{\downarrow}\mathcal{X} \to \mathcal{X}_\Omega{\downarrow}F$ maps objects $(A, UA \xrightarrow{\gamma} C)$ to $(A, A \xrightarrow{\gamma^\#} FC)$ and is the identity on morphisms. That φ is bijective is immediate but one has to check that φ maps morphisms (commuting squares) to commuting squares. This follows form $(-)^\#$ being natural.
"if": Given $\varphi : U{\downarrow}\mathcal{X} \to \mathcal{X}_\Omega{\downarrow}F$ define $(UA \xrightarrow{\gamma} C)^\#$ as $\varphi(A, UA \xrightarrow{\gamma} C)$. That $(-)^\#$ bijective is immediate and naturality follows from φ mapping morphisms (commuting squares) to commuting squares. $\square$

A particularly important property that a category $\mathcal{X}_\Omega$ of coalgebras may have is that the forgetful functor $U : \mathcal{X}_\Omega \to \mathcal{X}$ has a right adjoint $F : \mathcal{X} \to \mathcal{X}_\Omega$ (this implies the existence of a final coalgebra). We then say that $\mathcal{X}_\Omega$ admits a cofree construction and, for $C \in \mathcal{X}$, FC (together with the counit $\epsilon_C : UFC \to C$ given by the adjunction) is called the **cofree coalgebra** over C. That F is a right adjoint is equivalent to saying that FC and the counit $\epsilon_C : UFC \to C$ have the property: For all Ω-coalgebras A and all arrows $\gamma : UA \to C$

$$\begin{tikzcd} UFC \arrow[d, "\epsilon_C"'] & \\ C & UA \arrow[l, "\gamma"] \arrow[ul, dashed, "\gamma^\#"'] \end{tikzcd}$$

there is a unique coalgebra morphism $\gamma^\# : A \to FC$ such that the diagram commutes. (The dashed arrow is a coalgebra morphism (recall $\gamma^\# = U\gamma^\#$); the plain arrows are in the base category; commutativity of the diagram is commutativity in the base category.) In the case that C is the terminal element 1 of $\mathcal{X}$, $FC = F1$ is the **final coalgebra** or **terminal coalgebra**.[2]

The way to think about cofree constructions is the following (assume $\mathcal{X} = \mathbf{Set}$ for the next two paragraphs). In the case of $C = 1$ (i.e., a one element set) the property above reduces to the statement that, for all $A \in \mathcal{X}_\Omega$ there is a unique morphism $! : A \to F1$. Since we will see below (see example 1.1.10) that morphisms are functional bisimulations and we usually think of bisimulations as preserving behaviour, we can rephrase this property by saying that for all coalgebras A and all "states" $a \in A$ there is a *unique* element in $F1$ representing the behaviour of a, namely $!(a)$. That is, *for each possible behaviour that Ω-coalgebras can show, there is precisely one element of $F1$ representing it.*

In the case that $|C| > 1$ ($C = \{\}$ is left for the reader), we should think of C as a set of colours and of γ and ε_C as colourings of the transitions systems A and FC, respectively. Then cofreeness of FC means that for all coalgebras A and all colourings $\gamma : UA \to C$ there is a unique coalgebra-morphism $\gamma^\# : A \to FC$ respecting the colourings. That is, *for each possible behaviour that Ω-coalgebras can show where in addition to the observations specified by Ω there are new observations $\gamma : UA \to C$ allowed, there is precisely one element of FC representing this behaviour and the new observations γ.*

Example (Modal Logic) 1.1.4. We show how to interpret sets of colours and the cofree construction in modal logic. Let $\mathcal{X} = \mathbf{Set}$ and $\Omega X = \mathcal{P}_\kappa X$ where κ is a cardinal and $\mathcal{P}_\kappa X = \{Y \subset X : |Y| < \kappa\}$,[3] and let $C = \mathcal{P}P$, P a set of propositional variables. Then A is an Kripke frame with *degree of branching smaller than κ*. The functions $\gamma : UA \to C$ are valuations: every world in A is assigned the set of propositional variables which are true in this world. That is, A together with γ, written as (A, γ), is a Kripke model. And FC together with ϵ_C is a universal Kripke model in the sense that for every Kripke model (A, γ) there is a unique morphism $\gamma^\# : A \to FC$ respecting the valuations.

Let us mention the differences of the terminal coalgebra and the canonical model construction in modal logic. In the canonical model two elements are equal iff they satisfy the same finitary modal formulas. In the terminal coalgebra, two elements are equal iff they are bisimilar. In particular, there are usually different elements satisfying the same finitary formulas. If, however, the logic under consideration is finitary and characterises bisimulation then the terminal coalgebra is the canonical model, see e.g. Rößiger [105].

The examples above suggest that an Ω-coalgebra A together with an arrow $\gamma : UA \to C \in \mathcal{X}$ is a $(C \times \Omega)$-coalgebra and vice versa. To make this precise, suppose that a functor $\Omega : \mathcal{X} \to \mathcal{X}$ and $C \in \mathcal{X}$ are given. Let us write (A, γ) for a pair consisting of $A \in \mathcal{X}_\Omega$ and a $\gamma : UA \to C \in \mathcal{X}$. We define a morphism $f : (A, \gamma) \to (A', \gamma')$ to be a morphism $f : A \to A'$ such that $\gamma' \circ f = \gamma$. It is then immediate to show:

[2] The notions terminal and final are synonymous. We tend to speak of *terminal* objects but of *final* coalgebras or *final* semantics.

[3] This cardinality restriction on the powerset functor is necessary to guarantee the existence of a right adjoint to U. On the other hand this restriction is not essential in the sense that Aczel and Mendler [2] showed that the powerset functor can be extended from the category **Set** to the category **SET** of classes and set-continuous functions allowing for a cofree construction with cofree coalgebras having proper classes as carriers.

Proposition 1.1.5. *Let $\mathcal{X}$ be a category with binary products, $\Omega : \mathcal{X} \to \mathcal{X}$ and $C \in \mathcal{X}$. Then the category of pairs (A, γ), $A \in \mathcal{X}_\Omega$ and $\gamma : UA \to C \in \mathcal{X}$ defined above is isomorphic to the category $\mathcal{X}_{C\times\Omega}$ of $(C \times \Omega)$-coalgebras.*

Using the comma category $U{\downarrow}C$ (see section A.1) we express this observation in more categorical terms:

Corollary 1.1.6. *Let $\mathcal{X}$ be a category with binary products, $\Omega : \mathcal{X} \to \mathcal{X}$ and $C \in \mathcal{X}$. Then $U{\downarrow}C$ is isomorphic to $\mathcal{X}_{C\times\Omega}$.*

Example (Modal Logic) 1.1.7. Let $\Omega X = \mathcal{P}X$ and $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ the forgetful functor. Then $\mathbf{Set}_\Omega \simeq U{\downarrow}1$ is the category of Kripke *frames* and $\mathbf{Set}_{\mathcal{P}P\times\Omega} \simeq U{\downarrow}\mathcal{P}P$ is the category of those Kripke *models* which interpret the propositional variables in P.

Example 1.1.8 (Duality of algebras and Kripke frames). Thomason [116][4] describes a duality between the category of complete atomic boolean algebras with operators and the category of Kripke frames. This duality arises naturally from the coalgebraic perspective: It is not difficult to see that the dual of the category of coalgebras (Kripke frames) $\mathbf{Set}_\mathcal{P}$ is isomorphic to the category of complete atomic boolean algebras with operators.

1.1.2 Coalgebras for a Comonad

This section comments on the relationship of coalgebras for a functor and coalgebras for a comonad (see appendix A.7 for definitions). The following theorem is probably 'folklore'.

Theorem 1.1.9. *Let $\mathcal{X}$ be a category and $\Omega : \mathcal{X} \to \mathcal{X}$ a functor. Then the forgetful functor $U : \mathcal{X}_\Omega \to \mathcal{X}$ is comonadic iff it has right adjoint.*

Proof. We use (co-)Beck's theorem (see theorem A.7.1) stating that a functor is comonadic iff it has a right adjoint and creates split equalisers. We only have to show that $U : \mathcal{X}_\Omega \to \mathcal{X}$ creates split equalisers which immediately follows from the respective definitions: Let $X, Y, Z \in \mathcal{X}$ and $f, g : \eta \to \zeta \in \mathcal{X}_\Omega$ as indicated in the following diagram.

$$\begin{tikzcd}
\Omega X \arrow[r, "\Omega m"] & \Omega Y \arrow[r, shift left, "\Omega f"] \arrow[r, shift right, "\Omega g"'] & \Omega Z \\
X \arrow[u, dotted, "\xi"] \arrow[r, "m"] & Y \arrow[u, "\eta"'] \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Z \arrow[u, "\zeta"']
\end{tikzcd}$$

Assuming that in the lower row m is a split equaliser we have to show that there is a unique coalgebra structure ξ making $m : \xi \to \eta$ into an equaliser in $\mathcal{X}_\Omega$. Since m is split Ωm is a split equaliser as well. This gives existence and uniqueness of ξ. That ξ is indeed an equaliser follows from Ωm being mono (m is split mono). □

[4] I would like to thank Yde Venema for pointing out this reference.

1.1.3 Bisimulations

Bisimulations appeared in modal logic as p-morphisms (Segerberg [111]) and p-relations or zigzag relations (van Benthem [121, 122, 123]) and were rediscovered in concurrency theory (Park [89], Milner [84, 83]). Aczel [1] used bisimulations to define equality for non-well founded sets and to prove the existence of final coalgebras. The general definition of a bisimulation for coalgebras, given below, is due to Aczel and Mendler [2].

A **bisimulation** (R, p, q) (or (p, q) for short) between coalgebras A and B is an object $R \in \mathcal{X}$ and two arrows $p : R \to UA$, $q : R \to UB$ such that

1. (R, p, q) is a mono span in $\mathcal{X}$.[5]
2. there is a structure $\varrho : R \to \Omega R$ such that $p : (R, \varrho) \to A$, $q : (R, \varrho) \to B$ are morphisms in $\mathcal{X}_\Omega$

$$A \xleftarrow{\;p\;} (R, \varrho) \xrightarrow{\;q\;} B$$

The requirement that (R, p, q) is a mono span generalises $R \subset UA \times UB$ in **Set** to arbitrary categories $\mathcal{X}$.[6] But why does one not simply require that $((R, \varrho), p, q)$ is a mono span in $\mathcal{X}_\Omega$?

The reason is that, in general, *the structure map ϱ of R is not uniquely determined.* To make this clear consider the following example borrowed from Rutten [109]. Let A be the coalgebra for the (finite) powerset functor as depicted by the diagram

$$\begin{tikzcd} & s_0 \arrow[dl] \arrow[dr] & \\ s_1 & & s_2 \end{tikzcd}$$

Clearly, the largest bisimulation on A is $R = \{(s_0, s_0), (s_1, s_1), (s_1, s_2), (s_2, s_1), (s_2, s_2)\}$. It is now important to note that the structure map ϱ is not uniquely determined, as witnessed by the following examples:

$$\begin{tikzcd} (s_1, s_1) & (s_0, s_0) \arrow[l] \arrow[r] \arrow[dl] \arrow[dr] & (s_2, s_2) \\ (s_1, s_2) & & (s_2, s_1) \end{tikzcd} \qquad \begin{tikzcd} (s_1, s_1) & (s_0, s_0) \arrow[dl] \arrow[dr] & (s_2, s_2) \\ (s_1, s_2) & & (s_2, s_1) \end{tikzcd}$$

This has the following consequences:

- The largest bisimulation is not the product $A \times A$ (see Rutten [109]).
- In order to define notions like *largest bisimulation* or *union of bisimulations* one has to ignore the structures of the bisimulations (at least in the nondeterministic cases like, e.g., $\Omega = \mathcal{P}$).

---

[5] (R, p, q) is a mono span (= monic pair = a mono 2-source) in $\mathcal{X}$ iff for all $f, g \in \mathcal{X}$ with codomain R it holds that $p \circ f = p \circ g$ and $q \circ f = q \circ g$ imply $f = g$.

[6] In a category with binary products $(R, p : R \to A, q : R \to B)$ is a mono span iff the canonical morphism $\langle p, q \rangle : R \to A \times B$ is mono.

Let us note that in typical deterministic cases the structure of a bisimulation is indeed uniquely determined. For example, in the case that Ω *preserves mono spans*,

$$\begin{tikzcd} A \arrow[d, "\alpha"'] & R \arrow[l, "p"'] \arrow[r, "q"] \arrow[d, "\varrho"'] & B \arrow[d, "\beta"'] \\ \Omega A & \Omega R \arrow[l, "\Omega p"'] \arrow[r, "\Omega q"] & \Omega B \end{tikzcd}$$

$(\Omega p, \Omega q)$ is a mono span and hence ϱ is uniquely determined. A typical example of these kind of functors are the polynomial functors on **Set**. Another typical example is given by the pullback preserving functors,[7] in particular by the multiplicative functors.

Obvious questions arising from this observation are the following.

- The example above indicates that there may be a particular, namely largest, structure on the largest bisimulations. Is this always the case?
- Is it possible to characterise those signatures Ω such that the structures of (largest) bisimulations are uniquely determined? Is it possible to characterise those signatures Ω such that the product is the largest bisimulation?[8]
- Using uniqueness of the structure of a bisimulation to define *functors for deterministic coalgebras*, how does this notion relates to other notions of determinism?

Categories of Bisimulations

In the case $\mathcal{X} = \mathbf{Set}$ bisimulations on A and B are partially ordered by inclusion. In general, bisimulations on A and B are ordered by monos in $\mathcal{X}$. Given bisimulations (R, p, q) and (R', p', q') on coalgebras A and B, a morphism $m : (R, p, q) \to (R', p', q')$ is an arrow $m \in \mathcal{X}$ such that $p' \circ m = p$ and $q' \circ m = q$. Bisimulations on A and B together with their morphism form the **category of bisimulations** $\mathsf{Bisim}_\Omega^{\mathcal{X}}(A, B)$. If Ω and $\mathcal{X}$ are clear from the context we write $\mathsf{Bisim}(A, B)$. Note that a morphism in $\mathsf{Bisim}(A, B)$ is mono in $\mathcal{X}$. Consequently, the category $\mathsf{Bisim}(A, B)$ is thin, that is, there is at most one arrow between to bisimulations. Therefore, $\mathsf{Bisim}(A, B)$ is a pre-order which becomes a partial order if isomorphic objects (bisimulations) are identified.

There is an equivalent and simpler possibility to define a category of bisimulations. We can take spans in $\mathcal{X}_\Omega$ which are mono as spans in $\mathcal{X}$ as bisimulations when we define morphisms in a way that they ignore the structure of the spans. In detail: Define the objects of the category $\mathsf{Bisim}'(A, B)$ to be the spans in $\mathcal{X}_\Omega$ which are mono as spans in $\mathcal{X}$. Given spans $((R, \varrho), p, q)$ and $((R', \varrho'), p', q')$ on coalgebras A and B, a morphism $m : ((R, \varrho), p, q) \to ((R', \varrho'), p', q')$ is an arrow $m \in \mathcal{X}$ such that $p' \circ m = p$ and $q' \circ m = q$. The categories $\mathsf{Bisim}(A, B)$ and $\mathsf{Bisim}'(A, B)$ are equivalent (but generally not isomorphic).

[7] In a category with finite products, pullback preserving implies preserving mono spans.

[8] During the writing of this thesis, these questions have found a positive answer in the case of $\mathcal{X} = \mathbf{Set}$, see Gumm and Schröder [40], theorem 8.6.

Example (Modal Logic) 1.1.10. We show the well-known fact that the notion of bisimulation above coincides with the notion of bisimulation in modal logic.

Recall the conventions of example 1.1.1 where $\Omega X = \mathcal{P}P \times \mathcal{P}X$. Moreover, let A and B be Kripke models, $R \subset UA \times UB$, p and q the canonical projections and ϱ a function such that the diagram below commutes:

$$\begin{tikzcd} UA \arrow[d, "\alpha"'] & R \arrow[l, "p"'] \arrow[r, "q"] \arrow[d, "\varrho"'] & UB \arrow[d, "\beta"'] \\ \Omega UA & \Omega R \arrow[l, "\Omega p"'] \arrow[r, "\Omega q"] & \Omega UB \end{tikzcd}$$

Let $(a, b) \in R$, $(P_a, A_a) = \alpha(a)$, $(P_{(a,b)}, R_{(a,b)}) = \varrho(a, b)$, $(P_b, B_b) = \beta(b)$. That is, P_a, P_b are the propositional variables holding in a and b, respectively, and A_a, B_b are the sets of successors of a and b, respectively. Since, by definition of Ω, $\Omega p(P_{(a,b)}, R_{(a,b)}) = (P_{(a,b)}, \mathcal{P}p(R_{(a,b)}))$ and $\Omega q(P_{(a,b)}, R_{(a,b)}) = (P_{(a,b)}, \mathcal{P}q(R_{(a,b)}))$ it follows from the commutativity of the squares that $P_a = P_b$, which is the bisimulation condition on propositional variables. It remains to show the bisimulation condition on the successors, i.e., $(a, b) \in R \;\&\; a \to a' \;\Rightarrow\; \exists b' : (a', b') \in R \;\&\; b \to b'$ (and the other way round). Assume $a \to a'$, that is, $a' \in A_a$. Commutativity of the left-hand square implies $a' \in \mathcal{P}p(R_{(a,b)})$ which in turn implies (by definition of $\mathcal{P}p$) that there is b' such that $a' = p(a', b')$, i.e., $(a', b') \in R$. Commutativity of the right-hand square now implies $b' \in B_b$, i.e., $b \to b'$.

1.1.4 Largest Bisimulations

We discuss two ways to obtain largest bisimulations. Both use different proof techniques. The first requires that epis in the base category are split, the second that the signature functor Ω preserves weak pullbacks.[9] Since we later want to develop the theory of coalgebras without these requirements we will offer substitutes for largest bisimulations in section 1.2.

Recall that, given $A, B \in \mathcal{X}_\Omega$, the category $\mathsf{Bisim}(A, B)$ is thin, i.e., we have a preorder on bisimulations: $(R, p, q) \le (R', p', q')$ iff there is a morphism $m : (R, p, q) \to (R', p', q')$. This preorder replaces the usual partial order on bisimulations over **Set** given by $\subset$. We can now define the **largest bisimulation** on A and B as the largest object in $\mathsf{Bisim}_\Omega(A, B)$ w.r.t. the order $\le$. It is, if it exists, determined up to isomorphism.

Largest Bisimulation as Union of all Bisimulations

In the case that $\mathcal{X}$ has binary products, mono spans (R, p, q) are in one-to-one correspondence with monos $R \to UA \times UB$. The image of this mono is what we usually call a bisimulation. The largest bisimulation on A and B can then be defined as the union of these images for all bisimulations on A and B. There are several ways to describe the notion of a union of images categorically. An axiomatic approach via factorisation structures for sinks is presented in

[9] A weak pullback (P, p, q) is defined like a pullback but the mediating arrow into P need not be unique. See appendix A.3 for more information.

section A.5. Here, we give a more direct construction indicated by the following diagram (taken from Borceux [19], vol.1, section 4.2).

$$\begin{tikzcd}
\Sigma R_i \arrow[r, dotted, "e"] \arrow[dr, "h"] & R \arrow[d, dotted, "m"] \\
R_i \arrow[u, "in_i"] \arrow[r, "m_i"'] & UA \times UB
\end{tikzcd}$$

Let the $\{(R_i, m_i), i \in I\}$ be the class of all bisimulations on A and B. Then take the coproduct (disjoint union) ΣR_i and let h be given by the universal property of the coproduct. h is just a quotient identifying elements of the sum that correspond to equal elements in $UA \times UB$. Now, the image of h in $UA \times UB$ should be the largest bisimulation R. Categorically, one can identify R with the mono m obtained by factoring $h = m \circ e$ with m mono and e epi.

In the case of $\mathcal{X} = \mathbf{Set}$ it is obvious that we can factor every function h as $h = m \circ e$ for some injective map m and some surjective map e. This factorisation has the property that it is unique up to isomorphism, that is, given another epi-mono-factorisation $h = m' \circ e'$, there is a unique iso g such that

$$\begin{tikzcd}
\bullet \arrow[r, "e"] \arrow[d, "e'"'] & \bullet \arrow[d, "m"] \arrow[dl, dotted, "g"'] \\
\bullet \arrow[r, "m'"'] & \bullet
\end{tikzcd}$$

commutes. This uniqueness property is important, because it allows us to consider any mono m obtained by epi-mono-factoring h as the image of h. In the case of arbitrary categories $\mathcal{X}$ it is generally not true that factorisations in an epi followed by a mono are unique up to iso (if they exist at all). This problem is solved by using the notion of a *factorisation system* (see definition A.4.1).

We identified bisimulations (R, p, q) with monos $m : R \to UA \times UB$ which is possible if we assume $\mathcal{X}$ to have binary products. The other properties the category $\mathcal{X}$ has to have in order to allow the construction above are the following:

- $\mathcal{X}$ has small coproducts.
- There is up to isomorphism only a *set* (and not a proper class) of bisimulations R_i. This holds if we require that $\mathcal{X}$ is well-powered.
- $\mathcal{X}$ has a factorisation system.

We will meet these three conditions at several places in the first part of this work. The reason is, as above, that they allow to form unions of images of morphisms.

We now state theorem 5.5 of Rutten [108] on the existence of largest bisimulations. Reconsider the diagram

$$
\begin{tikzcd}
\Sigma R_i \arrow[r, dotted, "e"] \arrow[dr, "h"] & R \arrow[d, dotted, "m"] \\
R_i \arrow[u, "in_i"] \arrow[r, "m_i"'] & UA \times UB
\end{tikzcd}
$$

Proposition 1.1.11. *Let $\mathcal{X}$ be a wellpowered category with small coproducts, binary products, and a factorisation system (E, M) such that morphisms in E are split. Then, using the notation of the diagram above, $(R, \pi_A \circ m, \pi_B \circ m)$ is the largest bisimulation on A and B.*[10]

Proof. First, it is straightforward to show that (R, m) is the smallest relation containing all bisimulations on A and B (a proof can be found in Borceux [19], vol.1, prop. 4.2.6 or in Rutten [108]).

Second, we have to show that (R, m) is indeed a bisimulation. Clearly, it is a mono span in $\mathcal{X}$ but we have to exhibit a structure ϱ. We follow lemma 5.3 in Rutten [108]: By assumption the morphism e in the diagram above has a right inverse i, i.e., $e \circ i = \mathrm{id}_R$. Define $\varrho = \Omega e \circ \sigma \circ i$ where $\sigma : \Sigma R_i \to \Omega \Sigma R_i$ is the unique map making the $in_i : R_i \to \Sigma R_i$ into $\mathcal{X}_\Omega$-morphisms. $\square$

Remark 1.1.12 (On how to show that a given relation R is a bisimulation). In order to show that R is indeed a bisimulation we have to exhibit a structure $\varrho : R \to \Omega R$. Since we know that this structure is not uniquely determined in general, we cannot expect to obtain it from a universal construction (like limits, colimits, factorisation structures etc). There seem to be two general principles. The first one, using a not uniquely determined right inverse of a split epi, has been exemplified above. The other one constructs ΩR as a weak pullback. A typical example will be shown next.

Largest Bisimulations via Weak Pullback Preserving Signatures

Recall that in $\mathbf{Set}_\Omega$ a terminal coalgebra $F1$ has the property that bisimilarity of elements of the carrier set implies equality. In general, we expect that if $\mathcal{X}_\Omega$ has a terminal coalgebra $F1$ then the kernel of the unique morphism $! : A \to F1$ is the largest bisimulation on A. This holds indeed if $\mathcal{X}$ has pullbacks and Ω weakly preserves[11] them.

Proposition 1.1.13. *Let $\mathcal{X}$ be a category with pullbacks and $\Omega : \mathcal{X} \to \mathcal{X}$ be a functor weakly preserving them. Let $A, B \in \mathcal{X}_\Omega$ and $F1$ a terminal coalgebra in $\mathcal{X}_\Omega$. Then (R, π_1, π_2) is the*

[10] π_A, π_B are the projections from $UA \times UB$ to the components UA, UB.

[11] In a category with pullbacks an endofunctor preserves weak pullbacks iff it weakly preserves pullbacks.

greatest bisimulation on A and B iff the diagram below is a pullback in $\mathcal{X}$:

$$\begin{tikzcd} R \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] & UB \arrow[d, "!"] \\ UA \arrow[r, "!"] & UF1 \end{tikzcd}$$

Proof. The proof is straightforward. The main point is that ΩR is a weak pullback which gives the structure for R. For a details see the proof of proposition 1.2.2. ☐

Remark. The assumption that $\mathcal{X}$ has all pullbacks and they are weakly preserved by Ω could be weakened: Only the pullback in the diagram above is needed for the proof.

Remark. The existence of the terminal coalgebra is not really needed here, see corollary 1.2.3.

Remark. Note that the pullback is taken in $\mathcal{X}$ and not in $\mathcal{X}_\Omega$. This is important because limits in $\mathcal{X}_\Omega$ are usually more complicated to obtain than limits in $\mathcal{X}_\Omega$, see section 1.6.

Remark. This proposition gives one reason why the assumption of functors preserving weak pullbacks is useful: it serves as a technique to show the existence of a not uniquely determined structure map. Comparing this technique to the one using split epis considered before, one finds that the proof of proposition 1.1.13 makes (hidden) use of split epis: If P is a pullback of a cospan (f, g) and R a weak pullback of (f, g) then the unique morphism $R \to P$ is split epi (see Gumm and Schröder [42], theorem 4.5). Its right inverses give rise to the non unique morphisms into the weak pullback.

1.2 Cocongruences and Behavioural Equivalences

The aim being a general development of universal coalgebra without having recourse to the category of sets and without using that signatures preserve weak pullbacks or epis in the base category are split, we want to offer a better behaved alternative to the notion of bisimulation (see the discussion in section 1.1.4). We propose the notion of a cocongruence, defined as a cospan in $\mathcal{X}_\Omega$, and the notion of a behavioural equivalence, defined as an epi in $\mathcal{X}_\Omega$. (Recall that a morphism/span in $\mathcal{X}_\Omega$ is epi iff it is epi as a morphism/span in $\mathcal{X}$). The name 'cocongruence' was chosen due to the fact that cocongruences are dual to congruences for algebras as given in Rutten [108], section 13. Behavioural equivalences are essentially Aczel and Mendler's congruences but the term 'behavioural equivalence' seems to give a better intuition and does not conflict with other uses of 'congruence'.

As a consequence of our treatment we see why the property of a signature functor preserving weak pullbacks is convenient but not essential. If the signature preserves weak pullbacks, the kernel of a behavioural equivalence can be equipped with a coalgebra structure (proposition 1.2.14), i.e. it is a bisimulation in the sense of Aczel and Mendler. However, even if the kernel of a behavioural equivalence does not give rise to a bisimulation, it still gives us—as the name suggests—the right notion of behavioural equivalence (see example 1.2.5).

Recently these notions have also been investigated—in the case $\mathcal{X} = \mathbf{Set}$—in Wolter [125]. In that work, behavioural equivalences are called partitions and cocongruences are called compatible corelations.

1.2.1 Definitions

The basic idea of the definition of a cocongruence is to replace in the definition of a bisimulation the use of spans

$$A \xleftarrow{\;p\;} (R, \varrho) \xrightarrow{\;q\;} B$$

by the use cospans

$$A \xrightarrow{\;p\;} (R, \varrho) \xleftarrow{\;q\;} B.$$

In more detail: For $A, B \in \mathcal{X}_\Omega$, $R \in \mathcal{X}$, and arrows $p : UA \to R$, $q : UB \to R$ call (R, p, q) a cocongruence if

1. (R, p, q) is an epi cospan in $\mathcal{X}$,[12]
2. there is a structure map $\varrho : R \to \Omega R$ such that p, q are morphisms in $\mathcal{X}_\Omega$:

$$\begin{tikzcd} UA \arrow[r, "p"] \arrow[d, "\alpha"'] & R \arrow[d, "\varrho"'] & UB \arrow[l, "q"'] \arrow[d, "\beta"'] \\ \Omega UA \arrow[r, "\Omega p"] & \Omega R & \Omega UB \arrow[l, "\Omega q"'] \end{tikzcd}$$

[12] (R, p, q) is an epi cospan (= a epi 2-sink) in $\mathcal{X}$ iff for all $f, g \in \mathcal{X}$ with domain R it holds that $f \circ p = g \circ p$ and $f \circ q = g \circ q$ imply $f = g$.

The intuition behind this definition is that—due to morphisms in $\mathcal{X}_\Omega$ preserving behaviour—$p(a) = q(b)$ only if a, b have the same behaviour, that is, we may consider the cocongruence (R, p, q) as a relation between behavioural equivalent elements of the carriers of A and B.

Let us note some advantages of cocongruences compared to bisimulations.

- The largest cocongruence on two objects exists under rather general circumstances, e.g., if $\mathcal{X}$ has cointersections.[13] In particular, no special properties like epis being split or signatures preserving weak pullbacks are needed.
- In cases where largest cocongruence and largest bisimulation do not coincide, it is probably cocongruences (or behavioural equivalences) that will be preferred, see the discussion in examples 1.2.4 and 1.2.5.
- In contrast to bisimulations, the structure ϱ of a cocongruence is uniquely determined. This is due to (R, p, q) being an epi cospan in $\mathcal{X}$.
- Since the structure is uniquely determined and the forgetful functor preserves and reflects epi cospans,[14] cocongruences are just epi cospans in $\mathcal{X}_\Omega$.[15] Note that this gives a definition not mentioning the forgetful functor and the base category any more.[16] This will pay off in the treatment of universal coalgebra in section 1.4.

The last point gives rise to the actual definition:

Definition 1.2.1 (Cocongruences and Behavioural Equivalences). Let Ω be an endofunctor on a category $\mathcal{X}$ and $A, B \in \mathcal{X}_\Omega$. Then an epi cospan $(Q, p : A \to Q, q : B \to Q)$ in $\mathcal{X}_\Omega$ is called an Ω-*cocongruence* on A and B. An epi $e : A \to Q$ in $\mathcal{X}_\Omega$ is called an Ω-*behavioural equivalence* on A.

Remark. To explain this definition suppose $\mathcal{X} = \mathbf{Set}$. Given a cocongruence $(Q, p : A \to Q, q : B \to Q)$ (or a behavioural equivalence $e : A \to Q$) we define a relation $\sim_Q$ on $UA \times UB$ via $(a, b) \in \sim_Q \;\Leftrightarrow\; p(a) = q(b)$ (or $(a, b) \in \sim_Q \;\Leftrightarrow\; e(a) = e(b)$). Under the assumption that Ω preserves weak pullbacks, $\sim_Q$ is a bisimulation in the sense of the previous section (see proposition 1.2.2). Moreover, Q itself is the coalgebra which results from identifying the states related by $\sim_Q$.

Remark. The definition of behavioural equivalence is the obvious generalisation to arbitrary categories of what is called congruence in Aczel and Mendler [2]. We chose to use the term 'behavioural equivalence' instead for three reasons. First, the term 'congruence' conflicts with Rutten's definition mentioned above. Also, later in chapter 4 the slogan that the 'largest behavioural equivalence has to be congruence' would become cumbersome to express. Last, examples 1.2.4 and chapter 4 show the appropriateness of the term 'behavioural equivalence'.[17]

---

[13] For example, **Set** has cointersections. Also, categories of algebras usually have cointersections.

[14] That U preserves epi cospans is proved in the same way as U preserves epis, see Rutten [109]4.7.

[15] Dually, congruences in the sense of Rutten [109] are just mono spans in the category of algebras.

[16] Recall that, to achieve a similar simplification for bisimulations, we had to impose severe restrictions on the signature Ω.

[17] Still, 'observational equivalence' would be a good alternative.

Remark. To justify the term 'cocongruence' recall the notion of an Ω-congruence for algebras given in Rutten [109], section 13. Ω-cocongruences are precisely the Ω^{op}-congruences.

Cocongruences on two coalgebras A and B form the **category of cocongruences** denoted by $\mathsf{EpiCospan}(A, B)$ or $\mathsf{Cocong}(A, B)$. The morphisms in this category are just the usual morphisms of cospans in $\mathcal{X}_\Omega$: Given cocongruences (Q, p, q) and (Q', p', q') on coalgebras A and B, a morphism $f : (Q, p, q) \to (Q', p', q')$ is an arrow $f \in \mathcal{X}_\Omega$ such that $f \circ p = p'$ and $f \circ q = q'$. Note that such a morphism is epi. Also note that this definition is equivalent to defining the morphisms in $\mathsf{Cocong}(A, B)$ as arrows $f \in \mathcal{X}$ such that $f \circ p = p'$ and $f \circ q = q'$.

Behavioural equivalences on A form the **category of behavioural equivalences** denoted by $\mathsf{Epi}(A)$ or $\mathsf{BehEqu}(A)$. This is just the full subcategory of $A \downarrow \mathcal{X}_\Omega$ induced by the epis, i.e., given behavioural equivalences $p : A \to Q$ and $p' : A \to Q'$ a morphisms $f : p \to p'$ is a morphisms $f \in \mathcal{X}_\Omega$ (or, equivalently, a morphism $f \in \mathcal{X}$) such that $f \circ p = p'$.

Next, we show that, in the case that the signature Ω preserves weak pullbacks, cocongruences are indeed bisimulations in the standard sense. The proof is a standard argument, compare with the proof of proposition 1.1.13.

Proposition 1.2.2. *Let $\mathcal{X}$ be a category with pullbacks and Ω a functor on $\mathcal{X}$ preserving weak pullbacks. Then Ω-cocongruences and Ω-behavioural equivalences give rise to Ω-bisimulations via pullbacks in $\mathcal{X}$.*

Proof. Let (R, p, q) be an Ω-cocongruence (for behavioural congruences $e : A \to R$ do the following reasoning with $p = q = e$). Define $(\sim_R, f, g)$ as the pullback

$$\begin{tikzcd}
\sim_R \arrow[r, dotted, "g"] \arrow[d, dotted, "f"'] & UB \arrow[d, "q"] \\
UA \arrow[r, "p"'] & R
\end{tikzcd}$$

To see that it is a bisimulation we have to find a structure map σ. Consider

$$\begin{tikzcd}
\Omega \sim_R \arrow[rrr, "\Omega g"] \arrow[ddd, "\Omega f"'] & & & \Omega UB \arrow[ddd, "\Omega q"] \\
& \sim_R \arrow[ul, dotted, "\sigma"'] \arrow[r, "g"] \arrow[d, "f"'] & UB \arrow[ur, "\beta"] \arrow[d, "q"] & \\
& UA \arrow[r, "p"'] \arrow[dl, "\alpha"'] & R \arrow[dr, "\rho"] & \\
\Omega UA \arrow[rrr, "\Omega p"'] & & & \Omega R
\end{tikzcd}$$

Since Ω preserves weak pullbacks, the outer diagram is a weak pullback. Hence σ exists and $(\sim_R, f, g)$ is a bisimulation. $\square$

Remark. In case that R is a behavioural equivalence on A (i.e. $p = q$ in the proof above), $(\sim_R, f, g)$ is a bisimulation equivalence: In categories with pullbacks kernel pairs can be considered as equivalence relations, see Taylor [114], proposition 5.6.4.

Corollary 1.2.3. *Let Ω be a weak pullback preserving functor on a category $\mathcal{X}$ that has cointersections. Then the largest bisimulation exists.*

We have just shown that, in case that the signature Ω preserves weak pullbacks, cocongruences and behavioural equivalences are indeed bisimulations. Next we want to discuss an example showing the difference of theses notions.

1.2.2 An Example

We will consider the example of a not weak-pullback-preserving functor given by Aczel and Mendler [2] to explain the difference of bisimulation and behavioural equivalence (which was called 'congruence' in their paper). In particular we give two examples showing that, respectively,

bisimulations may fail to capture behavioural equivalence

and that

a bisimulation may fail to give rise to
a smallest bisimulation equivalence containing it.

Example 1.2.4. The functor $AM : \mathbf{Set} \to \mathbf{Set}$ is defined on objects as $AM(X) = \{(x, y, z) \in X^3 : |\{x, y, z\}| \leq 2\}$ and on morphisms in the obvious way. Aczel and Mendler consider the coalgebra $\alpha : UA \to AMUA$ where $UA = \{s_0, s_1\}$ and

$$\begin{aligned}\alpha(s_0) &= (s_0, s_0, s_1)\\ \alpha(s_1) &= (s_0, s_1, s_1)\end{aligned}$$

Recall that coalgebras for the functor $\Omega(X) = X^3$ can be considered as a kind of automata taking inputs from a three element set 3 (and having no output). For example, letting $3 = \{0, 1, 2\}$, $\alpha(s_0) = (s_0, s_0, s_1)$ can be interpreted as follows: In state s_0 one goes to state s_0 if one receives input 0 or 1 and one goes to state s_1 if one receives input 2. One can now view the cardinality restriction on the signature AM as imposing a constraint on the *implementation* of these automata: in every state at least two inputs have to give rise to the same successor state.

For an example of a behavioural equivalence that is not a bisimulation consider the coalgebra $\kappa : UQ \to AMUQ$ given by $UQ = \{*\}$ and κ the identity (Q is also the terminal coalgebra). The largest behavioural equivalence on A is given by

$$! : A \to Q$$

That is, $s_0, s_1 \in UA$ are behavioural equivalent. But they are not bisimilar because a bisimulation relating s_0, s_1 would have to have a structure that maps (s_0, s_1) as depicted in the diagram

$$\begin{tikzcd}
 & (s_0, s_1) \arrow[dl, "0"'] \arrow[d, "1"] \arrow[dr, "2"] & \\
(s_0, s_0) & (s_0, s_1) & (s_1, s_1)
\end{tikzcd}$$

which is not possible due to the cardinality constraint. In fact, there is no largest bisimulation on A. Hence bisimulations may fail to capture behavioural equivalence. This phenomenon is essentially due to the fact that the signature AM imposes an implementation constraint which cannot be satisfied by a largest bisimulation. On the other hand, from a behavioural point of view, this implementation constraint is not observable which is reflected by the fact that the largest behavioural equivalence does exist.

Example 1.2.5. The signature AM can also be used to show another deficiency of bisimulations, namely that a bisimulation may fail to give rise to a smallest bisimulation equivalence containing it. The following gives an example (which also shows that the composition of bisimulations may not be a bisimulation).

Consider the coalgebra $\beta : UB \to AMUB$ where $UB = \{s_0, s_1, s_2\}$ and

$$\begin{aligned}
\beta(s_0) &= (s_0, s_1, s_0) \\
\beta(s_1) &= (s_1, s_1, s_1) \\
\beta(s_2) &= (s_2, s_1, s_1)
\end{aligned}$$

There is a bisimulation relating s_0, s_1 and s_1, s_2. But there is no bisimulation equivalence containing it because there is no bisimulation relating s_0, s_2. Such a bisimulation would have to have a structure map that maps (s_0, s_2) as depicted in the diagram:

$$\begin{tikzcd}
 & (s_0, s_2) \arrow[dl, "0"'] \arrow[d, "1"] \arrow[dr, "2"] & \\
(s_0, s_2) & (s_1, s_1) & (s_0, s_1)
\end{tikzcd}$$

which is not a structure for AM.

1.2.3 Largest Behavioural Equivalences

Here we describe how union can be defined for behavioural equivalences and cocongruences and investigate when a largest behavioural equivalence exists. It will turn out that the existence of a largest behavioural equivalence does not depend on assumptions like the signature preserving weak pullbacks, epis in the base being split, or the existence of a final coalgebra. Instead, we only assume that the base has cointersections (see appendix A.1 or definition 1.4.2).

Definition 1.2.6 (union of behavioural equivalences/cocongruences).
Let Ω be a functor on $\mathcal{X}$ and $U : \mathcal{X}_\Omega \to \mathcal{X}$ the forgetful functor. Suppose that $\mathcal{X}$ has cointersections. Then the union of behavioural equivalences and the union of cocongruences is given by their colimit.

Remark. The notion 'union' in this context is a bit misleading but convenient. For example, in the case $\mathcal{X} = \mathbf{Set}$, the kernel of the union of behavioural equivalences is not given by the union of the kernels but by the equivalence generated by the union of the kernels.

Proposition 1.2.7. *Let Ω be a functor on $\mathcal{X}$, $U : \mathcal{X}_\Omega \to \mathcal{X}$ the forgetful functor and let $\mathcal{X}$ have cointersections. Let $A, B \in \mathcal{X}_\Omega$ and $D : I \to \mathsf{Cocong}(A, B)$ (or $D : I \to \mathsf{BehEqu}(A)$) be a (possibly large) diagram. Then the colimit of D exists.*

Corollary 1.2.8. *Under the assumptions of the proposition, $\mathsf{Cocong}(A, B)$ and $\mathsf{BehEqu}(A)$ are—up to equivalence of categories—complete lattices.*

Consequently, we can now define the notion of a largest behavioural equivalence.

Definition 1.2.9 (largest behavioural equivalence). Let Ω be a functor on $\mathcal{X}$ and $U : \mathcal{X}_\Omega \to \mathcal{X}$ the forgetful functor. Suppose that $\mathcal{X}$ has cointersections. The largest behavioural equivalence on a coalgebra A is the union of all behavioural equivalences on A.

Remark. Largest behavioural equivalences exist in $(\mathbf{Set^n})_\Omega$ for arbitrary functors Ω on $\mathbf{Set^n}$.

Remark. One can ask whether the largest behavioural equivalence $e : B \to A$ on a coalgebra B is extensional, that is, whether for all coalgebras C there is at most one coalgebra morphism $C \to A$. Under appropriate assumptions this is indeed the case, see proposition 1.4.10.

1.2.4 Bisimulation Equivalences vs. Behavioural Equivalences

Proposition 1.2.2 showed that if the signature preserves weak pullbacks then behavioural equivalences are bisimulation equivalences. Here we further investigate under what circumstances we get a bijection between these two notions (see corollary 1.2.16). We will do this by setting up an adjunction between $\mathsf{Bisim}(A, B)$ and $\mathsf{Cocong}(A, B)$ (and between $\mathsf{Bisim}(A, A)$ and $\mathsf{BehEqu}(A)$) and asking on what subcategories this adjunction becomes an equivalence. When bisimulations and cocongruences are considered as certain spans and cospans of the base category, this adjunction (or Galois connection) is just the familiar one between pullbacks and pushouts, see e.g. Herrlich and Strecker [54], exercise 21D.

It turns out that for the development of this section it is convenient to use the categories $\mathsf{Span}(A, B)$ of spans between A and B in $\mathcal{X}_\Omega$, $\mathsf{Cospan}(A, B)$ of cospans between A and B in $\mathcal{X}_\Omega$. Objects are (co)spans (R, p, q) and morphisms $f : (R, p, q) \to (R', p', q')$ are morphisms $f : R \to R'$ with $p' \circ f = p$, $q' \circ f = q$ ($f \circ p = p'$, $f \circ q = q'$).

Define two operations $Po : \mathsf{Span}(A, B) \to \mathsf{Cospan}(A, B)$ mapping each span in $\mathsf{Span}(A, B)$ to its pushout and $Pb : \mathsf{Cospan}(A, B) \to \mathsf{Span}(A, B)$ mapping each cospan to its pullback. In some cases these operations can also be defined for bisimulations and cocongruences: Po : $\mathsf{Bisim}(A, B) \to \mathsf{Cocong}(A, B)$ maps a bisimulation to its pushout in $\mathcal{X}$, $Pb : \mathsf{Cocong}(A, B) \to \mathsf{Bisim}(A, B)$ maps a cocongruence to its pullback in $\mathcal{X}$. (Note that the operators Pb and Po are defined here on the base category which helps for calculations in concrete examples.)

Proposition 1.2.10. *Let $\mathcal{X}$ be a category with pushouts and pullbacks. Then the operations $Po : \mathsf{Span}(A,B) \to \mathsf{Cospan}(A,B)$, $Pb : \mathsf{Cospan}(A,B) \to \mathsf{Span}(A,B)$ defined above are adjoint functors, $Po \dashv Pb$. Moreover, $PbPo$ is a closure operator on $\mathsf{Span}(A,B)$ and Po, Pb are an equivalence on the subcategories $PbPo(\mathsf{Span}(A,B))$ and $PoPb(\mathsf{Cospan}(A,B))$.*

Proof. Both Po and Pb are functorial. We show this for Po. Let $m : (R_1, p_1, q_1) \to (R_2, p_2, q_2) \in \mathsf{Span}(A,B)$, and $(Q_i, r_i, s_i) = Po(R_i, p_i, q_i)$, $i = 1, 2$. Define $Po(m)$ as the dotted arrow in the following diagram

$$\begin{tikzcd}
 & & \bullet \arrow[dr, "s_2"] \arrow[drr, "s_1"] & & \\
R_1 \arrow[r, "m"] \arrow[urr, "q_1"] \arrow[drr, "p_1"'] & R_2 \arrow[ur, "q_2"] \arrow[dr, "p_2"'] & & Q_2 & Q_1 \arrow[l, dotted] \\
 & & \bullet \arrow[ur, "r_2"] \arrow[urr, "r_1"'] & &
\end{tikzcd}$$

(Q_2, r_2, s_2) is a cocone for (R_1, p_1, q_1). Since (Q_1, r_1, s_1) is a pushout of (R_1, p_1, q_1), $Po(m)$ is uniquely defined. That Po preserves identity and composition follows from the uniqueness property. Functoriality of Pb is shown by a dual argument.

Due to the universal properties of pushouts and pullbacks, it holds that $\mathsf{Hom}(Po(p,q),(r,s)) \simeq \mathsf{Hom}((p,q), Pb(r,s))$, that is, Po is left-adjoint to Pb.

It follows that $PbPo$ defines a monad on $\mathsf{Span}(A,B)$ and that $PoPb$ defines a comonad on $\mathsf{Cospan}(A,B)$. Both (co)monads are idempotent because of $PoPbPo \simeq Po$ and $PbPoPb \simeq Pb$. It follows that they define (co)reflective subcategories of $\mathsf{Span}(A,B)$ and $\mathsf{Cospan}(A,B)$ and that Po and Pb are an equivalence on these subcategories. It also follows from the idempotency of the (co)monads that the operator $PbPo$ is a closure operator on $\mathsf{Span}(A,B)$ (and that $PoPb$ is an interior operator on $\mathsf{Cospan}(A,B)$). □

Remark. This proposition also holds if we replace the category of spans by mono spans and the category of cospans by epi cospans.

Remark. In the case $\mathcal{X} = \mathbf{Set}$ it holds that $PoPb \simeq \mathrm{Id}_{\mathsf{EpiCospan}(A,B)}$. The closure operator $PbPo$ has the following explicit description. Let (R, p, q) be a relation (with p, q the natural projections). Then $PbPo(R, p, q)$ is the smallest relation $\bar{R}$ containing R that is closed under the condition (let $a_1, a_2 \in UA$, $b_1, b_2 \in UB$)

$$\begin{tikzcd}
a_1 \arrow[r, dash] \arrow[d, dash] & b_1 \arrow[d, dash, dotted] \\
b_2 \arrow[r, dash] & a_2
\end{tikzcd}$$

$$a_1 \bar{R} b_1 \;\&\; a_1 \bar{R} b_2 \;\&\; a_2 \bar{R} b_2 \;\Rightarrow\; a_2 \bar{R} b_1 .$$

The proposition above also holds for bisimulations and cocongruences in case that the signature preserves weak pullbacks:

Proposition 1.2.11. *Let $\mathcal{X}$ be a category with pushouts and pullbacks and Ω be a weak pullback preserving functor on $\mathcal{X}$. Then the operations $Po : \mathsf{Bisim}(A, B) \to \mathsf{Cocong}(A, B)$ $Pb : \mathsf{Cocong}(A, B) \to \mathsf{Bisim}(A, B)$ defined above are adjoint, $Po \dashv Pb$. Moreover, $PbPo$ is a closure operator on Bisim and Po, Pb are an equivalence on the subcategories $PbPo(\mathsf{Span}(A, B))$ and $PoPb(\mathsf{Cospan}(A, B))$.*

Proof. Using proposition 1.2.10 we only have to check that the operations Pb and Po are well-defined. First note that the images of Pb are mono in $\mathcal{X}$ and the images of Po are epi in $\mathcal{X}$. It remains to show that these images can always be equipped with an appropriate structure map. This is obvious for Po (due to the universal property of pushouts). For Pb use that Ω preserves weak pullbacks and argue like in the proof of proposition 1.2.2. $\square$

Remark. In the case $\mathcal{X} = \mathbf{Set}$, the operator $PbPo$ has the same explicit description as in the remark to the previous proposition. Also, $PoPb \simeq \mathrm{Id}_{\mathsf{Cocong}(A,B)}$.

One way to summarise the proposition above is to say that—under the assumptions of the proposition—every cocongruence determines a bisimulation. To be more precise we give a definition.

Definition 1.2.12 (bisimulation determined by a cocongruence). Let $A, B \in \mathcal{X}_\Omega$ and (p, q) a cocongruence on A and B. If the pullback (r, s) of (p, q) in $\mathcal{X}$ exists and is a bisimulation then (r, s) is *the bisimulation determined by* (p, q).

We can describe the relationship between behavioural equivalences and bisimulations quite analogously. This time we get an adjunction between $\mathsf{Span}(A, A)$ and $A{\downarrow}\mathcal{C}$ (see appendix A.1). Let $Ceq : \mathsf{Span}(A, A) \to (A{\downarrow}\mathcal{C})$ be the operation mapping a span to its coequaliser and $Ker : (A{\downarrow}\mathcal{C}) \to \mathsf{Span}(A, A)$ be the operation mapping a morphism $A \to \bullet$ to its kernel pair. In some cases these operations can also be defined for bisimulations and behavioural equivalences: $Ceq : \mathsf{Bisim}(A, A) \to \mathsf{BehEqu}(A)$ gives the coequaliser of a bisimulation (R, p, q) in the base category $\mathcal{X}$ and $Ker : \mathsf{BehEqu}(A) \to \mathsf{Bisim}(A, A)$ calculates the kernel pair in $\mathcal{X}$ of a morphism in $\mathcal{X}_\Omega$. As before we get:

Proposition 1.2.13. *Let $\mathcal{X}$ be a category with pullbacks and coequalisers and $A \in \mathcal{X}$. The functors $Ceq : \mathsf{Span}(A, A) \to (A{\downarrow}\mathcal{C})$ and $Ker : (A{\downarrow}\mathcal{C}) \to \mathsf{Span}(A, A)$ defined above are adjoint, $Ceq \dashv Ker$. Moreover, $KerCeq$ is a closure operator and Ceq and Ker are an equivalence on the subcategories $KerCeq(\mathsf{Span}(A, A))$ and $CeqKer(A{\downarrow}\mathcal{C})$.*

Remark. This proposition also holds if we replace the category of spans by mono spans, the category $A{\downarrow}\mathcal{C}$ by $\mathsf{Epi}(A)$.

Proposition 1.2.14. *Let $\Omega : \mathcal{X} \to \mathcal{X}$ be a weak pullback preserving functor, $A \in \mathcal{X}_\Omega$, and $Ceq : \mathsf{Bisim}(A, A) \to \mathsf{BehEqu}(A)$, $Ker : \mathsf{BehEqu}(A) \to \mathsf{Bisim}(A, A)$ as above. Then $Ker \dashv Ceq$. Moreover, $KerCeq$ is the closure operator mapping a bisimulation to the least bisimulation equivalence containing it and Ker, Ceq are an equivalence on the subcategories $KerCeq(\mathsf{Bisim}(A, A))$ and $CeqKer(\mathsf{BehEqu}(A))$.*

One way to summarise the proposition above is to say that—under the assumptions of the proposition—every behavioural equivalence determines a bisimulation. To be more precise we give a definition.

Definition 1.2.15 (bisimulation determined by a behavioural equivalence). Let $A \in \mathcal{X}_\Omega$ and e a behavioural equivalence on A. If the kernel pair (r, s) of e in $\mathcal{X}$ exists and is a bisimulation then (r, s) is *the bisimulation determined by* e.

It is not difficult to see that, in case of $\mathcal{X} = \mathbf{Set}$, $KerCeq$ is the closure operator mapping a bisimulation to the least bisimulation equivalence containing it and that $CeqKer \simeq \mathrm{Id}_{\mathsf{BehEqu}(A)}$. It follows:

Corollary 1.2.16. *Let Ω be a weak pullback preserving endofunctor on* $\mathbf{Set}$. *Then the operations Po and Pb define a bijection between bisimulation equivalences and behavioural equivalences.*

Thus, in the case of a weak pullback preserving signature on **Set** we have the choice of whether we want to describe bisimulation equivalences as certain spans or as epis. But the description of equivalences as epis is simpler and generalises better to not weak-pullback-preserving signatures over other base categories.

1.3 Factorisation Systems for Coalgebras

It is one of the claims of this thesis that central technical notions to deal with coalgebras are *factorisation systems* and *factorisation structures for sinks.* Applications include the existence of limits in categories of coalgebras (see section 1.6), the existence of cartesian liftings in cofibrations of coalgebras (see [71]), proofs of generalised co-Birkhoff theorems (see chapter 2.5), and an axiomatic treatment of universal coalgebra independent from special properties of the category of sets (see section 1.4).

The reason for the use of factorisation structures is that in categories of coalgebras not based on sets, it is generally not clear how to define the notion of a subcoalgebra and the notion of a union of subcoalgebras.[18] As it turns out (see e.g. example 1.3.11), it is enough to axiomatically describe the properties that subcoalgebras and unions of subcoalgebras have to have. This is done using factorisation structures for sinks. This approach is inspired by the fact that factorisation structures have been used successfully to investigate (amongst others) categories of algebras, see Adámek, Herrlich, Strecker [4]. Our development will apply these ideas to coalgebras.

The main purpose of this section is to show that categories of coalgebras enjoy factorisation systems/structures under rather general circumstances. We show how to lift factorisation systems/structures in the base category to the category of coalgebras. Since the development is somewhat technical and probably not of independent interest the reader might want to skip this section and only note example 1.3.1, theorem 1.3.10, example 1.3.11, and corollary 1.3.15.

1.3.1 Factorisation Systems

This section shows how to lift factorisation systems for the base category to factorisation systems for the category of coalgebras. We conclude by applying these results to the base category of sets. The definition of a factorisation system is given in section A.4.

Let us first give an example, introduced in Gumm and Schröder [42], that shows that monos in $\mathbf{Set}_\Omega$ are not necessarily (isomorphic to) subcoalgebras.[19]

Example 1.3.1 (monos in $\mathbf{Set}_\Omega$ that are not injective). Recall example 1.2.4. The morphism ! is mono in $\mathbf{Set}_{\mathcal{AM}}$ but not injective.

Consequently, monos are in general not (isomorphic to) subcoalgebras. So one question is, whether we can characterise the class of morphisms that are isomorphic to subcoalgebras in $\mathbf{Set}_\Omega$ (theorem 1.3.10(1)). Another and more general question is, how to characterise subcoalgebras in categories of coalgebras over base categories $\mathcal{X}$ other than **Set**. In general, there is no satisfying way to capture categorically the notion of a sub-object (see, e.g., the discussion in Adámek, Herrlich, Strecker [4], chapters 7 and 8). For our purposes, however, it is enough to axiomatise—using the notion of a factorisation system—those properties of sub-objects that are needed for our development. We therefore assume a factorisation system (E, M) for $\mathcal{X}$ where the arrows in M are considered to be the sub-objects in $\mathcal{X}$. We then

[18] In $\mathbf{Set}_\Omega$, a morphism is (isomorphic to) a subcoalgebra if it is injective. But 'injective' is defined in terms of elements of sets. See also appendix A.2.

[19] A morphism $m : B \to A$ is isomorphic to a subcoalgebra $i : A_0 \hookrightarrow A$ iff there is an iso $f : B \to A_0$ such that $i \circ f = m$. i, or sometimes A_0, is called the image of m.

consider as a subcoalgebra $m : B \to A$ in $\mathcal{X}_\Omega$ all morphisms m such that $Um \in M$. In order for this approach to make sense, we expect the class of morphisms $U^{-1}M$ in $\mathcal{X}_\Omega$ to be part of a factorisation system as well. It is therefore the task of this section to investigate some conditions under which a factorisation system in the base category $\mathcal{X}$ can be lifted to the category of coalgebras $\mathcal{X}_\Omega$. The first step is to define when a forgetful functor creates factorisations.

Definition 1.3.2 (creating factorisations). Let $U : \mathcal{C} \to \mathcal{X}$ be a functor and (E, M) a factorisation system for $\mathcal{X}$. *U creates factorisations (w.r.t. (E, M))* iff for all $f : A \to B \in \mathcal{C}$ and every factorisation $UA \overset{e}{\to} X \overset{m}{\to} UB$ of Uf there is a unique $C \in \mathcal{C}$ and unique morphsims $e' : A \to C$, $m' : C \to B$ in $\mathcal{C}$ such that $UC = X$, $Ue' = e$, $Um' = m$, and $m' \circ e' = f$.

Remark. A functor creating factorisations is faithful.

Remark. That U creates factorisations expresses that we can caculate factorisations in $\mathcal{C}$ as factorisations in $\mathcal{X}$. But it does not imply that $(U^{-1}E, U^{-1}M)$ is a factorisation system since factorisations need not be unique up to *isomorphisms in $\mathcal{C}$* (they are, however, unique up to isomorphisms in $\mathcal{X}$). See proposition 1.3.6 for a condition guaranteeing that $(U^{-1}E, U^{-1}M)$ is a factorisation system.

The next two propositions give simple criterions for a forgetful functor to create factorisations.

Proposition 1.3.3. *Let $\Omega : \mathcal{X} \to \mathcal{X}$ be a functor, $U : \mathcal{X}_\Omega \to \mathcal{X}$ the forgetful functor and (E, M) a factorisation system for $\mathcal{X}$. Then U creates factorisations if $\Omega M \subset M$.*

Proof. Let $f : A \to B \in \mathcal{X}_\Omega$ and $UA \overset{e}{\to} X \overset{m}{\to} UB$ a factorisation of Uf. We have to show that X can be equipped uniquely with a coalgebra structure ξ such that e and m become coalgebra morphisms. This follows from unique diagonalisation and $\Omega M \subset M$ (let α, β be the structures of A, B, respectively):

$$\begin{tikzcd}
UA \arrow[r, "e"] \arrow[d, "\Omega e \circ \alpha"'] & X \arrow[d, "\beta \circ m"] \arrow[dl, dotted, "\xi"'] \\
\Omega X \arrow[r, "\Omega m"'] & \Omega UB
\end{tikzcd}$$

∎

Proposition 1.3.4. *Let $\Omega : \mathbf{Set} \to \mathbf{Set}$ be a functor. Then the forgetful functor $U : \mathcal{X}_\Omega \to \mathcal{X}$ creates factorisations w.r.t. $(Epi, Mono)$.*

Proof. Consider the proof of the previous proposition. If X is not empty, then m is a split mono (i.e. has a left inverse), hence Ωm is mono and the diagonal exists. If X is empty, the function with the empty graph is the unique diagonal. ∎

To show that factorisation systems are lifted by U one needs that morphisms in E are final or morphisms in M are initial. This assumption is usually satisfied in categories of coalgebras as shown by the next proposition.

Proposition 1.3.5. *Let Ω be an endofunctor on $\mathcal{X}$. A morphism in $\mathcal{X}_\Omega$ is final if it is epi in $\mathcal{X}$. A morphism g in $\mathcal{X}_\Omega$ is initial if Ωg is mono in $\mathcal{X}$.*

Proof. The proof is straight forward and can be found in Rutten [109] 2.4. $\square$

Remark. In $\mathbf{Set}_\Omega$ epis are final and strong monos are initial (for strong monos this will follow from theorem 1.3.10).

The next proposition shows that factorisation systems can be lifted under rather general circumstances.

Proposition 1.3.6. *Let $\Omega : \mathcal{X} \to \mathcal{X}$ be a functor, $U : \mathcal{X}_\Omega \to \mathcal{X}$ the forgetful functor, (E, M) a factorisation system for $\mathcal{X}$, and U creating these factorisations. Moreover, suppose that morphisms in $(U^{-1}E)$ are final or morphisms in $(U^{-1}M)$ are initial. Then $(U^{-1}E, U^{-1}M)$ is a factorisation system for $\mathcal{X}_\Omega$.*

Proof. $U^{-1}E, U^{-1}M$ are closed under isos. The existence of factorisations follows from U creating them. Concerning the unique diagonalisation, the diagonal exists in $\mathcal{X}$. It remains to show that the diagonal gives rise to a morphism in $\mathcal{X}_\Omega$. This follows from morphisms in $(U^{-1}E)$ being final or morphisms in $(U^{-1}M)$ being initial. $\square$

The following theorem shows that we can lift factorisation systems if $\mathcal{X}$ is a category of 'sets with structure' that has as a factorisation system which is lifted from **Set**.

Theorem 1.3.7. *Let $V : \mathcal{X} \to$ **Set** be a functor that creates factorisations w.r.t. the factorisation system $(Epi, Mono)$ for* **Set**. *Let Ω be a functor on $\mathcal{X}$ and $U : \mathcal{X}_\Omega \to \mathcal{X}$ the forgetful functor. Then U and UV create factorisations and $((UV)^{-1}(Epi), (UV)^{-1}(Mono))$ is a factorisation system for $\mathcal{X}_\Omega$.*

Proof. That U and UV create factorisations is proved as for proposition 1.3.3. That $((UV)^{-1}(Epi), (UV)^{-1}(Mono))$ is a factorisation system follows as for proposition 1.3.6 since morphisms in $(UV)^{-1}(Epi)$ are final (because they are epi in **Set**). $\square$

Remark. The theorem applies, for example, when $\mathcal{X}$ is a category of algebras for a functor or a monad.

Next, we show that in special cases the classes $U^{-1}E$, $U^{-1}M$ can be characterised without referring to U.

Theorem 1.3.8. *Let Ω be an endofunctor on $\mathcal{X}$ and let $U : \mathcal{X}_\Omega \to \mathcal{X}$ create factorisations w.r.t. $(E, M) = (Epi(\mathcal{X}), Mono(\mathcal{X}))$. Then $(U^{-1}E, U^{-1}M)$ is a factorisation system for $\mathcal{X}_\Omega$ and, moreover, $U^{-1}E = Epi(\mathcal{X}_\Omega)$ and $U^{-1}M = StrongMono(\mathcal{X}_\Omega) = ExtrMono(\mathcal{X}_\Omega)$.*

Proof. $U^{-1}Epi(\mathcal{X}) = Epi(\mathcal{X}_\Omega)$ holds because U preserves and reflects epis (proposition A.3.2). The two propositions above now imply that $(U^{-1}E, U^{-1}M)$ is a factorisation system for $\mathcal{X}_\Omega$. It remains to show[20] $ExtrMono(\mathcal{X}_\Omega) \subset U^{-1}Mono(\mathcal{X}) \subset StrongMono(\mathcal{X}_\Omega)$.
First, let $h : A \to B$ in $\mathcal{X}_\Omega$ be an extremal mono. h factors as $h = e \circ m$ with Ue epi and Um mono (since U creates factorisations). Since h is extremal, it follows that e is iso, therefore Uh mono.

[20]Recall $StrongMono \subset ExtrMono$, see appendix A.2.

Second, $U^{-1}Mono(\mathcal{X}) \subset StrongMono(\mathcal{X}_\Omega)$ is immediate from $(Epi(\mathcal{X}_\Omega), U^{-1}Mono(\mathcal{X}))$ being a factorisation system for $\mathcal{X}_\Omega$. □

Remark. We have shown a more general fact: Let $\mathcal{C}$ be a category with a factorisation system (Epi, N) such that $N \subset Mono$. Then $N = ExtrMono = StrongMono$.

Remark. Extremal monos in $\mathcal{X}_\Omega$ usually are monos in $\mathcal{X}$. More precisely, $ExtrMono(\mathcal{X}_\Omega) \subset U^{-1}M$ holds if $E \subset Epi(\mathcal{X})$ and $M \subset Mono(\mathcal{X})$ and U creates factorisations w.r.t. (E, M).

Remark. A coalgebra morphism which is mono in $\mathcal{X}$ is strong mono in $\mathcal{X}_\Omega$ whenever $(Epi(\mathcal{X}), Mono(\mathcal{X}))$ is a factorisation system for $\mathcal{X}$.

Another special case are base categories that have $(Epi, RegMono)$ as a factorisation system.

Theorem 1.3.9. *Let Ω be an endofunctor on a category $\mathcal{X}$ with pullbacks and equalisers and equalisers in $\mathcal{X}$ are split. Then $U : \mathcal{X}_\Omega \to \mathcal{X}$ creates factorisations w.r.t. $(Epi(\mathcal{X}), RegMono(\mathcal{X}))$. Moreover, $U^{-1}Epi(\mathcal{X}) = Epi(\mathcal{X}_\Omega)$ and $U^{-1}RegMono(\mathcal{X}) = RegMono(\mathcal{X}_\Omega)$.*

Proof. Since $\mathcal{X}$ has pushouts and equalisers, regular monos in $\mathcal{X}$ are the equalisers of their cokernel pair, hence split, hence preserved by Ω which implies that U creates factorisations (proposition 1.3.3).
The first equation is clear. For the second equation recall how $(Epi, RegMono)$-factorisations are calculated in a category with equalisers and pushouts (see proposition A.4.4) and use that U preserves and reflects pushouts and equalisers of cokernel pairs (proposition A.3.5). □

Remark. The theorem still holds if we only require $\mathcal{X}$ to have cokernel pairs and equalisers of cokernel pairs and these equalisers are split.

As a corollary we can now draw some conclusions on coalgebras over sets.

Theorem 1.3.10. *In $\mathbf{Set}_\Omega$ the following holds.*

1. *$(Epi, StrongMono)$ is a factorisation system. Moreover, Epi contains precisely the surjective coalgebra morphisms, StrongMono contains precisely the injective morphisms, and $(Epi, StrongMono)$-factorisations are calculated as $(Epi, Mono)$-factorisations in* **Set**.

2. *$ExtrMono = StrongMono = RegMono$.*

3. *If Ω preserves weak pullbacks then $Mono = StrongMono$.*

Proof. (1) and the first equation of (2) follow from the propositions and the first theorem. The second equation of (2) follows from the second theorem. (3) follows from proposition A.3.4 □

Remark. As the example of the beginning of this subsection shows, monos in $\mathbf{Set}_\Omega$ need not be extremal (otherwise they would be monos in **Set** by theorem 1.3.8.)

1.3.2 Factorisation Structures for Sinks

As indicated already, factorisation structures for sinks (see appendix A.5 for definitions and standard results) will be used to handle abstractly the notions of subcoalgebra and union of subcoalgebras.

The use that we will make of sinks can be illustrated by a typical example:

Example 1.3.11. Coalgebras cofree for a class $\mathcal{B} \subset \mathbf{Set}_\Omega$. (Rutten [108]). Suppose we are given a class of coalgebras $\mathcal{B} \subset \mathbf{Set}_\Omega$ for a functor Ω bounded by $C \in \mathbf{Set}$ (see definition 1.5.1) and we want to characterise the covariety generated by $\mathcal{B}$, i.e., the closure of $\mathcal{B}$ under sums, homomorphic images, and subcoalgebras. The wanted characterisation (given by Rutten [108], theorem 15.1) uses the notion of a coalgebra $F_\mathcal{B}C$ cofree for $\mathcal{B}$ which is defined as follows: The subcoalgebra $F_\mathcal{B}C$ of the cofree coalgebra FC is called *cofree for $\mathcal{B}$* iff for all $B \in \mathcal{B}$ every $f : B \to FC$ factors through $F_\mathcal{B}C \hookrightarrow FC$:

$$\begin{tikzcd} FC & F_{\mathcal{B}}C \arrow[l, hook'] \\ & B \arrow[ul, "f"] \arrow[u, dotted] \end{tikzcd}$$

Once the coalgebra $F_\mathcal{B}C$ is given, the covariety generated by $\mathcal{B}$ is characterised by the factorisation condition shown in the diagram. The question now is whether $F_\mathcal{B}C$ exists and how it can be constructed. The answer is that $F_\mathcal{B}C$ does exist and the construction is as follows. Take FC and let $(s_i : B_i \to FC)$ be the class of all morphisms in $\mathbf{Set}_\Omega$ with domain in $\mathcal{B}$. Then $F_\mathcal{B}C$ can be defined as the union of all images of the s_i. This construction works in **Set** because, in this case, we can calculate images and union of images using elements of sets. But, regarded more abstractly, what we actually need to obtain $F_\mathcal{B}C$ is the following:

the morphisms (s_i) factor uniquely as $m \circ (e_i)$ for some injective m.

We can then define $F_\mathcal{B}C$ as the domain of m. (It is a good exercice to check that definition A.5.1 together with basic properties of factorisation structures guarantee indeed that the domain of m is cofree for $\mathcal{B}$.)

We have investigated in the previous section how to lift factorisation systems from the base category to the category of coalgebras. Assuming that the category of coalgebras is wellpowered and has small coproducts we can extend this factorisation system to a factorisation structure for sinks (see proposition A.5.5). But we would like to know, moreover, that factoring sinks can be done in the base category. We therefore define, similar to the previous section, when a functor creates factorisations of sinks.

Definition 1.3.12 (creating factorisations of sinks). Let $U : \mathcal{C} \to \mathcal{X}$ be a functor and $\mathcal{X}$ be an $(\mathcal{E}, M)$-category. *U creates $(\mathcal{E}, M)$-factorisations* iff for all sinks $(s_i : A_i \to B)$ in $\mathcal{C}$ and all $(\mathcal{E}, M)$-factorisations $(Us_i) = m \circ (e_i)$ there is a unique $C \in \mathcal{C}$ and unique morphisms $e'_i : A_i \to C$, $m' : C \to B$ in $\mathcal{C}$ such that $Ue'_i = e_i$ and $Um' = m$.

We can continue as for factorisation systems:

Proposition 1.3.13. *Let Ω be a functor on $\mathcal{X}$ and $\mathcal{X}$ be an $(\mathcal{E}, M)$-category.*

1. *$U : \mathcal{X}_\Omega \to \mathcal{X}$ creates factorisations for sinks if $\Omega M \subset M$.*
2. *Sinks in $\mathcal{X}_\Omega$ are final if they are epi in $\mathcal{X}$. Sinks in $\mathcal{E}$ are epi iff $\mathcal{X}$ has equalisers and these are in M, see [4], 15.7.*
3. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ create $(\mathcal{E}, M)$-factorisations. Then $\mathcal{X}_\Omega$ is an $(U^{-1}\mathcal{E}, U^{-1}M)$-category if sinks in $U^{-1}\mathcal{E}$ are final or morphisms in $U^{-1}M$ are initial.*

One possibility to assemble the material presented on factorisation systems and factorisation structures for sinks is presented by the following theorem:

Theorem 1.3.14. *Let $\mathcal{X}$ be a wellpowered category that has equalisers and small coproducts. Let $(E_\mathcal{X}, M_\mathcal{X})$ be a factorisation system such that $M_\mathcal{X}$ contains the equalisers (regular monos) of $\mathcal{X}$. Let Ω be a functor on $\mathcal{X}$ such that $\Omega(M_\mathcal{X}) \subset M_\mathcal{X}$.[21] Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ be the corresponding forgetful functor. Then:*

- *$(E_\mathcal{X}, M_\mathcal{X})$ can be extended uniquely to a factorisation structure for sinks $(\mathcal{E}_\mathcal{X}, M_\mathcal{X})$. Moreover, sinks in $\mathcal{E}_\mathcal{X}$ are epi.*
- *$(E, M) = (U^{-1}E_\mathcal{X}, U^{-1}M_\mathcal{X})$ and $(\mathcal{E}, M) = (U^{-1}\mathcal{E}_\mathcal{X}, U^{-1}M_\mathcal{X})$ are factorisation structures for $\mathcal{X}_\Omega$. In particular, factorisations in $\mathcal{X}_\Omega$ are calculated as in the base category $\mathcal{X}$. Moreover, sinks in $\mathcal{E}$ are final.*

Proof. The unique extension follows from proposition A.5.5. Sinks in $\mathcal{E}_\mathcal{X}$ are epi because $\mathcal{X}$ has equalisers and they are in $M_\mathcal{X}$, see [4], 15.7.

(E, M) is a factorisation structure because of $\Omega(M_\mathcal{X}) \subset M_\mathcal{X}$. $(\mathcal{E}, M)$ is the unique extension of (E, M) to sinks: This extension exists because $\mathcal{X}_\Omega$ inherits coproducts and wellpoweredness from $\mathcal{X}$. This extension is indeed $(\mathcal{E}, M)$ because U preserves coproducts and (E, M)-factorisations. Sinks in $\mathcal{E}$ are final because they are epi as sinks in $\mathcal{E}_\mathcal{X}$ (compare Rutten [108], 2.4). □

Corollary 1.3.15. *Let Ω be a functor on **Set**. $\mathbf{Set}_\Omega$ is a $(EpiSink, StrongMono)$-category and factorisations in $\mathbf{Set}_\Omega$ are calculated as $(EpiSink, Mono)$-factorisations in **Set**.*

[21] The condition $\Omega(M_\mathcal{X}) \subset M_\mathcal{X}$ is not needed in case of $\mathcal{X} = \mathbf{Set}$, $M_\mathcal{X} = Mono$.

1.4 An Axiomatic Approach to Universal Coalgebra

The aim of this section is to show that Rutten's theory of universal coalgebra [109] can be developed without assuming the category of sets as a base category. The properties of a categoriy of coalgebras are determined by the forgetful functor. The main ingredient of the approach presented here is to require the forgetful functor to lift factorisation systems of the base category to the category of coalgebras (axiom 1). Since properties of the forgetful functor are usually harder to verify than properties of the base category or the signature functor additional requirements are only on the base category and the signature. A summary of the axioms can be found in section 1.4.4.

The motivation for founding universal coalgebra not on the category of sets but on arbitrary categories with a factorisation system comes mainly from the fact that the results of chapter 2 depend on the existence of appropriate factorisation systems but not on special properties of the category **Set**. Therefore, we give here a development of universal coalgebra supporting (to some extent) the generality of the results in chapter 2. Moreover, most of the proofs in Rutten [109] are categorical, that is, we only have to extract those properties of **Set** which are really needed in the proofs.

For the remainder of this section let $\mathcal{X}$ be a category, Ω an endofunctor on $\mathcal{X}$ and $U : \mathcal{X}_\Omega \to \mathcal{X}$ the forgetful functor.

Our main assumption is that the base category $\mathcal{X}$ has a factorisation system $(E_\mathcal{X}, M_\mathcal{X})$ such that $(E, M) = (U^{-1}E_\mathcal{X}, U^{-1}M_\mathcal{X})$ is a factorisation system for $\mathcal{X}_\Omega$. Moreover, we assume that factorisations w.r.t. (E, M) are calculated as factorisations in the base category. Going back to section 1.3.1 this can be achieved by requiring:

Axiom 1. *The base category $\mathcal{X}$ has a factorisation system $(E_\mathcal{X}, M_\mathcal{X})$ and the forgetful functor U creates factorsiations w.r.t. $(E_\mathcal{X}, M_\mathcal{X})$. Moreover, $E_\mathcal{X} \subset Epi(\mathcal{X})$, $M_\mathcal{X} \subset Mono(\mathcal{X})$, and morphisms in $U^{-1}M_\mathcal{X}$ are initial.*

Remark (and Definition). We define $(E, M) = (U^{-1}E_\mathcal{X}, U^{-1}M_\mathcal{X})$ and keep this notation for the remainder of this section 1.4.

- U creating factorisations was defined in definition 1.3.2 and ensures that factorisations can be calculated in the base category and that (E, M) is a factorisation system for $\mathcal{X}_\Omega$ (see proposition 1.3.6). In the case $\mathcal{X} = \mathbf{Set}$, $(E_\mathcal{X}, M_\mathcal{X}) = (Epi, Mono)$, forgetful functors create factorisations (proposition 1.3.4).

- For morphisms in M being initial see definition A.1.1. This property is needed to ensure that the structure on a subsystem is unique. In the case $\mathcal{X} = \mathbf{Set}$, $(E_\mathcal{X}, M_\mathcal{X}) = (Epi, Mono)$, morphisms in M are initial (remark to proposition 1.3.5).

Remark. Our choice of axiom 1 is not the only reasonable one. Assuming that U creates factorisations, in order to infer that (E, M) is a factorisation system one of the three conditions $E_\mathcal{X} \subset Epi(\mathcal{X})$, $\Omega M \subset M$, morphisms in M are initial is sufficient. So there is room to weaken axiom 1. Also, the condition that morphisms in M are initial is not essential for the topics of this thesis: it is only needed for proposition 2.5.6. The assumption $M_\mathcal{X} \subset Mono(\mathcal{X})$ is reasonable, but technically it is only needed later when we want to extend (E, M) to a factorisation structure for sinks, see axiom 4 and proposition 1.4.3.

Remark. If Ω is a functor such that $\Omega M_{\mathcal{X}} \subset M_{\mathcal{X}}$ then U creates factorisations (proposition 1.3.3) and morphisms in M are initial (proposition 1.3.5).

The main point about having a factorisation system is that it gives us canonical notions of quotient, sub-object, and image. In order not to interfere with the standard usage of 'subobject', which denotes simply monos (or an equivalence class thereof), we use the notion 'subsystem' from Rutten [109]. And quotients will also be called behavioural equivalences. This is the natural transfer of the notion of behavioural equivalence of definition 1.2.1 to a setting with factorisation system.

Definition 1.4.1 (behavioural equivalence, subsystem, quotient, image). In the context of a category of coalgebras $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfying axiom 1, the notions of behavioural equivalence and subsystem refer to morphisms in E and M, respectively. Behavioural equivalences are also called quotients. The image $m \in M$ of a morphism f is given by its (E, M)-factorisation $f = m \circ e$.

An important property of quotients is the following. Suppose we are given a possibly large family $(e_i : A \to B_i)_{i \in I}$ of quotients. We then ask whether there exists a quotient $e : A \to B$ which identifies everything which has to be idenitfied according to the quotients $(e_i : A \to B_i)_{i \in I}$ but not more. This can be expressed by the following universal property:

Definition 1.4.2 (cointersection). $e : A \to B$ is a *cointersection* of $(e_i : A \to B_i)_{i \in I}$ iff (1) e factors through all e_i and (2) every f which factors through all e_i also factors through e. A class of morphisms E *has cointersections* if the cointersection of any possibly large family $(e_i : A \to B_i)_{i \in I}$ of morphisms in E exsists and is itself in E. A category $\mathcal{X}$ has cointersections if $Epi(\mathcal{X})$ has cointersections.

(1) expresses that e identifies all that is identified by some of the e_i, (2) expresses that e does not identify more than imposed by $(e_i : A \to B_i)_{i \in I}$.

Axiom 2. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1 with $(E_{\mathcal{X}}, M_{\mathcal{X}})$ the factorisation system for $\mathcal{X}$. We then require $E_{\mathcal{X}}$ to have cointersections.*

Remark. The following remarks show that axiom 2 is satisfied under rather general circumstances.

1. $E_{\mathcal{X}}$ has cointersections whenever $\mathcal{X}$ has cointersections (proposition A.4.3(5) and $E_{\mathcal{X}} \subset Epi(\mathcal{X})$).

2. The category of sets has cointersections: Given a famlily of epis $(e_i : A \to A_i)$ their cointersection is given by quotienting A with the smallest equivalence relation containing the union of the kernels of the e_i.

3. A category has cointersections if it is cocomplete and cowellpowered. It follows that categories of algebras for a functor or a monad have cointersections.

Another property that is useful comes into play when want to do concrete caclculations with behavioural equivalences. It is then useful to represent them—similarly to bisimulations—as relations in the base category. This can be done by considering the kernel pair of a behavioural equivalence. The following axiom ensures that the kernel pair of a behavioural equivalence exists and that a behavioural equivalence is the coequaliser of its kernel pair. (This ensures that kernel pairs of behavioural equivalences and behavioural equivalence are—up to isomorphism—in a one-to-one correspondence.)

Axiom 3. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1 with (E, M) the factorisation system for $\mathcal{X}_\Omega$. We require $UE = RegEpi(\mathcal{X})$. Moreover, $\mathcal{X}$ has to have coequalisers and kernel pairs of coequalisers.*

The last axiom we will consider gives us a factorisation structure for sinks (definition A.5.1) which in turn provides us with arbitrary unions of subsystems (see section 1.3.2). This property is central to prove many useful features that categories coalgebras may have, see e.g. section 1.6 on limits of coalgebras and chapter 2.

Axiom 4. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1 with $(E_\mathcal{X}, M_\mathcal{X})$ the factorisation system on $\mathcal{X}$. We require $\mathcal{X}$ to have small coproducts and to be $M_\mathcal{X}$-wellpowered.*

Remark. As a consequence $\mathcal{X}_\Omega$ has small coproducts and is M-wellpowered.

Assuming axiom 4, $\mathcal{X}_\Omega$ is an $(\mathcal{E}, M)$-category (see definition A.5.1):

Proposition 1.4.3. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axioms 1 and 4. Then (E, M) can be uniquely extended to a factorisation structure for sinks $(\mathcal{E}, M)$ such that $\mathcal{X}_\Omega$ is an $(\mathcal{E}, M)$-category. Moreover, $(\mathcal{E}, M)$-factorisations are calculated as in the base category $\mathcal{X}$ and sinks in $\mathcal{E}$ are epi.*

Proof (axiom 1, 4). The extension of (E, M) to $(\mathcal{E}, M)$ is proposition A.5.5. The proof of this proposition shows that $(\mathcal{E}, M)$-factorisations are calculated as in the base category since U creates coproducts and (E, M)-factorisations. Using that arrows in $E_\mathcal{X}$ are epi according to axiom 1, it follows moreover that sinks in $\mathcal{E}$ are epi. □

The reminder of this section is devoted to develop the theory of universal coalgebras using the axioms above.

The Basic Theory

We review the parts of Rutten [109] which can be developed without any assumptions on $U : \mathcal{X}_\Omega \to \mathcal{X}$.

U is faithful.

U creates colimits, see [109]4.5, 4.6. Hence, U creates isos, in particular f is iso in $\mathcal{X}_\Omega$ iff it is in $\mathcal{X}$, see [109]2.3.

U preserves and reflects epis. U reflects monos and preserves them if Ω preserves weak pullbacks, see [109]4.7 and proposition A.3.4. Epis in $\mathcal{X}_\Omega$ are final, see [109]2.4.1 and monos

$m \in \mathcal{X}_\Omega$ are initial if Ωm mono, see [109]2.4.2 (the proviso Ωm mono can be dropped in the case $\mathcal{X} = \mathbf{Set}$ because monos with non-empty domain are split).

Graphs[22] of morphisms are bisimulations, see [109]2.5. The diagonal[23] of a coalgebra is a bisimulation, see [109]5.1, and the inverse of a bisimulation is a bisimulation, [109]5.2.

Let $A \in \mathcal{X}_\Omega$ and $m : X \to UA \in \mathcal{X}$. Then X is a subsystem of A (i.e., there is a unique $\xi : X \to \Omega X$ such that m is a morphism $(X, \xi) \to A$) iff the (X, m, m) is a bisimulation on A, see [109]6.2.

1.4.1 Behavioural Equivalences and Cocongruences

Up to this point, the notion of a functor preserving weak pullbacks had appeared only once, namely to show that a morphism which is mono in $\mathcal{X}_\Omega$ is also mono in $\mathcal{X}$. This is a property which is convenient, but in no way essential. Things change when we take a closer look at the notion of a bisimulation. Here, preservation of weak pullbacks comes in at several places.

Preservation of weak pullbacks is needed to ensure that

- composition of bisimulations is a bisimulation, see [109]5.4,
- all kernels of coalgebra morphisms are bisimulations,[24] see [109]5.7,
- the largest bismulation exists, see [109]5.6.

Concerning the last item, as showed in section 1.1.4, the proof of [109]5.6 does not need that Ω preserves weak pullbacks but uses instead that epis in **Set** are split. But this condition is rather strong and an axiomatic theory of coalgebras should not depend on it. Another possibility to show the existence of a largest bisimulation is to use a functor Ω preserving weak pullback as in corollary 1.2.3.

A property related to the second item is the following. If Ω preserves weak pullbacks then the pullback in $\mathcal{X}$ of a cospan in $\mathcal{X}_\Omega$ is a bisimulation, see [109]4.3. The proof is the same as for proposition 1.1.13.

Since we want to develop the general theory of coalgebras independently from the assumtions that signatures Ω preserve weak pullbacks or that epis in $\mathcal{X}$ are split we propose to use behavioural equivalences and cocongruences (see section 1.2) instead of bisimulations. In our setting, assuming axiom 1, we have defined behavioural equivalences to be quotients (definition 1.4.1). In order to obtain a notion of cocongruence which fits to the given factorisation system we define here cocongruences via quotients of coproducts.

Definition 1.4.4 (cocongruence). Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1 with (E, M) the factorisation system for $\mathcal{X}_\Omega$ and let $\mathcal{X}$ have binary coproducts. A *cocongruence* on two coalgebras A, B is a behavioural equivalence on $A + B$.

[22] A *graph* of a morphism $f : A \to B$ is the mono span (id_A, f).

[23] The *diagonal* of a coalgebra A is the span $(\mathrm{id}_A, \mathrm{id}_A)$.

[24] That is, more precisely, given a morphism $f : A \to B$ the kernel pair of Uf in $\mathcal{X}$ can be equipped with a structure such that it becomes a bisimulation. This has been shown in proposition 1.1.13.

Composition of cocongruences can be defined dually to the composition of bisimulations in Rutten [109]5.4. The construction in [109]5.4 needs both that the signature preserves weak pullbacks and that epis in the base are split. The dual construction is independent from this assumptions. One only needs to assume axiom 1 and the existence of binary coproducts.

Giving up that the signature preserves weak pullbacks, it does not hold any more that the kernel pairs of all coalgebra morphisms are bisimulations. But since we develop the theory using behavioural equivalences/cocongruences instead of bisimulations this does not matter anymore.

The largest behavioural equivalence exists assuming axioms 1 and 2. The largest cocongruence exists if we assume, moreover, the existence of binary coproducts.

Summary

We have shown that cocongruences and behavioural equivalences allow for technical results similar to those Rutten [109] showed for bisimulations but this time without assuming that signatures preserve weak pullbacks or that epis in the base are split.[25] How cocongruences and behavioural equivalences are related to coinductive proofs is discussed after proposition 1.4.11.

1.4.2 Base Categories with Factorisation Systems

We now turn to the part of universal coalgebra whose development depends on the existence of a factorisation system in the base category.

Subsystems

Recall the notion of a subsystem from definition 1.4.1.

Proposition 1.4.5 (Rutten [109]6.1). *The structure on a subsystem is uniquely determined.*

Proof (axiom 1). Let $(A, \alpha), (B, \beta) \in \mathcal{C}$, $m : (A, \alpha) \to (B, \beta)$ be a morphism in M and suppose that m is also a morphism $(A, \alpha') \to (B, \beta)$. By m being initial, id_A is a morphism $(A, \alpha) \to (A, \alpha')$, hence $\alpha = \alpha'$. □

Remark. Although axiom 1 refers to a factorisation system, this proposition depends only on the morphism m being initial, not on $m \in M$ for some factorisation system (E, M).
This is the only place in this thesis where the property of morphisms being initial comes into play.

Proposition [109]6.2 generalises to arbitrary base categories without any assumptions on $U : \mathcal{X}_\Omega \to \mathcal{X}$.

Given a morphism $f : A \to B \in \mathbf{Set}_\Omega$ and a subsystem $m_0 : B_0 \to B$ one can speak of $f(A)$ and $f^{-1}(B_0)$. Categorically, the image $f(A)$ of f is obtained via the (E, M)-factorisation

[25] That epis in the base are split is needed in [109]5.3,5.4,5.5,5.9.

$f = m \circ e$: We will consider m as the image of f. The preimage $f^{-1}(B_0)$ is given via the pullback:

$$\begin{tikzcd} f^{-1}(B_0) \arrow[r, "q"] \arrow[d, "p"'] & B_0 \arrow[d, "m_0"] \\ A \arrow[r, "f"] & B \end{tikzcd}$$

We call such a pullback *a pullback along a subsystem.* This construction is not so obvious in the general case. First, the pullback has to exist at all. Second, if it exists p should be in M in order to qualify $f^{-1}(B_0)$ as a subsystem of A. If $\mathcal{X}$ has pullbacks and Ω preserves them then these conditions are met (we calculate the pullback in the base category; moreover, $p \in M$ since $m_0 \in M$ and morphisms in M are stable under pullback).

Proposition 1.4.6 (Rutten [109]6.3). *Let $f : A \to B \in \mathcal{X}_\Omega$ and $m_0 : B_0 \to B$ be a subsystem.*

1. *The image of f is a subsystem.*

2. *If $\mathcal{X}$ has pullbacks along morphisms in $M_\mathcal{X}$ and Ω preserves them then $f^{-1}(B_0)$ is a coalgebra.*

Proof (axiom 1). (1) is immediate by axiom 1 and the definition of image. (2) is established using an argument similar to the one in the proof of proposition 1.2.2. Note that, in contrast to the proof in [109], the proof does not require epis in the base to be split. □

Remark. Ω preserves pullbacks along morphisms in $M_\mathcal{X}$ if it weakly preserves them and $\Omega M_\mathcal{X} \subset M_\mathcal{X}$.[26]. — Proof: One of the legs of the weakly preserved pullback is mono (recall that $M_\mathcal{X} \subset \mathit{Mono}(\mathcal{X})$ according to axiom 1), hence the pullback is also preserved.

Remark. Gumm and Schröder [42] characterise the functors Ω on **Set** that preserve pullbacks along monos. In particular, they show a kind of converse of proposition 1.4.6(2): If in $\mathbf{Set}_\Omega$ all pre-images of morphisms are subsystems then Ω preserves pullbacks along monos.

Isomorphism Theorems

The three isomorphism theorems in Rutten [109] will be treated next.

The first isomorphism theorem, see [109]7.1, becomes an immediate consequence of axiom 1, if we replace the use of kernels (bisimulations) by behavioural equivalences.

Similarly, using behavioural equivalence instead of bisimulation equivalence in [109]7.2 and 7.4, these theorems are immediate.

Concerning [109]7.3 it is not difficult to show the following correspondence between quotients of subsystems and subsystems of quotients (see also Gumm and Schröder [41]2.11).

[26]In the case $\mathcal{X} = \mathbf{Set}$, $M_\mathcal{X} = \mathit{Mono}$, the proviso $\Omega M_\mathcal{X} \subset M_\mathcal{X}$ is not needed

Proposition 1.4.7. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1 with (E, M) the factorisation system for $\mathcal{X}_\Omega$. If morphisms in E are stable under pullbacks then subsystems of quotients are quotients of subsystems (i.e. $SH \leq HS$ where S denotes closure under subsystems, H closure under quotients, and $SH \leq HS$ means $SH(\mathcal{B}) \subset HS(\mathcal{B})$ for all $\mathcal{B} \subset \mathcal{X}_\Omega$). If morphisms in M are stable under pushouts then quotients of subsystems are subsystems of quotients (i.e. $HS \leq SH$).*

Proof (axiom 1). Consider the diagrams which are a pullback and a pushout, respectively:

$$\begin{tikzcd}
\bullet \arrow[r, dotted] \arrow[d, dotted, two heads] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, two heads] \\
A \arrow[r, tail] & \bullet A
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\bullet \arrow[r, tail] \arrow[d, two heads] & B \arrow[d, dotted, two heads] \\
A \arrow[r, dotted] & \bullet \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

In the left-hand diagram A is a subsystem of a quotient of B. Since subsystems are stable under pullbacks and quotients are stable by assumption, A is also a quotient of a subsystem of B. The second assertion is proved analogously using the right-hand diagram. □

Remark. In $\mathbf{Set}_\Omega$ it holds that $HS \leq SH$.[27] — Proof: Recall that $\mathbf{Set}_\Omega$ is a $(Epi, StrongMono)$-category. The assertion now follows because strong monos in $\mathbf{Set}_\Omega$ are stable under pushouts (pushouts in $\mathbf{Set}_\Omega$ are calculated as pushouts in **Set**; strong monos in $\mathbf{Set}_\Omega$ are monos in **Set**; monos in **Set** are stable under pushouts; and monos in **Set** are strong monos in $\mathbf{Set}_\Omega$).

Remark. In $\mathbf{Set}_\Omega$ it holds $SH \leq HS$ if Ω preserves pullbacks. — Proof: Since Ω preserves pullbacks these are calculated as in **Set** where epis are stable under pullbacks (since they are split).

Extensional and Simple Coalgebras

We turn to Rutten [109], section 8. A coalgebra is called **simple** iff it has no proper quotients. In order to draw useful conlusions from this property we need to assume a proper interplay of quotients in $\mathcal{X}_\Omega$ and kernel pairs in $\mathcal{X}$, as required by axiom 3. An object $A \in \mathcal{X}_\Omega$ is called **extensional** iff for all $B \in \mathcal{X}_\Omega$ and all $f, g : B \to A$ it holds $f = g$.

The coinduction proof principle in Rutten [109] states that every bisimulation on a coalgebra A is a subset of the diagonal (identity relation) on A. This proof principle is satisfied by simple (and hence final) coalgebras and is typically used to prove equality of two elements in a final coalgebra. Rutten also shows the converse, namely, that in the case of a weak pullback preserving signature every coalgebra satisfying the coinduction proof principle is simple.

In case of signatures not preserving weak pullbacks the coinduction proof principle is still valid on simple coalgebras but it does not imply any more that a coalgebra is simple. We therefore generalise the coinduction proof principle. It is not difficult to guess that we now have to require that (not only bisimulations but) all kernels of behavioural equivalences are contained in, i.e. equal to, the diagonal (see also proposition 1.4.11).

[27] This is the dual assertion to $SH \leq HS$ in universal algebra, see e.g. Wechler [124].

Definition 1.4.8 (generalised coinduction proof principle). Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1. $A \in \mathcal{X}_\Omega$ satisfies the *generalised coinduction proof principle* iff the kernel pair in $\mathcal{X}$ of every behavioural equivalence on A exists and is (isomorphic to) the diagonal $(\mathrm{id}_{UA}, \mathrm{id}_{UA})$.

We can simplify notation using $\mathrm{id}_{UA} = U\mathrm{id}_A = \mathrm{id}_A$.

Proposition 1.4.9. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1 and 2. Then the following are equivalent.*

1. *A is simple.*
2. *A satisfies the generalised coinduction proof principle.*
3. *The largest behavioural equivalence on A is an iso.*

Proof (axiom 1, 2). Note that a coalgebra A is simple iff all behavioural equivalences on A are iso. (1) $\Leftrightarrow$ (2) follows from the fact that a behavioural equivalence is iso iff its kernel pair is ismorphic to the diagonal. (1) $\Leftrightarrow$ (3) follows from the fact that all behavioural equivalences on A are iso iff the largest behavioural equivalence on A is iso (assuming that the largest behavioural equivalence exists which it does by axiom 2). $\square$

To show that these three conditions are equivalent to A being extensional we need additional assumptions:

Proposition 1.4.10. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axioms 1 and 3 and let Ω weakly preserve kernel pairs of coequalisers. Then the following are equivalent for $A \in \mathcal{X}_\Omega$.*

1. *A is extensional.*
2. *A is simple.*

Proof (axiom 1, 3). We use that A is simple iff it satisfies the generalised coinduction proof principle. Let (E, M) be the factorisation system for $\mathcal{X}_\Omega$ which exists according to axiom 1.
(2) $\Rightarrow$ (1) holds because, as a consequence of axiom 3, $\mathcal{X}_\Omega$ has coequalisers and they are in E (to show extensionality let f, g be a parallel pair with codomain A; take their coequaliser which is a quotient, hence its kernel pair is the diagonal $(\mathrm{id}_A, \mathrm{id}_A)$, hence the quotient is mono, hence iso, hence $f = g$).
(1) $\Rightarrow$ (2) needs that $\mathcal{X}$ has kernel pairs of coequalisers and that Ω weakly preserves them (let e be a quotient in $\mathcal{X}_\Omega$ and (p, q) be the kernel pair of e in $\mathcal{X}$; since Ω weakly preserves kernel pairs of quotients, p, q are morphisms in $\mathcal{X}_\Omega$, hence $p = q$; it follows that $(\mathrm{id}_A, \mathrm{id}_A)$ is a kernel pair of e). $\square$

The generalised coinduction proof principle implies the one in Rutten [109] (assuming axiom 3).

Proposition 1.4.11. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axioms 1 and 3. Then $A \in \mathcal{X}_\Omega$ satisfies the general coinduction proof principle only if it satisfies the coinduction proof principle, i.e., for every bisimulation (R, p, q) on A it holds that $(R, p, q) \leq (UA, \mathrm{id}_A, \mathrm{id}_A)$.*

Proof (axiom 1, 3). Let (R, p, q) be a bisimulation on A and ϱ a structure map for the bisimulation. Let e be the coequaliser of $p, q : (R, \varrho) \to A$. e is a behavioural equivalence by axiom 3. It follows now from the generalised coinduction proof principle that $(UA, \mathrm{id}_A, \mathrm{id}_A)$ is a kernel pair of Ue. Hence there is an arrow $(R, p, q) \to (UA, \mathrm{id}_A, \mathrm{id}_A)$. □

The question remaining is how to use the generalised coinduction proof principle in proofs by coinduction. Suppose we want to prove that two states a, b of a simple coalgebra A are equal. Using proposition 1.4.11, we can still proceed as we are used to by finding a bisimulation on A relating a, b. But, as shown by example 1.2.4, even if a, b are indeed behaviourally equivalent, there may not exist any bisimulation relating these points (take s_0, s_1 in example 1.2.4). Of course, behavioural equivalent points are related by the relation that is given by the kernel pair of the largest behavioural equivalence but the question is whether we can characterise those relations which arise as kernel pairs of behavioural equivalences in a useful way. In the case of example 1.2.4 there is a simple solution to this problem: $\mathbf{Set}_{AM}$ is a (coreflective) subcategory of $\mathbf{Set}_\Omega$ with $\Omega X = X^3$. We can now characterise the kernel pairs of behavioural equivalences as bisimulation equivalences in the larger category $\mathbf{Set}_\Omega$.[28] This means that in order to show that two elements of an AM-coalgebra are behavioural equivalent it is enough to look for an Ω-bisimulation relating them.

This rises of course the question whether there is a general principle here at work. Is it the case that for any functor Ω' on **Set** possibly not preserving weak pullbacks there is a weak pullback preserving functor Ω on **Set** such that (1) $\mathbf{Set}_{\Omega'} \hookrightarrow \mathbf{Set}_\Omega$ and (2) for all $A \in \mathbf{Set}_{\Omega'}$ the largest Ω'-behavioural equivalence is also the largest Ω-behavioural equivalence? And has this question an answer for other base categories than **Set**? To explore these issues has to be left for future work.

Union of Subsystems

As mentioned already, factorisation structures for sinks are the appropriate tool to model abstractly unions of subsystems. We now list some of the consequences that one can show for categories of coalgebras which have a factorisation structure for sinks.

A first consequence is that the category of coalgebras has equalisers.

Proposition 1.4.12. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axioms 1 and 4. Then $\mathcal{X}_\Omega$ has equalisers and these are in M.*

Proof. This follows from sinks in $\mathcal{E}$ being epi (proposition 1.4.3) and proposition A.5.4. The proof given there also shows how equalisers are calculated using the sinks in $\mathcal{E}$. □

Similarly, pullbacks along morphisms in M do exist (for a proof see Adámek, Herllich, Strecker [4], theorem 15.14(3)).

Proposition 1.4.13. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axioms 1 and 4. Then $\mathcal{X}_\Omega$ has pullbacks along morphisms in M.*

[28] Note that the right adjoint to the inlcusion $\mathbf{Set}_{AM} \hookrightarrow \mathbf{Set}_\Omega$ preserves the final coalgebra.

Since we can show that a category of coalgebras with a factorisation structure for sinks has equalisers one may suspect that one can use a similar reasoning to show the existence of limits. This is indeed the case, assuming that the forgetful functor has a right adjoint, see section 1.6.

The following two corollaries generalise Rutten [109]6.4.

Corollary 1.4.14. *The collection of all subsystems of a coalgebra $A \in \mathcal{X}_\Omega$ is a complete lattice in which joins are given by factorisation of sinks and meets by intersections in $\mathcal{X}_\Omega$.*

Proof (axiom 1, 4). The collection of all subsystems of $A \in \mathcal{X}_\Omega$ is—up to equivalence—a poset. Let $m_i : A_i \to A$ be a collection of subsystems. Consider the $(\mathcal{E}, M)$-factorisation of the sink $(m_i) = m \circ (e_i)$. We have to show that m is the join of the m_i. First, that m is larger than all m_i is witnessed by the e_i. Second, that m is the smallest such follows from the unique diagonalisation property in definition A.5.1(3). The claim that the lattice is complete follows now since all joins exist. That meets are given by intersections follows from the general fact that in an $(\mathcal{E}, M)$-category M is closed under intersections: Given $m_i : A_i \to A \in M$, let $s_j : B_j \to A$ be the sink consisting of the morphisms which are smaller than all m_i and consider an $(\mathcal{E}, M)$-factorisation $(s_j) = n \circ (f_j)$. Using unique diagonalisation again, it is not difficult to see that n is indeed the intersection of the m_i. □

Remark. It follows from axiom 2 that $\mathcal{X}_\Omega$ has an initial object 0. This can be used to describe the bottom element of the lattice of subsystems of A: The bottom element is given by the image of the unique morphism $? : 0 \to A$, see the remark after the proof of proposition A.5.5.

Corollary 1.4.15. *If $\mathcal{X}$ has cointersections and if Ω preserves them (e.g. if Ω preserves multiple pullbacks) then meet in the complete lattice of subsystems is given by the intersection in $\mathcal{X}$.*

Proof (axiom 1, 4). We know that intersections in $\mathcal{X}_\Omega$ are calculated as intersections in $\mathcal{X}$ (since Ω preserves intersections they are created by U, see proposition A.3.1). That meet in the lattice is given by intersection in $\mathcal{X}_\Omega$ was shown in corollary 1.4.14. □

Given a coalgebra $A \in \mathcal{X}_\Omega$ and $\iota : X \to UA \in M$ we define the largest subsystem $[X]$ of A contained in X as follows. Let $(m_i : A_i \to A)$ be the sink consisting of all morphisms $m_i \in M$ such that Um_i factors through ι. Consider an $(\mathcal{E}, M)$-factorisation $(m_i) = m \circ (e_i)$. Then $[X]$ is defined to be the domain of m.

The corresponding generalisation of Rutten [109]6.5 is immediate.

1.4.3 Discussion

We have shown that factorisation systems are well suited to build an axiomatic theory of universal coalgebra. Here, we want to briefly discuss some further reasonable assumptions expressing that

- the signature preserves weak pullbacks,
- morphisms in $E_\mathcal{X}$ are split,
- the forgetful functor has a right adjoint.

signatures preserving weak pullbacks

If one requires the signature to preserve weak pullbacks one obtaines the well-known theory of bisimulations as in Rutten [109]. Another useful property of weak pullback preserving signatures is given by a theorem of Carboni, Kelly and Wood [23] stating that a signature Ω on sets can be extended to a strong relator if and only if Ω preserves weak pullbacks. This property can be used to develop a coalgebraic theory of simulations, see Baltag [10].

quotients in the base category being split

Gumm and Schröder [42, 40] have recently shown that in the case of the base category **Set** one can develop a large part of the theory of universal coalgebra without the assumption that the signatures preserve weak pullbacks. Instead they make extensive use of epis in **Set** being split. Using this fact together with some other properties of the category **Set** they were able to characterise properties of signatures in terms of the structure they induce on coalgebras. For example: A signature preserves mono spans iff the structure of a bisimulation is unique iff the largest bismulation is the product; a signature weakly preserves multiple pullbacks of cardinality κ iff the intersection of a family of cardinality κ of subcoalgebras is a subcoalgebra.

On the other hand, to require epis in the base to be split is a very strong assumption excluding, for example, many categories of algebras as base categories (in categories of algebras epis are usually not even regular (or surjective) and hence not split). This seems to indicate that their results will not be easy to generalise. On the other hand, using a factorisation system $(E_{\mathcal{X}}, M_{\mathcal{X}})$ for the base category, it should be possible to obtain some of their results (and Rutten [109]5.3,5.4,5.5,5.9) by assuming only that arrows in $E_{\mathcal{X}}$ are split. Since arrows in $E_{\mathcal{X}}$ are quotients this would amount to the axiom of choice.

So we would require

Axiom 5. *Let $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfy axiom 1 with $(E_{\mathcal{X}}, M_{\mathcal{X}})$ the factorisation system for $\mathcal{X}$. Then arrows in $E_{\mathcal{X}}$ are split.*

This axiom would certainly not be enough to give us all of Gumm and Schröder [42, 40]. In particular, one would have to axiomatically describe the properties needed to prove the corresponding generalisation of their beautiful lemma [42]5.2.

forgetful functors with right adjoint

Concerning the right adjoint of the forgetful functor we could add as an assumption:

$\mathcal{X}$ has a terminal object 1 and $U : \mathcal{X}_\Omega \to \mathcal{X}$ has a right adjoint F.

As a first consequence, $\mathcal{X}_\Omega$ has now a terminal coalgebra, namely $F1$. The terminal coalgebra is simple and extensional. Assuming axiom 2 and 1, the largest behavioural equivalence on a coalgebra A can be obtained by factoring the unique morphism $! : A \to F1$. In particular, the quotient of a coalgebra by the largest behavioural equivalence is extensional, that is, one obtaines proposition 1.4.10 without assuming axiom 3.

It may be interesting to note that conversly—under certain circumstances—the terminal coalgebra exists if every simple coalgebra (or every quotient by a largest behavioural equivalence) is extensional; see the proof of the final coalgebra theorem (noting especially the 'main lemma') in Aczel [1] and Aczel and Mendler [2].

An important consequence of the existence of a right adjoint (together with axioms 1 and 4) is that $\mathcal{X}_\Omega$ has limits if $\mathcal{X}$ has, see section 1.6.

To require the existence of a right adjoint to the forgetful functor, however, conflicts with the idea of the axiomatic approach to require—whenever possible—only simple properties of the base category and the signature. The existence of a right adjoint should be *proved* from such assumptions. The ordinary way to proceed, is to require the signature to be bounded, but this condition still involves the category of coalgebras. It may be replaced by requiring the signature to be accessible. See section 1.5 for a discussion of these notions.

1.4.4 Summary of Axioms

Table 1.1 shows a summary of the axioms discussed. The only axiom involving the category of coalgebras is axiom 1, basically stating that the forgetful functor creates factorisations. Otherwise, the requirements are all on the base category.

Let $\mathcal{X}$ be a category and Ω an endofunctor on $\mathcal{X}$.

1. $\mathcal{X}$ has a factorisation system $(E_{\mathcal{X}}, M_{\mathcal{X}})$, and U creates factorisations w.r.t. $(E_{\mathcal{X}}, M_{\mathcal{X}})$. Moreover $E_{\mathcal{X}} \subset Epi(\mathcal{X})$, $M_{\mathcal{X}} \subset Mono(\mathcal{X})$, and morphisms in $U^{-1}M_{\mathcal{X}}$ are initial.
2. $E_{\mathcal{X}}$ has cointersections.
3. $\mathcal{X}$ has coequalisers and kernel pairs of coequalisers and $E_{\mathcal{X}} = RegEpi(\mathcal{X})$.
4. $\mathcal{X}$ has small coproducts and is $M_{\mathcal{X}}$-wellpowered.
5. Arrows in $E_{\mathcal{X}}$ are split.

Table 1.1: List of Axioms

We recall that, essentially,

axiom 1 equips the category of coalgebras with a factorisation system that allows to calculate factorisations in the base category,

axiom 2 guarantees the existence of a largest behavioural equivalence,

axiom 3 allows to represent behavioural equivalences as relations (kernel pairs) in the usual way,

axiom 4 enables unions of subcoalgebras,

axiom 5 amounts to the axiom of choice.

Let us also note that in order to obtain final coalgebras one usually assumes that the signature functor is bounded. In our framework, this condition can be expressed by the one that the category of coalgebras is bounded, see definition 1.5.3.

6. $\mathcal{X}_{\Omega}$ is bounded (see definition 1.5.3)

The disadvantage of (6) is that it is not simply a condition on the base category or the signature functor. It was shown in Barr [12] for the case of $\mathcal{X} = \mathbf{Set}$ and by Power and Watanabe [91] for the general case that signatures which are *accessible* functors (see appendix A.8) give rise to final and cofree coalgebras. Thus one might prefer (6') to (6):

6'. $\Omega : \mathcal{X} \to \mathcal{X}$ is accessible.

Note that (6') implies (6): It follows from (6') that $\mathcal{X}_\Omega$ is accessible (Power and Watanabe [91], theorem 3.8) and then from proposition 1.5.7 (and axioms 1, 4) that $\mathcal{X}_\Omega$ is bounded. In the case of $\mathcal{X} = \mathbf{Set}$, it follows from a recent result by Adámek [3] that $(6')$ and (6) are equivalent (see section 1.5 for more information).

1.5 Bounded Categories and Final Coalgebra Theorems

This section deals with notions of smallness and boundedness for categories of coalgebras. We introduce the notion of a bounded category to generalise categories of coalgebras for a bounded functor. The motivation for doing this is that, first, bounded categories are precisely what is needed in chapter 2 to prove a bounded version of the covariety theorem and, second, it allows us to abstract from the category of sets. Here, we show that one obtains from this generalised boundedness condition final coalgebras theorems as usual. We follow Barr [12] in using the special adjoint functor theorem (theorem A.6.2) in order to show final coalgebra theorems.

Consider a category of coalgebras $\mathbf{Set}_\Omega$. Clearly, every coalgebra is the union of its subcoalgebras. The functor Ω is called bounded by C if the size of these subcoalgebras over which the union is taken can be bounded by $|C|$. This idea goes back at least to Aczel and Mendler [2] where it appears as the 'small subcoalgebra lemma'. We define the notion of a bounded functor as in [75] (which is the obvious generalisation of Rutten [109] 6.7 for signatures that do not preserve weak multiple pullbacks) and a further variation which we will call $<$-bounded here.[29]

Definition 1.5.1 (bounded functor). Let $\Omega : \mathbf{Set} \to \mathbf{Set}$ be a functor and κ a cardinal. The functor Ω is bounded (resp. $<$-bounded) by κ iff for all $A \in \mathbf{Set}_\Omega$ there is a family of subcoalgebras $(A_i)_{i\in I}$ such that $A = \bigcup\{A_i : i \in I\}$ and $|A_i| \leq \kappa$ (resp. $|A_i| < \kappa$) for all $i \in I$. A functor is called bounded by $C \in \mathbf{Set}$ iff it is bounded by $|C|$.

Remark. A functor is bounded by κ iff it is $<$-bounded by κ^+ (where κ^+ denotes the successor cardinal of κ). — This shows that using $<$-bounded we could dispense with 'bounded'. We won't do so, on the one hand because we do not want to interfere with common use of 'bounded', on the other hand because, somewhat surprisingly, the notion of $<$-bounded would not give us better results in chapter 2.

This definition of a bounded functor depends on the base category **Set** because we use cardinals to measure the size of sets. But the essence of this definition, namely that every coalgebra is the union of 'small' coalgebras, can be stated in a set-independent way. We first generalise the notion of a union of coalgebras.

Definition 1.5.2 (union). Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category and $(S_i)_{i\in I}$ a family of objects of $\mathcal{C}$. We say that $A \in \mathcal{C}$ is the union of $(S_i)_{i\in I}$ iff there is a sink $(e_i : S_i \to A)$ such that $(e_i) \in \mathcal{E}$ and all e_i in M.

Remark. This is the canonical definition of union if the category $\mathcal{C}$ has a factorisation structure for sinks $(\mathcal{E}, M)$ and the sinks in $\mathcal{E}$ are epi. We did not require sinks in $\mathcal{E}$ to be epi for two reasons. First, the theorems in chapter 2 which are based on the notion of a bounded category do not need that sinks in $\mathcal{E}$ are epi. Second, if we use boundedness in the context of the axioms 1 and 4 of the previous section then sinks in $\mathcal{E}$ are epi anyway (proposition 1.4.3).

[29] We would expect the notion of $<$-bounded to be the canonical candidate for the notion of a bounded functor but the results in chapter 2 would then be clumsier to state.

Remark. If the category $\mathcal{C}$ is not equipped with a factorisation structure $(\mathcal{E}, M)$ one can still try one of the following. First, one could simply say that $A \in \mathcal{C}$ is the union of $(S_i)_{i \in I}$ iff there is an extremal epi-sink $(e_i : S_i \to A)$, such that all e_i are mono. (If we consider the monos as subobjects, the condition that (e_i) is extremal is needed anyway: Otherwise (e_i) would factor through a proper subobject of A, that is, A shouldn't be considered as the union of the e_i.) Second, it seems sensible to further require that the sink (e_i) is a colimit for an appropriate diagram (this would imply that (e_i) is extremal).

The obvious generalisation of definition 1.5.1 is now as follows.

Definition 1.5.3 (bounded category, small objects). Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. $\mathcal{C}$ is called *bounded* iff there is a set of objects $\mathcal{S} \hookrightarrow \mathcal{C}$ such that every object in $\mathcal{C}$ is the union of a family $(S_i)_{i \in I}$ with $S_i \in \mathcal{S}$. $\mathcal{C}$ is called *bounded by* $A \in \mathcal{C}$ iff for all $S \in \mathcal{S}$ there is $m : S \to A$, $m \in M$. In a bounded category the objects in $\mathcal{S}$ are called *small.*

Remark. If $\mathcal{C}$ is bounded and has small coproducts then it is bounded by some $A \in \mathcal{C}$.

The notion of a bounded category subsumes the notion of a category of coalgebras for a bounded functor:

Proposition 1.5.4. *Let $\Omega : \mathbf{Set} \to \mathbf{Set}$ be a functor bounded by C. Then $\mathbf{Set}_\Omega$ is bounded by FC (where F is right adjoint to $U : \mathbf{Set}_\Omega \to \mathbf{Set}$).*

Proof. $\mathbf{Set}_\Omega$ is an $(\mathit{EpiSink}, \mathit{StrongMono})$-category (corollary 1.3.15). Let $\mathcal{S} = \{A \in \mathbf{Set}_\Omega : |UA| \leq |C|\}$. Since Ω is bounded, every coalgebra is the union of coalgebras in $\mathcal{S}$. It remains to show that $\mathbf{Set}_\Omega$ is bounded by FC. Since Ω is bounded, U has a right adjoint F (see e.g. Rutten [108]). For all $S \in \mathcal{S}$ there is an injective mapping $m : US \to C$. It follows that the lifting $m^\# : S \to FC$ is injective and hence strong mono. □

Note that, conversely, if $\mathbf{Set}_\Omega$ is bounded by FC, then Ω is bounded by UFC. From this it is not difficult to deduce

Corollary 1.5.5 ("bounded=bounded"). *Let $\Omega : \mathbf{Set} \to \mathbf{Set}$ be a functor. Then $\mathbf{Set}_\Omega$ is bounded iff Ω is bounded.*

Remark. A recent study of bounded functors over **Set** can be found in Gumm and Schröder [43]. In particular, they characterise bounded functors (and hence bounded categories $\mathbf{Set}_\Omega$).

We remarked already that a bounded category with sinks in $\mathcal{E}$ being epi has a set of generators (see section A.6). The precise relation of boundedness and a set of generators is given by the following proposition. (We say that a category $\mathcal{C}$ has a *set $\mathcal{G}$ of $\mathcal{E}$-generators* iff for all objects $A \in \mathcal{C}$ the sink consisting of all morphisms from objects in $\mathcal{G}$ to A is an $\mathcal{E}$-sink.)

Proposition 1.5.6. *An $\mathcal{E}$-cowellpowered $(\mathcal{E}, M)$-category $\mathcal{C}$ is bounded iff it has a set of $\mathcal{E}$-generators.*

Proof. "only if" is obvious from the respective definitions (this direction does not need $\mathcal{E}$-cowellpoweredness). To show "if" assume that $\mathcal{G}$ is the set of generators. We have to show that there is an appropriate set $\mathcal{S}$ of small objects. Let $\mathcal{S} = \{C \in \mathcal{C} :$

there is $G \in \mathcal{G}$ and a morphism $G \to C$ in $\mathcal{E}\}$. By $\mathcal{E}$-cowellpoweredness we can assume $\mathcal{S}$ to be a set. To show that any object $A \in \mathcal{C}$ is the union of small objects, consider the sink $(s_i : G_i \to A)$ consisting of all morphisms from all $G_i \in \mathcal{G}$ to A. (s_i) is an $\mathcal{E}$-sink by generatedness. Now factor the s_i as $s_i = m_i \circ e_i$. Since the domains of the m_i are in $\mathcal{S}$, it remains to show that the sink (m_i) is in $\mathcal{E}$. This follows from (s_i) in $\mathcal{E}$ and e.g. proposition A.5.3(6). $\square$

For cowellpowered $(\mathcal{E}, M)$-categories being bounded is a more general notion than being accessible (or locally presentable) (see A.8 for definitions):

Proposition 1.5.7. *Let $\mathcal{C}$ be an $\mathcal{E}$-cowellpowered $(\mathcal{E}, M)$-category. Then $\mathcal{C}$ is accessible only if $\mathcal{C}$ is bounded.*

Proof. (Similar to "if" of the previous proposition.) Since $\mathcal{C}$ is accessible it has a set $\mathcal{G}$ of objects such that every object $A \in \mathcal{C}$ is the colimit $s_i : G_i \to A$ of a diagram in the full subcategory generated by $\mathcal{G}$. Let $\mathcal{S}$ be the closure of $\mathcal{G}$ under $\mathcal{E}$-quotients. $\mathcal{S}$ can assumed to be a set because $\mathcal{C}$ is $\mathcal{E}$-cowellpowered. Factor $s_i = m_i \circ e_i$. The sink (s_i) is extremal epi (because it is a colimiting cocone). It follows that the sink (m_i) is extremal epi as well, hence $(m_i) \in \mathcal{E}$ (proposition A.5.3(4))which shows that A is a union of objects in $\mathcal{S}$. (The m_i are even a colimiting cocone for an appropriate diagram resulting from factoring the s_i.) $\square$

Remark. Under the assumptions of the proposition boundedness is weaker than accessibility: The unions do not have to be colimits and the objects in $\mathcal{S}$ do not have to be κ-presentable.

Remark ("bounded=accessible"). In the case of coalgebras $\mathbf{Set}_\Omega$, it has recently been shown by Adámek [3] that for κ a regular infinite cardinal the functor Ω is $<$-bounded by κ iff Ω is κ-accessible.

Remark. Accessible categories of coalgebras are investigated in Power and Watanabe [91].

If a functor Ω is bounded then $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint. We can generalise this to bounded categories.

Theorem 1.5.8. *Let $\mathcal{C}$ be a bounded, cowellpowered, cocomplete $(\mathcal{E}, M)$-category with sinks in $\mathcal{E}$ being epi. Suppose that $U : \mathcal{C} \to \mathcal{X}$ is a functor that preserves colimits. Then U has a right adjoint.*

Proof. The theorem is an immediate consequence of the special adjoint functor theorem (see A.6.2) because the set of small objects in a bounded category is a set of generators if sinks in $\mathcal{E}$ are epi. $\square$

As a corollary one obtains final coalgebra theorems. Recall axioms 1 and 4 from section 1.4.

Corollary 1.5.9 (Final Coalgebra Theorem). *Let Ω be an endofunctor on $\mathcal{X}$ such that $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfies axioms 1 and 4. Then $\mathcal{X}_\Omega$ is bounded implies that $U : \mathcal{X}_\Omega \to \mathcal{X}$ has a right adjoint and, in particular, $\mathcal{X}_\Omega$ has a final coalgebra if $\mathcal{X}$ has a terminal object.*

Proof. $\mathcal{X}_\Omega$ is cowellpowered and cocomplete because $\mathcal{X}$ is and the sinks of the factorisation structure of sinks given by axiom 4 are epi (see proposition 1.4.3). $\square$

Corollary 1.5.10 (Aczel and Mendler [2]). *Let $\Omega : \mathbf{SET} \to \mathbf{SET}$ be a* **Set***-based functor.*[30] *Then $\mathbf{SET}_\Omega$ has a final coalgebra. Moreover, $U : \mathbf{SET}_\Omega \to \mathbf{SET}$ has a right adjoint.*

Proof. In the terminology of Aczel and Mendler, **SET** and $\mathbf{SET}_\Omega$ are superlarge categories with large homsets and $\mathbf{Set}_\Omega$ is a large subcategory of $\mathbf{SET}_\Omega$. To apply theorem 1.5.8 we now consider small/large in the usual sense (see section A.1) as large/superlarge in the sense of Aczel and Mendler.[31] That is, we can apply the theorem if there is a large set (= class) of generators, if every object has only a large set of quotients (and not a superlarge set) and if colimits exist for large diagrams. That $\mathbf{Set}_\Omega$ is a class of generators follows from the small subcoalgebra lemma (Aczel and Mendler [2], lemma 2.2). Cowellpoweredness of $\mathbf{SET}_\Omega$ follows from cowellpoweredness of **SET**. Cocompleteness (for large diagrams) of $\mathbf{SET}_\Omega$ follows from cocompleteness of **SET** which in turn holds since a large coproduct of classes is still a class and large quotients exist if one assumes a strong enough axiom of choice. □

Remark. The proof does not depend on special properties of the category of sets. One obtains in the same manner a general final coalgebra theorem in the sense of Aczel and Mendler [2], theorem 7.4.

[30] A functor is called **Set**-based in [2] if for each class A and each $a \in \Omega A$ there is a set $A_0 \subset A$, $A_0 \in \mathbf{Set}$, and $a_0 \in \Omega A_0$ such that $a = (\Omega\iota)(a_0)$, where ι is the inclusion map $A_0 \hookrightarrow A$.

[31] Theorem 1.5.8 relies on the SAFT. The smallness conditions in the SAFT (*set* of generators, cowellpowered = *set* of quotients) are needed because cocomplete means that colimits exist only for diagrams $D : I \to \mathcal{C}$ with I a *set*. That is, SAFT holds for a superlarge category $\mathcal{C}$ if $\mathcal{C}$ has a large set of generators and large colimits and if every object has—up to isomorphism—only a large set of epis.

1.6 Limits in Categories of Coalgebras

In categories of coalgebras colimits usually exist and are constructed as in the base category. Here we show that under reasonable conditions also limits exist and how they are constructed.

We present two proofs. A first version for the special case of coalgebras over the base category **Set**. In this case we can leave the apparatus of factorisation structures implicit. The second proof for the general case is actually the same, the factorisation structures serving only the purpose to allow for a clear and **Set**-independent way to state the precise assumptions that are necessary to make the proof possible.

Finally we compare our results with previous completeness proofs in the literature.

1.6.1 Limits in Categories of Coalgebras over Set

Before proving the theorem we recall some notions concerning sinks, for more details see the appendix.

A *sink* $(B, (s_i)_{i \in I})$ consists of an object B and a collection of morphisms $s_i : A_i \to B$ with common codomain B. We frequently write sinks as (s_i) and composition of sinks with a morphism f as $f \circ (s_i) = (f \circ s_i)$. Also, we use the same terminology for sinks as for morphisms. For example, a sink is called epi, or an *epi sink*, iff $f \circ (s_i) = g \circ (s_i)$ implies $f = g$.

The following two facts are important to note. First, a sink (e_i) is epi in $\mathbf{Set}_\Omega$ iff it is epi in **Set**. Second, epi sinks are *final*, that is, for any epi sink $(e_i)_{i \in I}$ in $\mathbf{Set}_\Omega$ and any function f in **Set**, it holds: if $f \circ e_i$ in $\mathbf{Set}_\Omega$ for all $i \in I$, then $f \in \mathbf{Set}_\Omega$.

Remark 1.6.1. The theorem below improves the earlier result by Power and Watanabe [91] because we prove more than the theorem states, namely how the limits in $\mathbf{Set}_\Omega$ are constructed: As certain subcoalgebras of the cofree coalgebras. So this theorem can be used to obtain detailed information on limits. As an example consider Rutten's example [109] of the following coalgebra A for the (finite) powerset functor:

$$\begin{tikzcd} & s_0 \arrow[dl] \arrow[dr] & \\ s_1 & & s_2 \end{tikzcd}$$

(The carrier of A is $\{s_0, s_1, s_2\}$ and the transition relation is as depicted in the diagram.) The reader is invited to use the construction in the proof of the theorem to prove the following remarks on the product $A \times A$:

- As noted in Rutten [109], the product $A \times A$ is not the largest bisimulation because the product has 'too many states' (the largest bisimulation on A has 5).[32]

- $A \times A$ is finite (the construction in the proof of the theorem also allows to calculate the precise number of states in $A \times A$ though this requires a bit more work).

[32]The reason for the product being too big is that the largest bisimulation has different possible transition relations which all have to be embeddable in the product.

- Define A' by adding transitions from s_1 and s_2 to s_0 in the coalgebra A. Then $A' \times A'$ is infinite.

Theorem 1.6.2. *Let Ω be a functor on* **Set** *such that the underlying functor $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint F. Then $\mathbf{Set}_\Omega$ is complete.*

Proof. Let $D : \mathcal{I} \to \mathbf{Set}_\Omega$ be a diagram in $\mathbf{Set}_\Omega$. Let $c_i : L \to UDi$ be a limit of UD in **Set**. Consider the cofree coalgebra FL over L and let $\varepsilon : UFL \to L$ be the arrow given by the counit of the adjunction.

$$\begin{tikzcd}
UC \arrow[r, "Um"] & UFL \arrow[d, "\varepsilon"] \\
UA_j \arrow[u, "Ue_j"] \arrow[ur, "Ug_j^\#", "Us_j"'] \arrow[r, "g_j"] \arrow[dr, "Uf_j^i"'] & L \arrow[d, "c_i"] \\
& UDi
\end{tikzcd}$$

Let A_j be a coalgebra and $f_j^i : A_j \to Di$ a cone for the diagram D. Since L is a limit of UD, there is a unique $g_j : UA_j \to L \in \mathbf{Set}$ such that $Uf_j^i = c_i \circ g_j$. Since FL is cofree g_j lifts to a unique $g_j^* : A \to FL$ such that $\varepsilon \circ Ug_j^* = g_j$.

We have seen that every cone $f_j^i : A \to Di$ gives rise to a $g_j^* : A \to FL$. Consider the sink (s_j) consisting of all these g_j^*. We can now define the limit of D: Let C be the subcoalgebra whose carrier consists of all images of all s_j and let $m : C \to FL$ be the corresponding embedding. That C and m are well defined as elements of $\mathbf{Set}_\Omega$ follows from Rutten [108], theorems 6.3 and 6.4.

Note that by definition of m and C, there is a sink (e_j) such that $(s_j) = m \circ (e_j)$, the e_j being the same as the s_j but the codomain restricted to C. Moreover, since every element of UC is in the image of some s_j, the sink (e_j) is epi, hence final.

To find the limiting cone consider $l_i = c_i \circ \varepsilon \circ Um$. By definition of (s_j), we have for all $i \in \mathcal{I}$ that $l_i \circ (Ue_j) = c_i \circ \varepsilon \circ (Us_j) = (Uf_j^i)$ is a sink in $\mathbf{Set}_\Omega$, hence l_i in $\mathbf{Set}_\Omega$ by the fact that (e_j) is final. Furthermore, l_i is a cone for D because it is a cone for UD which in turn holds because Uf_j^i is a cone for UD (for all j) and the sink (Ue_j) is epi.

To complete the proof we have to show that every cone in $\mathbf{Set}_\Omega$ over D factors uniquely through $l_i : C \to Di$. The existence follows from the definition of C, uniqueness from m being mono. ☐

1.6.2 The general theorem

We use some terminology about factorisation structures, see appendix A.5.

Theorem 1.6.3. *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category and $U : \mathcal{C} \to \mathcal{X}$ a faithful functor with right adjoint F. Suppose that sinks in $\mathcal{E}$ are final in $\mathcal{C}$ and epi in $\mathcal{X}$. Then $\mathcal{C}$ has every type of limit that $\mathcal{X}$ has. In particular, $\mathcal{C}$ is complete if $\mathcal{X}$ is.*

Proof. The proof is almost literally the same as in the previous section. Only that the definition of C via union of images is now replaced by the following. Let (s_j) be as before. By assumption on $\mathcal{X}$, (s_j) factors as $m \circ (e_j)$ with m being mono and (e_j) being final. Define C to be the domain of m. The remainder of the proof is as above. $\square$

Remark. In the case that $\mathcal{C}$ is a category of coalgebras $\mathcal{X}_\Omega$ we can drop the assumption that sinks in $\mathcal{E}$ are final because this follows from sinks being epi in $\mathcal{X}$, see proposition 1.3.5.

Recalling axioms 1 and 4 from section 1.4 we get the following corollary.

Corollary 1.6.4. *Let Ω be an endofunctor on $\mathcal{X}$ and suppose that $U : \mathcal{X}_\Omega \to \mathcal{X}$ satisfies axioms 1 and 4 and has a right adjoint. Then $\mathcal{X}_\Omega$ has every type of limit that $\mathcal{X}$ has and the limit is constructed as in the proof of the theorem.*

Comparison with Other Results

Finally, let us compare our result with the ones in Power and Watanabe [91] and Worrell [127]. The result of Power and Watanabe states that if the base category $\mathcal{X}$ is locally presentable and Ω is accessible then $\mathcal{X}_\Omega$ is complete. (They also show that under these assumptions U has a right adjoint.) The result of Worrell (obtained by dualising a corresponding result on algebras for a monad) states that a category of coalgebras for a comonad is complete if it has equalisers and the base category is complete. (Here, the right adjoint of U is built into the notion of a comonad.)

We have seen that all three results involve in some form the existence of a right adjoint of the underlying functor U. A difference lies in the relationship of the limits in $\mathcal{X}$ and the limits in $\mathcal{X}_\Omega$. [91, 127] use completeness of the base category $\mathcal{X}$ to show completeness of $\mathcal{X}_\Omega$. (And [127] moreover needs that $\mathcal{X}_\Omega$ has equalisers.) We have a sharper result: for every type of limit in $\mathcal{X}$ we show how the corresponding limit in $\mathcal{X}_\Omega$ is obtained. This dualises the fact that 'algebraic' functors detect colimits, see Adámek, Herrlich, Strecker [4], 23.11.

During the writing of this section, a construction similar to theorem 1.6.3 has independently been given by Gumm and Schröder [40]. In fact, their proof is essentially the same as ours specialised to $\mathcal{X} = \mathbf{Set}$, $\mathcal{C} = \mathbf{Set}_\Omega$ and the limit under consideration being the product.

1.7 Hidden and Multiplicative Signatures

In specification formalisms using algebras and/or coalgebras one often restricts attention to special signature functors, namely, hidden signatures in case of algebras and multiplicative functors in case of coalgebras. We show here that we can characterise, roughly speaking, hidden signatures as functors on $\mathbf{Set^n}$ having a right adjoint (and these right adjoints are multiplicative) and multiplicative functors as functors on $\mathbf{Set^n}$ having a left adjoint (and these left adjoints are hidden signatures). As a consequence, categories of hidden algebras are isomorphic to categories of coalgebras for multiplicative functors.

This section is an extended version of the section 'Deterministic Functors' in [71].

1.7.1 Isomorphic Categories of Algebras and Coalgebras

An adjunction $\Sigma \dashv \Xi$ between functors Σ, Ξ, gives rise to an isomorphism between the categories of Σ-algebras and Ξ-coalgebras:

Theorem 1.7.1. *Let $\Xi : \mathcal{X} \to \mathcal{X}$ be a functor and Σ a left adjoint of Ξ. Then the category $\mathcal{X}_\Xi$ of Ξ-coalgebras is isomorphic to the category $\mathcal{X}^\Sigma$ of Σ-algebras.*

Proof. For $X, Y \in \mathcal{X}$, let $\varphi_{X,Y} : \mathcal{X}(X, \Xi Y) \to \mathcal{X}(\Sigma X, Y)$ be the natural isomorphism given by the adjunction. The required isomorphism between the category of Ξ-coalgebras and Σ-algebras is then given on objects by $(X, \xi : X \to \Xi X) \mapsto (X, \varphi_{X,X}(\xi) : \Sigma X \to X)$ and on morphisms by the identity (naturality of φ guarantees that coalgebra morphisms are indeed algebra morphisms). $\square$

This theorem seems not to be very interesting at first sight because the only functors Ξ on $\mathbf{Set}$ that have a left adjoint are of the form $\Xi X = X^A$ for some $A \in \mathbf{Set}$ (see Arbib and Manes [7] or lemma 1.7.13 below). The important idea that comes in now is to make (some of) the parameters of the functor explicit. For example take $\Omega X = B \times X$. Ω has no left adjoint but one can make the parameter B explicit by defining a functor Ξ on $\mathbf{Set}^2$ as $\Xi(X, B) = (B \times X, 1)$. Now, Ξ has a left adjoint given by $\Sigma(X, B) = (X, X)$ and, moreover, we still have the Ω-coalgebras as those Ξ-coalgebras for which the second component is B (this will be made precise in proposition 1.7.4).

We will call a signature functor Ω an algebraic signature for coalgebras whenever Ω can be extended to a functor that has a left adjoint in the way shown above. More precisely:

Definition 1.7.2 (algebraic signature for coalgebras). An endofunctor Ω on $\mathcal{X}$ is called an *algebraic signature for coalgebras* if there is a *parameter category* $\mathcal{L}$ with terminal object 1 and an object $L \in \mathcal{L}$ such that there is a functor $\Xi' : \mathcal{X} \times \mathcal{L} \to \mathcal{X}$ with $\Xi'(X, L) \simeq \Omega X$ and such that $\Xi : \mathcal{X} \times \mathcal{L} \to \mathcal{X} \times \mathcal{L}$, $(X, M) \mapsto (\Xi'(X, M), 1)$, has a left adjoint.

Before giving examples we want to make precise in which sense the Ω-coalgebras appear as Ξ-coalgebras. To this end recall that objects/morphisms in $\mathcal{X} \times \mathcal{L}$ are just pairs of objects/morphisms and identity and composition are defined componentwise. We can now say that $\mathcal{X}_\Omega$ is isomorphic to the subcategory of $(\mathcal{X} \times \mathcal{L})_\Xi$ where the carriers of the coalgebras have second component L (using the notation from the definition above) and the morphisms have second component id_L. These subcategories are called fibres over L:

Definition 1.7.3 (fibres). For a functor $\Xi : \mathcal{X} \times \mathcal{L} \to \mathcal{X} \times \mathcal{L}$ define a functor $p : (\mathcal{X} \times \mathcal{L})_\Xi \to \mathcal{L}$ as follows: On coalgebras p is the second projection on the carriers, on coalgebra morphisms p is the second projection. For $L \in \mathcal{L}$ the subcategory of $(\mathcal{X} \times \mathcal{L})_\Xi$ consisting of objects A with $pA = L$ and of morphisms f with $pf = \mathrm{id}_L$ is called the fibre of $(\mathcal{X} \times \mathcal{L})_\Xi$ over L. In the same way define fibres for the category $(\mathcal{X} \times \mathcal{L})^\Xi$ of Ξ-algebras.

Recall the definition of Ξ extending a functor Ω on $\mathcal{X}$ as in definition 1.7.2. We now make precise the relation of $\mathcal{X}_\Omega$ and $(\mathcal{X} \times \mathcal{L})_\Xi$.

Proposition 1.7.4. *Let $\mathcal{X}$, Ω, $\mathcal{L}$, L, Ξ as in definition 1.7.2. Then $\mathcal{X}_\Omega$ is isomorphic to the fibre of $(\mathcal{X} \times \mathcal{L})_\Xi$ over L.*

To summarise, the extension of Ω by parameters to a functor Ξ gives us a larger category of coalgebras which allows for new constructions (such as finding an isomorphic category of algebras). On the other hand nothing is lost, since the categories of Ω-coalgebras now appear as fibres. This point of view has been introduced and developed in Kurz and Pattinson [71].

Example 1.7.5. In all examples $\Omega : \mathbf{Set} \to \mathbf{Set}$, $\mathcal{X} = \mathbf{Set}$, $A, B, L, M \in \mathbf{Set}$.

1. Let $\Omega X = X^A$. Then Ω is an algebraic signature for coalgebras as witnessed by $\mathcal{L} = \mathbf{1}$, $\Xi = \Xi' = \Omega$.
2. Let $\Omega X = B$. Then Ω is an algebraic signature for coalgebras as witnessed by $\mathcal{L} = \mathbf{Set}$, $L = B$, and $\Xi'(X, M) = M$. The left adjoint of Ξ is $\Sigma(X, M) = (0, X)$.
3. Let $\Omega X = B \times X$. Then Ω is an algebraic signature for coalgebras for as witnessed by $\mathcal{L} = \mathbf{Set}$, $L = B$, and $\Xi'(X, M) = M \times X$. The left adjoint of Ξ is $\Sigma(X, M) = (X, X)$.
4. Let $\Omega X = X + X$. Then Ω is an algebraic signature for coalgebras because $\Omega X \simeq 2 \times X$.
5. The functors $\Omega X = X + X^2$, $\Omega X = 1 + X$ and $\Omega X = \mathcal{P}X$ are not algebraic signatures for coalgebras? — The general case seems to be difficult to answer but if we restrict the parameter category $\mathcal{L}$ to be $\mathbf{Set^n}$ for some $n \in \mathbb{N}$ the answer is given by example 1.7.18.

An example of algebraic signatures for coalgebras are multiplicative functors.

Definition 1.7.6 (multiplicative functors). Let $\mathcal{X}$ be a cartesian closed category. A multiplicative functor on $\mathcal{X}$ is a endofunctor Ω on $\mathcal{X}$ built according to the following abstract syntax

$$\begin{aligned} \Omega &::= \Omega \times \Omega \mid F \\ F &::= B \mid G \\ G &::= \mathrm{Id} \mid G^A \end{aligned}$$

The objects B are called *output parameters*.

Proposition 1.7.7. *Let $\mathcal{X}$ be a bicartesian closed category and Ω a multiplicative functor on $\mathcal{X}$. Then Ω is an algebraic signature for coalgebras.*

Proof. Every multiplicative functor can be written as $\Xi X = \prod_{i=1}^{m} X^{A_i} \times \prod_{j=1}^{n} B_j$. Making the parameters B_j explicit this can be written as a functor $\Xi : \mathcal{X} \times \mathcal{X}^n \to \mathcal{X} \times \mathcal{X}^n$ with

$$\Xi \begin{pmatrix} X \\ B_1 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} \prod_{i=1}^{m} X^{A_i} \times \prod_{j=1}^{n} B_j \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

This functor has a left adjoint Σ:

$$\Sigma \begin{pmatrix} X \\ B_1 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} A_i \times X \\ X \\ \vdots \\ X \end{pmatrix}$$

□

Remark. Note that we would get no adjunction if we made explicit also the parameters A_i. This is due to the A_i appearing contravariantly in Ξ-coalgebras and covariantly in Σ-algebras. Intuitively speaking, the parameters that have to be made explicit are the 'output' parameters.

Also, we can do the same for algebras, now making explicit the 'input' parameters or constants. For example, consider the algebras for the functor $H : \mathcal{X} \to \mathcal{X}, HX = C + A \times X$. H can be viewed as a functor $\Sigma' : \mathcal{X} \times \mathcal{X} \to \mathcal{X}, (X, C) \mapsto C + A \times X$ and also as a functor $\Sigma : \mathcal{X} \times \mathcal{X} \to \mathcal{X} \times \mathcal{X}, (X, C) \mapsto (C + A \times X, 0)$ where 0 denotes the initial object of $\mathcal{X}$. This last functor has a right adjoint.

Definition 1.7.8 (coalgebraic signature for algebras). An endofunctor H on $\mathcal{X}$ is called a *coalgebraic signature for algebras* if there is a category $\mathcal{L}$ with initial object 0 and there is $L \in \mathcal{L}$ such that there is a functor $\Sigma' : \mathcal{X} \times \mathcal{L} \to \mathcal{X}$ with $\Sigma'(X, L) = HX$ and such that $\Sigma : \mathcal{X} \times \mathcal{L} \to \mathcal{X} \times \mathcal{L}$, $(X, M) \mapsto (\Sigma'(X, M), 0)$, has a right adjoint.

In the following we show that an example of coalgebraic signatures for algebras are hidden signatures as in the approach of hidden algebra by Goguen and Malcolm [34].

Definition 1.7.9 (hidden signature). Let $\mathcal{X}$ be a category with binary products and binary coproducts. A hidden signature on $\mathcal{X}$ is a endofunctor H on $\mathcal{X}$ built according to the following abstract syntax

$$\begin{aligned} H &::= H + H \mid F \\ F &::= C \mid G \\ G &::= \mathrm{Id} \mid A \times G \end{aligned}$$

The objects C are called *constants.*

Remark. The important point about this definition is that in each summand of H there is at most one occurrence of Id. That is—in the terminology of the hidden algebra approach—the algebraic operations defined by H take at most one argument of 'hidden sort' X.

Remark. Note that this definition allows—as in the case of multiplicative functors—many-sorted hidden signatures because $\mathcal{X}$ itself may be a product of categories.

Proposition 1.7.10. *Let $\mathcal{X}$ be a bicartesian closed category. Then hidden signatures on $\mathcal{X}$ are coalgebraic signatures for algebras.*

Proof. Every hidden signature can be written as $HX = \sum_{j=1}^{n} C_j + \sum_{i=1}^{m} A_i \times X$. Making the parameters C_j explicit this can be written as a functor $\Sigma : \mathcal{X} \times \mathcal{X}^n \to \mathcal{X} \times \mathcal{X}^n$ with (0 denoting the initial object of $\mathcal{X}$)

$$\Sigma \begin{pmatrix} X \\ C_1 \\ \vdots \\ C_n \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{n} C_j + \sum_{i=1}^{m} A_i \times X \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

This functor has a right adjoint Ξ:

$$\Xi \begin{pmatrix} X \\ C_1 \\ \vdots \\ C_n \end{pmatrix} = \begin{pmatrix} \prod_{i=1}^{m} X^{A_i} \\ X \\ \vdots \\ X \end{pmatrix}$$

□

Let us summarise the results by the following theorems which follow from propositions 1.7.7 and 1.7.10.

Theorem 1.7.11. *Let $\mathcal{X}$ be a bicartesian closed category. A multiplicative functor Ω on $\mathcal{X}$ with n output sorts can be extended to a multiplicative functor Ξ on $\mathcal{X} \times \mathcal{X}^n$ (with no output parameters) such that*

1. *there is $(L_1, \dots L_n) \in \mathcal{X}^n$ such that the fibre of $(\mathcal{X} \times \mathcal{X}^n)_\Xi$ over $(L_1, \dots L_n)$ is isomorphic to $\mathcal{X}_\Omega$.*

Moreover,

2. *Ξ has a left adjoint Σ which is a hidden signature on $\mathcal{X} \times \mathcal{X}^n$ (with no constants),*
3. *$(\mathcal{X} \times \mathcal{X}^n)_\Xi$ and $(\mathcal{X} \times \mathcal{X}^n)^\Sigma$ are fibrewise isomorphic.*[33]

We will show below that in the case of signatures over set there holds a kind of converse to this theorem, namely that every signature that can be extended to a functor with left adjoint is multiplicative, see theorem 1.7.17.

Theorem 1.7.12. *Let $\mathcal{X}$ be a bicartesian closed category. A hidden signature H on $\mathcal{X}$ with n constants can be extended to a hidden signature Σ on $\mathcal{X} \times \mathcal{X}^n$ (with no constants) such that*

[33] *Fibrewise isomorphic* means in particular that for all $(L_1, \dots L_n) \in \mathcal{X}^n$ the fibre of $(\mathcal{X} \times \mathcal{X}^n)_\Xi$ over $(L_1, \dots L_n)$ is isomorphic to the fibre of $(\mathcal{X} \times \mathcal{X}^n)^\Sigma$ over $(L_1, \dots L_n)$.

1. *there is* $(L_1, \dots L_n) \in \mathcal{X}^n$ *such that the fibre of* $(\mathcal{X} \times \mathcal{X}^n)^\Sigma$ *over* $(L_1, \dots L_n)$ *is isomorphic to* $\mathcal{X}^H$.

Moreover,

2. Σ *has a right adjoint* Ξ *which is a multiplicative functor on* $\mathcal{X} \times \mathcal{X}^n$ *(with no output parameters),*

3. $(\mathcal{X} \times \mathcal{X}^n)_\Xi$ *and* $(\mathcal{X} \times \mathcal{X}^n)^\Sigma$ *are fibrewise isomorphic.*

We will show below that in the case of signatures over set there holds a kind of converse to this theorem, namely that every signature that can be extended to a functor with right adjoint is a hidden signature, see theorem 1.7.20.

1.7.2 Hidden and Multiplicative Signatures over Set

We first characterise those functors on $\mathbf{Set^n}$ that have a left or right adjoint, generalising theorem 5.7 in Arbib and Manes [7] [34] from $\mathbf{Set}$ to $\mathbf{Set^n}$.

Lemma 1.7.13. *Let* Σ *be a functor on* $\mathbf{Set^n}$. *Then the following are equivalent.*

1. Σ *has a right adjoint.*

2. Σ *preserves coproducts.*

3. *There is a* $(n \times n)$*-matrix* M *over* $\mathbf{Set}$ *such that* $\Sigma X = MX$, $X \in \mathbf{Set^n}$.[35]

Proof. "(1) $\Rightarrow$ (2)" is a standard result on adjoints.
"(2) $\Rightarrow$ (3)": Let $1 \le i \le n$. Write X_i for the i-th component of X and E^i for the vector in $\mathbf{Set^n}$ that has 0 everywhere but 1 in the i-th component. Then $\Sigma X = \Sigma(\sum_{1\le i \le n} X_i \times E^i) = \sum_{1\le i\le n} \Sigma(X_i \times E^i) = \sum_{1\le i\le n} \Sigma(\sum_{|X_i|} E^i) = \sum_{1\le i\le n} \sum_{|X_i|} \Sigma E^i = \sum_{1\le i\le n} X_i \times \Sigma E^i$, using that Σ preserves coproducts. Now define the components of M by letting M_{ij} be the j-th component of ΣE^i.
"(3) $\Rightarrow$ (1)": Let $\Sigma X = MX$ for some $(n \times n)$-matrix M over $\mathbf{Set}$. Then define a right adjoint Ξ of Σ by $X_i \mapsto \prod_{1 \le j \le n} X_j^{M_{ij}}$. $\square$

This lemma characterises functors on $\mathbf{Set^n}$ that have a left or right adjoint. Moreover, the proof of the lemma also shows that having a left adjoint means being multiplicative without output parameters.

Theorem 1.7.14 (characterisation of right adjoints). *Let* Ξ *be a functor on* $\mathbf{Set^n}$. *Then* Ξ *has a left adjoint iff* Ξ *is (isomorphic to) a multiplicative functor on* $\mathbf{Set^n}$ *without output parameters.*

[34] I would like to thank Bart Jacobs for pointing out this theorem.

[35] MX is matrix multiplication, thinking of X as a vector and using the operations $+, \times$ on sets as addition and multiplication.

Proof. " $\Rightarrow$ ": Consider a left adjoint Σ of Ξ. Ξ is isomorphic to a (or every) right adjoint of Σ. By (3) of the lemma, Σ is given by $\Sigma X = MX$ for some matrix M. Now the claim follows, since the proof of (3) $\Rightarrow$ (1) of the lemma shows that Σ has a multiplicative right adjoint. Moreover, in the right adjoint of Σ no output parameters occur.
" $\Leftarrow$ ": This is a consequence of the proof of proposition 1.7.7. □

And, similarly, we can say that having a right adjoint means being a hidden signature without constants.

Theorem 1.7.15 (characterisation of left adjoints). *Let Σ be a functor on* $\mathbf{Set^n}$. *Then Σ has a right adjoint iff Σ is (isomorphic to) a hidden signature without constants.*

Proof. " $\Rightarrow$ ":By (3) of the lemma, Σ is given by $\Sigma X = MX$ for some matrix M. It not difficult to see that Σ is therefore a hidden signature without constants.
" $\Leftarrow$ ": This is a consequence of the proof of proposition 1.7.10. □

If we restrict our attention to categories built from products of **Set** we can now characterise the functors Ω that can be extended to a functor Ξ having a left adjoint. We first define what is meant here by extending a functor Ω.

Definition 1.7.16 (n-sorted algebraic signature for coalgebras over Set). An n-sorted algebraic signature for coalgebras over **Set** is a functor $\Omega : \mathbf{Set^n} \to \mathbf{Set^n}$ such that there is $m \in \mathbb{N}$ and $L \in \mathbf{Set^m}$ such that there is a functor $\Xi' : \mathbf{Set^{n+m}} \to \mathbf{Set^n}$ with $\Xi'(X, L) \simeq \Omega X$ and such that $\Xi : \mathbf{Set^{n+m}} \to \mathbf{Set^{n+m}}$, $(X, M) \mapsto (\Xi'(X, M), 1)$, has a left adjoint.

We can now characterise the functors Ω that can be extended to a functor Ξ having a left adjoint as those functors that are multiplicative.

Theorem 1.7.17 (characterisation of multiplicative functors). *Let Ω be a functor on* $\mathbf{Set^n}$. *Then Ω is (isomorphic to) a multiplicative functor on* $\mathbf{Set^n}$ *iff Ω is a n-sorted algebraic signature for coalgebras over* **Set**.

Proof. " $\Leftarrow$ ": Using the notation of definition 1.7.16 we have to show that $\Xi'(X, L) \simeq \Omega X$ is multiplicative. This follows from a calculation using that Ξ has a left adjoint that is given, for some $m \in \mathbb{N}$, by a $(n+m, n+m)$-matrix M as in lemma 1.7.13(3).
" $\Rightarrow$ ": This is as in proposition 1.7.7. □

Example 1.7.18. We can now show that the functors $\Omega X = X + X^2$, $\Omega X = 1 + X$ and $\Omega X = \mathcal{P}X$ from example 1.7.5(5) are not algebraic signatures for coalgebras over **Set**. We do the case $\Omega X = X + X^2$, the other cases are proved using a similar cardinality argument. Suppose that $\Omega X = X + X^2$ is an algebraic signature for coalgebras over **Set**. It follows from theorem 1.7.17 that Ω is isomorphic to a multiplicative functor $\Xi' X = B \times X^A$ (every 1-sorted multiplicative functor is of the form $B \times X^A$ for some A, B). Clearly, A, B have to be finite sets, with cardinalities $a, b \in \mathbb{N}$, respectively. Since isomorphisms in **Set** are bijections it follows that $x + x^2 = bx^a$ for all for all $x \in \mathbb{N}$. But this contradicts the fact that two such numbers $a, b \in \mathbb{N}$ do not exist. A similar argument works in the other two cases (for $\Omega X = \mathcal{P}X$ use that $|\mathcal{P}X| = 2^{|X|}$).

Similarly, we can characterise the functors H that can be extended to a functor Σ having a right adjoint. We first define what we mean here by extending a functor H to a functor Σ

Definition 1.7.19 (n-sorted coalgebraic signature for algebras over Set). An n-sorted coalgebraic signature for algebras over $\mathbf{Set}$ is a functor $H : \mathbf{Set^n} \to \mathbf{Set^n}$ such that there is $m \in \mathbb{N}$ and $L \in \mathbf{Set^m}$ such that there is a functor $\Sigma' : \mathbf{Set^{n+m}} \to \mathbf{Set^n}$ with $\Sigma'(X, L) \simeq \Omega X$ and such that $\Sigma : \mathbf{Set^{n+m}} \to \mathbf{Set^{n+m}}$, $(X, M) \mapsto (\Sigma'(X, M), 0)$, has a right adjoint.

We can now characterise the functors H that can be extended to a functor Σ having a right adjoint as hidden signatures.

Theorem 1.7.20 (characterisation of hidden signatures). *Let H be a functor on $\mathbf{Set^n}$. Then H is a hidden signature on $\mathbf{Set^n}$ iff H is (isomorphic to) a n-sorted coalgebraic signature for algebras over* $\mathbf{Set}$.

Proof. " $\Leftarrow$ ": Using the notation of definition 1.7.19 we have to show that $\Sigma'(X, L) \simeq HX$ is a hidden signature. This follows from a calculation using that Σ is given, for some $m \in \mathbb{N}$, by a $(n + m, n + m)$-matrix M as in lemma 1.7.13(3).
" $\Rightarrow$ ": This is as in proposition 1.7.10. □

Conclusion

The results of this section shed a new light on hidden algebra and on the question of whether modal or equational logics are appropriate to specify coalgebras. Concerning hidden algebra, we can say now that hidden signatures are precisely those signatures which give rise to an adjunction as described in theorem 1.7.1 (see theorems 1.7.15 and 1.7.20). That is, hidden algebras are essentially coalgebras. Concerning the logics, the use of equational logic for coalgebras in the work of Corradini [29], Roşu [98], and in chapter 4 can now be explained by the fact that the signatures considered there are in fact 'algebraic' signatures for coalgebras. Algebraic signatures for coalgebras have been called deterministic functors in [71], they give rise to deterministic coalgebras. The considerations in this section give a new explanation of why equational logic is an appropriate language for deterministic coalgebras.

1.8 Tables of Dualities

We just list some dualities. Some obvious ones are:

Coalgebra	Algebra
Colimit	Limit
Right Adjoint	Left Adjoint
Cofree Coalgebra	Free Algebra
Final/Terminal Coalgebra	Initial Algebra

One interesting consequence of categorical duality is the duality of quotients and images. Note that, given a factorisation system (E, M) for a category $\mathcal{C}$, (M, E) is a factorisation system for $\mathcal{C}^{\mathrm{op}}$. First, again, the obvious ones:

Coalgebra	Algebra
Quotient	Image/Subalgebra
Image/Subcoalgebra	Quotient
Mono	Epi
Extremal Mono	Extremal Epi
Strong Mono	Strong Epi
Regular Mono	Regular Epi

And some perhaps less obvious ones:

Coalgebra	Algebra
Bisimulation	—
Cocongruence	Congruence
Behavioural Equivalence	Subalgebra
Largest Behavioural Equivalence	Smallest Subalgebra
Behaviour	Reachable Part

Finally, some of the dualities which are important in chapter 2 are listed below:

Coalgebra	Algebra
Subcoalgebra (S)	Quotient (H)
Quotient (H)	Subalgebra (S)
Coproduct (Σ)	Product (P)
Covariety	Variety
Coquasivariety	Quasivariety
Coreflective Subcategory	Reflective Subcategory

1.9 Conclusion and Future Directions of Research

We have answered open questions in universal coalgebra, namely how to construct limits in categories of coalgebras (section 1.6), and what the precise relationship is between hidden signatures for algebras and multiplicative signatures for coalgebras (section 1.7).

It has also been shown that the technique of factorisation systems which is at the heart of chapter 2 and section 1.6 also serves for an axiomatic treatment of universal coalgebra (section 1.4). To get rid of certain limitations like signatures preserving weak pullbacks or epis in the base category being split, behavioural equivalences and cocongruences have been introduced (section 1.2). The notion of a bounded category has been introduced in order to capture in an set-independent way the essence of coalgebras for a bounded functor (section 1.5).

What had to be left open for future research is to test these new concepts in applications. For example, we think that in cases where the notions of behavioural equivalence and bisimulation equivalence do not agree, the more relevant one will be behavioural equivalence. An immediate gain from this new concept is given in chapter 4 but further investigations will have to be made. Another step that has to be taken in the future, is to investigate concrete examples of coalgebras over other base categories than **Set**.

Chapter 2

Modal Logic and Coalgebras

The main insight of this chapter is that, in the presence of cofree coalgebras, one can give a semantics to formulas of modal logic by considering modal formulas as subcoalgebras of cofree coalgebras. This idea is used to show that

- Kripke semantics of modal logic is dual to the (algebraic) semantics of equational logic,
- one can characterise the expressive power of infinitary modal logics on Kripke frames (and more generally on coalgebras) by dualising the classical theorems characterising the expressive power of equational and implicational logic for algebras.
- Moreover, as a consequence of our approach, we see how to formally dualise Birkhoff-type theorems and how to interpret the duals in terms of modal logic and coalgebras.

Section 2.1 contains some preliminaries, section 2.2 collects basic facts on coreflective subcategories, section 2.3 presents the interpretation of equations as quotients of free algebras introduced in Banaschewski and Herrlich [11]. This allows us to see that the account of modal logic given in section 2.4 is categorically dual to the account of equations in [11]. Section 2.5 takes advantage of this observation by dualising the proofs of (generalised) Birkhoff theorems in [11]. Section 2.6 further develops the idea that "formulas are morphisms".

The following sections are devoted to apply the abstract co-Birkhoff results of section 2.5 to characterise the expressive power of infinitary modal logic on coalgebras and Kripke frames. Section 2.7 presents a common framework for the application of the co-Birkhoff results to coalgebraic logic (section 2.8) and infinitary modal logic (section 2.9).

The idea to interpret modal formulas as subcoalgebras as well as the slogan that modal logic is dual to equational logic goes back to [75, 74]. The formal justification of the slogan by a categorical analysis of equational and modal logic also appeared as [76].

2.1 Preliminaries

As the author had to learn from reactions of parts of the logic in computer science community, the use of category theory still needs to be defended and justified: As someone said about a previous version of this chapter, "I fail to see what one gains from restating basic algebraic and logical definitions in the categorical framework" and "Birkhoff's theorem has a short statement that is understood by every living mathematician; what reflective subcategories are is known to a tiny group of people".

In the following, we answer the first question concerning the gains from category theory and, then, try to convince the reader that the concept of a reflective subcategory is simple and really at the heart of *the proof* of Birkhoff's theorem. Finally, we present an extended example showing that the concept of a coreflective subcategory has an intuitive interpretation from the point of view of modal logic.

2.1.1 Why Using Category Theory

We will briefly discuss why it seemed necessary to develop this chapter in the language of category theory.

First, the gains from using category theory are threefold: More general results, simpler (and reusable) proofs, and—to the author's opinion most importantly—new insights in why certain results hold.

Second, a large part of this chapter is devoted to the question of duality of equational and modal logic. It is hardly conceivable to state this question, let alone to answer it, without the use of category theory: To make the question of duality precise we have to have a definition of what duality means; and it is one of the strong points of category theory that it provides us with a general and powerful formalisation of this concept.

But still it remains true that—apart from the duality issue—the most interesting results of this chapter, namely the characterisation of the expressive power of infinitary modal logics on Kripke frames, can be obtained by making less use of category theory (and this has been done in [75, 74]). In the following we therefore want to further explain what is gained from using category theory.

The main reason that the use of category theory pays off in proving (i.e. understanding) (co)Birkhoff theorems is that category theory allows us to extract the essential ingredients needed to prove these theorems.[1] That is, using category theory, we are able

- to take care that the statements of the theorems depend only on those assumptions that are really needed in the proofs,
- to concentrate on the key concepts making the proofs work.

It is not difficult to imagine that as a consequence we obtain more general results, simpler (and reusable) proofs and new insights. These points will be further illustrated by the discussion of Birkhoff theorems in the next subsection. In our context, more general results include

[1]Moreover, even if one does not use category theory explicitly it is still a powerful tool to organise the proofs, see the treatment in Wechler [124] for an example. Nevertheless, the elegance and simplicity of the proofs becomes more difficult to appreciate.

that the covariety theorem is proved without the assumption that the signature functor is bounded and that the results generalise to categories of coalgebras over other categories than **Set**.[2] Proofs get simpler because we replace arguments involving variables, terms, equations, formulas, etc by arguments using factorisation systems. In particular, it turns out that—roughly speaking—the only assumption that is needed to obtain co-Birkhoff results is the existence of a factorisation structure for sinks. The key concept making the proofs work is that of an M-coreflective subcategory (see section 2.2 for definition and results (in particular corollary 2.2.4) and section 2.1.3 for an example). Moreover, the concept of a M-coreflective subcategory is also responsible for being able to reuse the proof of the (co)variety theorem for different logics, see the next subsection.

2.1.2 Birkhoff's Variety Theorem and Reflective Subcategories

We recall the definition of reflective subcategories and discuss their use in the proof of Birkhoff's variety theorem. We also indicate that this proof can easily be adapted ('reused') to characterise the expressive power of Horn formulas and implications (quasivariety theorems). The approach to Birkhoff's and related theorems presented here is due to Banaschewski and Herrlich [11].

Recall that a *variety* is a subcategory closed under homomorphic images, subalgebras and products. Let $\mathbf{Set}^\Omega$ be the class of algebras for some signature Ω. Birkhoff's variety theorem states: A class (or subcategory) of Ω-algebras is equationally definable iff it is a variety.

How is this theorem proved? The easy direction is, as usual, to show that definability implies the closure properties. The interesting direction is to see why every variety is definable. The essential idea is to show that every variety is a reflective subcategory[3] and from this description as a reflective subcategory we easily see which equations define the variety.

To be more explicit, let Ω be a functor on **Set** and let $\mathbf{Set}^\Omega$ be the category of Ω-algebras. Furthermore suppose that $\mathbf{Set}^\Omega$ has free algebras FX over X for all sets X.[4]

Now, $\mathcal{B} \subset \mathbf{Set}^\Omega$ is defined to be a **reflective subcategory** of $\mathbf{Set}^\Omega$ iff for all $A \in \mathbf{Set}^\Omega$ there is an algebra $\varrho A \in \mathcal{B}$ and an algebra morphism $\eta_A : A \to \varrho A$ such that for all $B \in \mathcal{B}$ and all algebra morphisms $f : A \to B$ there is a unique $g : \varrho A \to B$ such that

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[dr, "f"'] & \varrho A \arrow[d, dotted, "g"] \\ & B \end{tikzcd}$$

commutes.[5] The morphisms η_A are called *reflection morphisms*.

[2] Note that in order to obtain a duality result we are forced to consider other categories than **Set** because algebras over **Set** are dual to coalgebras over complete atomic boolean algebras.

[3] Even if you don't call it 'reflective subcategory' you will have to use this concept in the *proof* of the variety theorem.

[4] Categorically speaking, the forgetful functor $U : \mathbf{Set}^\Omega \to \mathbf{Set}$ has a left adjoint F.

[5] Categorically speaking, the inclusion $\iota : \mathcal{B} \hookrightarrow \mathbf{Set}^\Omega$ has a left adjoint ϱ with unit η.

To explain the interest in this definition suppose that $\mathcal{B}$ is an equationally defined subcategory. That is, the given equations specify for each algebra $A \in \mathbf{Set}^{\Omega}$ a quotient ϱA which is in $\mathcal{B}$. The map η_A is the natural morphism onto the quotient. The factorisation property of the diagram expresses that ϱA is the quotient w.r.t. the *smallest* congruence relation generated by the equations defining $\mathcal{B}$.

This definition of a reflective subcategory now enables us to find for every variety a defining class of equations. First, it is not difficult to show (dualise theorem 2.2.2 and proposition 2.2.3) that every variety is a reflective subcategory closed under homomorphic images. Next, one observes that the kernel of the reflection morphism $\eta_{FX} : FX \to \varrho FX$ is an equivalence relation giving us equations Φ_X in variables over X (recall that the elements in the carriers of the free algebra FX are precisely the terms in variables X; consequently, equivalence relations on the carrier of FX correspond to equations $t = t'$ where t, t' are terms in variables X). Using that an algebra $B \in \mathbf{Set}^{\Omega}$ *satisfies the equations* Φ_X iff[6] for all algebra morphisms $f : FX \to B$ there is g such that

$$\begin{tikzcd} FX \arrow[r, "\eta_{FX}"] \arrow[dr, "f"'] & \varrho FX \arrow[d, dotted, "g"] \\ & B \end{tikzcd}$$

commutes, it is now easy to check that the class of all equations $\Phi = \bigcup_{X \in \mathbf{Set}} \Phi_X$ defines the variety $\mathcal{B}$ (dualise the proof of (3) $\Rightarrow$ (1) of theorem 2.5.4).

Having seen, how straight forward a proof of Birkhoff's variety theorem becomes once we are aware of the notion of a reflective subcategory, we want to take a closer look at the techniques needed to establish that varieties are precisely the reflective subcategories closed under homomorphic images. Looking at textbook presentations of proofs of Birkhoff's variety theorem (e.g., Burris and Sankappanavar [21] and Wechler [124]) one sees a lot of calculations using variables and terms which tend to obscure the general ideas. Moreover, in view of our interest in co-Birkhoff theorems, it is not at all clear how these arguments involving variables and terms (and hence special properties of **Set**) can be dualised to the case of coalgebras and modal logic. Fortunately, it turns out that arguments involving variables and terms are not needed at all to prove Birkhoff's theorem. Roughly speaking, it suffices to assume that every algebra morphism f can be uniquely factored as $f = m \circ e$ where e is a quotient and (the domain of) m is a subalgebra (the image of f).

That is, using factorisation systems, we free ourselves from the category **Set** (more general results), we do not need to use variables and terms any more (simpler proofs), and we can make precise to what extent so-called co-Birkhoff results are indeed dualising Birkhoff results (new insights).

Let us remark on how the use of factorisation systems is linked to reflective subcategories. Recall that in abstract categories we consider the notions of quotient and subobject as being relative to a given factorisation system (E, M) where morphisms in E are called E-quotients and morphisms in M are called M-subobjects. The notion of a variety now becomes a

[6]This relation of equations Φ_X and morphisms η_{FX} is discussed in more detail in section 2.3.

subcategory closed under E-quotients, M-subobjects and products. Similarly, we call a reflective subcategory E-reflective iff the reflection morphisms are in E. Now, one can prove under some additional technical assumptions (see corollary 2.2.4) that a subcategory is a variety iff it is an E-reflective subcategory closed under E-quotients. This relationship of E-reflective subcategories and factorisation systems (E, M) allows to prove generalised Birkhoff theorems for categories that have a factorisation system.

Last but not least, the concept of a reflective subcategory enables us to characterise the expressive power of Horn formulas and implications[7] by using almost the same proof as for Birkhoff's variety theorem. The statements corresponding to Birkhoff's theorem are: A class of algebras is definable by implications iff it is closed under subalgebras and products (i.e. iff it is a quasivariety). A class of algebras is definable by Horn formulas iff it is closed under directed colimits[8], subalgebras and products. To see how the case of implications is proved let us go back to our sketch of a proof of Birkhoff's variety theorem. Since quasivarieties are reflective subcategories (see corollary 2.2.4) it suffices to show that reflective subcategories are implicationally definable. This is not difficult once one realised that arbitrary reflection morphisms $\eta_A : A \to \varrho A$ (dropping the assumption $A = FX$) correspond to implications in much the same way as reflection morphism with free domain correspond to equations.[9] Similarly, in the case of Horn formulas, one has to show that reflection morphisms with finitely presentable domain correspond to implications with finitary premise.

2.1.3 M-Coreflective Subcategories: An Example

Having discussed equational definability and reflective subcategories, we give now an example of a coreflective subcategory of Kripke frames. In the previous subsection we dealt mainly with varieties, here we give an example of a co*quasi*variety. We also use the opportunity to recall the notion of a modal rule.

Let us first recall the notion of a M-coreflective subcategory in this setting. Let $\Omega : \mathbf{Set} \to \mathbf{Set}$ be a signature (e.g. $\Omega X = \mathcal{P}P \times \mathcal{P}X$ as in example 1.1.1). Recall from theorem 1.3.10 that $(Epi, StrongMono)$ is a factorisation system for $\mathbf{Set}_\Omega$, strong monos in $\mathbf{Set}_\Omega$ being precisely the injective coalgebra morphisms. Now, $\mathcal{B} \subset \mathcal{C}$ is a $StrongMono$-coreflective subcategory of $\mathbf{Set}_\Omega$ iff for all $A \in \mathbf{Set}_\Omega$ there is $\varrho A \in \mathcal{B}$ and a strong mono $\varepsilon_A : \varrho A \to A$ such that for all $B \in \mathcal{B}$ and all morphisms $f : B \to A$ there is a unique $g : B \to \varrho A$ such that

$$\begin{tikzcd}
A & \varrho A \arrow[l, "\varepsilon_A"'] \\
& B \arrow[ul, "f"] \arrow[u, dotted, "g"']
\end{tikzcd}$$

commutes.[10] The morphisms ε_A are called the coreflection morphisms (and they have to be

[7] An implication is a formula $\bigwedge_{i\in I} t_i = t'_i \;\Rightarrow\; t = t'$ and a Horn formula is an implication with finite index set I.

[8] Directed colimits are often called direct limits in universal algebra.

[9] For more details on equations and implications see section 2.3.

[10] Categorically speaking, the inclusion $\iota : \mathcal{B} \hookrightarrow \mathbf{Set}_\Omega$ has a right adjoint ϱ with counit ε.

strong monos in order to be injective).

Example (Modal Logic) 2.1.1. To illustrate the notions above and their connection to modal logic we give an example. Let $\Omega X = \mathbb{B} \times \mathcal{P}X$, where $\mathbb{B} = \{\mathit{true}, \mathit{false}\}$ is the set of Booleans. That is, every state $x \in X$ is assigned (b, Y), where b is a Boolean and $Y \subset X$. We interpret b as the truth value of a fixed proposition called *start* and Y as the set of successors of x. A modal language for this functor is built from the usual connectives, modal operators and propositional variables from a set P, plus a propositional constant *start*. An Ω-coalgebra $A = (UA, \alpha)$ is a Kripke frame together with a predicate interpreting *start*: $\pi_1 \circ \alpha : UA \to \mathbb{B}$ interprets *start* and $\pi_2 \circ \alpha : UA \to \mathcal{P}UA$ is a Kripke frame (with π_1, π_2 denoting the projections from the product $\mathbb{B} \times \mathcal{P}$ to its components). Satisfaction of modal formulas is defined as follows. Let $\gamma : UA \to \mathcal{P}P$ be a valuation of propositional variables, $x \in UA$ an element of the carrier, and $p \in P$ a propositional variable. Then (boolean cases as usual):

$$\begin{array}{lcl} A, \gamma, x \models \mathit{start} & \Leftrightarrow & \pi_1 \circ \alpha(x) = \mathit{true} \\ A, \gamma, x \models p & \Leftrightarrow & p \in \gamma(x) \\ A, \gamma, x \models \Box\varphi & \Leftrightarrow & \forall y \in \pi_2 \circ \alpha(x) : A, \gamma, y \models \varphi \end{array}$$

The states x satisfying the first clause are called states marked by *start*.

Next, we want to axiomatise a subclass of these Kripke frames by modal rules. A modal rule φ/ψ (where φ, ψ are modal formulas) is interpreted via

$$A \models \varphi/\psi \iff \forall \gamma : UA \to \mathcal{P}P (A, \gamma \models \varphi \;\Rightarrow\; A, \gamma \models \psi)$$

Modal axioms are rules with a true premise. Consider the following rules:

$$\begin{array}{ll} \text{(refl)} & \Box p \to p \\ \text{(trans)} & \Box p \to \Box\Box p \\ \text{(start)} & \mathit{start} \to \Box p \,/\, p \end{array}$$

The first two are the well-known axioms defining reflexivity and transitivity on Kripke frames. The third one is the start rule from Kröger [70]. In the presence of reflexivity and transitivity it expresses that every state has to be reachable from a state marked by *start*.

Call Φ the set of the three rules above and let $\mathcal{B}$ be the class of Kripke frames defined by Φ. We show that $\mathcal{B}$ is a *StrongMono*-coreflective subcategory. Define ϱA as the largest subcoalgebra of A satisfying Φ (that is, to find ϱA, first take the largest subcoalgebra of A that is reflexive and transitive and then cut off all states that are not reachable by a state marked by *start*). $\varepsilon_A : A \to \varrho A$ is the canonical embedding and it it is a strong mono since it is injective. Recalling the definition of a coreflective subcategory, it remains to show that for all $B \in \mathbf{Set}_\Omega$ satisfying Φ it holds that for all $f : B \to A$ there is a unique $g : B \to \varrho A$ such that

$$\begin{tikzcd} A & \varrho A \arrow[l, "\varepsilon_A"'] \\ & B \arrow[ul, "f"] \arrow[u, dotted, "g"] \end{tikzcd}$$

commutes. Consider an $(Epi, Strong\,Mono)$-factorisation $B \xrightarrow{e} \bar{B} \xrightarrow{m} A$ of f. Since rules are invariant under taking images (see corollary 2.4.6) it follows that $\bar{B} \models \Phi$. Moreover $m : \bar{B} \to A$ is a subcoalgebra of A and since ϱA is the largest subcoalgebra of A satisfying Φ, m factors through ε_A as $m = \varepsilon_A \circ g'$ for some g'. Now, $g = g' \circ e$ is the required morphism and g is uniquely determined since ε_A is mono.

Finally, let us note that $\mathcal{B}$ is closed under images and disjoint unions (coproducts) but not under subcoalgebras. Hence $\mathcal{B}$ is an example of a coquasivariety that is not a covariety.

2.2 M-Coreflective Subcategories and Covarieties

We present some results on coreflective subcategories. Of importance are theorem 2.2.2, corollary 2.2.4, and definition 2.2.8.

Coreflective subcategories relate to the more familiar notions of (co)(quasi)variety as follows. A *quasivariety* (of algebras over sets) is a subcategory closed under subalgebras and products. A *variety* is a quasivariety closed under images of morphisms. Dually, a *coquasivariety* (of coalgebras over sets) is a subcategory closed under images of morphisms and coproducts. A *covariety* is a coquasivariety closed under subcoalgebras. Now, (co)quasivarieties can be characterised as (co)reflective subcategories (see corollary 2.2.4). Moreover, M-(co)reflective subcategories are the appropriate generalisation of (co)quasivarieties in cases where we do not have (or do not want to mention) a base category of sets.

This subsection follows (dualises) the presentation in Adámek, Herrlich, Strecker [4], chapter 16. In particular we obtain the characterisation coquasivarieties as M-coreflective subcategories as a corollary of a more general based on the notion of a factorisation structure for sinks. The main reason why factorisation structures for sinks appear here is that they naturally capture the concept of a union of subcoalgebras: In the following, the category $\mathcal{C}$ should be thought of as a category of coalgebras; factoring a sink $(s_i : A_i \to B)$ as $A_i \xrightarrow{e_i} A \xrightarrow{m} B$, A should be understood as the union of the images of the s_i.

The definitions of factorisation system and factorisation structure for sinks are given in the appendix.

Definition 2.2.1 (M-coreflective subcategories). Let M be a class of morphisms of $\mathcal{C}$. A subcategory $i : \mathcal{B} \hookrightarrow \mathcal{C}$ is called *M-coreflective* iff it is full and replete and i has a right adjoint r with counit $\varepsilon^{\mathcal{B}}$ such that $\varepsilon^{\mathcal{B}}_A : irA \to A$ is in M for all $A \in \mathcal{C}$. The $\varepsilon^{\mathcal{B}}_A$ are called *coreflection morphisms* and r is called the *coreflection*.

The following theorem and proposition are theorem 16.8 in [4]. Because it is worth to see the proof in its dualised version we sketch it below.

Theorem 2.2.2. *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category and $\mathcal{B}$ be a full subcategory of $\mathcal{C}$. Then $\mathcal{B}$ is M-coreflective iff $\mathcal{B}$ is closed under $\mathcal{E}$-sinks.*

Proof. M-coreflective subcategories $\mathcal{B}$ are closed under $\mathcal{E}$-sinks: Let $(s_i : B_i \to A)$ be a sink in $\mathcal{E}$ with all B_i in $\mathcal{B}$. Let $m : A' \to A$ be a M-coreflection morphism. By coreflectiveness (s_i) factors through m. Hence m is iso by proposition A.5.3(6). Since $\mathcal{B}$ is replete, $A \in \mathcal{B}$.

Closure of $\mathcal{B}$ under $\mathcal{E}$-sinks implies M-coreflectiveness: For $A \in \mathcal{C}$ let (s_i) be the sink consisting of all morphisms with codomain A and domain in $\mathcal{B}$. Let $B_i \xrightarrow{e_i} A' \xrightarrow{m} A$ be a $(\mathcal{E}, M)$-factorisation of (s_i). Then A' is in $\mathcal{B}$ and m is a M-coreflection morphism for A. $\square$

Closure under $\mathcal{E}$-sinks may not be a familiar closure condition. But one can show that under reasonable assumptions it is equivalent to closure under small coproducts and quotients:

Proposition 2.2.3. *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category has small coproducts and is M-wellpowered. Then closure under $\mathcal{E}$-sinks is the same as closure under small coproducts and $\mathcal{E}$-quotients.*

Proof. For one direction note that coproducts and $\mathcal{E}$-quotients are special $\mathcal{E}$-sinks. (For coproducts this follows from the fact that the colimiting cocone is an extremal-epi sink and hence in $\mathcal{E}$ by proposition A.5.3(4).

For the converse let $(A, (s_i)_{i\in I})$ be an $\mathcal{E}$-sink. We show that A is a quotient of the coproduct of the objects in the domain of the sink. Let $B_i \xrightarrow{e_i} A_i \xrightarrow{m_i} A$ be $(\mathcal{E}, M)$-factorisations of each s_i. By wellpoweredness there are $J \subset I$, $e'_i : B_i \to \Sigma_{j\in J} A_j$, $g : \Sigma_{j\in J} A_j \to A$ such that $(s_i) = g \circ (e'_i)$. Our claim now follows from considering a $(\mathcal{E}, M)$-factorisation $\Sigma_{j\in J} A_j \xrightarrow{e} A' \xrightarrow{m} A$ of g. □

The following corollary characterises coquasivarieties as coreflective subcategories. For the reader of the axiomatic approach in chapter 1.4 we note that the assumptions of the following corollary correspond to axioms 1 and 4.

Corollary 2.2.4 (characterisation of M-coreflective subcategories). *Let (E, M) be a factorisation system for a category $\mathcal{C}$ having small coproducts and being M-wellpowered. Moreover, suppose that morphisms in M are mono. Then for a full subcategory $\mathcal{B} \hookrightarrow \mathcal{C}$ it holds: $\mathcal{B}$ is closed under E-quotients and small coproducts iff it is a M-coreflective subcategory of $\mathcal{C}$.*

Proof. By theorem 2.2.2 and propositions 2.2.3, A.5.5. □

Remark. Dually, categories of algebras closed under subalgebras and products (quasivarieties) are characterised as reflective subcategories.

As another corollary we can show that every full subcategory has a M-coreflective hull (following 16.20–16.23 in [4]).

Proposition 2.2.5. *The intersection of any collection of M-coreflective subcategories is M-coreflective.*

Proof. Let there be a collection of M-coreflective subcategories. Each of them is closed under $\mathcal{E}$-sinks, hence the intersection is closed under $\mathcal{E}$-sinks, hence the intersection is M-coreflective. □

Definition 2.2.6 (M-coreflective hull, $\mathcal{E}(-)$). Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category and $\mathcal{B}$ a full subcategory. The M-coreflective hull of $\mathcal{B}$ is denoted by $\mathcal{E}(\mathcal{B})$. By theorem 2.2.2 $\mathcal{E}(\mathcal{B})$ is the closure of $\mathcal{B}$ under $\mathcal{E}$-sinks.

Proposition 2.2.7. *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category and $\mathcal{B}$ a full subcategory. Then $\mathcal{E}(\mathcal{B})$ is the closure of $\mathcal{B}$ under $\mathcal{E}$-sinks with* domain *in $\mathcal{B}$.*

Proof. The proof relies on the fact that $\mathcal{E}$-sinks are closed under composition, see proposition A.5.3. □

Theorem 2.2.2 will allow us to characterise *modal rule* definable classes of coalgebras as those closed under small coproducts and quotients. A nice feature of the proof of this characterisation result is that the proof only refers to properties of the category of coalgebras without mentioning the base category or the forgetful functor. In order to characterise, however, classes of coalgebras definable by *modal formulas* we have to take into account that formulas are rules with cofree codomain, a notion depending on the forgetful functor (cofree coalgebras are given by the right adjoint to the forgetful functor). Interestingly—and dually to the case of algebras—it is possible to replace the notion of a cofree coalgebra by that of an injective coalgebra, a notion which is internal to the category of coalgebras.

Definition 2.2.8 ((enough) injective objects). Let $\mathcal{C}$ be a category and M a class of morphisms of $\mathcal{C}$. An object A in a category $\mathcal{C}$ is called M-injective iff for all $m : A_1 \to A_2 \in M$ and all $f : A_1 \to A$ there is $g : A_2 \to A$ such that $g \circ m = f$:

$$\begin{tikzcd} A_1 \arrow[r, "m"] \arrow[dr, "f"'] & A_2 \arrow[d, dotted, "g"] \\ & A \end{tikzcd}$$

$\mathcal{C}$ is said to have *enough injectives* iff every object is a M-subobject of an injective object.

To explain this definition let us assume that morphisms in M represent subcoalgebras. Then 'A injective' means, that, for all $m : A_1 \to A_2 \in M$, every morphism $f : A_1 \to A$ can be extended to all of A_2. Intuitively this means that A is able to represent any behaviour that may occur in A_2, a condition that resembles cofreeness. To be more precise, we show

Proposition 2.2.9. *The cofree coalgebras in* $\mathbf{Set}_\Omega$ *are StrongMono-injective.*

Proof. Let FC be a cofree coalgebra in $\mathbf{Set}_\Omega$ with colouring (counit) $\varepsilon_C : FC \to C$, let $m : A_1 \to A_2$ be a strong mono and $f : A_1 \to FC$ a morphism. The fact that m is strong mono in $\mathbf{Set}_\Omega$ implies that m is mono in $\mathbf{Set}$ (see corollary 1.3.10). It follows that there is $g : A_2 \to C$ such that $g \circ m = \epsilon_C \circ f$. By cofreeness, there is $g^\# : A_2 \to FC$ s.t. $g^\# \circ m = f$. □

The reason for the existence of g in the proof above is due to the following: In $\mathbf{Set}$ all objects C are injective w.r.t. all monos m. We can therefore generalise the proposition.

Proposition 2.2.10. *Let* $U : \mathcal{C} \to \mathcal{X}$ *be a functor with right adjoint* F *and let* M *be class of morphisms in* $\mathcal{C}$. *Then a cofree coalgebra* FC, $C \in \mathcal{X}$, *is* M*-injective if* C *is* UM*-injective.*

Remark. Under the assumptions of the proposition it holds:

1. If the unit η_A of the adjunction $U \dashv F$ is in M, then A is M-injective implies that A is a retract of FUA.

2. If C is UM-injective, then A is a retract of a cofree coalgebra FC implies that A is M-injective.

In particular, in $\mathbf{Set}_\Omega$ the *StrongMono*-injective objects are precisely the retracts of the cofree coalgebras. That in $\mathbf{Set}_\Omega$ the $\eta_A : A \to FUA$ are injective follows from $\varepsilon_{UA} \circ U\eta_A = \mathrm{id}_{UA}$ which holds for any adjunction and implies that $U\eta_A$ is split mono.

2.3 Equations and Implications as Quotients

We briefly review some of the basic ideas of Banaschewski and Herrlich [11]. For details see [11] and [4], or dualise the material in sections 2.2, 2.4.2, 2.5.1, 2.5.2.

Let $\mathbf{Set}^\Omega$ be the category of algebras for a signature Ω, let $U : \mathbf{Set}^\Omega \to \mathbf{Set}$ be the forgetful functor, $F \dashv U$ the free construction and Φ a set of equations over variables X. To see that the equations Φ can be categorically characterised by a surjective homomorphism (regular epi) e consider the following diagram

$$\begin{tikzcd} FX \arrow[r, "e"] \arrow[dr, "\alpha"'] & FX/\Phi \arrow[d, dotted, "\beta"] \\ & A \end{tikzcd}$$

where FX is the free algebra over variables X, FX/Φ is the quotient of FX w.r.t. to the smallest congruence generated by Φ, and $e : FX \to FX/\Phi$ is the canonical projection. Now, it is not difficult to see that an algebra A satisfies Φ if and only if for all morphisms $\alpha : FX \to A$[11] there is a morphism $\beta : FX/\Phi \to A$ such that the diagram commutes. In categorical terms: A satisfies the equations Φ iff A is **injective** w.r.t. the regular epi[12] e.

That is, sets of equations can be considered as regular epis with a free algebra as domain. And conversely, (the kernels) of regular epis give rise to sets of equations.

The case of implications $\bigwedge_{i\in I}(t_i = s_i \;\Rightarrow\; t = s)$ is even simpler. They just correspond to regular epis (dropping the assumption that the domain is free):

$$\begin{tikzcd} FX \arrow[r, "e_B"] \arrow[dr, "\gamma"'] & B \arrow[r, "e"] \arrow[d, "\alpha"] & C \arrow[dl, dotted, "\beta"] \\ & A & \end{tikzcd}$$

For a surjective $e : B \to C$ we can find a free algebra FX (just choose X large enough) and sets of equations Φ, Ψ such that $B \simeq FX/\Phi$ and $C \simeq FX/\Psi$. $e_B : FX \to B$ denotes the quotient onto B. Now, it is not difficult to show[13] that A is injective w.r.t. e iff for all valuations $v : X \to UA$ it holds

$$A, v \models \Phi \;\Rightarrow\; A, v \models \Psi.$$

[11] Recall that morphisms $\alpha : FX \to A$ correspond bijectively to valuations of variables $v : X \to UA$. This is formally due to F being a left adjoint to the forgetful functor.

[12] Recall that in most common categories of algebras the *regular epis* are precisely the surjective homomorphisms.

[13] Let $v : X \to UA$ and $\gamma : FX \to A$ be arrows related by the adjunction $F \dashv U$. "$\Rightarrow$": $A, v \models \Phi$ implies that γ factors as $\alpha \circ e_B$ for some α. By injectivity of A w.r.t. e, α factors through e. Hence γ factors through $e \circ e_B$, i.e. $A, v \models \Psi$. "$\Leftarrow$": Given α, let $\gamma = \alpha \circ e_B$. If $A, v \models \Phi$, then $A, v \models \Psi$, i.e. γ factors as $\beta \circ e \circ e_B$ for some β. Since e_B is epi, it follows $\alpha = \beta \circ e$, i.e. A is injective w.r.t. e.

Conversely, for every surjective homomorphisms $e : B \to C$ we can find appropriate sets of equations: choose for Φ the kernel of e_B and for Ψ the kernel of $e \circ e_B$.

Finally, one has to convince oneself (easy excercise in logic) that for all sets of equations Φ, Ψ we can find a set of implications I and that for all sets of implications I we can find sets of equations Φ, Ψ such that for all valuations $v : X \to UA$

$$A, v \models \Phi \;\Rightarrow\; A, v \models \Psi \quad \text{iff} \quad A, v \models I$$

Similarly, implications with finitary premise (Horn formulas) can be considered as regular epis with finitely presentable domain.

More generally, Banaschewski and Herrlich showed that their characterisation results for implicational logics do not depend on the epis being regular but only on the existence of an factorisation structure for sources. Such a factorisation structure exists in particular if the category has regular epi/mono factorisations, has products and is cowellpowered (see [11, 4] or dualise proposition A.5.5).

2.4 Modal Logic is Dual to Equational Logic

In the previous section we have seen that—semantically—equations and implications are quotients and satisfaction is injectivity. This section shows that, dually, modal formulas and rules are subobjects and satisfaction is projectivity.

The first subsection shows that the 'formulas as subcoalgebra' interpretation agrees with the traditional Kripke semantics of modal logic.

The second subsection introduces an abstract notion of modal logic where formulas and rules actually *are* subobjects. This point of view proves to be fruitful for semantical investigations of modal logic. In section 2.7, when we go back to modal logics given by some language we reintroduce the distinction between formulas and rules (given by the language) and their semantics (given by morphisms).

The third subsection shows that for this abstract notion of modal logic the standard preservation results still hold.

2.4.1 Modal Formulas and Rules as Subcoalgebras

It is shown how formulas and rules of modal logic can be considered to be certain monomorphism (subcoalgebras) in the corresponding category of coalgebras.

Let Ω be a functor on $\mathbf{Set}$ such that $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint F with counit ε, that is, for all sets C there is a cofree coalgebra FC with colouring $\varepsilon_C : UFC \to C$. In order to explain the relationship of formulas and subcoalgebras of cofree coalgebras, consider the diagrams below:

$$\begin{tikzcd} FC \arrow[d, dashed, "\epsilon_C"'] & \\ C & A \arrow[l, dashed, "\gamma"] \arrow[ul, "\gamma^\#"'] \end{tikzcd} \qquad \begin{tikzcd} FC \arrow[d, dashed, "\epsilon_C"'] & FC|\varphi \arrow[l, "m_\varphi"'] \\ C & A \arrow[l, dashed, "\gamma"] \arrow[ul, "g"'] \arrow[u, "h"'] \end{tikzcd}$$

(The dashed arrows denote arrows in $\mathbf{Set}$, the plain arrows denote coalgebra morphisms; commutativity of the diagrams is to be understood as commutativity in $\mathbf{Set}$.)

The first diagram is intended to recall that an Ω-coalgebra FC (and an arrow $\varepsilon_C : UFC \to C$) is called *cofree over* C iff for all Ω-coalgebras A and all arrows $\gamma : UA \to C$ there is a unique morphism $\gamma^\# : A \to FC$ such that the diagram commutes. As example 1.1.4 shows, we can think of FC and A as Kripke frames, of (FC, ε_C) and (A, γ) as Kripke models (with the colourings ε_C and γ being valuations of propositional variables) and of coalgebra morphisms as p-morphisms.

The second diagram illustrates the interpretation of formulas as subcoalgebras of cofree coalgebras. Suppose that a modal formula φ and an Ω-coalgebra (Kripke frame) A are given. According to the definition of satisfaction in modal logic

$$A \models \varphi \ \text{ iff } \ A, \gamma, x \models \varphi \text{ for all } x \in UA \text{ and all valuations } \gamma : UA \to C$$

Let $FC|\varphi$ be the largest $C \times \Omega$-subcoalgebra of (FC, ϵ_C) whose elements satisfy φ and let m_φ be the natural embedding. Then the above condition for $A \models \varphi$ can equivalently be expressed[14] by saying that for all $\gamma : UA \to C$ there is $h : A \to FC|\varphi$ such that $\epsilon_C \circ m_\varphi \circ h = \gamma$, or equivalently,

$$A \models \varphi \text{ iff for all } g : A \to FC \text{ there is } h : A \to FC|\varphi \text{ such that } m_\varphi \circ h = g.$$

That is, categorically speaking,

$$A \models \varphi \text{ iff } A \text{ is } \textbf{projective} \text{ w.r.t. to } m_\varphi.$$

Conversely, every subcoalgebra $m : B \to FC$ corresponds to a modal formula if the modal logic under consideration is expressive enough to define the the carrier of B as a subset of the cofree coalgebra.

We have seen that we can generalise modal formulas to subcoalgebras with cofree codomain and modal satisfaction to projectivity. To see how modal rules φ/ψ correspond to just subcoalgebras recall that $A \models \varphi/\psi$ iff

$$\forall \gamma : UA \to C\, .((\forall x \in UA\, .\, A, \gamma, x \models \varphi) \;\Rightarrow\; (\forall x \in UA\, .\, A, \gamma, x \models \psi)).$$

Now consider the following diagram

$$\begin{tikzcd}
FC & FC|\varphi \arrow[l, "m_\varphi"'] & FC|(\varphi \wedge \psi) \arrow[l, "m"'] \\
 & & A \arrow[ul, "g"] \arrow[u, "h"']
\end{tikzcd}$$

where $FC|\varphi$ and $FC|(\varphi\wedge\psi)$ are the subcoalgebras defined by φ and $\varphi\wedge\psi$ and m_φ, m are the corresponding embeddings. It is not difficult to see that this can equivalently be expressed by saying that for all $g : A \to FC|\varphi$ there is $h : A \to FC|(\varphi \wedge \psi)$ such that $m \circ h = g$. That is, again, A satisfies φ/ψ iff A *is projective w.r.t.* m.

Conversely, in the case that all subcoalgebras of cofree coalgebras are modally definable, we can find an appropriate modal rule for every subcoalgebra m.

2.4.2 Modal Logic for $(\mathcal{E}, M)$-Categories

In this section we define the notion of a modal logic for categories $\mathcal{C}$ having a factorisation structure for sinks $(\mathcal{E}, M)$ (see definition A.5.1). The category $\mathcal{C}$ should be thought of as a category of coalgebras. Factoring a sink $(s_i : A_i \to B)$ as $A_i \xrightarrow{e_i} A \xrightarrow{m} B$, A (or $m : A \to B$) should be understood as the union of the images of the s_i. We abstract from the base category, from signatures, and forgetful functors. The notion of a cofree coalgebra FC which relies on the right adjoint F to the forgetful functor is replaced by the notion of an injective object (definition 2.2.8).

[14] Recall that coalgebra morphisms are bisimulations and that modal formulas are invariant under bisimulations.

That the notion of a modal logic for an $(\mathcal{E}, M)$-category really encompasses the standard notion of a modal logic follows from the considerations in section 2.4.1 and the fact that epi sinks and injective morphisms are a factorisation structure for sinks for categories of coalgebras over set (see theorem 1.3.10 and corollary 1.3.15). Most importantly,

this notion of a modal logic dualises
the notion of equational logic in Banaschewski and Herrlich [11].

Note that in the following definition and up to section 2.7 we deliberately do not distinguish between syntax and semantics, that is, formulas and rules *are* morphisms in a category.

Definition 2.4.1 (modal logic for $(\mathcal{E}, M)$-categories).
Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. A *modal rule* (or rule for short) is a morphism in M. A *modal formula* (or formula) is a morphism in M that has an injective codomain. For $A \in \mathcal{C}$, $m : B \to B' \in M$ and a morphism $f : A \to B'$ (sometimes called a valuation) we define: $A, f \models m$ iff there is $g : A \to B$ such that $m \circ g = f$:

$$\begin{tikzcd} B' & B \arrow[l, "m"'] \\ & A \arrow[ul, "f"] \arrow[u, dotted, "g"'] \end{tikzcd}$$

We define $A \models m$ iff $A, f \models m$ for all f, that is, iff A is projective w.r.t. m.

Remark. The definition above calls every morphism with M-injective codomain a formula. Later, from section 2.7 on, we will restrict our attention to those M-injective codomains which are moreover cofree. All the following results hold also for this setting.

Definition 2.4.2 (Mod, Th, Ru). Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. For a class $\mathcal{B}$ of objects we denote by $\mathsf{Ru}(\mathcal{B})$ and $\mathsf{Th}(\mathcal{B})$ the class of rules and formulas, respectively, satisfied by all objects in $\mathcal{B}$. For $\Phi \subset M$, $\mathsf{Mod}(\Phi)$ denotes the class of objects (also called models henceforth) of $\mathcal{C}$ satisfying every rule in Φ.

It is sometimes convenient to consider Mod, Ru and Th as operators on categories: we indentify a subclass of objects of $\mathcal{C}$ with the corresponding full subcategory and a subclass of M with the category generated by it.

Definition 2.4.3 (modally definable subcategories). We call a full subcategory $\mathcal{B}$ of $\mathcal{C}$ *modal rule definable* (or just rule definable) iff $\mathcal{B} = \mathsf{Mod}(\Phi)$ for some $\Phi \subset M$. We call $\mathcal{B}$ *modally definable* iff $\mathcal{B} = \mathsf{Mod}(\Phi)$ for some $\Phi \subset M$ consisting only of formulas.

A first example of a rule definable class are the models of all rules: the empty sinks in $\mathcal{E}$.

Example 2.4.4. Let $\mathcal{E}_{\{\}} = \{A \in \mathcal{C} : (A, \{\}) \in \mathcal{E}\}$ be the class of empty sinks in $\mathcal{E}$. Then $\mathcal{E}_{\{\}} = \mathsf{Mod}(M)$.

Proof. $\mathcal{E}_{\{\}} \models M$ is immediate by (3) of remark A.5. For the converse, let $A \models M$ and consider the empty sink $(A, \{\})$. By (2) of remark A.5 there is $m : A \to A' \in M$ with $A' \in \mathcal{C}_{\mathcal{E}}$. Because of $A \models m$, id_A factors as $m \circ g$ for some g, that is, m is split epi. Since m is mono by proposition A.5.3(1), m is iso. Hence $A \in \mathcal{E}_{\{\}}$ by (1) of definition A.5.1. □

Note that, by (2) of remark A.5, $\mathsf{Mod}(M)$ is never empty. This seems to be counterintuitive at first sight: The class of all modal rules (or formulas) is generally inconsistent and has, therefore, an empty class of models. The reason for this mismatch is that, modally speaking, we did not exclude the empty Kripke frame as a possible model. It follows from the remark after proposition A.5.5 that instantiating our framework with usual Kripke frames yields a class $\mathcal{E}_{\{\}}$ that consists only of the empty Kripke frame.

2.4.3 Preservation Results

Here we investigate which constructions on models preserve satisfaction of formulas. The results parallel the ones known from modal logic where formulas are preserved under images, disjoint unions, and substructures, see van Benthem [122], chapter 2.

We first show preservation results for rules (which also holds for formulas).

Proposition 2.4.5 (rules are preserved under $\mathcal{E}$-sinks).
Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. Then every rule definable subcategory is closed under $\mathcal{E}$-sinks.

Proof. Let $\mathcal{B} = \mathsf{Mod}(\Phi)$ and let $(e_i : A_i \to A) \in \mathcal{E}$ with A_i in $\mathcal{B}$, $m : B \to B' \in \Phi$ and $f : A \to B'$. We have to show that f factors through m. Because of $A_i \models m$ there is a sink $(s_i : A_i \to B)$ such that $m \circ (s_i) = f \circ (e_i)$. Now, by the unique diagonalisation property there is $g : A \to B$ s.t. $m \circ g = f$, that is, A is projective w.r.t. m. Hence $A \in \mathcal{B}$. □

It follows:

Corollary 2.4.6 (rules are preserved under $\mathcal{E}$-quotients). *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. Then every rule definable subcategory is closed under $\mathcal{E}$-quotients.*

Corollary 2.4.7 (rules are preserved under colimits).
Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. Then every rule definable subcategory is closed under colimits.

Proof. A direct proof is easy but one can also argue as follows. Every colimiting cocone is an *ExtrEpi*-sink and every *ExtrEpi*-sink is in $\mathcal{E}$. □

As to be expected formulas are also preserved under taking substructures.

Proposition 2.4.8 (formulas are preserved under M-subobjects).
Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. Then every formula-definable subcategory is closed under M-subobjects.

Proof. Let $\mathcal{B} = \mathsf{Mod}(\Phi)$ for some class Φ of formulas. Let $B \in \mathcal{B}$, $m : A \to B \in M$, and $\varphi \in \Phi$. Consider the following diagram.

$$\begin{tikzcd} A_0 \arrow[r, "\varphi"] & A_1 \\ A \arrow[u, dotted, "g"] \arrow[ur, dotted, "f"] \arrow[r, "m"'] & B \arrow[u, dotted, "h"'] \end{tikzcd}$$

We have to show $A \models \varphi$, i.e., $\forall f : A \to A_1 . \exists g : A \to A_0 . \varphi \circ g = f$. Since φ is a formula, A_1 is M-injective, hence there is h s.t. $h \circ m = f$. Because of $B \models \varphi$ there is $g' : B \to A_0$ s.t. $\varphi \circ g' = h$. Define $g = g' \circ m$. It follows $\varphi \circ g = f$, hence $A \models \varphi$. $\square$

Example (Modal Logic) 2.4.9. It follows in particular that modal rules (and hence formulas) of standard modal logic are preserved under images and disjoint unions of Kripke frames and that formulas are preserved under taking subframes.

2.5 Co-Birkhoff-Theorems

Birkhoff's variety theorem states that the equationally definable classes of algebras are precisely the varieties (i.e. classes closed under homomorphic images, subalgebras and products). It has been generalised in two different directions. First, the proof was generalised to characterise implicationally definable classes. Second, the notion of an algebra was generalised, for example allowing for operations with infinitary many arguments (like joins in complete semilattices) or to algebras not over sets but over other categories. A common framework for these developments has been given in Banaschewski and Herrlich [11] which offers a general and axiomatic description of categories that admit Birkhoff theorems.

In the following we will understand under a *Birkhoff theorem* any theorem characterising the expressive power of some logic for algebras which stands in the tradition sketched above. Dually, a *co-Birkhoff theorem* is a theorem characterising the expressive power of some logic for coalgebras that is obtained (essentially) by dualising a Birkhoff theorem.

Subsections 2.5.1 and 2.5.2 dualise theorems of Banaschewski and Herrlich [11]. Subsection 2.5.3 uses the notion of a bounded category (see definition 1.5.3) yielding a proof which does not dualise the usual proof of a (bounded) variety theorem (compare e.g. Adámek and Rosický [5], chapter 3).

2.5.1 Rule definable subcategories

We show that the rule definable subcategories are the M-coreflective ones. The idea of the proof is simple: Show that every M-coreflective subcategory is defined by its coreflection morphisms.

Result and proof dualise proposition 1 in Banaschewski and Herrlich [11] (also theorem 16.14 in Adámek, Herrlich and Strecker [4]) characterising implicationally definable subcategories.

Theorem 2.5.1 (characterisation of rule definable subcategories). *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category and $\mathcal{B}$ a full subcategory $\mathcal{C}$. Then the following are equivalent.*

1. *$\mathcal{B}$ is modal rule definable*
2. *$\mathcal{B}$ is closed under $\mathcal{E}$-sinks*
3. *$\mathcal{B}$ is M-coreflective.*

Proof. By proposition 2.4.5, every rule definable category is closed under $\mathcal{E}$-sinks. By theorem 2.2.2, $\mathcal{B}$ is also M-coreflective. To show that (3) implies (1) let $\mathsf{Ru}'(\mathcal{B})$ be the subclass of M consisting of all coreflection morphisms for $\mathcal{B}$. We show that $\mathsf{Ru}'(\mathcal{B})$ defines $\mathcal{B}$, i.e., that $\mathsf{ModRu}'(\mathcal{B}) = \mathcal{B}$.

"$\subset$": Let $A \in \mathcal{C}$ with $A \models \mathsf{Ru}'(\mathcal{B})$ and consider the coreflection $m : A' \to A \in M$. Because of $m \in \mathsf{Ru}'(\mathcal{B})$, id_A factors through m, hence m is split epi and, since by proposition A.5.3 m is mono, it is iso. Hence $A \in \mathcal{B}$.

"$\supset$": immediate from the universal property of the coreflection morphisms in $\mathsf{Ru}'(\mathcal{B})$. □

As a corollary we can characterise the closure operator ModRu as closure under $\mathcal{E}$-sinks (see definition 2.2.6).

Corollary 2.5.2 (characterisation of ModRu). *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category and $\mathcal{B}$ a full subcategory of $\mathcal{C}$. Then $\mathsf{ModRu}(\mathcal{B}) = \mathcal{E}(\mathcal{B})$.*

Proof. "$\supset$" is proposition 2.4.5. "$\subset$": This is a standard argument using properties[15] of the operators Mod and Ru. Recall the definition of Ru' in the proof of theorem 2.5.1. Using $\mathcal{B} \subset \mathcal{E}(\mathcal{B})$, $\mathsf{Ru}'\mathcal{E}(\mathcal{B}) \subset \mathsf{Ru}\mathcal{E}(\mathcal{B})$, and theorem 2.5.1, we conlcude $\mathsf{ModRu}(\mathcal{B}) \subset \mathsf{ModRu}\mathcal{E}(\mathcal{B}) \subset \mathsf{ModRu}'\mathcal{E}(\mathcal{B}) = \mathcal{E}(\mathcal{B})$. □

Corollary 2.5.3 (co-quasi-variety theorem). *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category that has small coproducts and is M-wellpowered. Then a full subcategory $\mathcal{B}$ is rule definable iff it is closed under $\mathcal{E}$-quotients and small coproducts. Moreover, $\mathsf{ModRu}(\mathcal{B}) = \mathrm{H}\Sigma(\mathcal{B})$.*

Proof. By theorems 2.5.1, 2.5.2 and 2.2.2. The last statement follows because $\mathrm{H}\Sigma(\mathcal{B}) = \mathcal{E}(\mathcal{B})$. □

2.5.2 Modally definable subcategories

We show that the modally definable subcategories are those M-coreflective ones which are closed under M-subobjects. The idea of the proof is similar to the previous section: Show that these subcategories are defined by its coreflection morphisms with injective codomain.

Result and proof dualise proposition 3 in Banaschewski and Herrlich [11] (also theorem 16.17 in Adámek, Herrlich, Strecker [4]).

Theorem 2.5.4 (characterisation of modally definable subcategories). *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category with enough injectives and $\mathcal{B}$ be a full subcategory $\mathcal{C}$. Then the following are equivalent.*

1. *$\mathcal{B}$ is modally definable.*
2. *$\mathcal{B}$ is closed under $\mathcal{E}$-sinks and M-subobjects.*
3. *$\mathcal{B}$ is M-coreflective and closed under M-subobjects.*

Proof. (1) implies (2) follows from propositions 2.4.5 and 2.4.8, (2) implies (3) from theorem 2.2.2. To show (3) implies (1) let $\mathsf{Th}'(\mathcal{B})$ be the subclass of M consisting of all coreflection morphisms $A' \to A$ with A injective. We show that $\mathsf{Th}'(\mathcal{B})$ defines $\mathcal{B}$, i.e., $\mathsf{ModTh}'(\mathcal{B}) = \mathcal{B}$.

"$\subset$": Let $B \in \mathcal{C}$ with $B \models \mathsf{Th}'(\mathcal{B})$. Consider an injective A, a $n : B \to A \in M$ and the coreflection $m : A' \to A \in M$. Because of $m \in \mathsf{Th}'(\mathcal{B})$, n factors through m as $m \circ f = n$. By proposition A.5.3(7), $f \in M$, hence $B \in \mathcal{B}$.

"$\supset$": immediate from the universal property of the coreflection morphisms in $\mathsf{Th}'(\mathcal{B})$. □

[15] These properties are due to the fact that Mod and Ru form a Galois connection, i.e., $\mathsf{Mod}\Phi \subset \mathcal{B} \Leftrightarrow \Phi \supset \mathsf{Ru}(\mathcal{B})$ for classes of rules Φ. Also, $\mathsf{Mod}\Psi \subset \mathcal{B} \Leftrightarrow \Psi \supset \mathsf{Th}(\mathcal{B})$ for classes of formulas Ψ. In particular, ModRu and ModTh are closure operators. For more on Galois connections see e.g. Wechler [124].

Corollary 2.5.5 (co-variety-theorem). *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category with enough injectives that has small coproducts and is M-wellpowered. Then a full subcategory $\mathcal{B}$ of $\mathcal{C}$ is modally definable iff it is closed under M-subobjects, $\mathcal{E}$-quotients and small coproducts.*

As before we can phrase this result as characterising the closure operator ModTh:

Corollary 2.5.6 (co-variety-theorem). *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category with enough injectives that has small coproducts and is M-wellpowered and let $\mathcal{B}$ be a full subcategory of $\mathcal{C}$. Then $\mathsf{ModTh}(\mathcal{B})$ is the closure of $\mathcal{B}$ under M-subobjects, $\mathcal{E}$-quotients and small coproducts.*

2.5.3 Co-Variety-Theorem for Bounded Categories

If the category $\mathcal{C}$ is bounded (definition 1.5.3) by an injective object A one can show that modally definable subcategories are defined by a single formula, namely the coreflection morphism of A.

Theorem 2.5.7. *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category bounded by an injective object A. Let $\mathcal{B}$ be full subcategory of $\mathcal{C}$. Then the following are equivalent.*

1. *$\mathcal{B}$ is modally definable.*
2. *$\mathcal{B}$ is closed under $\mathcal{E}$-sinks and M-subobjects.*
3. *$\mathcal{B}$ is M-coreflective and closed under M-subobjects.*
4. *$\mathcal{B}$ is definable by a single formula $m \in M$.*

Proof. Using theorem 2.5.4 it remains to show that (3) implies (4). Let $m : A' \to A$ be the coreflection morphism of the bounding object A. We show that m defines $\mathcal{B}$, i.e., $\mathsf{Mod}\{m\} = \mathcal{B}$. "$\supset$" is again by the universal property of the coreflection. To show '$\subset$", let $B \in \mathcal{C}$ with $B \models m$. B is the union of some B_i (see definition 1.5.2). By definition of $\mathcal{C}$ being bounded there are $m_i : B_i \to A \in M$. Since m is a formula and formulas are preserved under M-subobjects, it follows $B_i \models m$. Hence the morphisms m_i factor through m which in turn implies that the B_i are M-subobjects of A' and therefore $B_i \in \mathcal{B}$. Since the union of the B_i is a sink in $\mathcal{E}$ it follows $B \in \mathcal{B}$. □

Corollary 2.5.8 (covariety theorem). *Let $\mathcal{C}$ be a bounded $(\mathcal{E}, M)$-category with enough injectives that has small coproducts and is M-wellpowered. Then a full subcategory $\mathcal{B}$ of $\mathcal{C}$ is a covariety iff it is modally definable by a single formula (= morphism in M).*

Proof. A bounded $(\mathcal{E}, M)$-category with enough injectives that has small coproducts is bounded by an injective object. Now apply the theorem above. □

Corollary 2.5.9 (Rutten [109]). *Let Ω : **Set** $\to$ **Set** be a bounded functor. Then a full subcategory $\mathcal{B}$ of $\mathbf{Set}_\Omega$ is a covariety iff it is definable by a subcoalgebra of a cofree coalgebra.*

Proof. $\mathbf{Set}_\Omega$ is an $(\mathcal{E}, M)$-category (theorem 1.3.10) and morphisms in M are subcoalgebras. Moreover, $\mathbf{Set}_\Omega$ has enough injectives and we can choose as these injectives the cofree coalgebras. Also, bounded functors Ω give rise to bounded categories $\mathbf{Set}_\Omega$. Now apply "2 $\Leftrightarrow$ 4" of the theorem. It remains to show that the covarieties are precisely the subcategories closed under M-subobjects and $\mathcal{E}$-sinks. This follows again from theorem 2.2.2 and proposition 2.2.3 using the fact that $\mathbf{Set}_\Omega$ has small coproducts and is wellpowerd. $\square$

Remark. To see the connection with Rutten [109], theorems 17.3 and 17.5: Let $S \in \mathbf{Set}_\Omega$ be a subcoalgebra $S \overset{m}{\hookrightarrow} S_C$ of a cofree coalgebra S_C. The class $\mathcal{K}(S)$ which is considered in [109] is our $\mathsf{Mod}\{m\}$.

Remark. In contrast to what is stated in Rutten [109] one does not have to require that Ω preserves weak pullbacks.

2.5.4 Modally Definable Subcategories are Comonadic

'Categories of algebras for a monad' generalise 'equationally definable classes of algebras for a signature'. Dually, categories of coalgebras for a comonad generalise the idea of modally definable subcategories as shown by the next theorem.

Theorem 2.5.10. *Let $U : \mathcal{C} \to \mathcal{X}$ be comonadic and $\mathcal{C}$ be an $(\mathcal{E}, M)$-category such that for all $f \in \mathcal{C}$ it holds that $Uf \in \mathit{SplitMono}(\mathcal{X}) \;\Rightarrow\; f \in M$. Then every modally definable subcategory $\mathcal{B} \subset \mathcal{C}$ subcategory is the category of coalgebras for a comonad.*

Proof. We show that the restriction V of U to $\mathcal{B}$ is comonadic. We use Beck's theorem (see theorem A.7.1). Since the inclusion $\mathcal{B} \hookrightarrow \mathcal{C}$ has a right adjoint (see definition 2.2.1 and theorem 2.5.4) it remains to show that V creates split equalisers. (It may be helpful to consider the diagram in the proof of theorem 1.1.9.) Let η, ζ be coalgebras in $\mathcal{B}$, $f, g : \eta \to \zeta$ be morphisms and m a split equaliser in $\mathcal{X}$ of f, g. Since U creates split equalisers there is a unique coalgebra $\xi \in \mathcal{C}$ such that m is an equaliser in $\mathcal{C}$ of f, g. Since $m \in \mathit{SplitMono}(\mathcal{X})$, ξ is a M-subobject of η, hence in $\xi \in \mathcal{B}$ ($\mathcal{B}$ is closed under subobjects since it is modally definable). It follows that V creates split equalisers. $\square$

Corollary 2.5.11. *Let Ω be an endofunctor on $\mathbf{Set}$ and $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ have a right adjoint. Then every modally definable subcategory $\mathcal{B} \subset \mathbf{Set}_\Omega$ is the category of coalgebras for a comonad.*

Proof. Since U has a right adjoint, $\mathbf{Set}_\Omega$ is comonadic (theorem 1.1.9). Now, apply the theorem using that $\mathbf{Set}_\Omega$ is an $(\mathit{EpiSink}, \mathit{StrongMono})$-category (see theorem 1.3.15). $\square$

2.6 More on Formulas and Rules as Morphisms

In this section we have a look at the interplay of logical notions concerning formulas (rules) and categorical notions concerning morphisms. Also, it is shown that under mild assumptions the notion of a rule as a morphism and the notion of a rule in the sense of modal logic (see definition 2.6.1) coincide. These results are needed in the next section when we want to give a semantics to concrete modal logics by interpreting formulas of the language as morphisms with cofree codomain. The reader may well skip this section and only go back when the material is needed.

2.6.1 Basic Definitions

In this section proofs are omitted when they are direct consequences of projectivity as satisfaction and the respective definitions.

Fix an $(\mathcal{E}, M)$-category $\mathcal{C}$.

According to definition 2.4.1, a rule is just a morphism in M. But the usage of the term "rule" in modal logic suggests the following definition. Corollaries 2.6.3, 2.6.13 show that both definitions coincide.

Definition 2.6.1 (rules (logically)). For $m_1, m_2 \in M$ with common codomain B, we define

$$A \models m_1/m_2 \qquad \text{iff} \qquad \forall f : A \to B\,.\ A, f \models m_1 \ \Rightarrow\ A, f \models m_2.$$

If $m_1, m_2 \in M$ are formulas then m_1/m_2 is called a (logical) *rule*.

W.r.t. rules, composition of morphisms behaves like modus ponens in the following sense:

Proposition 2.6.2. *$m_2 = m_1 \circ m$ implies*

$$A, f \models m_1/m_2 \qquad \textit{iff} \qquad A, f \models m.$$

Remark. m_1/m_2 and m are equivalent (see definition 2.6.5 below).

In the case that there are enough formulas every rule in the sense of definition 2.4.1 is a rule in the sense of definition 2.6.1 (see definition 2.2.8 for 'enough injective objects'):

Corollary 2.6.3. *Let $\mathcal{C}$ have enough injectives. Then for all $m \in M$ there is a rule m_1/m_2 such that $A \models m$ iff $A \models m_1/m_2$.*

Proof. Let $m : A_1 \to A_2$. Since $\mathcal{C}$ has enough injectives there is $m_1 : A_2 \to A_3 \in M$ with A_3 M-injective. Now let $m_2 = m_1 \circ m$ and apply the proposition. □

Remark. The corollary holds in particular for categories $\mathbf{Set}_\Omega$ that admit a cofree construction.

The converse (that every logical rule is a morphism in M) is proved at the end of the section.

We now express some logical notions in categorical terms.

Definition 2.6.4 (true morphism). A morphism t in $\mathcal{C}$ is called *true* iff all objects of $\mathcal{C}$ are projective w.r.t. t.

Remark. A morphism is true iff it is split epi ('if': Every object is projective w.r.t. a split epi; 'only if': The identity factors through a true morphism).

Definition 2.6.5 (equivalence). Let $e : A_2 \to A_1 \in \mathcal{C}$ and $m_2, m_1 \in M$ with codomain A_2, A_1, respectively.

$$\begin{tikzcd} A_2 \arrow[d, "t"'] & \bullet \arrow[l, "m_2"'] \\ A_1 & \bullet \arrow[l, "m_1"] \end{tikzcd}$$

Then m_2, m_1 are called *equivalent w.r.t. t*, written as $m_1 \equiv_t m_2$, iff for all $B \in \mathcal{C}$, $f : B \to A_2$

$$B, f \models m_2 \;\Leftrightarrow\; B, t \circ f \models m_1.$$

If m_1, m_2 are equivalent w.r.t. a true morphism we write $m_1 \equiv_\top m_2$.
Let m_2, m_1 be morphisms or logical rules (definition 2.6.1). Then m_2, m_1 are called *equivalent*, written as $m_1 \equiv m_2$, iff for all $B \in \mathcal{C}$, $B \models m_2 \;\Leftrightarrow\; B \models m_1$.

Remark. Let t, m_2, m_1 be as in the definition above. It is important to note that $m_1 \equiv_\top m_2$ is a stronger notion than $m_1 \equiv m_2$. The first means

$$\forall f\,(B, f \models m_2 \;\Leftrightarrow\; B, t \circ f \models m_1),$$

whereas the second is—since t is true—equivalent to

$$\forall f : B, f \models m_2 \;\Leftrightarrow\; \forall f : B, t \circ f \models m_1.$$

Definition 2.6.6 (invariant morphism). A morphism $m : A \to B \in M$ is called *invariant* iff $A \models m$.

The notion "invariant" is explained by the following observation: In the case of $\mathcal{C} = \mathbf{Set}$ and $m : A \hookrightarrow B$, m invariant means $f(A) \subset A$ for all $f : A \to B$.

We continue with three propositions linking these notions.

Proposition 2.6.7. *Let $m_1 : A_1 \to B$, $m_2 : A_2 \to B$ be invariant morphisms in M. Then $m_1 \equiv m_2$ implies that m_1, m_2 are isomorphic (i.e. isomorphic as objects in $\mathcal{C} \downarrow B$).*

Proposition 2.6.8.

1. *A pullback of a true morphism is true.*

2. *$m \in M$ is true iff m is iso.*

3. Equivalence w.r.t. a true morphism implies equivalence.

Proof. (1) and (3) are immediate from satisfaction as projectivity and the respective definitions. For (2) let $m : A \to B$. Since m is true, id_B factors through m, that is, m is split epi. Since m is mono by assumption, it follows that m is iso. □

The converse of (3) does not hold in general (see the remark after definition 2.6.5).

Proposition 2.6.9. *Let $m_2, m_1 \in M$, $h \in \mathcal{C}$ and consider the diagram D below:*

$$\begin{tikzcd} A_2 \arrow[d, "h"'] \arrow[dr, phantom, "\mathsf{D}"] & B_2 \arrow[l, "m_2"'] \arrow[d, dotted, "h'"] \\ A_1 & B_1 \arrow[l, "m_1"] \end{tikzcd}$$

There is (a unique) h' making the diagram into a pullback iff $m_1 \equiv_h m_2$.

Proof. Recall that $m_1 \equiv_h m_2$ is defined as

$$\forall A, \forall f : A \to A_2 \quad A, e \circ f \models m_1 \;\Leftrightarrow\; A, f \models m_2.$$

We call " $\Leftarrow$ " and " $\Rightarrow$ " the respective implications.

"only if": We have to show $m_1 \equiv_h m_2$. " $\Leftarrow$ " follows because D commutes. " $\Rightarrow$ " follows because D is a pullback.

"if": Using " $\Leftarrow$ " and $B_2, m_2 \models m_2$ it follows that there is h' such that $m_1 \circ h' = h \circ m_2$. Using " $\Rightarrow$ " it follows that D is a pullback. □

Corollary 2.6.10. *Consider the diagram D of the proposition above.*

1. $\forall A \in \mathcal{C}\ (A \models m_1 \;\Rightarrow\; A \models m_2)$ *if D is a pullback.*
2. $\forall A \in \mathcal{C}\ (A \models m_1 \;\Leftarrow\; A \models m_2)$ *if D commutes and h is true.*

In particular: if h is true and D is a pullback then $m_1 \equiv m_2$.

2.6.2 Categorical Connectives

In this subsection we express conjunctions $m_1 \wedge m_2$ and rules m_1/m_2 in categorical terms.

Definition 2.6.11 (conjunctions and rules (categorically)). Let $m_1, m_2 \in M$ and

$$\begin{tikzcd} \bullet \arrow[d, "m_1"'] & \bullet \arrow[l, "m_2'"'] \arrow[d, "m_1'"] \\ B & \bullet \arrow[l, "m_2"] \end{tikzcd}$$

be a pullback. Define

$$m_1 \wedge m_2 = m_1 \circ m_2' \qquad \text{and} \qquad m_1//m_2 = m_2'.$$

If $m_1, m_2 \in M$ are formulas then $m_1//m_2$ is called a (categorical) *rule*.

It might be more accurate to write $m_1//(m_1 \wedge m_2)$ instead of $m_1//m_2$ but, up to equivalence (definition 2.6.5), this makes no difference.

Proposition 2.6.12.

1. *$A, f \models m_1 \wedge m_2$ iff $A, f \models m_1$ and $A, f \models m_2$.*
2. *$A, f \models m_1//m_2$ iff $A, f \models m_1 \Rightarrow A, f \models m_2$,*

In particular, $A \models m_1//m_2$ iff $A \models m_1/m_2$.

It follows now that logical rules can be expressed by morphisms:

Corollary 2.6.13. *Let $\mathcal{C}$ have pullbacks. Then for all formulas $m_1, m_2 \in M$ there is $m \in M$ such that $A \models m_1/m_2$ iff $A \models m$.*

Let us summarise corollaries 2.6.3, 2.6.13 in the following proposition.

Proposition 2.6.14. *Let $\mathcal{C}$ be a category with pullbacks and enough injectives. Then the three notions of a rule according to definitions 2.4.1, 2.6.1, 2.6.11 are equivalent.*

As long as we only work with *formulas* the notion of "equivalence" of morphisms is an appropriate notion of equivalence. But to show the next proposition on equivalence of *rules* the stronger notion of "equivalence w.r.t." is needed.

The next proposition shows that equivalent morphisms give rise to equivalent rules.

Proposition 2.6.15. *Let $m_1 : A_1 \to A$, $m_2 : A_2 \to A$, $n_1 : B_1 \to B$, $n_2 : B_2 \to B$ in M and $t : B \to A$ in $\mathcal{C}$. Then $n_1 \equiv_t m_1$, $n_2 \equiv_t m_2$ and t true implies $n_1//n_2 \equiv_\top m_1//m_2$.*

Proof. We use proposition 2.6.9. Consider the following diagram consisting of 5 parts labelled N,W,S,E,C:.

$$\begin{tikzcd}
B_2 \arrow[dr, "n_2"] \arrow[ddd, "t_2"'] \arrow[drr, phantom, "\mathsf{N}"] \arrow[ddr, phantom, "\mathsf{W}"] & & & B_0 \arrow[lll, "n_2//n_1"'] \arrow[dl, "n_1//n_2"'] \arrow[ddd, dotted, "t_0"] \arrow[ddl, phantom, "\mathsf{E}"] \\
& B \arrow[d, "t"'] \arrow[dr, phantom, "\mathsf{C}"] & B_1 \arrow[l, "n_1"'] \arrow[d, "t_1"] & \\
& A & A_1 \arrow[l, "m_1"] & \\
A_2 \arrow[ur, "m_2"] \arrow[urr, phantom, "\mathsf{S}"] & & & A_0 \arrow[lll] \arrow[ul, "m_1//m_2"']
\end{tikzcd}$$

Note that N, S are pullbacks by definition of the operation $/\!/$ and that there are (unique) t_2, t_1 such that W, C are pullbacks.

t_1 is true because t is true and C is a pullback. We have to show that there is t_0 such that E is a pullback. We obtain t_0 because $(B_0, t_2 \circ (n_2/\!/n_1), t_1 \circ (n_1/\!/n_2)$ is a cone for the pullback S. To show that E is indeed a pullback suppose that $(Z, f : Z \to A_0, g : Z \to B_1)$ is a cone. Using that W, N are pullbacks one gets $h : Z \to B_0$. It is easy to check that $t_0 \circ h = f$ and $n_1/\!/n_2 \circ h = g$. h is uniquely determined because $n_1/\!/n_2$ is mono. $\square$

2.6.3 On Coreflection Morphisms

Recall theorem 2.2.2 stating that every M-coreflective subcategory is definable. The proof shows that the defining class of morphisms is $\Phi = \{m \mid m \text{ a coreflection morphism}\}$. This shows that we can expect the coreflection morphisms to play a special role.

We first show that every invariant morphism m is a coreflection morphism, namely for the coreflective subcategory $\mathsf{Mod}\{m\}$.

We consider subcategories defined by single morphisms. Given $m \in M$, recall from the proof of theorem 2.2.2 that the subcategory $\mathcal{B} = \mathsf{Mod}\{m\}$ is determined by all coreflection morphisms, i.e., $\mathsf{Mod}\{m\} = \mathsf{Mod}\Phi$ for $\Phi = \{m \mid m \text{ a coreflection morphism}\}$. Obviously, all coreflection morphisms are determined by m and, moreover, m and Φ are equivalent. This observation is useful because it may allow simplifications by substituting a set of rules by a single rule. In this section we investigate whether, given $m \in M$ there is a simple description of the morphisms in Φ not making reference to the coreflection.

For the purposes of this subsection let $m \in M$, $\mathcal{B} = \mathsf{Mod}\{m\}$, $\iota : \mathcal{B} \to \mathcal{C}$ be the inclusion functor and $\varrho : \mathcal{C} \to \mathcal{B}$ a M-coreflection, i.e., a right adjoint to ι. The counit of the adjunction is denoted by ε, i.e., the morphisms $\varepsilon_A : \iota\varrho A \to A$ are in M.

We first show that m itself is a coreflection morphism iff m is invariant (see definition 2.6.6).

Proposition 2.6.16. *Let $m : A' \to A$ and consider a coreflection $\varrho : \mathcal{C} \to \mathsf{Mod}\{m\}$. Then m is a coreflection morphism iff m is invariant.*

Proof. "only if" is immediate. For "if", let (s_i) be the sink consisting of all $B \to A$ with $B \in \mathsf{Mod}\{m\}$. By definition, (s_i) factors through m, that is, $(s_i) = m \circ (t_i)$ for some sink (t_i). Also, (s_i) has an $(\mathcal{E}, M)$-factorisation $n \circ (e_i)$. It follows that there is a diagonal

$$\begin{tikzcd} \bullet \arrow[r, "e_i"] \arrow[d, "t_i"'] & \bullet \arrow[d, "n"] \arrow[dl, dotted, "d"'] \\ A' \arrow[r, "m"'] & A \end{tikzcd}$$

Clearly, d is mono. Since m is invariant, m factors through n, i.e., there is i with $s_i = m$ and $t_i = \mathrm{id}$ such that the diagram commutes. It follows that d is iso. Since n is a coreflection morphism also m is. $\square$

In the typical application of the next proposition $t : A \to A'$ will be a surjective morphism relating a larger A to a smaller A'. The proposition can then be interpreted as saying that the coreflection morphism on the larger structure determines the coreflection morphisms on the smaller ones.

Proposition 2.6.17. *Let $\mathcal{B} \subset \mathcal{C}$ be a full coreflective subcategory with coreflection $\varrho : \mathcal{C} \to \mathcal{B}$ and counit ε. Consider*

$$\begin{tikzcd} A \arrow[d, "t"'] & \bullet \arrow[l, "\varepsilon_A"'] \arrow[d, "\varrho t"] \\ A' & \bullet \arrow[l, "\varepsilon_{A'}"'] \end{tikzcd}$$

If t true, then $\varepsilon_{A'} \circ \varrho t$ is an $(\mathcal{E}, M)$-factorisation of $t \circ \varepsilon_A$.

Proof. Let s be the sink consisting of all $B \to A$ with $B \in \mathcal{B}$. Since ε_A is a coreflection morphism, s has an $(\mathcal{E}, M)$-factorisation $\varepsilon_A \circ \mathsf{e}$. Since t is true, $t \circ \mathsf{s}$ is a sink consisting of all $B \to A'$ with $B \in \mathcal{B}$. Since $\varepsilon_{A'}$ is a coreflection morphism, $t \circ \mathsf{s}$ has an $(\mathcal{E}, M)$-factorisation $\varepsilon_{A'} \circ \mathsf{e}'$. Now let $e \circ m$ be an $(\mathcal{E}, M)$-factorisation of $t \circ \varepsilon_A$. Consider

$$\begin{tikzcd} \bullet \arrow[r, "\mathsf{e}'"] \arrow[d, "e \circ \mathsf{e}"'] & \bullet \arrow[d, "\varepsilon_{A'}"] \arrow[dl, dotted, "d"'] \\ \bullet \arrow[r, "m"] & \bullet \end{tikzcd}$$

where the diagonal fill-in d is an iso. Now, $\varepsilon_{A'} \circ \varrho t$ is a factorisation because

$$\begin{tikzcd} \bullet \arrow[r, "e"] \arrow[d, "\varrho t"'] & \bullet \arrow[d, "m"] \arrow[dl, dotted, "d"'] \\ \bullet \arrow[r, "\varepsilon_{A'}"] & \bullet \end{tikzcd}$$

commutes. ☐

Corollary 2.6.18. *Consider*

$$\begin{tikzcd} A \arrow[d, "t"'] & \bullet \arrow[l, "m"'] \arrow[d, "e'"] \\ A' & \bullet \arrow[l, "m'"'] \end{tikzcd}$$

Let $m \in M$ be invariant, t true, and $m' \circ e'$ an $(\mathcal{E}, M)$-factorisation of $t \circ m$. Then there is a coreflection $\varrho : \mathcal{C} \to \mathsf{Mod}\{m\}$ with counit ε such that $m = \varepsilon_A$, $m' = \varepsilon_{A'}$, and $e' = \varrho t$.

Proof. Follows from the two propositions above and the factorisation $m' \circ e'$ being essentially unique. □

The next proposition is in some sense a converse to the previous one. The question is how a coreflection morphism on a smaller structure determines the coreflection morphisms on larger structures. In general this is not possible, but we will look at a case where we know that one morphism determines all coreflection morphisms, namely where the subcategory under consideration is $\mathsf{Mod}\{m\}$.

Proposition 2.6.19. *Consider*

$$\begin{tikzcd} A' \arrow[d, "t"'] & B' \arrow[l, "m'"'] \arrow[d, "t'"] \\ A & B \arrow[l, "m"] \end{tikzcd}$$

Let $m \in M$ be invariant and t be true. If the diagram is a pullback and m' is invariant then m' is a coreflection morphism for the subcategory $\mathsf{Mod}\{m\}$.

Proof. Assume that the diagram is a pullback and let $\varepsilon_{A'} : B'' \to A'$ be a coreflection morphism for $\mathsf{Mod}\{m\}$. Since m is invariant it is a coreflection morphism for $\mathsf{Mod}\{m\}$ by proposition 2.5.6. It follows that $t \circ \varepsilon_{A'} = m \circ g$ for some $g : B'' \to B$, that is, $(B', \varepsilon_{A'}, g)$ is a cone for the pullback. Hence $\varepsilon_{A'}$ factors through m'. To see that m' factors through $\varepsilon_{A'}$ note that it follows from m' invariant and t true that $B' \models m$, i.e., $B' \in \mathsf{Mod}\{m\}$. □

2.7 Modal Logics for Coalgebras

This section investigates modal logics for categories $\mathcal{X}_\Omega$ whose forgetful functor $U : \mathcal{X}_\Omega \to \mathcal{X}$ has a right adjoint F. We assume that $\mathcal{X}_\Omega$ is an $(\mathcal{E}, M)$-category. As injectives we can now take the coalgebras FC cofree over C. A formula with colours in C will be interpreted as a subcoalgebra of FC. Moreover, given a coalgebra A, there is now a one-to-one correspondence between colourings $\gamma : UA \to C$ and morphisms $\gamma^\# : A \to FC$.[16]

2.7.1 The Logics

The notion of logic that is presented here has been chosen to be general enough to encompass standard notions of modal logic as well as Moss' coalgebraic logic. We do not give a particular syntax but describe a standardised interface between modal logics for coalgebras and their semantics.

The first idea to describe modal logics for coalgebras is simply the following (think of $\mathcal{C}$ as $\mathcal{X}_\Omega$).

Definition 2.7.1 (modal logic for coalgebras (bounded case)). Let $U : \mathcal{C} \to \mathcal{X}$ be a functor with right adjoint F and $\mathcal{C}$ be an $(\mathcal{E}, M)$-category bounded by FC for some $C \in \mathcal{X}$. A modal logic for $\mathcal{C}$ consists of a class of formulas $\mathcal{L}$ and a function $[\![-]\!] : \mathcal{L} \to M$ mapping formulas to morphisms in M with codomain FC. The satisfaction relation for the logic is given by

$$A \models \varphi \text{ iff } A \text{ is projective w.r.t. } [\![\varphi]\!]$$

where $A \in \mathcal{C}$ and $\varphi \in \mathcal{L}$.

The definition above works fine as long as we can restrict our attention to formulas on a single set of colours. This is possible if the category is bounded and if we are only interested in characterising the expressive power of modal formulas and not of modal rules. Otherwise we have to work with a proper class of 'sets of colours' $C \in \mathcal{X}$. In particular, we have to consider classes of formulas $\mathcal{L}_C$ and functions $[\![-]\!]_C$ for different $C \in \mathcal{X}$. Therefore, we need to be able to deal with 'weakening' w.r.t. the 'contexts' C: Consider a formula in modal logic, e.g., $\varphi = \Box p \to p$ which can be interpreted w.r.t. different sets of variables, e.g., $\{p\}$ and $\{p, q\}$. Now, we have to take care that the given semantics of φ in context $\mathcal{P}(\{p\})$ and of φ in context $\mathcal{P}(\{p, q\})$ are equivalent.

Spelling all this out in detail is a bit messy. But fortunately—following a suggestion by Dirk Pattinson—the material can be organised in a more satisfying manner using techniques of fibred category theory (see Jacobs [63]). The basic idea is to replace $[\![-]\!] : \mathcal{L} \to M$ in

[16] Note the duality to algebra where C is a set of variables, FC a free algebra, and $\gamma : C \to UA$ a valuation of the variables.

definition 2.7.1 by the following 'fibred picture'

$$\begin{tikzcd} \mathcal{L} \arrow[r, "{[\![-]\!]}"] \arrow[d, "c"'] & \hat{M} \arrow[d, "col"] \\ \mathbb{C} \arrow[r, "{[\![-]\!]_{\mathbb{C}}}"'] & \mathcal{X} \end{tikzcd}$$

where the language is given by a fibration $c : \mathcal{L} \to \mathbb{C}$ which maps a formula φ in context to its context (e.g. $\Box p \to p$ to $\mathcal{P}(\{p\})$). The semantics of the language is given by two functors $[\![-]\!]$, $[\![-]\!]_{\mathbb{C}}$ mapping formulas to morphisms and contexts to sets of colours. $\hat{M}$ contains the morphisms in M with cofree codomain and *col* maps a morphism $A \to FC$ to C.

Definition 2.7.2 ($col : \hat{M} \to \mathcal{X}$). Let $U : \mathcal{C} \to \mathcal{X}$ be a functor with right adjoint F and $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. Define $\hat{M}$ as the category having as objects morphisms in M with cofree codomain. Arrows $(n : B \to FD) \to (m : A \to FC)$ are pairs (f, p) with $f : B \to A \in \mathcal{X}_\Omega$ and $p : D \to C$ such that

$$\begin{tikzcd} B \arrow[r, "f"] \arrow[d, "n"'] & A \arrow[d, "m"] \\ FD \arrow[r, "Fp"'] & FC \end{tikzcd}$$

commutes. *col* maps $m : A \to FC$ to C and an arrow (f, p) to p. Morphisms $m : A \to FC$ are said to be *over* C, arrows (f, p) over p.

Remark. $\hat{M}$ is the full subcategory of $\mathrm{Id}_{\mathcal{C}}{\downarrow}F$ generated by morphisms in M.

Proposition 2.7.3. *$col : \hat{M} \to \mathcal{X}$ is a fibration: col is a functor and, given $p : D \to C \in \mathcal{X}$ and $m \in \hat{M}$ over C, the cartesian lifting (f, p) of p in m is given by the following pullback:*

$$\begin{tikzcd} \bullet \arrow[r, "f"] \arrow[d, "n"'] \arrow[dr, phantom, "\lrcorner", very near start] & \bullet \arrow[d, "m"] \\ FD \arrow[r, "Fp"'] & FC \end{tikzcd}$$

Proof. The proof is the same as for the so-called codomain fibration (see Jacobs [63]). Only recall that the pullback exists because pullbacks along morphisms in M exist in $(\mathcal{E}, M)$-categories (Adámek, Herrlich, Strecker [4], theorem 15.14(3)) and that morphisms in M are preserved by pullbacks. $\Box$

For readers not familar with fibred category theory it might be useful to read the following definition after example 2.7.9. The intended interpretation of the next definition is $\mathcal{C} = \mathcal{X}_\Omega$ but the more general version comes without any extra cost.

Definition 2.7.4 (modal logic for Ω-coalgebras). Let $U : \mathcal{C} \to \mathcal{X}$ be a functor with right adjoint F and $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. A modal logic for $\mathcal{C}$ is a fibration $c : \mathcal{L} \to \mathbb{C}$ and a fibred functor $([\![-]\!], [\![-]\!]_{\mathbb{C}})$

$$\begin{tikzcd} \mathcal{L} \arrow[r, "{[\![-]\!]}"] \arrow[d, "c"'] & \hat{M} \arrow[d, "col"] \\ \mathbb{C} \arrow[r, "{[\![-]\!]_{\mathbb{C}}}"'] & \mathcal{X} \end{tikzcd}$$

The satisfaction relation is given for a formula $\varphi \in \mathcal{L}$, $A \in \mathcal{C}$, and a valuation $\gamma : UA \to [\![c(\varphi)]\!]_{\mathbb{C}}$ by

$$A, \gamma \models \varphi \ \text{ iff } \gamma^{\#} \text{ factors through } [\![\varphi]\!].$$

Satisfaction for modal rules φ/ψ, $\varphi, \psi \in \mathcal{L}$, $c(\varphi) = c(\psi)$ is given by

$$A, \gamma \models \varphi/\psi \ \text{ iff } A, \gamma \models \varphi \ \Rightarrow \ A, \gamma \models \psi.$$

Remark 2.7.5 (Notation and Terminology). Objects $\varphi \in \mathcal{L}$ are called *formulas in context* or, briefly, *formulas*. $\mathbb{C}$ is a called a category of *contexts*. Contexts are denoted by Γ, Δ. The intended interpretation is that $c(\varphi)$ denotes the 'set of colours' occurring in the formula $\varphi \in \mathcal{L}$. More precisely, we say that *φ is a formula in colours C* if $[\![c(\varphi)]\!]_{\mathbb{C}} = C$; $C \in \mathcal{X}$ is called a *set of colours* if there is a context $\Gamma \in \mathbb{C}$ such that $[\![\Gamma]\!]_{\mathbb{C}} = C$. It is often convenient to write $\varphi[\Gamma]$ for a formula in $\mathcal{L}$ to indicate that $c(\varphi[\Gamma]) = \Gamma$. Arrows $q : \Delta \to \Gamma \in \mathbb{C}$ are called *substitutions*.

Remark 2.7.6 (Satisfaction of Rules). Note that, by proposition 2.6.12, we can express satisfaction of rules $\varphi[\Delta]/\psi[\Delta]$, categorically as

$$A, \gamma \models \varphi[\Delta]/\psi[\Delta] \text{ iff } \gamma^{\#} \text{ factors through } [\![\varphi[\Delta]]\!] /\!/ [\![\psi[\Delta]]\!].$$

In particular, $A \models \varphi[\Delta]/\psi[\Delta]$ iff $A \models [\![\varphi[\Delta]]\!] /\!/ [\![\psi[\Delta]]\!]$.

Remark 2.7.7 (Cartesian Liftings). The requirement that c is a fibration gives for every $q : \Delta \to \Gamma \in \mathbb{C}$ and every $\varphi[\Gamma]$ a formula $\psi[\Delta]$ and an arrow (cartesian lifting) $\psi[\Delta] \to \varphi[\Gamma]$. The intended interpretation of $\psi[\Delta]$ is that it is obtained by performing the substitution q on $\varphi[\Gamma]$.

Remark 2.7.8 (Fibred Functor). Intuitively, the requirement that $([\![-]\!], [\![-]\!]_{\mathbb{C}})$ is a fibred functor expresses that the semantics of formulas given by $([\![-]\!], [\![-]\!]_{\mathbb{C}})$ is compatible with substitutions $q : \Delta \to \Gamma \in \mathbb{C}$ and, in particular, weakening of formulas does not change their

meaning.
Technically, the requirement can be unfolded to the statement that the square on the upper right side of the following diagram is a pullback in $\mathcal{C}$ for all $q : \Delta \to \Gamma \in \mathbb{C}$ and all $\varphi[\Gamma] \in \mathcal{L}$

$$\begin{tikzcd}
 & & & \bullet \arrow[r, "f"] \arrow[d, "{[\![\psi[\Delta]]\!]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \bullet \arrow[d, "{[\![\varphi[\Gamma]]\!]}"] \\
\psi[\Delta] \arrow[r, "\bar{q}"] & \varphi[\Gamma] & {} \arrow[r, mapsto, "{[\![-]\!]}"] & FD \arrow[r, "Fp"'] & FC \\
\Delta \arrow[r, "q"'] & \Gamma & {} \arrow[r, mapsto, "{[\![-]\!]_{\mathbb{C}}}"] & D \arrow[r, "p"'] & C
\end{tikzcd}$$

where

1. $\bar{q} : \psi[\Delta] \to \varphi[\Gamma]$ is a cartesian lifting of $q : \Delta \to \Gamma$,

2. $(f, Fp) = [\![\bar{q}]\!]$,

3. $p : D \to C = [\![q : \Delta \to \Gamma]\!]_{\mathbb{C}}$.

It follows from proposition 2.6.9 and the square being a pullback that $[\![\psi[\Delta]]\!]$ and $[\![\varphi[\Gamma]]\!]$ are equivalent w.r.t. Fp (see definition 2.6.5). This now implies that—semantically—'$\psi[\Delta]$ is obtained by performing the substitution q on $\varphi[\Gamma]$':[17]

$$A, \delta \models \psi[\Delta] \quad \Leftrightarrow \quad A, p \circ \delta \models \varphi[\Gamma],$$

for $A \in \mathcal{C}$ and a valuation $\delta : UA \to D$.

Before looking at the following example recall that $\mathbf{Set}_\Omega$ is an $(\mathit{EpiSink}, \mathit{StrongMono})$-category and that strong monos in $\mathbf{Set}_\Omega$ are precisely the subcoalgebras (see theorem 1.3.10 and corollary 1.3.15).

Example (Modal Logic) 2.7.9. We show how the definition above applies to modal logic in the case $\mathcal{X} = \mathbf{Set}$ and $\Omega X = \mathcal{P}_\kappa X$ (see example 1.1.4). We define $\mathbb{C}$ to be the category with objects $\{\mathcal{P}P : P \in \mathbf{Set}\}$ and morphisms $\iota^{-1} : \mathcal{P}Q \to \mathcal{P}P$ where $\iota : P \hookrightarrow Q$. Let $\mathcal{L}$ be pairs (φ, P), written as $\varphi[\mathcal{P}P]$, where φ is a formula of (standard) modal logic and P is a set of propositional variables containing the propositional variables occurring in φ. c maps a formula (φ, P) to $\mathcal{P}P$. c is defined to be a fibration in a trivial sense: The fibres $\mathcal{L}_{\mathcal{P}P}$ over $\mathcal{P}P$ are discrete; cartesian arrows are, for each $\iota^{-1} : \mathcal{P}Q \to \mathcal{P}P$ and for each formula $\varphi[\mathcal{P}P]$, an arrow $\bar{\iota} : \varphi[\mathcal{P}Q] \to \varphi[\mathcal{P}P]$ (note that the domain of the cartesian lifting has the same formula as the codomain, only the context is enlarged). $[\![-]\!]_{\mathbb{C}}$ is inclusion and $[\![-]\!]$ maps a formula

[17] Note that $(p \circ \delta)^\# = Fp \circ \delta^\#$.

$\varphi[\mathcal{P}P]$ to the largest subcoalgebra of $(F\mathcal{P}P, \varepsilon_{\mathcal{P}P})$ satisfying φ. On morphisms $\bar{\iota}$ we define $[\![-]\!]$ as follows. Consider the following diagram.

$$\begin{tikzcd} \bullet \arrow[r, dotted, "f"] \arrow[d, "{[\![\varphi[\mathcal{P}Q]]\!]}"'] & \bullet \arrow[d, "{[\![\varphi[\mathcal{P}P]]\!]}"] \\ F\mathcal{P}Q \arrow[r, "Fp"'] & F\mathcal{P}P \end{tikzcd}$$

First note that $p = \iota^{-1} : \mathcal{P}Q \to \mathcal{P}P$ is surjective, hence Fp is true.[18] Moreover, it is an almost trivial fact in modal logic that the interpretation of a formula does not depend on the interpretation of variables which do not appear in the formula. This implies that for all $A \in \mathcal{X}_\Omega$ and all $\delta : UA \to \mathcal{P}Q$:

$$A, \delta \models \varphi[\mathcal{P}Q] \quad \Leftrightarrow \quad A, p \circ \delta \models \varphi[\mathcal{P}P].$$

This statement is equivalent to saying that $[\![\varphi[\mathcal{P}Q]]\!]$ and $[\![\varphi[\mathcal{P}P]]\!]$ are equivalent w.r.t. Fp. It now follows from proposition 2.6.9 that there exists (a unique) f making the diagram above into a pullback. Defining $[\![\bar{\iota}]\!] = (f, p)$ now makes $([\![-]\!], [\![-]\!]_{\mathbb{C}})$ into a fibred functor.
It remains to check that the satisfaction relation defined in standard modal logic conincides with the one given by definition 2.7.4. This was shown in section 2.4.1.

Remark 2.7.10 (Logics with derivability). In the example above, the fibres are discrete because we are—in the investigations of this chapter—not interested in syntactic derivability but only in semantic consequence. However, our notion of a modal logic is general enough to allow for arrows $\psi[\Gamma] \to \varphi[\Gamma]$ expressing derivability. $([\![-]\!], [\![-]\!]_{\mathbb{C}})$ being a fibred functor then means that $\psi[\Gamma] \to \varphi[\Gamma]$ in $\mathcal{L}$ only if for all $A \in \mathcal{X}_\Omega$ and all valuations $\gamma : UA \to [\![\Gamma]\!]_{\mathbb{C}}$ it holds that $A, \gamma \models \psi \;\Rightarrow\; A, \gamma \models \varphi$. This means that derivability as expressed by $\psi[\Gamma] \to \varphi[\Gamma]$ is correct w.r.t. to the semantics given by $([\![-]\!], [\![-]\!]_{\mathbb{C}})$. (This kind of derivability corresponds to a modal calculus with necessitation but without substitution rule as e.g. given in appendix B.)

One usually expects modal formulas to be invariant under bisimulations. The following proposition, which is almost immediate from $\mathcal{C}$ being an $(\mathcal{E}, M)$-category, shows that **formulas in colours C are invariant under $C \times \Omega$-bisimulations**.

Proposition 2.7.11. *Let $\mathcal{L}$ be a modal logic for $U : \mathcal{C} \to \mathcal{X}$ as in definition 2.7.4. Let $f : A \to B \in \mathcal{C}$ be a morphism in $\mathcal{E}$,[19] $C \in \mathcal{X}$, $\varphi \in \mathcal{L}$ be a formula in colours C, and $\gamma : UA \to C$, $\delta : UB \to C$ with $\delta \circ f = \gamma$. Then*

$$A, \gamma \models \varphi \;\Leftrightarrow\; B, \delta \models \varphi.$$

[18] For the notions of true and equivalent morphisms see definitions 2.6.4 and 2.6.5.
[19] Recall from sections 1.2 and 1.4 that f is a behavioural equivalence, which is a bisimulation equivalence if the signature preserves weak pullbacks.

Proof. Consider the following diagram

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[dr, "\gamma^{\#}"] & B \arrow[d, "\delta^{\#}"] \\
\bullet \arrow[r, "\llbracket\varphi\rrbracket"'] & FC
\end{tikzcd}$$

For " $\Leftarrow$ " we have to show that $\gamma^{\#}$ factors through $\llbracket\varphi\rrbracket$ if $\delta^{\#}$ does, which is immediate. For " $\Rightarrow$ " we have to show that $\delta^{\#}$ factors through $\llbracket\varphi\rrbracket$ if $\gamma^{\#}$ does. This follows from $\llbracket\varphi\rrbracket \in M$, $f \in \mathcal{E}$, and $(\mathcal{E}, M)$ being a factorisation structure. $\square$

2.7.2 Expressiveness

According to definition 2.7.4, a modal logic for coalgebras may consist of a single formula "true". To avoid these boring cases we need to impose conditions that guarantee some expressive power. The definitions below are designed to ensure maximal expressive power (i.e. *every* morphism in M (with cofree codomain) can be expressed). This is a rather strong condition that will require infinite conjunctions in concrete examples (see sections 2.8, 2.9). To analyse logics with less expressive power (in particular, finitary logics) will be left for future work.

The idea of the following definition is to call a modal logic expressive if every morphism in M with cofree codomain can be expressed by the logic. As it turns out in the proof of the covariety theorem it is enough to require that *invariant* morphisms (see definition 2.6.6) can be expressed.

Definition 2.7.12 (expressive). A modal logic for Ω-coalgebras $\mathcal{L}$ is called *expressive* iff for all invariant $m \in \hat{M}$ there is $\varphi \in \mathcal{L}$ such that $\llbracket\varphi\rrbracket$ and m are equivalent.

This condition does not guarantee that all morphisms in M can be expressed as rules of $\mathcal{L}$. Therefore (recall the definition 2.6.11 of $//$):

Definition 2.7.13 (rule-expressive). Let $\mathcal{L}$ be a modal logic for Ω-coalgebras. Then $\mathcal{L}$ is *rule-expressive* iff for every $m \in M$ there are $\varphi[\Delta], \psi[\Delta] \in \mathcal{L}$ such that m and $\llbracket\varphi[\Delta]\rrbracket // \llbracket\psi[\Delta]\rrbracket$ are equivalent.

Remark. Recall from remark 2.7.6 that $A \models \llbracket\varphi[\Delta]\rrbracket // \llbracket\psi[\Delta]\rrbracket$ iff $A \models \varphi[\Delta]/\psi[\Delta]$.

Proposition 2.7.14. *Let Ω be a functor on* **Set**. *A modal logic for Ω-coalgebras $\mathcal{L}$ is rule-expressive if for all $C \in$* **Set** *there is $\Delta \in \mathbb{C}$ and a surjection $p : \llbracket\Delta\rrbracket_{\mathbb{C}} \to C$ such that for all $m \in M$ over C there is $\varphi[\Delta] \in \mathcal{L}$ such that $\llbracket\varphi[\Delta]\rrbracket$ and m are equivalent w.r.t. Fp.*

Proof. Let $m \in M$. Using corollary 2.6.3 we find m_1, m_2 with codomain FC (for some $C \in$ **Set**) such that $A \models m$ iff $A \models m_1/m_2$. By proposition 2.6.12, $A \models m_1/m_2$ iff $A \models m_1//m_2$. By assumption we find $\Delta \in \mathbb{C}$, a surjection $p : \llbracket\Delta\rrbracket_{\mathbb{C}} \to C$ and formulas $\varphi_1[\Delta], \varphi_2[\Delta] \in \mathcal{L}$ such that $\llbracket\varphi_1[\Delta]\rrbracket \equiv_{Fp} m_1$, $\llbracket\varphi_2[\Delta]\rrbracket \equiv_{Fp} m_2$. Note that Fp is true since p is surjective. It follows from proposition 2.6.15 that $\llbracket\varphi_1[\Delta]\rrbracket // \llbracket\varphi_2[\Delta]\rrbracket \equiv_{\top} m_1//m_2$. Therefore, m and $\llbracket\varphi[\Delta]\rrbracket // \llbracket\psi[\Delta]\rrbracket$ are equivalent. $\square$

Example 2.7.15. Consider the coalgebras for the functor $\Omega X = \mathcal{P}_\omega(L \times X)$. Coalgebras for this functor are finitly branching labeled transition systems. Adding countably many propositional variables to Hennessy-Milner logic makes this logic (rule) expressive.

2.7.3 Co-Birkhoff Theorems

This section applies the co-Birkhoff theorems of section 2.5 to characterise the expressive power of expressive modal logics for coalgebras.

The theorems are stated for the case $\mathcal{X} = \mathbf{Set}$ but also hold for all $U : \mathcal{X}_\Omega \to \mathcal{X}$ having a right adjoint, satisfying axioms 1 and 4 from section 1.4, having enough $M_\mathcal{X}$-injective objects in the base category (see proposition 2.2.10; $M_\mathcal{X}$ is from the factorisation system in the base category), and satisfying $f \circ g \in M_\mathcal{X} \;\Rightarrow\; g \in M_\mathcal{X}$.[20] We then have to choose the functor $[\![-]\!]_\mathbb{C} : \mathbb{C} \to \mathcal{X}$ in such a way that contexts in $\mathbb{C}$ are mapped to $M_\mathcal{X}$-injective objects in $\mathcal{X}$.

As a corollary to theorem 2.5.1 we obtain:

Theorem 2.7.16 (co-quasivariety theorem). *Let Ω be an endofunctor on* $\mathbf{Set}$ *such that the forgetful functor $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint and let $\mathcal{L}$ be a rule-expressive modal logic for Ω-coalgebras. For a full subcategory $\mathcal{B}$ of $\mathbf{Set}_\Omega$ are then equivalent:*

1. *$\mathcal{B}$ is definable by rules of $\mathcal{L}$,*
2. *$\mathcal{B}$ is a StrongMono-coreflective subcategory of $\mathbf{Set}_\Omega$,*
3. *$\mathcal{B}$ is closed under quotients and small coproducts.*

Proof. (1) $\Rightarrow$ (2) and (2) $\Rightarrow$ (3) are as in theorem 2.5.1 using proposition 2.2.3. For (3) $\Rightarrow$ (1) let $A \in \mathbf{Set}_\Omega$ and $m_A : A' \to A$ be the corresponding M-coreflection morphism. Using expressiveness we can find φ_A, ψ_A such that $A \models m_A \;\Leftrightarrow\; A \models \varphi_A/\psi_A$. Let $\mathsf{Ru}' = \{\varphi_A/\psi_A : A \in \mathbf{Set}_\Omega\}$. Now, $\mathsf{ModRu}'(\mathcal{B}) = \mathcal{B}$ follows from proposition 2.5.1. $\square$

Similarly, as a corollary to theorem 2.5.4:

Theorem 2.7.17 (covariety theorem). *Let Ω be an endofunctor on* $\mathbf{Set}$ *such that the forgetful functor $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint and let $\mathcal{L}$ be an expressive modal logic for Ω-coalgebras. Then the following are equivalent for a full subcategory $\mathcal{B} \subset \mathbf{Set}_\Omega$:*

1. *$\mathcal{B}$ is definable by formulas of $\mathcal{L}$,*
2. *$\mathcal{B}$ is closed under subcoalgebras, quotients and small coproducts.*
3. *$\mathcal{B}$ is a StrongMono-coreflective subcategory and is closed under subcoalgebras.*

Proof. (1) $\Rightarrow$ (2) and (2) $\Rightarrow$ (3) are as in theorem 2.5.4 (also use proposition 2.2.3). For (3) $\Rightarrow$ (1) consider for all $\Gamma \in \mathbb{C}$ the cofree coalgebras $FC = F([\![\Gamma]\!]_\mathbb{C})$. Let $m_C : F_\mathcal{B}C \to FC$ be the corresponding M-coreflection morphisms[21] and $\mathsf{Th}'(\mathcal{B}) = \{m_C \mid \exists \Gamma \in \mathbb{C} : C = [\![\Gamma]\!]_\mathbb{C}\}$. Since the m_C are invariant and the logic is expressive, there is $\Phi \subset \mathcal{L}$ equivalent to $\mathsf{Th}'(\mathcal{B})$. It follows $\mathsf{Mod}(\Phi) = \mathcal{B}$ from theorem 2.5.4. $\square$

[20] In categories with binary products this last requirement is equivalent to $E_\mathcal{X} \subset \mathit{Epi}(\mathcal{X})$, see Adamek, Herrlich, Strecker [4], proposition 14.11.

[21] The reader might want to note that the coalgebras $F_\mathcal{B}C$ are the coalgebras which are cofree for $\mathcal{B}$ as in example 2.5.6.

Similarly, as a corollary to theorem 2.5.7:

Theorem 2.7.18 (covariety theorem for bounded categories). *Let Ω be an endofunctor on* **Set** *such that the forgetful functor $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint and let $\mathcal{L}$ be an expressive modal logic for Ω-coalgebras. Moreover, assume that the category $\mathbf{Set}_\Omega$ is bounded by a cofree coalgebra FC for some set of colours $C \in \mathbf{Set}$.*[22] *Then the following are equivalent for a full subcategory $\mathcal{B} \subset \mathbf{Set}_\Omega$:*

1. *$\mathcal{B}$ is definable by a single formula in colours C,*
2. *$\mathcal{B}$ is closed under subcoalgebras, quotients and small coproducts.*
3. *$\mathcal{B}$ is a StrongMono-coreflective subcategory and is closed under subcoalgebras.*

The proof is immediate from theorem 2.5.7. Moreover, it is possible to weaken the expressiveness requirement a little: For a category $\mathbf{Set}_\Omega$ bounded by a cofree coalgebra FC a modal logic is already expressive if it can express all morphisms $m \in M$ with codomain FC.

Proposition 2.7.19. *Let Ω be an endofunctor on* **Set** *such that the forgetful functor $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint and let $\mathcal{L}$ be a modal logic for Ω-coalgebras. Moreover, assume that the category $\mathbf{Set}_\Omega$ is bounded by a cofree coalgebra FC for some set of colours $C = [\![\Gamma]\!]_{\mathbb{C}}$, $\Gamma \in \mathbb{C}$. Then $\mathcal{L}$ is expressive if for all $m \in M$ with codomain FC there is $\varphi[\Gamma] \in \mathcal{L}$ such that $[\![\varphi[\Gamma]]\!]$ and m are equivalent.*

Proof. We show that for all $D \in \mathbf{Set}$ and all m_D with codomain FD there is $\varphi[\Gamma]$ such that $[\![\varphi[\Gamma]]\!]$ and m_D are equivalent. By theorem 2.5.7 we know that there is $m \in M$ with codomain FC such that $\mathsf{Mod}\{m\} = \mathsf{Mod}\{m_D\}$. By assumption, there is $\varphi[\Gamma]$ such that $[\![\varphi[\Gamma]]\!]$ and m are equivalent. Hence $[\![\varphi[\Gamma]]\!]$ and m_D are equivalent. $\square$

[22] Recall that this is in particular the case if the functor Ω is bounded by C, see proposition 1.5.4.

2.8 Coalgebraic Logic

One motivation for proving co-Birkhoff theorems for coalgebras is that the work of Lawrence Moss on coalgebraic logic [87] provides us with an expressive modal logic for coalgebras for a large class of functors Ω on **Set**. To show this in detail is the topic of the present section. For a brief summary of coalgebraic logic see appendix C.

In order to show that coalgebraic logic provides us with an example of a modal logic for coalgebras as defined in section 2.7, we have to augment coalgebraic logic with colours or propositional variables. This can be done in two ways explained below.

2.8.1 Coalgebraic Logic with Colours

In order to get a version of coalgebraic logic that allows us to state a co-Birkhoff theorem we want to add colours to the logic. How should we do that? Consider the algebraic case: Given a signature Ω, an equation with variables from X is a pair of terms for the signature $X + \Omega$. Analogously, a formula for Ω-coalgebras with colours from C is a formula for the signature $C \times \Omega$. To interpret $C \times \Omega$-formulas in Ω-coalgebras we use (see proposition 1.1.5) that an Ω-coalgebra A and a valuation $\alpha : UA \to C$ determine a $C \times \Omega$-coalgebra (A, α).

This allows for the following definition:

Definition 2.8.1 (Coalgebraic logic with colours). Let Ω be a functor, and C a set. The coalgebraic logic for Ω with colours from C, denoted by $\mathcal{CL}_{C,\Omega}$, is given by the set of colours C, the class of formulas $\mathcal{CL}_{C\times\Omega}$ and by the relation $\models_{C,\Omega}$ defined by

$$A, \alpha, x \models_{C,\Omega} \varphi \;\Leftrightarrow\; x \models_{C\times\Omega} \varphi,$$

where A an Ω-coalgebra, $\alpha : UA \to C$, $x \in UA$, $\varphi \in \mathcal{CL}_{C\times\Omega}$, and $\models_{C\times\Omega}$ the semantics of coalgebraic logic defined above w.r.t. the $C \times \Omega$-coalgebra (A, α).

It is not difficult to see that $\mathcal{CL}_{C,\Omega}$ meets the requirements of a modal logic for coalgebras as in definition 2.7.1. It remains to ensure its expressiveness.

In [87], *uniform* functors are defined and it is proven that for uniform functors Ω the logic $\mathcal{CL}_\Omega$ has characterising formulas: Let $F1$ be the final Ω-coalgebra and $t \in UF1$. Then there is $\varphi_t \in \mathcal{CL}_\Omega$ such that $s \models_\Omega \varphi_t \;\Leftrightarrow\; s = t$.

Now suppose that $C \times \Omega$ is uniform and has a final $C \times \Omega$-coalgebra (FC, ϵ_C). Let S be a subcoalgebra of (FC, ϵ_C). Define $\varphi = \bigvee_{t\in US} \varphi_t$. It follows $FC, \epsilon_C, x \models \varphi \;\Leftrightarrow\; x \in US$. Hence $\mathcal{CL}_{C\times\Omega}$ can express every subcoalgebra of (FC, ϵ_C). It follows now from propositions 2.7.19 and 1.5.4 that $\mathcal{CL}_{C\times\Omega}$ is expressive if $C \times \Omega$ is bounded by C. We have just shown that coalgebraic logic with colours meets the assumptions of our covariety theorem 2.7.18. Therefore:

Theorem 2.8.2. *Let Ω be a standard and weak pullback preserving functor on* **Set** *bounded by C and let $C\times\Omega$ be uniform. Then a class K of Ω-coalgebras is a covariety iff it is definable by a class of formulas of $\mathcal{CL}_{C\times\Omega}$.*

2.8.2 Coalgebraic Logic with Propositional Variables

To prove a co-quasivariety theorem for coalgebraic logic we need sets of colours of arbitrary large cardinality. Unfortunately, we cannot simply extend the definition of the previous subsection by letting $\mathcal{L} = \bigcup\{\mathcal{CL}_{C\times\Omega} : C \text{ a set}\}$ because the semantics of a formula $\varphi \in \mathcal{L}$ will depend on the choice of C. For example, let $\varphi = (c_1, \mathit{true}) \vee \neg(c_1, \mathit{true})$ and choose a set C with $c_1 \in C$. Now, in the case that C contains only two elements, φ is true (because every state is either coloured with c_1 or not), whereas for $|C| > 2$, φ is false (for some valuation). It is possible to handle this by carefully choosing the arrows in $\mathbb{C}$ and their cartesian liftings in $\mathcal{L}$ (compare definition 2.7.4), but it is easier to use propositional variables as we are used to in modal logic.

Definition 2.8.3 (coalgebraic logic with propositional variables). Let $\mathbf{P}$ a class of propositional variables, and P be a sub*set* of $\mathbf{P}$. The coalgebraic logic $\mathcal{CL}_{P,\Omega}$ for Ω with variables from P, is given as in definition C.0.1 plus a clause for propositional variables:

$$p \in P \quad \Longrightarrow \quad p \in \mathcal{CL}_{P,\Omega}.$$

$\mathcal{CL}_{\mathbf{P},\Omega}$ is $\bigcup\{\mathcal{CL}_{P,\Omega} : P \subset \mathbf{P}, P \text{ a set}\}$. A model is given by a $\mathcal{P}P \times \Omega$-coalgebra, i.e., an Ω-coalgebra A and a valuation $\gamma : UA \to \mathcal{P}P$. The semantics $\models_{P,\Omega}$ is given as in definition C.0.2 plus

$$p \in \gamma(x) \quad \Longrightarrow \quad x \models_{P,\Omega} p.$$

For $\varphi \in \mathcal{CL}_{\mathbf{P},\Omega}$ and $x \in UA$, we define $A, x \models_{\mathbf{P},\Omega} \varphi$ iff for some $P \subset \mathbf{P}$ such that $\varphi \in \mathcal{CL}_{P,\Omega}$ it holds $\forall \gamma : UA \to \mathcal{P}P \,.\, A, \gamma, x \models_{P,\Omega} \varphi$.

It is not difficult to see that $\mathcal{CL}_{\mathbf{P},\Omega}$ meets the requirements of a modal logic for coalgebras as in definition 2.7.4. To ensure that it is rule-expressive we require the functors $\mathcal{P}P \times \Omega$ to be uniform. In [87], *uniform* functors are defined and it is proven that for uniform functors Ω the logic $\mathcal{CL}_\Omega$ has characterising formulas. This can be used to deduce from theorem 2.7.17:

Theorem 2.8.4 (co-quasivariety theorem). *Let Ω be a standard and weak-pullback preserving functor on* **Set** *such that $\mathcal{P}P \times \Omega$ is uniform for all sets P and such that $U : \mathbf{Set}_\Omega \to$* **Set** *has a right adjoint. Then a class K of Ω-coalgebras is definable by rules of $\mathcal{CL}_{\mathbf{P},\Omega}$ iff it is closed under quotients and small coproducts.*

2.9 Infinitary Modal Logic

In this section we apply the results of section 2.7 to analyse the expressive power of infinitary modal logics on Kripke frames (with degree of branching restricted to some infinite regular cardinal).

For the remainder of this section let κ be an infinite regular cardinal. We denote with $\mathbf{KF}_\kappa$ the class of Kripke frames with degree of branching smaller than κ, i.e., $\mathbf{KF}_\kappa$ is $\mathbf{Set}_{\mathcal{P}_\kappa}$. Note that $\mathcal{P}_\kappa$ is bounded (see definition 1.5.1) by κ (for $\kappa = \aleph_{\alpha+1}$ for some ordinal α, $\mathcal{P}_\kappa$ is even bounded by $\aleph_\alpha$). The proof of these facts is similar to an argument in the proof of proposition 1.3 in Barr [12].

We denote with $\mathcal{ML}_\infty$ the infinitary modal logic built from a proper class of propositional variables, the constant $\bot$, the operators $\neg, \Box$ and conjunctions $\bigwedge$ over any set of formulas. $\bigvee$ and $\Diamond$ are defined as the usual abbreviations. $\mathcal{ML}_\kappa(P)$ denotes the subset of $\mathcal{ML}_\infty$ where conjunctions are taken only over sets of cardinality not greater than κ and where propositional variables are from the set P.

The following fact on infinitary modal logic follows from Baltag's lemma in Barwise and Moss [13], 11.13.

Lemma 2.9.1. *Let κ be an infinite regular cardinal, $|P| \leq \kappa$, $A \in \mathbf{KF}_\kappa$, $\alpha : UA \to \mathcal{P}P$, $a \in A$. Then there is $\varphi^a \in \mathcal{ML}_\kappa(P)$ such that for all $B \in \mathbf{KF}$, $\beta : UB \to \mathcal{P}P$, $b \in B$ it holds that $B, \beta, b \models \varphi^a$ only if (A, α, a) and (B, β, b) are bisimilar.*

We can now show the following theorem which has appeared in a weaker form and with a different proof in [75].

Theorem 2.9.2 (definability by rules). *Let κ be an infinite regular cardinal and κ' be the cardinal of the carrier of the final $\mathcal{P}_\kappa$-coalgebra. Then $\mathcal{B} \subset \mathbf{KF}_\kappa$ is definable by rules of $\mathcal{ML}_\infty$ iff $\mathcal{B}$ is closed under subframes and disjoint unions.*

Proof. Treating $\mathcal{ML}_\infty$ as in example 2.7.9 shows that $\mathcal{ML}_\infty$ is a modal logic for $\mathcal{P}_\kappa$-coalgebras in the sense of definition 2.7.4. The objects of $\mathbb{C}$ are $\{\mathcal{P}P : P \text{ a set}\}$. The morphisms are $\iota^{-1} : \mathcal{P}Q \to \mathcal{P}P$ where $\iota : P \hookrightarrow Q$. $\mathcal{L}_{\mathcal{P}P}$ is $\mathcal{ML}_\infty(P)$. $\mathcal{ML}_\infty$ is the union of all $\mathcal{ML}_\infty(P)$, $P \in \mathbf{Set}$. That $\mathcal{ML}_\infty$ is rule-expressive follows from the lemma. Now apply theorem 2.7.16. $\Box$

As for algebras, in the case of definability by formulas (equations) we can restrict the cardinality of the set of colours (variables) and the cardinality of the disjunctions (number of equations). (A less categorical proof of the following theorem in the style of [75] is available as [74].)

Theorem 2.9.3 (definability by formulas). *Let κ be an infinite regular cardinal, P a set with $|P| \leq \kappa$ and $|\mathcal{P}P| \geq \kappa$ and κ' be the least regular cardinal which is larger than the carrier of the cofree $\mathcal{P}_\kappa$-coalgebra over $\mathcal{P}P$. Then $\mathcal{B} \subset \mathbf{KF}_\kappa$ is definable by formulas of $\mathcal{ML}_{\kappa'}(P)$ iff $\mathcal{B}$ is closed under subframes, disjoint unions and p-morphic images*[23]*.*

[23] *p-morphism* is the traditional name in the literature on modal logic for morphisms in $\mathbf{Set}_\mathcal{P}$.

Proof. To apply theorem 2.7.17 we need to show that $\mathcal{ML}_{\kappa'}(P)$ is expressive. According to theorem 2.7.18 we have to show that every subcoalgebra A of the cofree coalgebra $F\mathcal{P}P$ can be expressed by a $\varphi \in \mathcal{ML}_{\kappa'}(P)$. For all $a \in A$ let φ^a be as in the lemma. Let $\varphi = \bigvee_{a \in A} \varphi^a$. Since $\kappa' \geq \kappa$, $\varphi \in \mathcal{ML}_{\kappa'}(P)$. □

Obviously, this theorem implies that a class of frames in $\mathbf{KF}_\kappa$ is $\mathcal{ML}_\infty$-definable iff it is closed under subframes, disjoint unions and p-morphic images. But it also shows why infinitary logic is needed: First, we need enough propositional variables P such that every generated subframe can be embedded into $F\mathcal{P}P$ (see the proof of theorem 2.7.18). Second, we need infinitary disjunctions of the cardinality of $F\mathcal{P}P$ to achieve expressiveness.

It seems that in order to know more about the size of $F\mathcal{P}P$ we have to make further set-theoretic assumptions. For example, it follows from Martin's axiom (or the stronger continuum hypothesis) that 2^ω is regular (see Martin and Solovay [82]). A calculation similar to the one of the proof of proposition 1.3 in Barr [12] shows that, in the case of $\kappa = \omega$, we can then choose $P = \omega$ and $\kappa' = 2^\omega$. Similarly, assuming the general continuum hypothesis, in the case of $\kappa = \aleph_{\alpha+1}$ for some ordinal α, we can choose $P = \aleph_\alpha$ and $\kappa' = \aleph_{\alpha+1}$.

2.10 Related and Future Work

For bounded signature functors on **Set**, it was shown in Rutten [108], theorem 15.1, that covarieties are determined by a subcoalgebra of a *cofree coalgebra* (see also example 1.3.11). Therefore, in order to prove a covariety theorem, it remained to find a logic for coalgebras allowing to specify subcoalgebras of cofree coalgebras. But first, a different problem was solved, namely to characterise those subcategories which are determined by a subcoalgebra of the *final coalgebra*.

Roşu [98] (also [100, 101]) uses equational logic, but this result only holds for a restricted class of signature functors (those called algebraic in chapter 1.7). He also proves a generalisation to abstract categories[24], leaving open, however, what logical systems could take the place of a logic for coalgebras. Moreover, his characterisation results are not purely algebraic since the characterisation of definable classes of coalgebras involves closure under 'representative inclusions'[25], a notion which depends on the logic used to instantiate the framework.

In order to compare Roşu's theorem ([98], theorem 4.15)[26] with the theorem of Gumm and Schröder ([41], proposition 6.1)[27] and the work of this chapter, we show a definability theorem for modal logics with formulas involving no propositional variables (i.e. $C = 1$ in our terminology).

Theorem 2.10.1. *Let* $\mathbf{Set}_\Omega$ *be a category of coalgebras with final coalgebra* Z *and let* $\mathcal{L}$ *be a modal logic for coalgebras without propositional variables allowing to define all subcoalgebras of* Z*. Then* $\mathcal{B} \subset \mathbf{Set}_\Omega$ *is definable by formulas in* $\mathcal{L}$ *iff* $\mathcal{B}$ *is closed under domains of morphisms, quotients, and coproducts.*

Proof. " $\Rightarrow$ ": Closure under quotients and coproducts holds for all classes definable by modal logic (section 2.4.3). Closure under domains of morphisms follows from the fact that formulas without propositional variables correspond to subcoalgebras of the *final* coalgebra.
" $\Leftarrow$ ": Let $\mathcal{B}$ be closed under domains of morphisms, quotients, and coproducts. Every coalgebra in $\mathcal{B}$ has an image in Z and closure under quotients and coproducts implies that the union of these images gives rise to a subcoalgebra $i : S \hookrightarrow Z$ which is in $\mathcal{B}$. Let Φ be the set of formulas in $\mathcal{L}$ defining S. We show that Φ defines $\mathcal{B}$. Assume $B \models \Phi$, i.e. the unique morphism $B \to Z$ factors through i. Then $B \in \mathcal{B}$ follows by closure under domains of morphisms. □

[24] This result assumes that the category under consideration is equipped with a 'weak inclusion system', a requirement essentially to the same effect as the existence of a factorisation system.

[25] Given an arbitrary logic with semantics in terms of objects of a category $\mathcal{C}$, Roşu defines that $\mathcal{B} \subset \mathcal{C}$ is closed under *representative inclusions* iff for all $B \in \mathcal{B}$ and all $f : B \hookrightarrow C$ in $\mathcal{C}$ such that B and C satisfy the same formulas it follows $C \in \mathcal{B}$.

[26] Roşu's theorem considers logics (for objects in an abstract category) whose formulas are invariant under domains of morphisms, quotients and coproducts. It then states that a class is definable iff it is closed under domains of morphisms, quotients, coproducts, and representative inclusions.

[27] Gumm and Schröder's theorem considers logics (for coalgebras over set) which characterise each element of a coalgebra up to bisimulation. It then states that a class is definable iff it is closed under total bisimulations, subcoalgebras, and coproducts.

Remark. Moss' coalgebraic logic [87] (see also appendix C) is an example of a logic satisfying the requirements of the theorem (under some mild assumptions on the signature). For another example, consider the case of $\Omega X = \mathcal{P}_\omega(L \times X)$. Then Ω-coalgebras are finitely branching labelled transition systems and Hennessy-Milner logic with countable conjunctions and disjunctions up to cardinality 2^ω is expressive.[28]

Remark. Following the approach of sections 2.4 and 2.5, this theorem can easily be generalised to abstract categories.

Remark. This theorem essentially specialises Roşu's characterisation theorem ([98], theorem 4.15) to logics which are expressive in the sense that they can define all subcoalgebras of the final coalgebra. This allows to eliminate the requirement of closure under representative inclusions. The requirement of closure under representative inclusions may, however, still be useful if the logic under consideration is not expressive enough.

Remark. Since closure under domains of morphisms and quotients is equivalent to closure under bisimulations, this theorem is essentially the characterisation theorem by Gumm and Schröder ([41], proposition 6.1). Gumm and Schröder also mention that Moss' coalgebraic logic is a logic satisfying the requirements of the theorem.

We summarise the discussion above:

- Both the characterisation results in Gumm and Schröder [41] and Roşu [98] deal with the case—in the terminology of modal logic—in which there are no propositional variables. This case is of special importance in computer science because the typical applications of modal logic only involve propositional constants and no propositional variables (or, to put it differently, in computer science usually Kripke models and not Kripke frames are used as semantic structures).

- In the case that the logic under consideration is expressive (in the sense that every subcoalgebra of the final coalgebra is definable)

  - closure under 'representative inclusions' can be omitted from Roşu's characterisation theorem,
  - Roşu's theorem becomes equivalent to (the appropriate generalsations of) the theorem by Gumm and Schröder [41] and of theorem 2.10.1

Let us go back to the quest for logics for co-Birkhoff theorems. The first suggestion regarding a logic for coalgebras involving colours is due to Gumm [45]. Let $\mathbf{Set}_\Omega$ be a category of coalgebras such that $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ has a right adjoint F. Then elements $\varphi \in FC$ are called coequations and satisfaction is defined as follows:

$$A \models \varphi \quad \Leftrightarrow \quad \varphi \notin \gamma(UA) \quad \text{for all } \gamma : A \to FC$$

[28] Countable conjunctions to characterise each element of a given carrier and a disjunction to discribe the union of all these elements. The impact of finiteness constraints on expressiveness is left for future research.

Intuitively, a coequation φ specifies a behaviour that has to be avoided for any possible colouring γ. Gumm [45] shows that coequations give rise to a covariety theorem.

Our notion of a modal logic for coalgebras 2.7 is general enough to comprise coequations: Just map any formula $\varphi \in FC$ to the largest subcoalgebra $i : S \hookrightarrow FC$ not containing φ.

Since satisfaction of equations and implications can be characterised as injectivity (section 2.3), it is no surprise that the idea of using the dual notion of projectivity to define satisfaction for coalgebras was discovered independently by several people. Roşu [99] has an approach based on dualising work by Andreka and Nemeti (see e.g. [6]). Hughes [56] announces an abstract co-Birkhoff theorem and the recent Awodey and Hughes [8] is based as our work on Banaschewski and Herrlich [11]. The work of this chapter is distinguished from these approaches by offering modal logic as a dual to equational logic and by exploiting this to prove new theorems about the expressive power of modal logics.

Another recent work on the dual of Birkhoff's theorem is Goldblatt [39]. This work goes back to the setting of Roşu [98] in considering equational logic for mulitplicative signature functors. Goldblatt introduces boolean combinations of equations as a logic for coalgebras and shows that a class is definable iff it is closed under domains of morphisms, quotients, coproducts, and ultrafilter enlargements. Though the theorem only applies to a restricted class of signatures, it is the first truly algebraic characterisation of the expressive power of a *finitary* logic for coalgebras.

We would like to conclude our discussion of related work by mentioning that recently an important further step in the work on modal logic and coalgebra has been taken by Dirk Pattinson [90]. Up to now, the situation has been the following. On the one hand, we have Moss' coalgebraic logic [87] providing a logic for coalgebras depending in a generic way[29] on the signature. Coalgebraic logic shares many features with classical modal logic, but does lack the intuitive modal operators. On the other hand, there were approaches for restricted classes of signatures based on ad hoc translations of coalgebras to Kripke models ([73, 102, 103, 104, 66, 67]). Pattinson's work now shows a way to derive modal logics from signatures in a generic way and keeping the modal operators.

Concerning future work, we only want to mention two points here. First, applications of our framework to coalgebras over other base categories than sets should be worked out. Second, we want to find abstract principles (depending in a generic way on the signature) to characterise the expressive power of *finitary* logics for coalgebras.

[29]Syntax and semantics of coalgebraic logic are given uniformly for all weak-pullback preserving functors.

Part II

Applications of Coalgebras

Chapter 3

Modal Logic and Coalgebra: A Case Study

This chapter is a slightly revised version of [77].

3.1 Introduction

Coalgebras have been used in Reichel [93] and Jacobs [62] to formalise the notion of classes and objects in object-oriented programming. As for algebras, they use equational logic to specify coalgebras (i.e., classes and objects). An account of the connection between equational specifications and coalgebras is given by Hensel and Reichel [53] and Jacobs [61]. In this chapter we study the use of modal logic for the specification of the kind of coalgebras arising in the work of Reichel and Jacobs.

Apart from the natural relation of coalgebras and modal logic discussed in the first part of this thesis, there is another reason for the use of modal logic which is due to coalgebras being used here to describe classes. Roughly speaking, given a coalgebra $(S, f : S \to \Omega S)$ (S a set (of 'states'), Ω a functor, f a function) the state of an object is represented by an element $s \in S$. Now, looking for a logic to specify methods we should respect the idea of encapsulation: We do not want to talk about states, which are supposed to be non-observable, but only about observable behaviour. Modal logic is an obvious choice: Formulas of modal logic are evaluated in states but generally do not refer explicitly to specific states. Compared to equational logic a conceptual advantage is that equations between states can be avoided.

We will discuss two approaches to use the ideas of modal logic. First, given a certain kind of functor, find a translation of the corresponding coalgebras to Kripke models. Then apply results of modal logic to the Kripke models and transfer them back to coalgebras. This has the advantage that the well-developed machinery of modal logic can easily be used but the drawback that the translation does not generalise to arbitrary functors. Therefore, the second approach is to use the coalgebraic logic of Moss [87]. This logic has the advantage that its syntax is derived from the functor itself and does not depend on a non-canonical detour via Kripke models.

A third approach, due to Martin Rößiger [105], may be viewed as an intermediate one. By analysing functors as syntax trees, he manages to give a systematic description of modal logics for a larger class of functors than it is done in this chapter. On the other side, as an advantage to Moss' approach, the logics still contain the intuitive modal operators. Comparing to our work, one should note that Rößiger's results could also be obtained via a translation of coalgebras into Kripke models, albeit a more complicated one (see Rößiger [104]).

The chapter is organised as follows. Section 2 reviews the essentials on coalgebras as models for classes. Section 3 introduces our modal language for coalgebras and relates modally definable classes of coalgebras to final coalgebras. Section 4 shows that the expressive power of the logic allows to define elements of coalgebras up to bisimulation and then a complete calculus for the logic is given. We also comment on the relation of the canonical model for a modal logic and the final coalgebra. Section 5 shows by examples that the logic allows for natural proofs of properties of programs. Section 6 relates our modal language to Moss' coalgebraic logic. Section 7 discusses the approach of this chapter in view of object oriented programming.

3.2 Coalgebras and OO

We show how coalgebras may be used to describe objects and classes, and introduce the examples that are used in this chapter. We mainly follow Jacobs [62].

We only consider coalgebras in the category $\mathbf{Set}_\Omega$, i.e., a coalgebra is a pair $(S, f : S \to \Omega S)$ where S a set, Ω a functor on **Set** and f a function. Coalgebras can be used to describe classes and objects. The functor Ω specifies the type of the methods, f the effect of the methods. The state of an object is represented by an element $s \in S$, $f(s)$ describing the results of the methods when sent to s.

The functors we consider are of the form

$$\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \ldots (B_n + C_n \times S)^{A_n}.$$

How this describes the number and type of the methods of the class will become clear in the following example. Let us consider a **one-cell buffer** with two operations *store* and *read*. *store* is supposed to put an element in the buffer, *read* should output the current element or yield an error message if the buffer is empty. Writing A for the set of elements that the buffer may contain and 1 for the one-element set containing the error message, the signature becomes

$$\mathit{store} : S \times A \to S, \mathit{read} : S \to 1 + A \times S.$$

Using the isomorphism $(S \times A \to S) \simeq (S \to S^A)$ we can write *store* and *read* as one function

$$\langle \mathit{store}, \mathit{read} \rangle : S \to S^A \times (1 + A \times S).$$

That is, the functor Ω in our example is given by

$$\Omega(S) = S^A \times (1 + A \times S).$$

The second example we will consider is a **LIFO-queue** with two operations $\mathit{in} : S \times A \to S$ and $\mathit{out} : S \to 1 + A \times S$ where again A denotes the set of possible elements to be stored in the queue and 1 is the set containing the error message. Both examples have the same functor. That their behaviour is different will be expressed by the modal logic presented in section 3.3.

One of the advantages of viewing transition systems as coalgebras is that, once the functor is given, there is a canonical notion of bisimulation. This is due to the fact that functors are not only defined on sets but also on functions: Let $\Omega(S) = (B_1+C_1\times S)^{A_1}\times \dots (B_n+C_n\times S)^{A_n}$ and $h : S \to S'$ be a function, then $\Omega h : \Omega S \to \Omega S'$ takes a tuple of functions $(g_1, \dots g_n), g_i \in A_i \to B_i + C_i \times S$ to a tuple of functions $(g'_1, \dots g'_n), g'_i \in A_i \to B_i + C_i \times S'$, where the g'_i are defined as follows (a, b, c, t are in A_i, B_i, C_i, S, respectively):

$$g'_i(a) = \begin{cases} b & \text{if } g_i(a) = b \\ (c, h(t)) & \text{if } g_i(a) = (c, t) \end{cases}$$

Now, we can define a *homomorphism* (or *morphism* for short) between two coalgebras $(S, f), (S', f') \in \mathbf{Set}_\Omega$ to be a function $h : S \to S'$ s.t. $f' \circ h = \Omega h \circ f$. It is not difficult to show that this definition of a morphism of coalgebras corresponds to the usual definition of a functional bisimulation. A *bisimulation* between two coalgebras $(S, f), (T, g)$ is a relation $R \subset S \times T$ such that there exists a function $r : R \to \Omega R$ together with two coalgebra morphisms $(R, r) \to (S, f)$ and $(R, r) \to (T, g)$. R is a *total bisimulation* iff these two coalgebra morphisms are surjective. We write $(S, s) \simeq^\Omega (T, t)$ iff there is a bisimulation between $(S, f), (T, g)$ containing (s, t).

To simplify notation we assume the sets B_i, C_i to be disjoint. Also, we will write S for (S, f) if there is no danger of confusion.

3.3 A Specification Language for Coalgebras

First, we define the modal language that we will use to specify coalgebras, we give a semantics and we show—using the examples from section 3.2—that it allows for an intuitive formalisation of properties. We also show that modal logic avoids equations between states in a natural way. Second, it is shown that formulas are invariant under morphisms and modally definable classes of coalgebras are related to final coalgebras.

3.3.1 Language, Semantics, Examples

We first need a modal language. The set of propositions and the set of modal operators are determined by the functor $\Omega(S) = (B_1+C_1\times S)^{A_1}\times \dots (B_n+C_n\times S)^{A_n}$ in the following way. For each component $(B_i + C_i \times S)^{A_i}$ and each $a \in A_i$ there is a modal operator $[i, a]$. And for each $d \in B_i + C_i$ we have an atomic proposition $(i, a) = d$ (read as "output of message i with argument value a is d"). In the case that A contains only one element we prefer to write simply $[i]$ and $i = d$. This gives a modal language $\mathcal{L}^\Omega$.

As an illustration consider the example of the **one-cell buffer** of section 2. (Recall that $n = 2$, $A_1 = A$ and A_2 a one element set.) Suppose we want to specify that a *store* into the empty buffer stores indeed, that a *store* into a full buffer has no effect and that a *read* empties the buffer. Then—writing $[\mathit{store}(a)]$ for $[1, a]$, $[\mathit{read}]$ for $[2]$, $\mathit{read} = d$ for $2 = d$, and *error* for the element of 1—we can formalise the conditions above as follows.

$$
\begin{gathered}
read = error \to [store(a)]read = a \\
read = a \to [store(b)]read = a \\
[read]read = error
\end{gathered}
$$

Note that the above expressions are not strictly speaking formulas of our modal language. They are axiom schemes that yield formulas for all $a, b \in A$.

Having gained some intuition, here is the definition of the language and its semantics.

Definition 3.3.1 (The modal language $\mathcal{L}^\Omega$). Let Ω be a functor on **Set** of the form $\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \ldots (B_n + C_n \times S)^{A_n}$. Then the set of atomic propositions $\mathbf{P}$ for Ω consists of propositions $(i, a) = d$ for all $1 \leq i \leq n, a \in A_i, d \in B_i + C_i$. The modal language obtained from $\mathbf{P}$ by adding the constant $\perp$, boolean connectives and modal operators $[i, a]$ for all $1 \leq i \leq n, a \in A_i$ is called $\mathcal{L}^\Omega$. $\langle i, a\rangle\varphi$ is an abbreviation for $\neg[i, a]\neg\varphi$.

Note that given the modal language we automatically have a semantics in terms of Kripke models, see appendix B. Here we give a semantics of the language in terms of coalgebras. The two semantics are related in section 3.4.1.

Definition 3.3.2 (Semantics $\models^\Omega$ of $\mathcal{L}^\Omega$). Let (S, f) be a Ω-coalgebra, $s \in S$, φ a formula of $\mathcal{L}^\Omega$, and $\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \ldots (B_n + C_n \times S)^{A_n}$. The semantics for boolean connectives is as usual. For propositions and modal operators:

$$
\begin{array}{lll}
s \models^\Omega (i, a) = d & \text{iff} & (\pi_i \circ f)(s)(a) = d \text{ or} \\
 & & \text{there is } t \in S \text{ s.t. } (\pi_i \circ f)(s)(a) = (d, t) \\
s \models^\Omega [i, a]\varphi & \text{iff} & \text{for all } c \in C_i, t \in S \ (\pi_i \circ f)(s)(a) = (c, t) \ \Rightarrow \ t \models^\Omega \varphi
\end{array}
$$

Note that the "or" on the right-hand side of the first clause corresponds to the "+" in $(B_i + C_i \times S)^{A_i}$.

The satisfaction relation $\models^\Omega$ is always relative to a specific coalgebra (S, f). If we want to emphasise this, we write $(S, f), s \models^\Omega \varphi$ (or sometimes simply $S, s \models^\Omega \varphi$). For a theory of (S, f) in state s we write $\mathsf{Th}^\Omega(S, s) = \{\varphi : S, s \models^\Omega \varphi\}$.

Next, we want to show how to specify and prove properties of *newly created objects*. The natural way to do this in our approach is to add a predicate *New* to the language having the newly created states as its extension. Such a predicate may easily be expressed in the coalgebraic setting by considering the functor $\mathrm{Bool}\times\Omega(S)$ instead of $\Omega(S)$, where Bool denotes the set of truth values $\{true, false\}$. In other words, the characteristic function $S \to \mathrm{Bool}$ of *New* is added to the description of the class. This is seen to be compatible with the functors being of the form

$$\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \ldots (B_n + C_n \times S)^{A_n}$$

since we may choose A_1 to be a one-element set, C_1 to be empty, and $B_1 = \mathrm{Bool}$. In the corresponding language $\mathcal{L}^\Omega$ we write *New* to denote the atomic proposition $1 = true$. For an example consider the LIFO-queue below.

Before analysing more of the properties of this logic, we want to emphasise that modal logic gives us the appropriate language when we are not interested in particular states but only in states up to bisimulation. To make this clear let us specify the **LIFO-queue** from section 3.2. We first require

$$[in(a)]out = a$$
$$New \to out = error$$

meaning that *out* yields a after input of a and that a newly created queue is empty. But something still lacks. How can we express that doing an *out* after an *in* gives us the "same" queue as before? Of course we do not want to say that we really get the same queue. Since states are not observable what should be said is: doing *out* after *in* gives us a queue that has the same *behaviour* as the queue before. This can be expressed in our formalism by writing down the following axiom scheme:[1]

$$[in(a)][out]\varphi \leftrightarrow \varphi$$

Note that *specifying the above property with equational logic forces us to use not only equations between attributes but between states* (cf. the discussion in [62]). *Modal logic avoids the direct access to states in a natural way.* That the above axiom scheme expresses indeed the intended constraint on the behaviours is implied by theorem 3.4.1.

Last, we want to make the connection to Rößiger [105]. In his remark 2.5 it is shown that for the functors considered here the two logics coincide. In the LIFO-example, disregarding the *New* (i.e. taking as functor $\Omega(S) = S^A \times (1 + A \times S)$), the remaining two axioms from above become

$$[\times_1 a](\times_2 +_2 \times_1)a,$$
$$[\times_1 a][\times_2 +_2 \times_2]\varphi \leftrightarrow \varphi.$$

3.3.2 Some Properties of the Logic

As to be expected, validity of formulas is invariant under morphisms and bisimulations:

Proposition 3.3.3. *Given two Ω-coalgebras and a morphism $h : (S, f) \to (T, g)$ we have:*

$$(S, f), s \models^{\Omega} \varphi \iff (T, g), h(s) \models^{\Omega} \varphi$$

Proof. The proof is the usual induction on the structure of the formulas. Let us look at two cases. Let $a \in A_i, c \in C_i$. First, φ is $(i, a) = c$: $S, s \models^{\Omega} (i, a) = c$ iff $\exists s' \in S : (\pi_i \circ f)(s)(a) = (c, s')$ iff $\exists s' \in S : (\pi_i \circ \Omega h \circ f)(s)(a) = (c, h(s'))$ iff $\exists t' \in T : (\pi_i \circ g \circ h)(s)(a) = (c, t')$ iff $T, h(s) \models^{\Omega} (i, a) = c$.

Second, φ is $[i, a]\psi$, $s' \in S, t' \in T$. Then $S, s \models^{\Omega} [i, a]\psi$ iff $(\pi_i \circ f)(s)(a) = (c, s') \Rightarrow S, s' \models^{\Omega} \psi$ iff $(\pi_i \circ f)(s)(a) = (c, s') \Rightarrow T, h(s') \models^{\Omega} \psi$ iff $(\pi_i \circ \Omega h \circ f)(s)(a) = (c, h(s')) \Rightarrow T, h(s') \models^{\Omega} \psi$ iff $(\pi_i \circ g \circ h)(s)(a) = (c, t') \Rightarrow T, t' \models^{\Omega} \psi$ iff $T, h(s) \models^{\Omega} \psi$. ☐

[1] The scheme denotes the set of all its instances with formulas of $\mathcal{L}^{\Omega}$ substituted for φ (and elements of A substituted for a).

Then as a corollary of the above proposition we get that if $(S, s) \simeq^{\Omega} (T, t)$ then also $\mathsf{Th}^{\Omega}(S, s) = \mathsf{Th}^{\Omega}(T, t)$.

Next, let us take a look at modally definable classes of coalgebras. Let Ω be a functor as described above and $\Phi \subset \mathcal{L}^{\Omega}$ a set of formulas. Consider the class

$$\mathrm{Mod}(\Phi) = \{(S, f) \in \mathbf{Set}_{\Omega} : \text{ for all } s \in S, \varphi \in \Phi : (S, f), s \models^{\Omega} \varphi\}$$

of (coalgebra-) models of Φ. Obviously, $\mathrm{Mod}(\Phi)$ gives rise to a full subcategory of $\mathbf{Set}_{\Omega}$. And it is closely related to the coalgebra T_{Φ} that is defined as the largest subcoalgebra of the final coalgebra T whose carrier is contained in $\{t \in T : t \models^{\Omega} \Phi\}$:[2] As shown by the next theorem, specifications in modal logic work by defining subcoalgebras of the final coalgebra.

Theorem 3.3.4. *Let $\Phi \subset \mathcal{L}^{\Omega}$. Then $S \in \mathrm{Mod}(\Phi)$ iff there is a (necessarily unique) morphism $S \to T_{\Phi}$.*

Proof. "only if:" Let T be the final coalgebra in $\mathbf{Set}_{\Omega}$ (that exists and is explicitly described in [62]). There is a unique morphism $h : S \to T$. Since $S \models \Phi$ the image of S is contained in the carrier of T_{Φ}, therefore h factors uniquely through T_{Φ},[3] giving rise to a unique morphism $S \to T_{\Phi}$.
"if:" Immediate by definition of T_{Φ} and proposition 3.3.3. ☐

A class of models K is said to be *closed under bisimulations* whenever from $S \in K$ and R a total bisimulation between S, S' it follows $S' \in K$. It is called a covariety when it is closed under images, subcoalgebras and disjoint unions. We follow Gumm and Schröder [41] and call covarieties that are closed under bisimulations *complete covarieties*. With this definition we get as an immediate corollary:

Corollary 3.3.5. *Let $\Phi \subset \mathcal{L}^{\Omega}$. Then $\mathrm{Mod}(\Phi)$ is a complete covariety.*

Proof. That $\mathrm{Mod}(\Phi)$ is a covariety follows from the theorem above and Rutten [108], theorem 15.1. Closure under bisimulations is an obvious consequence of proposition 3.3.3.

☐

Gumm and Schröder [41] analyse logics where the converse (i.e. any complete covariety is definable by a set of formulas) also holds. See the paragraphs following theorem 3.4.1 for a discussion on how this relates to our case.

3.4 Transferring Results from Modal Logic

One reason why it is nice to use modal logic as a specification language is that the theory of modal logics gives us a lot of tools to design appropriate logics, to prove results about these logics, and to work with the logics (interactive theorem provers, model checkers). To get access to this area we give a translation of coalgebras into Kripke models and show that the coalgebraic semantics of $\mathcal{L}^{\Omega}$ coincides with Kripke semantics. This translation may then

[2] Existence of T_{Φ} follows from Theorem 6.4 in [108]
[3] Theorem 7.1 in [108]

be used to transfer results from modal logic. In this chapter it is used to show that $\mathcal{L}^\Omega$ allows us to define behaviours up to bisimulation and to give a complete axiomatisation of $\mathcal{L}^\Omega$. In connection with the completeness proof we also discuss the relationship of the canonical model for $\mathcal{L}^\Omega$ and the final Ω-coalgebra.

3.4.1 The Translation

The logic $\mathcal{L}^\Omega$ has been given a semantics in terms of coalgebras. On the other hand, like any modal logic in the style of appendix B, it has also a semantics w.r.t. Kripke models. The connection between both is given by a translation of the category of Ω-coalgebras into the category of Kripke models for $\mathcal{L}^\Omega$. It is a full and faithful embedding preserving and reflecting all interesting logical properties.

Let Ω be a functor on **Set** of the form $\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \dots (B_n + C_n \times S)^{A_n}$. The category K^Ω of Ω-Kripke models is given as follows. Let $\mathbf{P}$ be the set of atomic propositions for the functor Ω (see definition 3.3.1). Then a Ω-Kripke model is a Kripke model $(W, \mathcal{R}, V)$ where W is a set, $\mathcal{R}$ is a family $(R_{i,a})_{1 \le i \le n, a \in A_i}$ of relations and $V : \mathbf{P} \to \mathcal{P}(W)$ a mapping from propositions into the powerset of W. Since the methods are supposed to be functions we have the following restrictions on the relations and on the extensions of the propositions. For all $w \in W, 1 \le i \le n, a \in A_i, b \in B_i, c \in C_i$ it holds:

1. $w \in V((i,a) = b) \;\Rightarrow\; w$ has no $R_{i,a}$-successor,
2. $w \in V((i,a) = c) \;\Rightarrow\; w$ has exactly one $R_{i,a}$-successor,
3. in w holds exactly one proposition of $\{(i,a) = d : d \in B_i + C_i\}$.

The morphisms in the category K^Ω are the p-morphisms (see appendix B). Note that K^Ω is a full subcategory of the category of Kripke models for $\mathcal{L}^\Omega$. In the following we will see that $\mathbf{Set}_\Omega$ and K^Ω are equivalent (even isomorphic).

We define two functors $sk : \mathbf{Set}_\Omega \to K^\Omega$ and $ks : K^\Omega \to \mathbf{Set}_\Omega$ which will be isomorphisms that preserve in particular logical equivalence and bisimilarity.

$sk : \mathbf{Set}_\Omega \to K^\Omega$ maps every Ω-coalgebra (S, f) to the Ω-Kripke model $(S, \mathcal{R}, V)$ where

$$
\begin{aligned}
(s,t) \in R_{i,a} \;&\Leftrightarrow\; \exists c \in C_i : (\pi_i \circ f)(s)(a) = (c,t) \\
s \in V((i,a) = b) \;&\Leftrightarrow\; (\pi_i \circ f)(s)(a) = b \\
s \in V((i,a) = c) \;&\Leftrightarrow\; \exists t \in S : (\pi_i \circ f)(s)(a) = (c,t)
\end{aligned}
$$

$ks : K^\Omega \to \mathbf{Set}_\Omega$ maps every Ω-Kripke model $(S, \mathcal{R}, V)$ to the Ω-coalgebra (S, f) where

$$
\begin{aligned}
(\pi_i \circ f)(s)(a) = b \;&\Leftrightarrow\; s \in V((i,a) = b) \\
(\pi_i \circ f)(s)(a) = (c,t) \;&\Leftrightarrow\; s \in V((i,a) = c) \text{ and } (s,t) \in R_{i,a}
\end{aligned}
$$

On morphisms sk and ks are the identity.

sk and ks are isomorphisms that preserve all interesting properties. We need in particular:

$$
\begin{array}{ll}
\text{(a)} & S, s \models^\Omega \varphi \;\Leftrightarrow\; sk(S), s \models \varphi \\
\text{(b)} & (M,s) \simeq (N,t) \;\Rightarrow\; (ks(M), s) \simeq^\Omega (ks(N), t) \\
\text{(c)} & ks \circ sk = id_{\mathbf{Set}_\Omega}
\end{array}
$$

3.4.2 Logical equivalence implies bisimilarity

Logical equivalence implies bisimilarity. The proof uses the well-known fact that this property holds for image-finite Kripke models.

Theorem 3.4.1. $\mathsf{Th}^\Omega(S,s) = \mathsf{Th}^\Omega(T,t) \;\Rightarrow\; (S,s) \simeq^\Omega (T,t)$.

Proof. Suppose $\mathsf{Th}^\Omega(S,s) = \mathsf{Th}^\Omega(T,t)$. Then $\mathsf{Th}(sk(S),s) = \mathsf{Th}(sk(T),t)$ by (a). Since $sk(S)$ and $sk(T)$ are so-called *image-finite* Kripke models, logical equivalence implies bisimilarity, that is, $(sk(S),s) \simeq (sk(T),t)$. By (b) and (c) we get $(S,s) \simeq^\Omega (T,t)$. □

The argument remains valid if we would allow the functors Ω to be built from the finite powerset functor.[4] More generally, the above argument is possible whenever the translation into Kripke models yields a class of models having the so-called Hennessy-Milner property. For detailed discussions of this concept see Goldblatt [38] and Hollenberg [55].

Note also that our expressiveness result is not strong enough to get the converse of corollary 3.3.5. To achieve this we would either need a "global" version of the above theorem saying $\mathsf{Th}^\Omega(S) = \mathsf{Th}^\Omega(T) \;\Rightarrow\; S \simeq^\Omega T$ or a logic with infinite conjunctions and disjunctions. See Gumm and Schröder [41] for a proof in the latter case.

3.4.3 Axiomatisations

Our next aim is to give a complete axiomatisation of the logic $\mathcal{L}^\Omega$. The strategy is to use again the translation of coalgebras into Kripke models. First we need to find the set Σ^Ω of axioms. If we then can show that $K^\Omega \models \varphi \;\Rightarrow\; \Sigma^\Omega \models \varphi$ we get a completeness result for $\models^\Omega$ from the completeness result of modal logic. Unfortunately, to find a strong enough set Σ^Ω, the restriction to finite output sets B_i, C_i is needed. We give a complete axiomatisation for this restricted case using the canonical model for $\mathcal{L}^\Omega$ and comment on the relation of the final coalgebra and the canonical model. Then we show why the restriction to finite output sets is needed, and sketch some possible ways to overcome this.

Let again Ω be given by $\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \ldots (B_n + C_n \times S)^{A_n}$. To find Σ^Ω recall the definition of K^Ω in section 3.4.1. The restrictions on the models expressed there (see (i)-(iii)) have now to be formulated using modal formulas. Recall that $[i,a]\bot$ expresses that a world has no successor and that $[i,a]\varphi \to \langle i,a\rangle\varphi$ expresses that a world has at least one successor, cf. [36]. Furthermore $\langle i,a\rangle\varphi \to [i,a]\varphi$ expresses (on frames and also on the canonical model, see the proof of the theorem below) that every world has at most one successor. We therefore get the following axioms ((Ax1) corresponding to (i) and (Ax2),(Ax3) to (ii)):

(Ax1) $\quad (i,a) = b \to [i,a]\bot$ for all $b \in B_i$
(Ax2) $\quad (i,a) = c \to ([i,a]\varphi \to \langle i,a\rangle\varphi)$ for all $c \in C_i, \varphi \in \mathcal{L}^\Omega$
(Ax3) $\quad \langle i,a\rangle\varphi \to [i,a]\varphi$

Next we have to express that each method yields exactly one output value. At this point (see (Ax5)) we need that all the sets B_i, C_i are finite.

[4] For example functors described by $\Omega ::= C \mid Id \mid \Omega + \Omega \mid \Omega \times \Omega \mid \Omega^C \mid \mathcal{P}_\omega$ where C a constant functor and $\mathcal{P}_\omega$ the finite covariant powerset functor.

(Ax4) $(i,a) = d \to \neg(i,a) = d'$ for all $d \neq d', d, d' \in B_i + C_i$
(Ax5) $\bigvee_{d \in B_i + C_i} (i,a) = d$

Let Σ^Ω be the set of $\mathcal{L}^\Omega$-formulas defined by the five axiom schemes above.

Theorem 3.4.2 (Completeness for $\models^\Omega$). *Let there be a functor on* **Set** $\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \ldots (B_n + C_n \times S)^{A_n}$ *with all the B_i, C_i finite sets and let φ be a $\mathcal{L}^\Omega$-formula and Γ be a set of $\mathcal{L}^\Omega$-formulas. Then $\Gamma \models^\Omega \varphi \Leftrightarrow \Gamma \cup \Sigma^\Omega \vdash \varphi$.*

Proof. " $\Leftarrow$ " is a standard correctness proof. For " $\Rightarrow$ ", using the translation of section 3.4.1, it is enough to show that $\Gamma \cup \Sigma^\Omega \nvdash \varphi$ implies the existence of a model $M \in K^\Omega : M \models \Gamma \ \& \ M \not\models \varphi$. We define $M^\Omega = (W^\Omega, \mathcal{R}^\Omega, V^\Omega)$ to be the canonical model[5] of $\Gamma \cup \Sigma^\Omega$. That is, W^Ω is the set of maximal $(\Gamma \cup \Sigma^\Omega)$-consistent sets of $\mathcal{L}^\Omega$-formulas, $wR^\Omega_{i,a}v \ \Leftrightarrow \ (\forall \psi \in \mathcal{L}^\Omega : [i,a]\psi \in w \ \Rightarrow \ \psi \in v)$, $w \in V^\Omega(p) \ \Leftrightarrow \ p \in w$. The canonical model has the property that $M^\Omega, w \models \psi \ \Leftrightarrow \ \psi \in w$. Therefore $\Gamma \cup \Sigma^\Omega \nvdash \varphi$ implies that there is $w \in W$ such that $M^\Omega, w \not\models \varphi$. Also, by construction $M^\Omega \models \Gamma$. It remains to show that $M^\Omega \in K^\Omega$. That (Ax1,2,4,5) enforce the intended conditions on M^Ω should be obvious. That (Ax3) implies that any world in M^Ω has at most one $R^\Omega_{i,a}$ successor is an easy exercise that may be found in [36]. $\square$

We have shown that the canonical model M^Ω for $\mathcal{L}^\Omega$ is in K^Ω and hence can be thought of as a Ω-coalgebra. Is M^Ω also the final coalgebra? This is indeed the case (cf. Rößiger[105], theorem 7.1): given any coalgebra S define a function $h : S \to M^\Omega$ via $f(s) = \mathsf{Th}(S, s)$. It is easy to show that h is well defined and satisfies the conditions (i) and (iii) of the definition of a p-morphism. For condition (ii) use that any world in M^Ω has at most one $R^\Omega_{i,a}$ successor. Uniqueness of h is obvious. Hence M^Ω is final in K^Ω. But it is important to note that for more general functors the canonical model is *not* the final coalgebra. The reason is that the canonical model can be understood as the disjoint union of all models quotiented by logical equivalence whereas the final coalgebra is the quotient w.r.t. bisimulation which, generally, is a finer equivalence relation. Consequently, in cases where the logic is too weak to characterise worlds up to bisimulation, the existence of a morphism into the canonical model will fail (but if it exists it is unique).

Back to the completeness result, we have shown that if B_i, C_i are finite the calculus for $\mathbf{K}_{\mathcal{L}^\Omega}$ as presented in appendix B together with Σ^Ω as axioms is a strongly complete calculus for $\mathcal{L}^\Omega$. We will call it $\mathbf{C}^\Omega$.

How can we get rid of the restriction that the sets of output values B_i, C_i are finite? Possible ideas are to use modal predicate logic or to allow infinite conjunctions. But then it could be that completeness is lost. Here, we want to make a different proposal. It seems natural to allow that methods do not yield any output unless they are forced to do so by the axioms of the specification. In other words, the fact that methods are not partial is not any more expressed automatically by the syntax but has to be specified in the axiomatic part. We then do not need (Ax5) any more. This may seem to render the task of specifying more complicated but looking at our two examples from above we see that no changes to the axioms are necessary.

[5] See e.g. Goldblatt [36] for details.

To be more explicit, things work out as follows. Define K^{Ω^-} as K^{Ω} but allowing models that have states where $\neg(i,a) = d$ holds for all $d \in B_i + C_i$. Define Σ^{Ω^-} as Σ^{Ω} but without (Ax5). We then can allow the sets B_i, C_i to be arbitrary. The proof of the theorem can be adopted almost literally. We therefore get a strongly complete calculus $\mathbf{C}^{\Omega^-}$ for $\models^{\Omega^-}$.

To summarise from a more practical point of view: Proofs that do not use (Ax5) yield formulas that are also valid for non-finite output sets. The above completeness result together with the examples (see below) indicates that the loss of (Ax5) is not important.

3.5 Proving Properties of Specifications

In this section we use the calculus for $\mathcal{L}^{\Omega}$ as given in the preceding subsection to show some example proofs. Consider the LIFO-queue example from section 2 and the following valid formulas:

(*)	$[in(a)]\neg out = error$
(**)	$[in(a_1)]\dots[in(a_k)][out]^i out = a_{k-i},\ 0 \le i < k$
(***)	$New \to [in(a_1)]\dots[in(a_k)][out]^k out = error,\ 0 \le k$

where $[out]^i$ is an abbreviation for i-times $[out]$. The meaning of (*) is that after input of a the queue is not empty. (**) says that after k inputs and $i < k$ outputs a further output yields the $(k-i)$-th input value. (***) states that starting with a newly created queue and making k inputs and then as many outputs gives back an empty queue.

We give derivations in the calculus $\mathbf{C}^{\Omega^-}$. The last column tells how the current line was derived. For example, (prop)(dist)(nec)(2) in the derivation below means that we applied rule (nec) to the formula derived in line 2, then used axiom (dist) and then applied some propositional reasoning (in that case only (mp)) to derive the actual formula.

First for (*).

(1)	$[in(a)]out = a$	axiom from the spec
(2)	$out = a \to \neg out = error$	(Ax4) from Σ^{Ω^-}
(3)	$[in(a)]out = a \to [in(a)]\neg out = error$	(prop)(dist)(nec)(2)
(4)	$[in(a)]\neg out = error$	(mp)(1,3)

For (**) let us see how things work for the instance

$$[in(a_1)][in(a_2)][in(a_3)][in(a_4)][out][out]out = a_2.$$

The derivation is as follows:

(1)	$[in(a_4)][out][out]out = a_2 \leftrightarrow [out]out = a_2$	spec
(2)	$[in(a_3)][out]out = a_2 \leftrightarrow out = a_2$	spec
(3)	$[in(a_2)]out = a_2$	spec
(4)	$[in(a_2)][in(a_3)][out]out = a_2 \leftrightarrow [in(a_2)]out = a_2$	(prop)(dist)(nec)(2)
(5)	$[in(a_2)][in(a_3)][in(a_4)][out][out]out = a_2 \leftrightarrow$ $[in(a_2)][in(a_3)][out]out = a_2$	(prop)(dist)(nec)(1)
(6)	$[in(a_2)][in(a_3)][in(a_4)][out][out]out = a_2$	(prop)(3,4,5)
(7)	$[in(a_1)][in(a_2)][in(a_3)][in(a_4)][out][out]out = a_2$	(nec)(6)

For (***) we first show a lemma for all $k \geq 0$:

$$(\mathrm{L}k) \quad [in(a_1)] \ldots [in(a_k)][out]^k \varphi \leftrightarrow \varphi$$

The case $k = 0$ is an axiom from the specification. Suppose that we have proved (Lk). We show how to derive (L$(k+1)$).

(1)	$[in(a_1)][out]\varphi \leftrightarrow \varphi$	(spec)
(2)	$[in(a_2)] \ldots [in(a_{k+1})][out]^k [out]\varphi \leftrightarrow [out]\varphi$	Lk
(3)	$[in(a_1)][in(a_2)] \ldots [in(a_{k+1})][out]^k [out]\varphi \leftrightarrow [in(a_1)][out]\varphi$	(prop)(dist)(nec)(2)
(L$(k+1)$)	$[in(a_1)][in(a_2)] \ldots [in(a_{k+1})][out]^{k+1}\varphi \leftrightarrow \varphi$	(prop)(3)(1)

It is now immediate to show (***):

(1)	$New \to out = error$	(spec)
(2)	$out = error \leftrightarrow [in(a_1)] \ldots [in(a_k)][out]^k out = error$	(Lk)
(3)	$New \to [in(a_1)] \ldots [in(a_k)][out]^k out = error$	(prop)(1)(2)

3.6 Coalgebraic Logic as a Specification Language

So far, our treatment of specifications of objects and classes using modal logic was inspired by regarding them as coalgebras. But the results we proved were obtained by viewing coalgebras as Kripke models. In this respect, the general question underlying our approach is: to what extent can coalgebras be considered as Kripke models? Unfortunately, giving a uniform translation from coalgebras to transition systems for a larger variety of functors including exponentiation and powerset does seem to yield rather complicated Kripke models.

It is therefore natural to ask for a logic that depends in a canonical way on the functor and is thus truly coalgebraic. Such a logic has recently been developed by Moss [87]. We show examples of specifications in coalgebraic logic and give a translation from $\mathcal{L}^\Omega$ into coalgebraic logic.

3.6.1 Specifications in Coalgebraic Logic

For a review of coalgebraic we refer to appendix C.

We first look at the examples. We do not consider here the axioms concerning the newly created objects. Consider the three axioms of the one-cell buffer in section 3. In coalgebraic logic we can write them as ($*$ being the error message):

$$(true^A, *) \to (\lambda a.(true^A, (a, true)), *)$$
$$(true^A, (a, true)) \to (\lambda b.(true^A, (a, true)), (a, true))$$
$$\bigvee\{(true^A, z) : z \in 1 + A \times \{(true^A, *)\}\}$$

where $true^A$ is the constant function $A \to \{true\}$.

At first sight the main difference is that we have no direct access to the single components *store* and *read* (recall that *store* corresponds to the first component, *read* to the second). This is also the reason for the infinite disjunction in the third clause.

Let us take a closer look at the first clause. The premise $(true^A, *)$ tells that *read* yields *error* and specifies nothing about the *store*. The conclusion $(\lambda a.(true^A, (a, true)), *)$ says that we are in a state where the *store* is in accordance with $\lambda a.(true^A, (a, true))$ and the *read* yields *error*. Now, some thought shows that $\lambda a.(true^A, (a, true))$ means that storing a gives a state where $(true^A, (a, true))$ holds. And this formula describes exactly those states where *read* yields a. The reader is invited to check this, paying special attention to the third clause of the definition of $\models_\Omega$.

The LIFO-example. The axioms become:

$$\bigvee_{z \in 1+A\times\{true\}}(\lambda a.(true^A, (a, true)), z)$$
$$(\bigvee_{z \in 1+A\times\{true\}}(\lambda a.(true^A, (a, \varphi)), z)) \leftrightarrow \varphi$$

Note that the infinite disjunctions are needed to express that the properties are independent from the first element in the queue. The next section shows that there is a way to give access to the single components and thereby eliminating the disjunctions from the specifying formulas and reintroducing the modal operators.

3.6.2 Translating Modal Logic into Coalgebraic Logic

Coalgebraic Logic gives a general way to get a logic for coalgebras. But one disadvantage is that it lacks the intuitive box and diamond operators of modal logic. Translating $\mathcal{L}^\Omega$ into $\mathcal{CL}_\Omega$ means to render into coalgebraic logic the modal operators.

Definition 3.6.1. (Translation T from $\mathcal{L}^\Omega$ to $\mathcal{CL}_\Omega$) The boolean operators are translated in the obvious way. For propositions and modal operators (using $a, a' \in A_i, b \in B_i, c \in C_i, \varphi \in \mathcal{L}^\Omega$):

$$T((i,a) = b) = \bigvee\{(g_1, \dots g_n) : g_j \in A_j \to B_j + C_j \times \{true\}, g_i(a) = b\}$$
$$T((i,a) = c) = \bigvee\{(g_1, \dots g_n) : g_j \in A_j \to B_j + C_j \times \{true\}, g_i(a) = (c, true)\}$$
$$\begin{aligned} T([i,a]\varphi) = \bigvee\{(g_1, \dots g_n) :\ & g_j \in A_j \to B_j + C_j \times \{true\} \text{ for all } j \neq i, \\ & g_i(a) \in B_i + C_i \times \{T(\varphi)\}, \\ & g_i(a') \in B_i + C_i \times \{true\} \text{ for all } a' \neq a\} \end{aligned}$$

The next proposition gives a characterisation of the translation of $\langle i, a\rangle$.

Proposition 3.6.2.

$$\begin{aligned} \models_\Omega T(\langle i,a\rangle\varphi) \leftrightarrow \bigvee\{(g_1, \dots g_n) : g_j \in A_j \to B_j + C_j \times \{true\} & \textit{ for all } j \neq i, \\ g_i(a) \in C_i \times \{T(\varphi)\}&, \\ g_i(a') \in B_i + C_i \times \{true\} & \textit{ for all } a' \neq a\} \end{aligned}$$

The next theorem states that specifications in the language $\mathcal{L}^\Omega$ can also be considered as specifications in coalgebraic logic.

Theorem 3.6.3. *Let Ω be a functor on the category* **Set** *of the form $\Omega(S) = (B_1 + C_1 \times S)^{A_1} \times \dots (B_n + C_n \times S)^{A_n}$, (S, f) a Ω-coalgebra and $\varphi \in \mathcal{L}^\Omega$. Then*

$$\textit{for all } s \in S \ :\ s \models^\Omega \varphi \iff s \models_\Omega T(\varphi).$$

Proof. By induction on the structure of φ.

$(i,a) = b$. "$\Rightarrow$:" Suppose $s \models^{\Omega} (i,a) = b$. We have to find w in $\Omega(\models_{\Omega})$ such that $(\Omega\pi_1)(w) = f(s)$ and $(\Omega\pi_2)(w)$ a formula of the disjunction. Define w to be a tuple $(w_1, \dots w_n)$ such that for all $1 \le j \le n$, $w_j : A_j \to B_j + C_j \times (S \times \mathcal{CL}_{\Omega})$ s. t.

$$w_j(a) = \begin{cases} b & \text{if } (\pi_j \circ f)(s)(a) = b \\ (c,(t,true)) & \text{if } (\pi_j \circ f)(s)(a) = (c,t) \end{cases}$$

"$\Leftarrow$:" Suppose there is a $w \in \Omega(\models_{\Omega})$ such that $(\Omega\pi_1)(w) = f(s)$ and $(\Omega\pi_2)(w) = g$ for a formula g of the disjunction. Then $g = (g_1, \dots g_n)$ and $g_i(a) = b$. Therefore $(\pi_i \circ f)(s)(a) = (\pi_i \circ \Omega\pi_1)(w)(a) = b$.

$(i,a) = c$. Similar argument.

$[i,a]\psi$. "$\Rightarrow$:" Choose $w = (w_1, \dots w_n)$ such that for all $1 \le j \le n, j \ne i$

$$w_j(a) = \begin{cases} b & \text{if } (\pi_j \circ f)(s)(a) = b \\ (c,(t,true)) & \text{if } (\pi_j \circ f)(s)(a) = (c,t) \end{cases} \quad \text{and}$$

$$w_i(a) = \begin{cases} b & \text{if } (\pi_i \circ f)(s)(a) = b \\ (c,(t,T(\psi))) & \text{if } (\pi_i \circ f)(s)(a) = (c,t) \end{cases}$$

As above $\Omega\pi_1(w) = f(s)$ and $\Omega\pi_2(w)$ is a formula of the disjunction. It remains to show that w is indeed in $\Omega(\models_{\Omega})$. This follows from the definition of $\models^{\Omega}$ and the induction hypothesis.

"$\Leftarrow$:" Suppose $(\pi_i \circ f)(s)(a) = (c,t)$. We have to show $t \models^{\Omega} \psi$. Let g be a formula of the disjunction $T([i,a]\psi)$ with $s \models_{\Omega} g$. It follows that there is a $w \in \Omega(\models_{\Omega})$ s.t. $\Omega\pi_1(w) = f(s), \Omega\pi_2(w) = g$. Because of $(\pi_i \circ f)(s)(a) = (c,t)$ and the definition of g we get $(\pi_i \circ \Omega\pi_1(w))(a) = (c,t)$ and $(\pi_i \circ \Omega\pi_2(w))(a) = (c, T(\psi))$, hence $(\pi_i \circ w)(a) = (c,(t,T(\psi)))$. Therefore $t \models_{\Omega} T(\psi)$. By induction hypothesis it follows $t \models^{\Omega} \psi$.

□

3.7 Coalgebras, Modal Logic, and Object Orientation

The modal logic presented in this chapter was designed to show that modal logics provide a natural language to speak about coalgebras. It was not the aim to propose a language capable of specifying fully fledged object oriented systems. This section discusses possibilities and problems of developments in this direction.

First let us hint at possible extensions of our logic dealing with the issues of verification, temporal specifications and inheritance. Concerning verification, our formalism only allows to specify properties of methods but not to verify correctness of implementations. This could be achieved by extending our logic by features of propositional dynamic logic (PDL)[6] as follows. PDL is a modal logic that has modal operators $[\alpha]$ for each statement α of a given imperative programming language. As in our logic, the intended meaning of a modal formula $[\alpha]\varphi$ is: after all executions of the statement α proposition φ holds. In addition, PDL has algebraic operations on the modalities corresponding to the operations allowing to form statements. For example $[\alpha], [\beta]$ being modalities of PDL, there is also a modality $[\alpha;\beta]$, corresponding

[6] For more information on PDL see e.g. Goldblatt [36].

to the composition operator ";" in the programming language. The intended meaning of the operations on the modalities is expressed by certain axioms (for example $[\alpha;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi$). In this way, using PDL as a logic on the level of the programming language and thinking of the formulas of our logic as special PDL formulas, we could use the (complete) calculus for PDL to verify that some implementation meets the specifications written in our logic.

We might also be interested in extending our logic with modal operators that allow to specify safety and liveness properties. In principle, the coalgebras being isomorphic to Kripke models, it is straight forward to use any of the many temporal logics designed for Kripke models like linear temporal logic or CTL*. For example, we may add to our logic two operators $\bigcirc$ and $\Box$ and interpret $\bigcirc$ by[7] $R_{\bigcirc} = \bigcup_{i,a} R_{i,a}$ and $R_{\Box}$ by the reflexive and transitive closure of $R_{\bigcirc}$. Hence $\bigcirc\varphi$ means "whatever is the method to be executed next, after its execution φ will hold" and $\Box$ is the corresponding always operator. But there is a technical problem due to the possibility of infinitely many modal operators $[i,a]$: To get a reasonable calculus we need to be able to infer $\bigcirc\varphi$ from the infinitely many premises $\{[i,a]\varphi\text{: for all } i,a\}$. Admitting a calculus with infinitary rules (but no infinitary formulas) and using as an essential ingredient the infinitary rule $\{[i,a]\varphi : \text{for all } i,a\} \vdash \bigcirc\varphi$, it is possible to give a complete calculus for a temporal logic for coalgebras with operators $\bigcirc$ and $\Box$. The completeness proof uses the technique developed in Segerberg [112] (but see also Goldblatt [37]).

How reasoning about inheritance could be integrated into our framework remains to be investigated. One approach that seems promising is to build on work by Uustalu [120] who gives modal logics to treat various aspects of inheritance.

Having discussed possibilities to deal with verification, temporal specifications, and inheritance we come to the main point concerning coalgebras and the OO-paradigm. It is still an open question how to deal with concurrency and communication inside the coalgebraic approach. Note that this point is also related to the question of how to model shared objects: Once we have a communication mechanism shared objects can be treated just as objects linked to the owners by message passing. And vice versa, shared objects could be used to model communication.

In order to further illustrate the issue we compare the coalgebraic approach with the one using distributed temporal logic (DTL), see [33] for an overview and further references and [47] for using DTL to give a complete semantics to the object oriented specification language TROLL [30]. The DTL approach is based on the view of objects as autonomous sequential agents capable of synchronous communication. Each object is modelled by a labelled sequential event structure. Synchronous communication is modelled by objects sharing the same event. Formulas of the logic are interpreted only locally w.r.t. a specific object. But formulas allow to express that an object shares an event with another object. For example, a formula local to object i may specify certain properties that have to hold for a different object j at the time when a common event is shared between i and j. Also, treating actions like transactions in database theory it is possible to give a logic invariant under change of granularity and thus supporting refinement proofs [31].

We think that the coalgebraic approach and the object-as-agent approach are in some sense orthogonal. The coalgebraic approach emphasises classes, e.g. it leads to canonical

[7]Recall from section 3.4.1 that the $R_{i,a}$ are the relations interpreting the operators $[i,a]$.

notions of invariance and bisimulation for a class. Using PVS, it has been used by Jacobs and others [64, 52, 65] to implement verification tools for (parts of) Java. The features handled are mainly those related to the class structure of the system like inheritance, overriding, and late binding. Another way to put it is to say that the coalgebraic approach as understood today is well suited to deal with the static features of an object oriented system[8] but it is not clear how to integrate dynamic aspects like communication and concurrency. This is also highlighted by the work of Cenciarelli et al. [24] who give an event based semantics of Java dealing with threads and concurrency. Whereas this semantics is conceptually not too far away from the DTL approach it has, up to now, resisted all efforts to combine it with a coalgebraic approach.

The object-as-agent approach emphasises objects. It allows for a natural treatment of communication and concurrency but seems to be less suited to deal with aspects related to classes. Also it has not yet led to verification tools.

One way to combine both worlds has already been presented in Reichel [93]. We think it is promising to continue this line of research and to try to develop a logical framework allowing to deal with both worlds in coherent way. There is the possibility that modal logics may play a major role in it, allowing to be used for different purposes as describing imperative programs, specifying objects and classes, communication and concurrency. As mentioned already, this would give us the very powerful machinery of modal logic at hand, with all its results concerning completeness, definability, decidability. And—important for specifying and verifying programs—with all its tools like interactive theorem provers and model checkers.

3.8 Conclusion

We have seen that by conceiving of coalgebras as Kripke models modal logic may be used as a logic for coalgebras. It just has been sketched what may be achieved by this approach. The kind of the functors considered should be extended to include at least the powerset functor. Concerning the specification of classes and objects it would be interesting to include temporal reasoning allowing for the specification of safety and liveness properties (see Jacobs [66] for a recent paper). Further topics include inheritance, refinement, compositionality, communication and verification. Also extensions of the logic by quantification or infinitary conjunctions should be considered.

[8] Late binding may be considered a dynamic concept but it can be handled in a static way, see [60].

Chapter 4

Algebraic and Coalgebraic Specifications

This is a revised version of [72]. The main difference is that, in section 3, we make use of behavioural equivalences (see chapter 1.2) instead of bisimulations.

4.1 Introduction

In this chapter we propose a framework, called (Ω, Ξ)-structures, for the algebraic extension of coalgebraic specifications of state-based systems (in particular, of object-oriented programs). The underlying ideas stem from the (algebraic) framework of observational logic presented in [50] and from similar ideas of swinging data types (Padawitz [88]) and recent extensions of hidden algebra (see Diaconescu [32] and Goguen and Roşu [35]. We show that the basic principles of observational logic can be transferred into the coalgebraic setting thus leading to a flexible extension of current coalgebraic specification techniques (cf. Reichel [93], Jacobs [62]).

The specific goals of our approach are to integrate constants and n-ary operations, to allow arbitrary first-order formulas for specifying observational properties of systems, to use a loose semantics approach in order to obtain sufficient flexibility for the choice of implementations, to support modularity, and to provide a sound and complete proof system for the verification of observational properties.

The starting point of our study is a consideration of standard coalgebraic specification techniques in the case where a polynomial functor $\Xi : \mathbf{Set} \to \mathbf{Set}$ is used to represent the possible operations on a (non-observable) state space X. As a simple example let us consider the following usual operations on bank accounts

$$bal : X \to \mathbb{Z}, \qquad update : X \times \mathbb{Z} \to X,$$

which are extracted from the functor

$$\Xi X = \mathbb{Z} \times X^{\mathbb{Z}}$$

as the projections of the transition function $\beta : X \to \mathbb{Z} \times X^{\mathbb{Z}}$ (whereby, for update, we use the fact that functions $X \to X^{\mathbb{Z}}$ correspond to functions $X \times \mathbb{Z} \to X$). According to the definition of Ξ *both* operations *bal* and *update* are used to define an indistinguishability relation for bank accounts (formally expressed by Ξ-bisimulation). Thereby two bank accounts a and b are indistinguishable (in the following also called observationally equivalent), if each of the observable experiments $\_.bal$, $\_.update(n).bal$, $\_.update(n_1).update(n_2).bal, \ldots$ yields the same result whether applied to a or to b.

We believe that using both operations, *bal* and *update*, for determining the observational equivalence of accounts imposes unnecessary complexity (for instance, for the construction of the final Ξ-coalgebra) and does also not express our intuition of observationally equivalent accounts since the essential information carried by an account is simply given by its balance whereas the update operation is just a method which does not reveal any new information. On the contrary, the *update* operation has to *respect the observational equality* of accounts (since, obviously, if two accounts have the same balance and then are credited by the same amount they should have again the same balance after the operation is performed).

As a consequence of this discussion we propose to split the set of operations of a specification into "true" observers (in the following simply called observers) and the "other" operations (in the following simply called operations).[1] Thereby it is the task of the specifier to choose the observers in such a way that they determine an appropriate observational equivalence for the objects under consideration. This is quite analogous to the specification of abstract data types and functional programs where also a decision has to be made which operations are to be considered as constructors for the data and which operations have to be defined by induction on the constructors.

Technically, this splitting in observers and operations is achieved by using two functors $\Omega, \Xi : \mathbf{Set} \to \mathbf{Set}$ such that Ξ defines a coalgebra structure (for the observers) and Ω defines an algebra structure (for the operations). Thereby it is assumed that the observers have only one argument of a state sort[2] while the operations can contain constants and n-ary operations on states.[3] Typically, the operations will be defined by coinduction w.r.t. the observers. For instance, the signature of bank accounts can be represented by the two functors

$$\Omega X = X \times \mathbb{Z}, \quad \Xi X = \mathbb{Z},$$

representing $update : X \times \mathbb{Z} \to X$ and $bal : X \to \mathbb{Z}$, respectively. A coinductive definition of *update* is $x.update(n).bal = x.bal + n$.

Another important point supporting our analysis concerns the question of modular observational specifications. The following example shows that the distinction between observers and operations is necessary if we want to be able to do modular observational specifications. Here, 'observational' means that the logic does not distinguish between models and their behaviours and, in particular, that equality on states is interpreted by observational equivalence. Modularity then requires that the observational equivalence of a single component is preserved

[1] This splitting is also the basic idea of the algebraic frameworks [88, 50, 32, 35], in contrast to hidden algebra [34] where all operations having a state sort (hidden sort) as argument are implicitly regarded as observers.

[2] To our knowledge, this restriction is also assumed in all other approaches which require the existence of final structures.

[3] In the current presentation a map $f : X \to A$ from state to data has to be an observer. This restriction is not essential: one could allow the set of data A to be one component of the many-sorted state space which would make it possible to consider f as an algebraic operation.

if this component is combined with other components. This can only be guaranteed if no "new" observers for the given component are introduced by the overall system. For instance, suppose we have a component for persons with observers $name, address$: person $\rightarrow$ string and we want to import this component into another component for accounts. Then the account-component must not introduce a new observer, say *birthdate* for persons (otherwise, importing the person-component would change the observational equivalence of the person-component which in turn would not allow us any more to transfer to accounts properties proved for persons). The account-component may only introduce observers for accounts, like *bal* : account $\rightarrow$ int or *owner* : account $\rightarrow$ person. However, it is important to note that the splitting of the signatures in observers and operations allows the account-component to contain arbitrary *non-observer* operations with arguments of type person like $change\_owner$: account, person $\rightarrow$ account (since these operations do not contribute to the definition of the observational equivalence).

The chapter is divided into two parts. Sections 2 to 4 introduce the framework of (Ω, Ξ)-structures, sections 5 to 7 show how this framework can be applied to the observational approach of algebraic specifications. In more detail the content is the following.

Section 2 contains notational conventions and recalls some technical preliminaries.

In Section 3 an (Ω, Ξ)-structure is defined as an algebra-coalgebra pair $(\alpha : \Omega X \rightarrow X, \beta : X \rightarrow \Xi X)$ such that the operations of the algebra part respect the observational equivalence determined by the observers of the coalgebra part (i.e. the operations are compatible with the largest bisimulation induced by Ξ).[4] Several characterisations of (Ω, Ξ)-structures are given which show the adequacy of this notion. Finally, we discuss some consequences of defining algebraic operations coinductively.

Section 4 requires some familiarity with monads and fibred category theory but the results needed later are stated without using these notions. It is shown that the features of (Ω, Ξ)-structures that make them suitable models for an observational approach to specifications can be categorically expressed by the fact that the operation mapping an (Ω, Ξ)-structure to its behaviour is a fibred idempotent monad. As a consequence we obtain—under a natural condition on the satisfaction relation—that the categories of (Ω, Ξ)-structures give rise to institutions. This in turn enables a modular approach to specifications.

Section 5 gives an extended example illustrating that for certain functors Ω, Ξ the framework of section 3 specialises to observational algebraic specifications.

In section 6, it is shown that for these functors and a special choice of signature morphisms one obtains an institution for observational specifications of (Ω, Ξ)-structures. In particular, we consider specifications $Sp = (\Omega, \Xi, Ax)$ with a set Ax of first-order axioms and we define the (loose) semantics of Sp as the class of all (Ω, Ξ)-structures (α, β) which Ξ-satisfy the axioms Ax. This means that (α, β) satisfies Ax up to Ξ-bisimilarity of elements which allows us to focus on observable properties and to abstract from internal (non-visible) properties of states. As a consequence of the distinction of observers and operations we obtain a straightforward method for coinductive specifications of the operations by a complete case distinction w.r.t.

[4] Algebra-coalgebra pairs are also considered in Malcolm [80], but without assuming the above compatibility requirement for (Ω, Ξ)-structures and with another morphism notion. It is, however, interesting to observe that the technical postulates used to achieve the results of [80] indeed force algebra-coalgebra pairs to be (Ω, Ξ)-structures.

the given observers.

For proving observable properties of a specification we present in section 7 a sound, complete and modular proof system.

4.2 Notation and Technical Preliminaries

Recall that, given a category $\mathcal{C}$ and two functors $\Omega, \Xi : \mathcal{C} \to \mathcal{C}$ and an object $X \in \mathcal{C}$, arrows $\alpha : \Omega X \to X$ and $\beta : X \to \Xi X$ are called *algebras* and *coalgebras*, respectively. An *algebra morphism* $f : \alpha \to \alpha'$ of algebras $\alpha : \Omega X \to X$, $\alpha' : \Omega X' \to X'$ is an arrow $f : X \to X'$ in $\mathcal{C}$ such that $f \circ \alpha = \alpha' \circ \Omega f$; a *coalgebra morphism* $f : \beta \to \beta'$ of coalgebras $\beta : X \to \Xi X$, $\beta' : X' \to \Xi X'$ is an arrow $f : X \to X'$ in $\mathcal{C}$ such that $\Xi f \circ \beta = \beta' \circ f$. Algebras and coalgebras form categories $\mathcal{C}^{\Omega}$ and $\mathcal{C}_{\Xi}$, respectively. Following Malcolm [80] we call a pair (α, β) of an algebra $\alpha : \Omega X \to X$ and a coalgebra $\beta : X \to \Xi X$ on the same object X an *algebra-coalgebra pair*. Algebra-coalgebra-pair morphisms are morphisms that are both algebra and coalgebra morphisms.

In this chapter, $\mathcal{C}$ will mostly be the category $\mathbf{Set^n}$, $n \in \mathbb{N}$, of n-sorted sets. More precisely, an object $X \in \mathbf{Set^n}$ is a family $(X_i)_{1 \le i \le n}$ and an arrow $f : X \to Y$ is a family of functions $(f_i : X_i \to Y_i)_{1 \le i \le n}$. The identity arrow and composition are defined componentwise. A feature of the category $\mathbf{Set^n}$ that we use tacitly is that epis (monos with componentwise nonempty source) have a right (left) inverse (i.e. they are split) and are hence preserved by functors.

Recall that a Ξ-*bisimulation* on a coalgebra $\beta : X \to \Xi X$ is a relation $R \subset X \times X$[5] such that there is an arrow $\gamma : R \to \Xi R$ that makes the left-hand diagram below commute; an Ω-*congruence* on an algebra $\alpha : \Omega X \to X$ is a relation $R \subset X \times X$ such that there is an arrow $\delta : \Omega R \to R$ that makes the right-hand diagram below commute. (π_1, π_2 are the canonical projections.)

$$\begin{tikzcd} X \arrow[d, "\beta"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"'] & X \arrow[d, "\beta"] \\ \Xi X & \Xi R \arrow[l, "\Xi\pi_1"'] \arrow[r, "\Xi\pi_2"] & \Xi X \end{tikzcd} \qquad \begin{tikzcd} \Omega X \arrow[d, "\alpha"'] & \Omega R \arrow[l, "\Omega\pi_1"'] \arrow[r, "\Omega\pi_2"] \arrow[d, "\delta"'] & \Omega X \arrow[d, "\alpha"] \\ X & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & X \end{tikzcd}$$

According to this definition, an Ω-congruence need not be an equivalence relation, but it has to be substitutive, i.e. it is compatible with the algebraic operations α. For example, fix a set A and let $\alpha : A \times X \to X$ be an algebra. Then R is an Ω-congruence on α iff for all $a \in A$, for all $x, y \in X$ it holds that $xRy \Rightarrow \alpha(a,x) R \alpha(a,y)$.

Recall from chapter 1.2 that a Ξ-behavioural equivalence e on a coalgebra β is just an epi with domain β. The *largest behavioural equivalence* on a coalgebra β exists for arbitrary functors Ξ on $\mathbf{Set^n}$ and will be denoted by $e : \beta \to \bar{\beta}$. Also recall that we can identify behavioural equivalences $e : \beta \to \bar{\beta}$ with the kernel pair (R, π_1, π_2) of e. We also call the kernel pair of a morphism simply its *kernel*.

Recall from chapter 1.2.4 that in the case that Ξ preserves *weak pullbacks* the notions of largest behavioural equivalence and largest bisimulation can be identified:

[5] The notions of relation and subset w.r.t. $\mathbf{Set^n}$ are defined componentwise.

Lemma 4.2.1. *Let $\beta : X \to \Xi X$ be a Ξ-coalgebra, and suppose that Ξ preserves weak pullbacks. Then, given a largest behavioural equivalence $e : \beta \to \bar{\beta}$ on β, $R \subset X \times X$ is the largest bisimulation on β iff the diagram below is a pullback in $\mathbf{Set^n}$:*

$$\begin{tikzcd} R \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "e"] \\ X \arrow[r, "e"] & \bar{X} \end{tikzcd}$$

Conversely, given the largest bisimulation $R \subset X \times X$ on β, e is the largest behavioural equivalence on β iff e is the coequaliser of (π_1, π_2).

Finally, in section 4.4 we will need a lemma about idempotent monads.

Lemma 4.2.2. *Let $\mathcal{C}$ be a category, B an operation on the objects of $\mathcal{C}$, and, for each $M \in \mathcal{C}$, let $\eta_M : M \to BM$ be epi in $\mathcal{C}$. Moreover, assume that for each $f : M \to BN$ there is $f^\#$ such that*

$$\begin{tikzcd} BM \arrow[r, dotted, "f^\#"] & BN \\ M \arrow[u, "\eta_M"] \arrow[ur, "f"'] & \end{tikzcd}$$

commutes. Then $(B, \eta, (-)^\#)$ is an idempotent monad.

Proof. By definition of a monad as a Kleisli triple, we have to check for all $f : M \to BN, g : L \to BM$ the laws (i) $f = f^\# \circ \eta_M$, (ii) $(\eta_M)^\# = \mathrm{id}_{BM}$, (iii) $(f^\# \circ g)^\# = f^\# \circ g^\#$. First note that $f^\#$ is uniquely determined since η_M is epi. (i) holds by assumption, (ii) follows from uniqueness, and (iii) follows using (i) and uniqueness ($f^\# \circ g = (f^\# \circ g)^\# \circ \eta_L$ and $f^\# \circ g = f^\# \circ (g^\# \circ \eta_L)$). To see that the monad is idempotent we have to show that the 'multiplication' $(\mathrm{id}_{BM})^\# : BBM \to BM$ is iso. By (i), $(\mathrm{id}_{BM})^\# \circ \eta_{BM} = \mathrm{id}_{BM}$, that is, η_{BM} is split mono, hence iso, hence $(\mathrm{id}_{BM})^\#$ is iso. □

4.3 (Ω, Ξ)-structures

As discussed in the introduction we are interested in structures of the kind $\Omega X \to X \to \Xi X$ where the algebraic part respects the behavioural equivalence expressed by the coalgebraic part.

The structure of this section changed a little w.r.t. [72] because we make use of the notion of a largest behavioural equivalence of chapter 1.2.3. The main advantage of the new definition of (Ω, Ξ)-structures is that it works for any functors on $\mathbf{Set^n}$: We do not need to require neither that functors preserve weak pullbacks nor that a final coalgebra exists.

4.3.1 Basic Definitions and Results

We start with the definition of behaviour.

Definition 4.3.1 (Behaviour). Let Ω, Ξ be functors on $\mathbf{Set^n}$ and $(\alpha : \Omega X \to X, \beta : X \to \Xi X)$ an algebra-coalgebra pair. Consider the largest behavioural equivalence $e : \beta \to \bar{\beta}$ on β. Then $(\bar{\alpha}, \bar{\beta})$ is called the behaviour of (α, β) iff the following diagram commutes:

$$\begin{tikzcd} \Omega X \arrow[r, "\alpha"] \arrow[d, "\Omega e"'] & X \arrow[r, "\beta"] \arrow[d, "e"'] & \Xi X \arrow[d, "\Xi e"] \\ \Omega \bar{X} \arrow[r, dotted, "\bar{\alpha}"] & \bar{X} \arrow[r, "\bar{\beta}"] & \Xi \bar{X} \end{tikzcd}$$

Remark 4.3.2. The behaviour of an algebra-coalgebra pair exists iff $\bar{\alpha}$ exists such that the left-hand square commutes. The existence of $\bar{\alpha}$ expresses that the algebraic operations α are compatible with the largest behavioural equivalence e on β. This means in particular that the operations α do not contribute to making observations.

Proposition 4.3.3. *Let $\Omega X \xrightarrow{\alpha} X \xrightarrow{\beta} \Xi X$ be an algebra-coalgebra pair. Then its behaviour—if it exists—is uniquely determined (up to isomorphism of algebra-coalgebra pairs).*

Proof. Uniqueness of $\bar{\beta}$ is immediate by the definition of a largest behavioural equivalence (see definition 1.2.9), uniqueness of $\bar{\alpha}$ follows from Ωe epi (which, in turn, is due to the fact that e is epi and epis in $\mathbf{Set^n}$ are split). $\square$

We now define the central notion of this chapter.

Definition 4.3.4 ((Ω, Ξ)-structures). Let Ω, Ξ be functors on $\mathbf{Set^n}$ and $(\alpha : \Omega X \to X, \beta : X \to \Xi X)$ an algebra-coalgebra pair. Then (α, β) is an (Ω, Ξ)-structure (on X) iff the behaviour $(\bar{\alpha}, \bar{\beta})$ of (α, β) exists.

Since every (Ω, Ξ)-structure has a behaviour we can define:

Definition 4.3.5. Let M be an (Ω, Ξ)-structure. We denote the operation that maps an (Ω, Ξ)-structure to its behaviour by B and the operation that maps an (Ω, Ξ)-structure to the unique algebra-coalgebra-pair morphism $\eta_M : M \to BM$ by η.

We show that definition 4.3.4 is equivalent to the original definition given in [51, 72] in the case that the final coalgebra exists.

Proposition 4.3.6. *Let Ω, Ξ be functors on $\mathbf{Set^n}$ and let $\pi : Z \to \Xi Z$ be a final Ξ-coalgebra. Then an algebra-coalgebra pair $(\alpha : \Omega X \to X, \beta : X \to \Xi X)$ is an (Ω, Ξ)-structure iff there is an arrow $h : \Omega Z \to Z$ such that the following diagram commutes (! denotes the unique morphism from the coalgebra β to the final coalgebra π):*

$$\begin{tikzcd} \Omega X \arrow[r, "\alpha"] \arrow[d, "\Omega !"'] & X \arrow[r, "\beta"] \arrow[d, "!"'] & \Xi X \arrow[d, "\Xi !"] \\ \Omega Z \arrow[r, dotted, "h"] & Z \arrow[r, "\pi"] & \Xi Z \end{tikzcd}$$

Proof. Let us write $X \xrightarrow{e} \bar{X} \xrightarrow{m} Z$ for the factorisation of $! : \beta \to \pi$ which is unique up to iso. e is the largest behavioural equivalence on β. Consider the following diagram:

$$\begin{tikzcd}
\Omega X \arrow[r, "\alpha"] \arrow[d, "\Omega e"'] & X \arrow[r, "\beta"] \arrow[d, "e"'] & \Xi X \arrow[d, "\Xi e"] \\
\Omega \bar{X} \arrow[r, dotted, "\bar{\alpha}"] \arrow[d, "\Omega m"'] & \bar{X} \arrow[r, "\bar{\beta}"] \arrow[d, "m"'] & \Xi \bar{X} \arrow[d, "\Xi m"] \\
\Omega Z \arrow[r, dotted, "h"] & Z \arrow[r, "\pi"] & \Xi Z
\end{tikzcd}$$

For the "if" part suppose that h exists and define $\bar{\alpha}$ to be the unique diagonal fill-in (m is mono and Ωe is epi). For the "only if" part consider the components $m_i : \bar{X}_i \to Z_i$, $1 \leq i \leq n$. If $\bar{X}_i$ is empty then h_i is any map, if $\bar{X}_i \neq \{\}$ then let j_i be a left inverse of m_i (i.e. $j_i \circ m_i = \mathrm{id}_{\bar{X}_i}$) and define $h_i = m_i \circ \bar{\alpha}_i \circ \Omega j_i$. □

Remark. h is in general not uniquely determined. But it follows from proposition 4.3.3 that the restriction of h to the image of $\Omega!$ is unique.

The intuition that Ω-operations of (Ω, Ξ)-structures are compatible with Ξ-observations is made precise by the following proposition (which, as shown in theorem 4.3.8, is even a characterisation of (Ω, Ξ)-structures):

Proposition 4.3.7. *Let (α, β) be an (Ω, Ξ)-structure on $X \in$ $\mathbf{Set^n}$. The largest Ξ-behavioural equivalence on the coalgebra β is an Ω-congruence on the algebra α.*

Proof. The (kernel of the) largest behavioural equivalence e on β is given by the pullback

$$\begin{tikzcd}
R \arrow[r, "\pi_2"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "e"] \\
X \arrow[r, "e"] & \bar{X}
\end{tikzcd}$$

Hence $e \circ \pi_1 = e \circ \pi_2$. Using $\bar{\alpha} \circ \Omega e = e \circ \alpha$, it follows $e \circ (\alpha \circ \Omega\pi_1) = e \circ (\alpha \circ \Omega\pi_2)$. Since R is a pullback there is a mapping (even a unique one) $\delta : \Omega R \to R$ making R into a Ω-congruence. □

Next, we prove the converse of proposition 4.3.7 and thereby give a second characterisation of (Ω, Ξ)-structures.[6]

Theorem 4.3.8. *Let $\Omega X \xrightarrow{\alpha} X \xrightarrow{\beta} \Xi X$ be an algebra-coalgebra pair and suppose that Ω preserves weak pullbacks. Then (α, β) is an (Ω, Ξ)-structure iff the largest Ξ-behavioural equivalence on β is an Ω-congruence on α.*

[6] Theorem 4.3.8 is closely related to the result of Rutten and Turi [106] saying (very roughly) that a final semantics has an equivalent initial semantics if bisimulation is congruence.

Proof. The "only if" part was proved as proposition 4.3.7. For the converse we show that the behaviour of (α, β) exists. Let $e : \beta \to \bar{\beta}$ be the largest behavioural equivalence on β. To define $\bar{\alpha} : \Omega\bar{X} \to \bar{X}$, we fix a right inverse i of e (i.e. $e \circ i = id_{\bar{X}}$) and let $\bar{\alpha} = e \circ \alpha \circ \Omega i$.

We have to show that e is an algebra morphism, i.e. $\bar{\alpha} \circ \Omega e = e \circ \alpha$. Let (R, π_1, π_2) be the kernel pair of e and $\Omega R \stackrel{\delta}{\to} R$ the arrow making R into a congruence. Consider the following three layered diagram:

$$\begin{tikzcd}
\Omega\bar{X} \arrow[d, "\Omega i"'] & \Omega X \arrow[l, "\Omega e"'] \arrow[d, "r"'] \arrow[dr, "id"] & \\
\Omega X \arrow[d, "\alpha"'] & \Omega R \arrow[l, "\Omega\pi_1"'] \arrow[r, "\Omega\pi_2"] \arrow[d, "\delta"'] & \Omega X \arrow[d, "\alpha"] \\
X \arrow[dr, "e"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & X \arrow[dl, "e"] \\
& \bar{X} &
\end{tikzcd}$$

Recall that R is a pullback. Therefore (Ω preserving weak pullbacks) ΩR is a weak pullback. Together with $\Omega e \circ \Omega i \circ \Omega e = \Omega e \circ \mathrm{id}_{\Omega X}$ this shows that there is $r : \Omega X \to \Omega R$ such that the topmost layer commutes. The second layer commutes since R is a congruence and the third by definition of R. Now, going from the top to the bottom yields $e \circ \alpha \circ \Omega i \circ \Omega e = e \circ \alpha$, i.e. $\bar{\alpha} \circ \Omega e = e \circ \alpha$. $\square$

Using lemma 4.2.1 we can express the result above in terms of bisimulations:

Corollary 4.3.9. *Let $\Omega X \stackrel{\alpha}{\to} X \stackrel{\beta}{\to} \Xi X$ be an algebra-coalgebra pair and suppose that Ω, Ξ preserve weak pullbacks. Then (α, β) is an (Ω, Ξ)-structure iff the largest Ξ-bisimulation on β is an Ω-congruence on α.*

In order to obtain a category of (Ω, Ξ)-structures we still need an appropriate notion of morphism. Of course, the obvious choice is that of an algebra-coalgebra-pair morphism (see section 4.2). But algebra-coalgebra-pair morphisms do not reflect the relationships between the *behaviours* of structures: From the point of view of the specifier, a notion of morphism is required that implies that (Ω, Ξ)-structures are isomorphic if they have the same behaviour. These two approaches give rise to two different categories. Moreover, we also want to consider the category that consists only of behaviours.

Definition 4.3.10 ($\mathbf{Str}^{\Omega}_{\Xi}, \mathbf{bStr}^{\Omega}_{\Xi}, \overline{\mathbf{Str}}^{\Omega}_{\Xi}$). $\mathbf{Str}^{\Omega}_{\Xi}$ is the category of (Ω, Ξ)-structures with algebra-coalgebra-pair morphisms. $\mathbf{bStr}^{\Omega}_{\Xi}$ has the same objects as $\mathbf{Str}^{\Omega}_{\Xi}$ but the morphisms between two (Ω, Ξ)-structures M, N are given by the algebra-coalgebra-pair morphisms between the behaviours BM, BN. $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ is the full replete[7] subcategory of $\mathbf{Str}^{\Omega}_{\Xi}$ defined by the behaviours of $\mathbf{Str}^{\Omega}_{\Xi}$.

[7] That is, $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ has as objects all objects of $\mathbf{Str}^{\Omega}_{\Xi}$ isomorphic to a behaviour and as morphisms between two such objects the morphisms of $\mathbf{Str}^{\Omega}_{\Xi}$.

Intuitively, the difference of the three categories is as follows. $\mathbf{Str}^{\Omega}_{\Xi}$ is the natural category of (Ω,Ξ)-structures. But in $\mathbf{Str}^{\Omega}_{\Xi}$ it is generally not the case, that structures with the same behaviour are isomorphic. Since this seems natural from a specification point of view, one may prefer to work in $\mathbf{bStr}^{\Omega}_{\Xi}$ instead. $\mathbf{bStr}^{\Omega}_{\Xi}$ contains all implementations but regarded from a behavioural point of view.[8] It should be intuitively clear (for a proof see corollary 4.4.8) that the category of behaviours $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ is equivalent to $\mathbf{bStr}^{\Omega}_{\Xi}$. But they are not isomorphic, the difference being precisely that $\mathbf{bStr}^{\Omega}_{\Xi}$ also contains all possible implementations whereas $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ only contains the behaviours.

In section 4.4.1 we will show that B (see definition 4.3.5) gives rise to a monad on $\mathbf{Str}^{\Omega}_{\Xi}$ with $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ being the category of Eilenberg-Moore algebras and $\mathbf{bStr}^{\Omega}_{\Xi}$ the Kleisli category.

4.3.2 Coinductive Definitions

In this section we will suppose that a final Ξ-coalgebra exists because, usually, to define by coinduction means to define morphisms into the final coalgebra. Nevertheless, this section could also be developed using the notion of a largest behavioural equivalence instead of a final coalgebra.

As indicated in the introduction, in our setting a typical style of writing specifications is to define the algebraic structure via coinduction using the coalgebraic signature Ξ. For example, in the introduction we called $x.update(n).bal = x.bal + n$ a coinductive definition of the *update*-operation. We now want to justify this informal terminology by relating axioms like $x.update(n).bal = x.bal + n$ to the formal account of coinduction as presented in Rutten [109] or Jacobs and Rutten [58].

There, the coalgebra $f : X \to \Xi X$ is said to be a coinductive definition of the function $\alpha : X \to Z$ if $Z \xrightarrow{\pi} \Xi Z$ is the final coalgebra and α is the unique coalgebra morphism, see the left-hand diagram below:

$$\begin{tikzcd} X \arrow[r, "\alpha"] \arrow[d, "f"'] & Z \arrow[d, "\pi"] \\ \Xi X \arrow[r, "\Xi\alpha"'] & \Xi Z \end{tikzcd} \qquad \begin{tikzcd} \Omega X \arrow[r, "\alpha"] \arrow[d, "f"'] & X \arrow[d, "\beta"] \\ \Xi\Omega X \arrow[r, "\Xi\alpha"'] & \Xi X \end{tikzcd}$$

In our context, we want to define the algebraic operations $\alpha : \Omega X \to X$ on a coalgebra $\beta : X \to \Xi X$ coinductively. First, let β be the final coalgebra and consider the right-hand diagram above. Then any function $f : \Omega X \to \Xi\Omega X$ provides a coinductive definition of algebraic operations $\alpha : \Omega X \to X$. To see what f has to be in our example (α as *update*) recall $\Omega X = X \times \mathbb{Z}$, $\Xi X = \mathbb{Z}$, $\Xi\alpha = \mathrm{id}_{\mathbb{Z}}$, $\beta = bal$. It is easy to see that $f(x,n) = x.bal + n$ defines the operation *update*.

Second, suppose that β is (isomorphic to) a subcoalgebra of π. Now, every function $f : \Omega X \to \Xi\Omega X$ defines a unique morphism $\alpha' : \Omega X \to Z$. Moreover, α' (and hence f) determines a morphism $\alpha : \Omega X \to X$ if and only if α' factors through $! : \beta \to \pi$. In this case,

[8] In the case of the special functors Ω, Ξ in section 4.6 the morphisms in $\mathbf{bStr}^{\Omega}_{\Xi}$ can be described explicitly as certain relations, see the observational homomorphisms in [50], 3.9.

the algebraic operations α are uniquely determined by $\alpha' = ! \circ \alpha$ (since $!$ is mono), see the left-hand diagram below:

$$\begin{tikzcd}
\Omega X \arrow[r, dotted, "\alpha"] \arrow[d, "f"'] & X \arrow[r, "!"] \arrow[d, "\beta"'] & Z \arrow[d, "\pi"] \\
\Xi\Omega X \arrow[r, dotted, "\Xi\alpha"'] & \Xi X \arrow[r, "\Xi !"'] & \Xi Z
\end{tikzcd}
\qquad
\begin{tikzcd}
\Omega X \arrow[r, "\Omega e"] \arrow[d, "f"'] & \Omega\bar{X} \arrow[d, dotted, "\bar{f}"] \\
\Xi\Omega X \arrow[r, "\Xi\Omega e"'] & \Xi\Omega\bar{X}
\end{tikzcd}$$

Third, let β be any Ξ-coalgebra and suppose that $\alpha' : \Omega X \to Z$ factors through X as $\alpha' = ! \circ \alpha$. Then α is unique up to bisimulation.[9] But it may well be that α is not compatible with observational equivalence, i.e., that (α, β) is not an (Ω, Ξ)-structure. (The reason is that an arbitrary f may distinguish between observably equivalent states.) We therefore need a condition forcing f to depend only on observable properties of states. This can be done as follows.

Definition 4.3.11 (Coinductive definition of (Ω, Ξ)-structures). A coinductive definition of (Ω, Ξ)-structures consists of a function $f : \Omega X \to \Xi\Omega X$ for each coalgebra $\beta : X \to \Xi X$ such that there is a function $\bar{f} : \Omega\bar{X} \to \Xi\Omega\bar{X}$ making the right-hand diagram above commute (where $\bar{X}$ is the carrier of the behaviour $\bar{\beta}$ of β and $e : \beta \to \bar{\beta}$ the corresponding morphism, see definition 4.3.1).

Let $f : \Omega X \to \Xi\Omega X$ be a coinductive definition of (Ω, Ξ)-structures, $\pi : Z \to \Xi Z$ the final coalgebra, and $\alpha' : f \to \pi$. We say that an (Ω, Ξ)-structure (α, β) on X *is defined by* $f : \Omega X \to \Xi\Omega X$ iff $\alpha' = ! \circ \alpha$ (where $! : \beta \to \pi$).

The following proposition generalises the second point above to arbitrary coalgebras.

Proposition 4.3.12. *Let $f : \Omega X \to \Xi\Omega X$ be a coinductive definition of (Ω, Ξ)-structures, $\pi : Z \to \Xi Z$ the final coalgebra, and $\alpha' : f \to \pi$. Then a coalgebra β on X gives rise to an (Ω, Ξ)-structure (α, β) defined by f iff α' factors through $! : \beta \to \pi$. Moreover the (Ω, Ξ)-structure is unique up to Ξ-bisimulation.*

Proof. Assume α' factors through $! : \beta \to \pi$. Let α be such that $\alpha' = ! \circ \alpha$. Uniqueness up to bisimulation is clear from the respective definitions. It remains to show that (α, β) is an (Ω, Ξ)-structure. We write $\bar{X}$ for the image of $!$ and $! = m \circ e$ for the corresponding factorisation. We show that the behaviour (α, β) exists, i.e., that there is $\bar{\alpha} : \Omega\bar{X} \to \bar{X}$ with $\bar{\alpha} \circ \Omega e = e \circ \alpha$. First, by the existence of an $\bar{f} : \Omega\bar{X} \to \Xi\Omega\bar{X}$ it follows that there is $\alpha'' : \Omega\bar{X} \to Z$ such that $\alpha' = \alpha'' \circ \Omega e$. Also, $\alpha' = m \circ (e \circ \alpha)$ (by definition of α) and, hence, $\alpha'' \circ \Omega e = m \circ (e \circ \alpha)$. Now, since m mono and Ωe epi there is a "diagonal fill-in" $\bar{\alpha} : \Omega\bar{X} \to \bar{X}$ such that $\bar{\alpha} \circ \Omega e = e \circ \alpha$. $\square$

The first part of the discussion above showed that coinductive definitions of (Ω, Ξ)-structures always have a model, namely the final coalgebra itself. This shows the following important property of coinductive definitions.

[9] We call two functions $\alpha_1, \alpha_2 : Y \to X$ equal up to bisimulation iff $! \circ \alpha_1 = ! \circ \alpha_2$ (where Y a set, X the carrier of a coalgebra, $!$ the corresponding morphism into the final coalgebra).

Proposition 4.3.13. *Coinductive definitions of* (Ω, Ξ)*-structures are consistent.*

A final remark on the nature of coinductive definitions of (Ω, Ξ)-structures: The discussion above showed that the class of models of such a definition is determined by those Ξ-coalgebras $\beta : X \to \Xi X$ such that the morphisms α' factor through !. That is, a coinductive definition imposes closure conditions on a coalgebra X (closure under the operations specified by α'), in other words, forces the coalgebra to contain enough "good" elements. In this respect our approach here differs from other approaches like Jacobs [61], Gumm [45], and chapter 2 and 3 where specifications force coalgebras to avoid "bad" elements.

4.4 The Behaviour Monad and Institutions

We show that the operation mapping an (Ω, Ξ)-structure to its behaviour is an idempotent monad. This observation has several interesting consequences. First, it allows to relate the three categories of (Ω, Ξ)-structures in a satisfying manner. Second, it will allow us to find a general condition under which the categories $\mathbf{Str}^{\Omega}_{\Xi}, \mathbf{bStr}^{\Omega}_{\Xi}$ give rise to institutions. Last but not least, the framework developed below suggests that the notion of an idempotent monad is at the heart of the behavioural approach to specification and may open the way to an axiomatic theory of behavioural specifications.

Readers not familiar with monads or fibred category may want to look only at corollary 4.4.8, subsection 4.4.2, and corollary 4.4.22 which only are needed to continue with section 4.6. The categorical background can be found in Mac Lane [78] (monads), Jacobs [63] (fibred category theory) and Borceux [19] (both).

4.4.1 The Behaviour Monad

The operation B on $\mathbf{Str}^{\Omega}_{\Xi}$ mapping an (Ω, Ξ)-structure to its behaviour was defined in 4.3.5. To show that B is a monad (in particular, a functor) we need the following proposition (see also 4.3.5 for the definition of η_M):

Proposition 4.4.1. *Let M, N be (Ω, Ξ)-structures. Then every morphism $f : M \to BN$ in $\mathbf{Str}^{\Omega}_{\Xi}$ determines a unique $f^{\#} : BM \to BN$ such that $f = f^{\#} \circ \eta_M$.*

Proof. Let $M = \Omega X \xrightarrow{\alpha} X \xrightarrow{\beta} \Xi X$, $BM = \Omega\bar{X} \xrightarrow{\bar{\alpha}} \bar{X} \xrightarrow{\bar{\beta}} \Xi\bar{X}$, $BN = \Omega\bar{Y} \xrightarrow{\bar{\gamma}} \bar{Y} \xrightarrow{\bar{\delta}} \Xi\bar{Y}$. To define $f^{\#}$ let $\pi : Z \to \Xi Z$ be the final coalgebra, $m \circ \eta_M$ the unique factorisation of $! : \beta \to \pi$ through $\bar{\beta}$, let $i : \bar{\delta} \to \pi$, and consider the following diagram in $\mathbf{Set^n}$:

$$\begin{tikzcd} X \arrow[r, "\eta_M"] \arrow[d, "f"'] & \bar{X} \arrow[d, "m"] \arrow[dl, dotted, "f^{\#}"'] \\ \bar{Y} \arrow[r, "i"'] & Z \end{tikzcd}$$

The square commutes since both ways are coalgebra morphisms to the final coalgebra. η_M is epi and i is mono ($\bar{Y}$ being the carrier of a behaviour). Therefore, $f^{\#}$ is the unique diagonal fill-in with $f = f^{\#} \circ \eta_M$. That $f^{\#}$ is a morphism in $\mathbf{Str}^{\Omega}_{\Xi}$ can easily be seen using that η_M and $\Omega\eta_M$ are epi. $\square$

Definition 4.4.2 (Behaviour Monad B on $\mathbf{Str}^{\Omega}_{\Xi}$). The behaviour monad B on $\mathbf{Str}^{\Omega}_{\Xi}$ is defined to be the Kleisli triple $(B, \eta, (-)^{\#})$ where B and η are as in 4.3.5 and $(-)^{\#}$ is the uniquely defined operation on morphisms of $\mathbf{Str}^{\Omega}_{\Xi}$ described in the proposition above.

It follows from lemma 4.2.2 that B is indeed a monad and, moreover, idempotent:

Proposition 4.4.3. *B is an idempotent monad.*

The following property of idempotent monads will be useful.

Proposition 4.4.4. *Let $\mathsf{B} = (B, \eta, (-)^{\#})$ be an idempotent monad on $\mathcal{C}$. Then there is a natural iso $\mathcal{C}(M, BN) \simeq \mathcal{C}(BM, BN)$.*

Proof. The iso $\varphi_{MN} : \mathcal{C}(M, BN) \to \mathcal{C}(BM, BN)$ is given by $\varphi_{MN}(f) = f^{\#}$ and $\varphi^{-1}_{MN}(g) = g \circ \eta_M$. □

The proposition above holds for all idempotent monads but the behaviour monads $(B, \eta, (-)^{\#})$ arising in the framework of (Ω, Ξ)-structures have, moreover, the property that the η_M are split epi. The (proof of the) following theorem shows that these monads are in bijective correspondence to the natural isos $\varphi_{MN} : \mathcal{C}(M, BN) \to \mathcal{C}(BM, BN)$ for full endofunctors B (the bijection is given by the proof of proposition 4.4.4 and "if" below).

Theorem 4.4.5. *Let B be an endofunctor on $\mathcal{C}$. Then B can be extended to an idempotent monad $\mathsf{B} = (B, \eta, (-)^{\#})$ with the η_M being split epis iff B is full and there is a natural iso $\varphi_{MN} : \mathcal{C}(M, BN) \to \mathcal{C}(BM, BN)$.*

Proof. "only if": Use proposition 4.4.4. To show that B is full let $k : BL \to BM$. We have to find k' with $Bk' = k$. Since η_L is epi, $k = (k \circ \eta_L)^{\#}$. Since η_M is split epi, there is k' with $\eta_M \circ k' = k \circ \eta_L$, hence $Bk' = (\eta_M \circ k')^{\#} = (k \circ \eta_L)^{\#} = k$.
"if":Let $f : M \to BN$, $g : L \to BM$, $l : L \to M$. Define $f^{\#} = \varphi_{MN}(f)$ and $\eta_M = \varphi^{-1}_{MM}(\mathrm{id}_{BM})$. We have to check (i) $f = f^{\#} \circ \eta_M$, (ii) $(\eta_M)^{\#} = \mathrm{id}_{BM}$, (iii) $(f^{\#} \circ g)^{\#} = f^{\#} \circ g^{\#}$, (iv) $Bl = (\eta_M \circ l)^{\#}$,[10] and to show that the η_M are split epi (which implies idempotency, see lemma 4.2.2). (ii) is immediate. For the other conditions we need to know that naturality of φ means (let $l : L \to M$, $h : M \to BN$, $p : N \to P$) that (a) $(h \circ l)^{\#} = h^{\#} \circ Bl$ and (b) $(Bp \circ h)^{\#} = Bp \circ h^{\#}$. (iv) is an instance of (a). That B is full implies (c) $\exists f' : f^{\#} = Bf'$. Now, (iii) follows from (c) and (b): $(f^{\#} \circ g)^{\#} = (Bf' \circ g)^{\#} = Bf' \circ g^{\#} = f^{\#} \circ g^{\#}$. (i) follows from (iii), (ii), and φ_{MN} injective. Finally, we show that η_M are split epi. Since B is full there is f' such that $Bf' = \mathrm{id}^{\#}_{BM}$. It follows $\mathrm{id}_{BM} = \mathrm{id}^{\#}_{BM} \circ \eta_{BM} = Bf' \circ \eta_{BM} = \eta_M \circ f'$, that is, η_M is split epi. □

Next, we show that $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ is isomorphic to the category of Eilenberg-Moore algebras of the behaviour monad B and that $\mathbf{bStr}^{\Omega}_{\Xi}$ is isomorphic to the Kleisli category.

Proposition 4.4.6. *$\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ is isomorphic to the category $(\mathbf{Str}^{\Omega}_{\Xi})^{\mathsf{B}}$ of Eilenberg-Moore algebras for the behaviour monad B.*

[10] (iv) is usually the definition of B on morphisms. Since we assume here that B is a functor already, we have to show that B satisfies (iv).

Proof. We define an isomorphism $\varphi : \overline{\mathbf{Str}^{\Omega}_{\Xi}} \to (\mathbf{Str}^{\Omega}_{\Xi})^{\mathsf{B}}$. Let M be an (Ω, Ξ)-structure isomorphic to a behaviour. Then $\eta_M : M \to BM$ is iso. Define $\varphi(M) = \eta_M^{-1} : BM \to M$. $\varphi(M)$ is indeed an Eilenberg-Moore algebra. Obviously, φ is injective on objects. To see that φ is onto let $\theta : BM \to M \in (\mathbf{Str}^{\Omega}_{\Xi})^{\mathsf{B}}$. By definition of $(\mathbf{Str}^{\Omega}_{\Xi})^{\mathsf{B}}$ it holds $\theta \circ \eta_M = id_M$ and by idempotency of B we have that θ is iso (see [19], vol.2, proposition 4.2.3). Therefore, $\theta = \eta_M^{-1}$ which shows that φ is onto. On morphisms, φ is the identity. ☐

Proposition 4.4.7. $\mathbf{bStr}^{\Omega}_{\Xi}$ *is isomorphic to the Kleisli category* $(\mathbf{Str}^{\Omega}_{\Xi})_{\mathsf{B}}$ *of the behaviour monad* B.

Proof. The Kleisli category $(\mathbf{Str}^{\Omega}_{\Xi})_{\mathsf{B}}$ has the same objects as $\mathbf{Str}^{\Omega}_{\Xi}$ and morphisms $(\mathbf{Str}^{\Omega}_{\Xi})_{\mathsf{B}}(M,N) = \mathbf{Str}^{\Omega}_{\Xi}(M, BN)$. The isomorphism $(\mathbf{Str}^{\Omega}_{\Xi})_{\mathsf{B}} \simeq \mathbf{bStr}^{\Omega}_{\Xi}$ is given on objects as the identity and on morphisms via proposition 4.4.4: $(\mathbf{Str}^{\Omega}_{\Xi})_{\mathsf{B}}(M,N) = \mathbf{Str}^{\Omega}_{\Xi}(M, BN) \simeq \mathbf{Str}^{\Omega}_{\Xi}(BM, BN) = \mathbf{bStr}^{\Omega}_{\Xi}(M,N)$. ☐

The following is an immediate corollary of the two propositions above.

Corollary 4.4.8. *The functor* $K : \mathbf{bStr}^{\Omega}_{\Xi} \to \overline{\mathbf{Str}^{\Omega}_{\Xi}}$ *mapping structures to their behaviours is full and faithful and, moreover, an equivalence of categories.*

Proof. K is the comparison functor [78] between the Kleisli category and the Eilenberg-Moore category, hence full and faithful. It is easy to check that idempotency of the monad implies that K is an equivalence (using that the Eilenberg-Moore algebras are of the kind $\theta : BM \to M$, θ iso). ☐

We have shown that the operation B mapping models to their behaviours is an idempotent monad. This statement seems to contain most features of B relevant to the behavioural approach to specifications. Nevertheless behaviours have another important property, namely that there is at most one morphism from a given structure into a behaviour. Behaviours inherit this uniqueness property from the uniqueness property of final coalgebras. We will call every idempotent monad with this property a *behaviour functor* (a notion borrowed from [50], theorem 3.8). One way to state the uniqueness property is the following:

Definition 4.4.9 (Behaviour Functor). A behaviour functor on a category $\mathcal{C}$ is an idempotent monad B on $\mathcal{C}$ with the property that the Kleisli category is thin.[11]

4.4.2 Institutions of (Ω, Ξ)-Structures

In this section we define signature morphisms and investigate under what conditions we obtain institutions of (Ω, Ξ)-structures.

To fix notation we recall the notion of institution here. We follow Tarlecki [113]. An institution $(\mathbf{Sig}, \mathsf{Mod}, \mathsf{Sen}, \models)$ consists of a category of *signatures* $\mathbf{Sig}$, a functor $\mathsf{Mod} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$ giving for each signature $\Sigma \in \mathbf{Sig}$ a category of Σ-*models*, a functor $\mathsf{Sen} : \mathbf{Sig} \to \mathbf{Set}$ giving for each signature $\Sigma \in \mathbf{Sig}$ a set of *sentences* $\mathsf{Sen}(\Sigma)$, and for each signature $\Sigma \in \mathbf{Sig}$

[11] That is, for all $X, Y \in \mathcal{C}$ there is at most one morphism $X \to BY$ in $\mathcal{C}$.

a *satisfaction relation* $\models_\Sigma \subset \mathsf{Mod}(\Sigma) \times \mathsf{Sen}(\Sigma)$ such that for all $\sigma : \Sigma \to \Sigma'$, all $\varphi \in \mathsf{Sen}(\Sigma)$, and all $M' \in \mathsf{Mod}(\Sigma')$

$$\mathsf{Mod}(\sigma)(M') \models_\Sigma \varphi \quad \Leftrightarrow \quad M' \models_{\Sigma'} \mathsf{Sen}(\sigma)(\varphi).$$

This condition is called the *satisfaction condition* of institutions. Moreover, we require that $M_1 \models_\Sigma \varphi \Leftrightarrow M_2 \models_\Sigma \varphi$ whenever M_1, M_2 are isomorphic. We usually write $M \models \varphi$ instead of $M \models_\Sigma \varphi$ because the signature Σ can always be inferred from the signature of the model M.[12] We abbreviate $\mathsf{Mod}(\sigma)$ by σ^* when Mod is clear from the context. σ^* is called the *reduct functor*.

Also, let us note that institutions can equivalently be defined by replacing the functor Mod by a split fibration. This observation will be used in section 4.4.2.

Let us briefly illustrate the satisfaction condition in the context of behavioural specifications. Suppose that we have a logic containing an equality predicate on states and that equality on states shall be interpreted as observational equivalence. Then the satisfaction condition can be read as stating that observational equivalence on states is invariant under transforming signatures. We will see that this can be achieved, abstractly, by assuming that the reduct functors preserve behaviours (see definition 4.4.13 and corollary 4.4.22) and, concretely, by restricting signature morphisms according to the slogan 'no new observations on old sorts' (see definition 4.6.3).

Signatures

In order to get sufficient flexibility we have to consider signature morphisms between functors (Ω, Ξ) and (Ω', Ξ') on different categories. Typically, (Ω, Ξ) will be defined on $\mathbf{Set^n}$ and (Ω', Ξ') on $\mathbf{Set^{n'}}$ for $n' \geq n$. This allows us, for instance, to build up larger specifications from smaller ones.

Definition 4.4.10 (Signatures and Signature Morphisms). A pair (Ω, Ξ) of functors on a category $\mathcal{C}$ is called a signature. If we want to make the category explicit, we also write $(\mathcal{C}, \Omega, \Xi)$ or (n, Ω, Ξ) if $\mathcal{C}$ is $\mathbf{Set^n}$. A *signature morphism* $\sigma : (\mathcal{C}, \Omega, \Xi) \to (\mathcal{C}', \Omega', \Xi')$ is given by a functor $V : \mathcal{C}' \to \mathcal{C}$ and two natural transformations $\varrho : \Omega V \to V\Omega'$, $\tau : V\Xi' \to \Xi V$. We write $\sigma = (V, \varrho, \tau)$. The identity morphism is given by the identity functor and the identity natural transformations. Composition is defined in the obvious way: Given $\sigma_1 : (\mathcal{C}_1, \Omega_1, \Xi_1) \to (\mathcal{C}_2, \Omega_2, \Xi_2)$ and $\sigma_2 : (\mathcal{C}_2, \Omega_2, \Xi_2) \to (\mathcal{C}_3, \Omega_3, \Xi_3)$ with $\sigma_1 = (V_1, \varrho_1, \tau_1)$ and $\sigma_2 = (V_2, \varrho_2, \tau_2)$, then $\sigma_2 \circ \sigma_1 = (V_1 V_2, \varrho_2 \circ \varrho_1, \tau_1 \circ \tau_2)$.

Definition 4.4.11 (Sig). The category **Sig** has objects (n, Ω, Ξ), $n \in \mathbb{N}$, and morphisms as described in the definition above. Since the parameter n is implicit in the functors Ω, Ξ we generally write (Ω, Ξ) for signatures in **Sig**.

The functor V represents the part of the signature morphism relating the sorts. Typically, V will be a projection functor $\mathbf{Set^{n'}} \to \mathbf{Set^n}$, $n' \geq n$, induced by an injective mapping on sorts $\{1, \ldots n\} \to \{1, \ldots n'\}$. The natural transformations ϱ and τ represent the part of the

[12] This notation is also suggested by the fibred perspective of section 4.4.2: there, $\models$ will indeed be a single relation, not a collection of relations indexed by signatures.

signature morphism relating the function symbols. This definition of a signature morphism is the natural one in the sense that each morphism σ induces a corresponding *reduct functor* σ^* on models:

Definition 4.4.12 (Reduct Functor). Let $\mathsf{Mod} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$ be a functor mapping signatures to categories of algebra-coalgebra pairs, let $\sigma : (\mathcal{C}, \Omega, \Xi) \to (\mathcal{C}', \Omega', \Xi')$ be in $\mathbf{Sig}$ and $M' = \Omega' X \xrightarrow{\alpha'} X \xrightarrow{\beta'} \Xi' X$ be in $\mathsf{Mod}(\mathcal{C}', \Omega', \Xi')$. We then define the reduct of M' w.r.t. σ as $\sigma^*(M') =$

$$\Omega V X \xrightarrow{\varrho_X} V\Omega' X \xrightarrow{V\alpha'} VX \xrightarrow{V\beta'} V\Xi' X \xrightarrow{\tau_X} \Xi V X$$

which is an algebra-coalgebra pair $(V\alpha' \circ \varrho_X, \tau_X \circ V\beta')$ over $VX \in \mathcal{C}$. On morphisms $f' : M' \to N'$, $\sigma^*(f') = Vf'$.

It is routine to check that σ^* is a functor between categories of algebra-coalgebra pairs and, moreover, the corresponding mapping $\mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$, $\sigma \mapsto \sigma^*$ is functorial. This looks like being on the right track towards an institution but we have to make sure that the reduct functors preserve the compatibility of the algebraic operations of an (Ω, Ξ)-structure (that is, they map indeed (Ω, Ξ)-structures to (Ω, Ξ)-structures). Moreover, we have to find a condition that allows to show that the satisfaction condition of institutions is met. Interestingly, both aspects can be dealt with by the same condition, namely that the reduct functors preserve behaviours.

Definition 4.4.13. A functor σ^* between two categories of algebra-coalgebra pairs is said to *preserve behaviours* iff σ^* maps behaviours to structures isomorphic to a behaviour. On categories of (Ω, Ξ)-structures this can be expressed by stating that for all M it holds $B\sigma^* M \simeq \sigma^* BM$.

According to the categories $\mathbf{Str}^{\Omega}_{\Xi}, \mathbf{bStr}^{\Omega}_{\Xi}, \overline{\mathbf{Str}}^{\Omega}_{\Xi}$ of definition 4.3.10 we can now define three operations $\mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$ mapping signatures to categories of structures and signature morphisms to functors.

Definition 4.4.14 ($\mathsf{Str}, \mathsf{bStr}, \overline{\mathsf{Str}}$). Suppose that for each $\sigma \in \mathbf{Sig}$, σ^* preserves behaviours and let k_M be a choice of isomorphisms $B\sigma^* M \simeq \sigma^* BM$. For $\sigma : (\Omega, \Xi) \to (\Omega', \Xi')$ define $\mathsf{Str}, \mathsf{bStr}, \overline{\mathsf{Str}} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$:

- $\mathsf{Str} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT},\ (\Omega, \Xi) \mapsto \mathbf{Str}^{\Omega}_{\Xi},\ \sigma \mapsto \sigma^*$,
- $\overline{\mathsf{Str}} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT},\ (\Omega, \Xi) \mapsto \overline{\mathbf{Str}}^{\Omega}_{\Xi},\ \sigma \mapsto \sigma^*$,
- $\mathsf{bStr} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT},\ \mathsf{bStr}(\Omega, \Xi) = \mathbf{bStr}^{\Omega}_{\Xi},\ \mathsf{bStr}(\sigma)(N) = \sigma^*(N),\ \mathsf{bStr}(\sigma)(f) = (k_N)^{-1} \circ \sigma^*(f) \circ k_M$ for all $f : M \to N \in \mathbf{bStr}^{\Omega'}_{\Xi'}$.

The definition of bStr depends on a choice of the isomorphisms k_N but any other choice would yield a functor bStr naturally isomorphic to the original one.

Proposition 4.4.15. $\mathsf{Str}, \overline{\mathsf{Str}}, \mathsf{bStr}$ *are functors.*

Proof. In the case of Str we have to show that $\sigma^*(M)$ is an (Ω, Ξ)-structure for all $M \in \mathbf{Str}^{\Omega'}_{\Xi'}$. We have to show that $\sigma^*(M)$ has a behaviour. But this follows from M having a behaviour and σ^* preserving it. The case of $\overline{\mathsf{Str}}$ is even simpler. For bStr we have to take into consideration that morphisms $M \to N \in \mathbf{bStr}^{\Omega'}_{\Xi'}$ are morphisms $BM \to BN \in \mathbf{Str}^{\Omega'}_{\Xi'}$. □

Recalling the definition of institution it still remains to work out when the satisfaction condition for the these functors hold. But before doing so we will investigate the categories of (Ω, Ξ)-structures from the point of view of fibred category theory.

Fibrations of (Ω, Ξ)-structures

The aim of this section is to extend the characterisations of the categories $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ (proposition 4.4.6) and $\mathbf{bStr}^{\Omega}_{\Xi}$ (proposition 4.4.7) to the *functors* $\overline{\mathsf{Str}}, \mathsf{bStr} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$.

To be able to do this we need the language of fibred category theory (see Jacobs [63] for an account on fibred category theory and a discussion of indexed categories vs. fibrations). The main idea is that functors $\mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$ can equivalently be described as a certain functors $\mathcal{E} \to \mathcal{B}$ called split fibrations.

For our purposes, the definition of a fibration can best be explained by stating the properties that are necessary (and sufficient) for transforming a functor $\mathcal{E} \to \mathcal{B}$ into an equivalent functor $\mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$. A *cloven fibration* is a functor $p : \mathcal{E} \to \mathcal{B}$ that has a *cleavage* $*$ which assigns to every morphism $\sigma : I \to J$ in the base $\mathcal{B}$ and every object $N \in \mathcal{E}$ over J (i.e., $pN = J$) a *cartesian lifting* $*(\sigma, N) : \sigma^*(N) \to N$. These liftings have to be closed under composition and have to satisfy the following universal property. For all f over σ (i.e., $pf = \sigma$) there is a unique f_1 such that $f = *(\sigma, N) \circ f_1$. Using uniqueness, σ^* can be extended from objects to morphisms: for $f : M \to N$ in the fibre over J (i.e., $pf = \mathrm{id}_J$), let $\sigma^*(f)$ be the unique morphism such that $f \circ *(\sigma, M) = *(\sigma, N) \circ \sigma^*(f)$. Moreover, again by uniqueness, it follows that σ^* is functorial. A *fibration* is defined similarly, also having cartesian liftings but not necessarily a uniform choice (cleavage) of them. The important point about the cleavage is that it allows to define the functors σ^*. A cloven fibration is called a *split fibration* (and the cleavage is called a *splitting*) if the induced assignment $\mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$, $\sigma \mapsto \sigma^*$ is functorial.

Next, we describe how—via the Grothendieck construction—a functor $F : \mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$ can be viewed as a split fibration $p : \mathcal{E} \to \mathcal{B}$. The objects of $\mathcal{E}$ are pairs $(I, M), I \in \mathcal{B}, M \in F(I)$.[13] The morphisms of $\mathcal{E}$ are pairs (σ, f) with $\sigma : I \to J \in \mathcal{B}$ and $f : M \to F(\sigma)(N)$ for $M \in F(I)$ and $N \in F(J)$. Composition is $(\sigma, f) \circ (\varrho, g) = (\sigma \circ \varrho, (F\varrho)(f) \circ g)$. $p : \mathcal{E} \to \mathcal{B}$ is defined as the first projection.

Before giving an example let us recall some standard terminology. $\mathcal{E}$ is called the *total category* and $\mathcal{B}$ the *base category*. The fibre over $I \in \mathcal{B}$ is the category with objects $M \in \mathcal{E}$ s.t. $pM = I$ and morphisms $f \in \mathcal{E}$ s.t. $pf = \mathrm{id}_I$. If $pM = I$ and $pf = \sigma$ we speak of M over I and f over σ. Our main example is obtained by applying the Grothendieck construction to the functor $\mathsf{Str} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$, yielding a split fibration $p : \mathbf{Str} \to \mathbf{Sig}$ where the total category $\mathbf{Str}$ contains all (Ω, Ξ)-structures for all signatures $(\Omega, \Xi) \in \mathbf{Sig}$. The categories $\mathbf{Str}^{\Omega}_{\Xi}$ are now the fibres over $(\Omega, \Xi) \in \mathbf{Sig}$.

For what we want to do, the fibrational approach is appropriate because we can extend the behavioural monads from the single fibres $\mathbf{Str}^{\Omega}_{\Xi}$ to the total category $\mathbf{Str}$ but not to all of

[13] That is, the class of objects of $\mathcal{E}$ is obtained as the disjoint union of the objects of all $F(I)$, $I \in \mathcal{B}$.

CAT. We first show that there is a canonical such extension if we have behavioural functors on every fibre and the functors σ^* preserve behaviours. We then show that performing the Eilenberg-Moore and the Kleisli construction on the total category **Str** gives us splits fibrations which correspond to the functors $\overline{\mathsf{Str}}$ and bStr.

Definition 4.4.16. Let $p : \mathcal{E} \to \mathcal{B}$ be a fibration with cleavage $*$ and $\mathsf{B} = (B, \eta, (-)^{\#})$ be fibrewise a behaviour functor on $\mathcal{E}$ (definition 4.4.9). Suppose that for all $\sigma : I \to J \in \mathcal{B}$, for all N in the fibre over J, there is an isomorphism $k^\sigma_N : B\sigma^* N \to \sigma^* BN$ (i.e. σ^* preserves behaviours).[14] We define B on the total category as follows. Every $f : M \to N \in \mathcal{E}$ factors as $f = *(\sigma, N) \circ f_1$ for a unique f_1 over I. Define $Bf = *(\sigma, BN) \circ k^\sigma_N \circ Bf_1$.

Let us note that this extension of a monad (or a functor) to the total category is also possible for functors that are not behaviour functors and for (cloven) fibrations that are not split. We would then have to add some coherence condition involving the k^σ_M and other natural transformations.

Proposition 4.4.17. *Under the assumptions of the definition above,* B *is a fibred monad on* p.

Proof. One shows that B is a functor whenever the following coherence conditions are satisfied: $k^{\mathrm{id}}_M = \mathrm{id}_{BM}$, $\sigma^*(k^\varrho_M) \circ k^\sigma_{\varrho^* M} = k^{\varrho\circ\sigma}_M$. But, B being a behaviour functor, morphisms into behaviours are unique. Together with the k-s being iso it follows that the equations above hold. The same kind of reasoning can be used to show that B is a monad. Here, the coherence conditions are (with $\mathsf{B} = (B, \eta, \mu)$) $k^\sigma_M \circ \mu_{\sigma^* M} = \sigma^* \mu_M \circ k^\sigma_{BM} \circ Bk^\sigma_M$ and $k^\sigma_M \circ \eta_{\sigma^* M} = \sigma^* \eta_M$. To show that B is fibred we have to prove that (i) $pB = p$ and (ii) B preserves cartesian morphisms. (i) is clear by the fibrewise definition of B. (ii) follows from the k-s being isos. □

For a fibred monad B on $p : \mathbf{Str} \to \mathbf{Sig}$, the Eilenberg-Moore construction and the Kleisli construction on the total category **Str** yield split fibred categories again. The following proposition is a slight variation on exercise 1.7.9 in Jacobs [63].

Proposition 4.4.18 ($p_{\mathsf{B}}, p^{\mathsf{B}}$). *Let* $p : \mathcal{E} \to \mathcal{B}$ *be a split fibration and* B *a fibred monad on* p. *Then the Kleisli category* $\mathcal{E}_{\mathsf{B}}$ *and the Eilenberg-Moore category* $\mathcal{E}^{\mathsf{B}}$ *are split fibrations over* $\mathcal{B}$ *denoted, respectively, by* $p_{\mathsf{B}} : \mathcal{E}_{\mathsf{B}} \to \mathcal{B}$ *and* $p^{\mathsf{B}} : \mathcal{E}^{\mathsf{B}} \to \mathcal{B}$.

Proof. Let $*$ be the splitting of p. We will give the definitions of the splittings $\#$ of p^{B} and $+$ of p_{B} skipping the lengthy but routine verifications. Since $\mathsf{B} = (B, \eta, (-)^{\#})$ is fibred we have for all $\sigma : I \to J \in \mathcal{B}$, for all M in the fibre over J, an iso $k^\sigma_M : B\sigma^* M \to \sigma^* BM$ with $*(\sigma, BM) \circ k^\sigma_M = B(*(\sigma, M))$. We define for all $\eta^{-1}_M \in \mathcal{E}^{\mathsf{B}}$ (see the proof of proposition 4.4.6) $\sigma^{\#}(\eta^{-1}_M) = \sigma^*(\eta^{-1}_M) \circ k^\sigma_M$ and $\#(\sigma, \eta^{-1}_M) = *(\sigma, M)$. For all $M \in \mathcal{E}_{\mathsf{B}}$ we define $\sigma^+(M) = \sigma^*(M)$, $+(\sigma, M) = \eta_M \circ *(\sigma, M)$. □

Theorem 4.4.19. *Let* p *be the split fibration corresponding to the functor* Str. *Then the functors* bStr *and* $\overline{\mathsf{Str}}$ *are the functors obtained by performing, respectively, the Kleisli and the Eilenberg-Moore construction for the behaviour monad* B *on* p.

[14] By definition 4.4.9 morphisms into behaviours are unique. Hence k^σ is natural.

Proof. First, use the Grothendieck construction to obtain the split fibration $p : \mathbf{Str} \to \mathbf{Sig}$ corresponding to the functor Str. Then, by proposition 4.4.17, the behaviour monads on $\mathbf{Str}^{\Omega}_{\Xi}$ can be lifted to a monad B on $\mathbf{Str}$ which, moreover, is a fibred monad on p. Using the cartesian liftings $\#$ and $+$ of the proof of proposition 4.4.18, we obtain the fibrations p^B, p_B which in turn are converted back to functors $\mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$. To obtain $\overline{\mathsf{Str}}$ and bStr it only remains to apply the isomorphisms shown in the proofs of propositions 4.4.6 and 4.4.7. □

We want to close this section with some remarks on what can be gained by using a fibred approach to institutions. *First*, the fibred approach is conceptually appealing. It allowed us to lift the behaviour monad to the total category and to apply the common categorical constructions of Eilenberg-Moore and Kleisli. This seems to be the most natural way to understand why we will be able to obtain institutions in our setting (see theorem 4.4.21). Another point is that the essential condition to obtain institutions, namely that the functors σ^* preserve behaviours, has a very natural counterpart in the fibrational setting: it corresponds to the requirement that the behaviour monad is fibred. *Second*, there are technical advantages. We obtain elegant categorical proofs of proposition 4.4.15 and corollary 4.4.22. We admit that this alone would not justify to introduce the apparatus of fibred category theory, but the authors are sure that it will pay off in future research on this topic. *Third*, and more generally, it seems to be against the spirit of the notion of institution to require the assignment $\mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$, $\sigma \mapsto \sigma^*$ to be functorial, where what we typically care about would be to require the equality $\sigma^*\varrho^* = (\varrho \circ \sigma)^*$ only up to natural isomorphism.[15] This could be achieved by replacing the functor $\mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$ by a pseudo-functor, or simpler, by dropping the assumption on the fibration to be split. The treatment of this section, for example, could easily be adapted for non-split fibrations.

Satisfaction Relations

We have shown that for the categories $\mathbf{Str}$, $\mathbf{bStr}$, $\overline{\mathbf{Str}}$ we get functors Str,bStr, $\overline{\mathsf{Str}}$. It remains to investigate under what conditions the satisfaction condition of institutions holds. The crucial point is that the reduct functors preserve behaviours or, in the language of fibred categories, that the behaviour monad is fibred.

The situation is the following. We suppose that we have a suitable logic (called the *standard logic*) to specify behaviours. The task is to extend this logic to a logic called the *behavioural logic* in such a way that models and their behaviours are not distinguished by formulas of the logic. That is, the behavioural satisfaction relation $\models\!\!\!\models$ is required to satisfy $M \models\!\!\!\models \varphi$ iff $BM \models \varphi$.

Definition 4.4.20 (Behavioural Satisfaction). Let $\mathcal{E}$ be a class (with elements called models), $B : \mathcal{E} \to \mathcal{E}$ an operation (with BM called the behaviour of M), $\mathcal{L}$ a class (with elements called formulas) and $\models, \models\!\!\!\models \subset \mathcal{E} \times \mathcal{L}$ two relations (called satisfaction relations). Then $\models\!\!\!\models$ is behavioural w.r.t. $\models$ (and w.r.t. B) iff for all $M \in \mathcal{E}, \varphi \in \mathcal{L}$

$$M \models\!\!\!\models \varphi \ \text{ iff } \ BM \models \varphi.$$

Remark. A behavioural satisfaction relation $\models\!\!\!\models$ is uniquely determined by $\models$ and B.

[15] Or even only up-to some weaker notion of equivalence, see [113].

The following theorem gives a general recipe which allows us—given a fibred monad B on a split fibration p—to obtain from an institution $(p^{\mathsf{B}}, \mathsf{Sen}, \models)$ two new institutions $(p, \mathsf{Sen}, \models\!\!\models)$ and $(p_{\mathsf{B}}, \mathsf{Sen}, \models\!\!\models)$. The interest in this theorem comes from the following interpretation. Let $\mathcal{E}$ be the total category of models of the fibration $p : \mathcal{E} \to \mathcal{B}$. The theorem then tells that *if there is given a 'standard' logic $(p^{\mathsf{B}}, \mathsf{Sen}, \models)$ on 'behaviours' $\mathcal{E}^{\mathsf{B}}$ then one also obtains 'behavioural' logics $(p, \mathsf{Sen}, \models\!\!\models)$ and $(p_{\mathsf{B}}, \mathsf{Sen}, \models\!\!\models)$ on 'standard' models $\mathcal{E}$ and $\mathcal{E}_{\mathsf{B}}$.*

Theorem 4.4.21. *Let $p : \mathcal{E} \to \mathcal{B}$ be a split fibration and B a fibred monad on p, let $\mathsf{Sen} : \mathcal{B} \to$ **Set** be a functor with $\mathcal{L}$ the class of all formulas over objects in $\mathcal{B}$, and let $\models \subset \mathcal{E} \times \mathcal{L}$ such that $(p^{\mathsf{B}}, \mathsf{Sen}, \models)$ is an institution. Then $(p, \mathsf{Sen}, \models\!\!\models)$ as well as $(p_{\mathsf{B}}, \mathsf{Sen}, \models\!\!\models)$ are institutions if $\models\!\!\models$ is behavioural w.r.t. $\models$ and B.*

Proof. We have to show that for p and p_{B} the satisfaction condition of institutions holds. That is, in the case of p, that for all $\sigma : I \to J \in \mathcal{B}$, all $\varphi \in \mathsf{Sen}(I)$, and all $M \in \mathcal{E}$ in the fibre over J it holds that $M \models\!\!\models \mathsf{Sen}(\sigma)(\varphi)$ iff $\sigma^*(M) \models\!\!\models \varphi$. We have

$$\begin{array}{l}\sigma^*(M) \models\!\!\models \varphi \;\Leftrightarrow\; B\sigma^*(M) \models \varphi \;\Leftrightarrow\; \sigma^*(BM) \models \varphi \;\Leftrightarrow\\ BM \models \mathsf{Sen}(\sigma)(\varphi) \;\Leftrightarrow\; M \models\!\!\models \mathsf{Sen}(\sigma)(\varphi).\end{array}$$

The equivalences are due to, respectively, $\models\!\!\models$ being behavioural w.r.t. $\models$, B being fibred (i.e., the reduct functors preserve behaviours), the satisfaction condition for $\models$, and again $\models\!\!\models$ being behavioural. In the case of p_{B} the same reasoning is valid. $\square$

As a corollary we obtain:

Corollary 4.4.22. *Consider the categories of (Ω, Ξ)-structures $\mathbf{Str}^{\Omega}_{\Xi}, \mathbf{bStr}^{\Omega}_{\Xi}, \overline{\mathbf{Str}}^{\Omega}_{\Xi}$ (definition 4.3.10) and the respective operations $\mathsf{Str}, \mathsf{bStr}, \overline{\mathsf{Str}} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$ (definition 4.4.14). Let $\mathsf{Sen} : \mathbf{Sig} \to \mathbf{Set}$ be a functor and $\models, \models\!\!\models$ satisfaction relations. If for all $\sigma : (\Omega, \Xi) \to (\Omega', \Xi') \in \mathbf{Sig}$, all $M \in \mathbf{Str}$*

1. *the reduct functors σ^* preserve behaviours (definition 4.4.13),*
2. *$\models\!\!\models$ is behavioural w.r.t. $\models$ and B (definitions 4.4.20 and 4.3.5),*
3. *$(\mathbf{Sig}, \mathsf{Sen}, \overline{\mathsf{Str}}, \models)$ is an institution,*

then $(\mathbf{Sig}, \mathsf{Sen}, \mathsf{Str}, \models\!\!\models)$, $(\mathbf{Sig}, \mathsf{Sen}, \mathsf{bStr}, \models\!\!\models)$ are institutions as well.

Proof. Follows from the theorem. An independent proof can be given using proposition 4.4.15 and checking the satisfaction condition as in the proof of the theorem. $\square$

4.5 Example of Observational Specifications

This section gives an example of the kind of specifications that will be considered in the remainder of this chapter.

Figure 4.1 presents a specification of bank accounts which has observers *bal*, *owner*, and *undo*. The intended meaning of *undo* is to reconstruct the previous state of an account after the performance of an *update* or a *change_owner*. Hence, by using *undo* one can potentially

spec PERSON
  observers
    $\_.name$: person $\to$ string
    $\_.address$: person $\to$ string
  operations
    $\_.change\_address$: person, string $\to$ person

spec ACCOUNT2
  import PERSON
  observers
    $\_.bal$: account $\to$ int
    $\_.owner$: account $\to$ person
    $\_.undo$: account $\to$ account
  operations
    new : string $\to$ account
    $\_.update\_$: account, int $\to$ account
    $\_.change\_owner\_$: account, person $\to$ account
    $\_.paycharge$: account $\to$ account

Figure 4.1: Bank Account Example

reveal more information (namely the account's history) than simple observation of the 'attributes' *bal* and *owner* would provide. Thus *undo* has indeed to be declared as an observer and not as an operation. In contrast, the intended meaning of *new*, *update*, *change_owner*, and *paycharge* is that of operations that respect the observational equivalence defined by *bal*, *owner*, and *undo*, that is, applying *update*, *change_owner*, *paycharge* to indistinguishable accounts is to be expected to keep them indistinguishable. Also note that without the distinction of algebraic operations from coalgebraic observations it would not be possible to model the signature above because of *change_owner*.

Writing X, Y for sets of sorts PERSON and ACCOUNT2, respectively, a model of PERSON is an (Ω, Ξ)-structure

$$X \times \text{string} \to X \to \text{string} \times \text{string}$$

and a model of type ACCOUNT2 an (Ω', Ξ')-structure

$$\begin{pmatrix} X \times \text{string} \\ \text{string} + Y \times \text{int} + Y \times X + Y \end{pmatrix} \to \begin{pmatrix} X \\ Y \end{pmatrix} \to \begin{pmatrix} \text{string} \times \text{string} \\ \mathbb{Z} \times X \times Y \end{pmatrix}$$

A typical example of a signature morphism is the obvious $\sigma : (\Omega, \Xi) \to (\Omega', \Xi')$ including PERSON in ACCOUNT2. Note that σ does not introduce new observers on old sorts: ACCOUNT2 has a new algebraic operation *change_owner* involving PERSON but does not specify a new observer on PERSON. Consequently, including PERSON in ACCOUNT2 does not change the observational equivalence of PERSON. Technically, this can be expressed by saying that the reduct functor of σ preserves behaviours (definition 4.4.13) which in turn (see corollary 4.4.22) will allow us to build an institution in the next section.

4.6 An Institution for Observational Specifications

In this section we show that, for a special choice of functors, (Ω, Ξ)-structures specialise to algebras in the sense of behavioural algebraic specifications. Moreover, these functors Ω, Ξ enable us to use first order logic as a specification language in a straight forward way. The aim of this section is to establish conditions (i)–(iii) of corollary 4.4.22 which in turn will give us an institution for modular observational specifications.

4.6.1 Signatures

For the remainder of the chapter, we consider only functors Ω and Ξ of the following special format.

For elements $X \in \mathbf{Set^n}$ we write X_s to denote component s, assuming $1 \leq s \leq n$. The s, $1 \leq s \leq n$, are called *state sorts.* For $w \in \{1, \dots n\}^*$, that is $w = s_1 \dots s_k$, let $X_w = X_{s_1} \times \dots X_{s_k}$.

The algebraic functors Ω are then given—componentwise—as a sum of finite products:

$$(\Omega X)_s = \sum_{i \in I} C_i \times X_{w_i}, \quad w_i \in \{1, \dots n\}^*$$

where the C_i are a finite number of arbitrary (but fixed) sets. In particular, we allow operations of any arity on states as long as these operations are not used as observers.

Ξ is a functor of the kind

$$(\Xi X)_s = \prod_{j \in J_1} (X_{s_j})^{A_j} \times \prod_{j \in J_2} B_j^{A_j},$$

where A_j, B_j are a finite number of arbitrary (but fixed) sets. The A_j, B_j, C_i are called parameter sets, the B_j output sets.

This special format of Ξ is called a *multiplicative functor.* It means that an observer either takes as input a state in X_s and an element in A_j and produces a new state or it takes as input a state in X_s and an element in A_j and produces an observable output of type B_j. In [71] it was shown that coalgebras for multiplicative functors are (isomorphic to) algebras for hidden signatures. Let us also note that the given format for signatures Ω, Ξ is more liberal than it might seem at first sight. For example, the parameter sets may be 0 (empty), 1 (singleton) or products of (other parameter) sets.

The functors Ω, Ξ define a signature that allows to name the components of $\alpha : \Omega X \to X$, $\beta : X \to \Xi X$ via the categorical laws $\alpha(s) = [\alpha(s) \circ in_1; \dots]$ and $\beta(s) = \langle \pi_1 \circ \beta(s), \dots \rangle$. It is nevertheless convenient to name the single components explicitly. This is done by introducing the sets $Opns(\Omega)$, $Obs(\Xi)$, called operations and observers, see the definition below.

Furthermore, for specifications, we need terms referring to standard operations on the parameter sets. And we need to use theorems concerning the parameter sets. Similarly to the hidden algebra approach (see e.g. [34]), we therefore assume that the parameter sets form a many-sorted algebra $\mathcal{D}$ (called the underlying *data algebra*) with respect to a signature Σ that has one sort for each parameter set A_j, B_j, C_i (for simplicity the sorts corresponding to A_j, B_j, C_i are also named A_j, B_j, C_i) and has operation symbols $Opns(\Sigma)$. It is required that $Opns(\Sigma)$ is disjoint from $Opns(\Omega) \cup Obs(\Xi)$ and that every element of a parameter set is denotable by some ground Σ-term. Given a logic based on the terms formed from $Opns(\Sigma)$ and variables, we will write $\mathsf{Th}(\mathcal{D})$ for the set of formulas valid in $\mathcal{D}$.

Definition 4.6.1 ($Opns(\Omega)$, $Obs(\Xi)$, (Ω, Ξ)-terms).
Let Ω, Ξ be functors as above. The many-sorted set $Opns(\Omega) \in \mathbf{Set^n}$ has components $Opns(\Omega)_s$ which consist—for all $1 \leq s \leq n$—of typed function symbols $f_i : C_i \times X_{w_i} \to X_s$ for every $i \in I$. The many-sorted set $Obs(\Xi) \in \mathbf{Set^n}$ has components $Obs(\Xi)_s$ which consist—for all $1 \leq s \leq n$—of typed function symbols $g_j : X_s \times A_j \to X_{s_j}, j \in J_1$ and $h_j : X \times A_j \to B_j, j \in J_2$. The many-sorted set $Terms(\Omega, \Xi)$ of (Ω, Ξ)-terms is formed in the usual way using a countable many-sorted set of variables Var and the function symbols of $Opns(\Omega) \cup Obs(\Xi) \cup Opns(\Sigma)$.

In the example of section 4.5 we have (considering PERSON as state sort 1, ACCOUNT2 as state sort 2, and renaming the function symbols f_i, g_j, h_j):

$$Opns(\Omega')_1 = \{change\_address\}, \qquad Obs(\Xi')_1 = \{name, address\},$$
$$Opns(\Omega')_2 = \{new, update, change\_owner, paycharge\}, \qquad Obs(\Xi')_2 = \{bal, owner, undo\}.$$

$Opns(\Sigma)$ may include further operations on strings and integers.

Terms of special importance are the contexts:

Definition 4.6.2 (Ξ-context). The many-sorted set $Cont(\Xi, B)$ of observable Ξ-contexts of (output) sort B has components $Cont(\Xi, B)_s$ which consist—for all state sorts $1 \leq s \leq n$—of the terms of output sort B formed from the set of function symbols $Obs(\Xi)$, variables of parameter sort, and a special variable z_s of state sort s. $Cont(\Xi)_s$ is the union of the $Cont(\Xi, B)_s$ for all output sorts B. Substitution of a term t in the context c for the variable z_s is denoted by $c[t]$.

In the example of section 4.5 we have (considering PERSON as state sort 1, ACCOUNT2 as state sort 2):

$$\begin{aligned}
Cont(\Xi, \text{string})_1 &= \{name(z_1), address(z_1)\}, \\
Cont(\Xi, \text{int})_2 &= \{bal(undo^n(z_2)) : n \geq 0\}, \\
Cont(\Xi, \text{string})_2 &= \{name(owner(undo^n(z_2))), address(owner(undo^n(z_2))) : n \geq 0\},
\end{aligned}$$

where $undo^n(z_2)$ is the obvious abbreviation.

Next we define a category of signatures. It is a subcategory of the category defined in definition 4.4.11 and will replace it in the sequel, hence we keep the old name.

Definition 4.6.3 (Sig). The category **Sig** has as objects the signatures (n, Ω, Ξ) with Ω, Ξ as defined at the beginning of this section. A morphism between (n, Ω, Ξ) and $(n', \Omega', \Xi'), n \leq n'$, is given by, first, a functor $V : \mathbf{Set^{n'}} \to \mathbf{Set^n}$ which is induced by an injection $\sigma_{\text{sorts}} : \{1, \ldots n\} \to \{1, \ldots n'\}$, second, mappings $\varrho_s : Opns(\Omega)_s \to Opns(\Omega')_{\sigma_{\text{sorts}}(s)}$, $\tau_s : Obs(\Xi')_{\sigma_{\text{sorts}}(s)} \to Obs(\Xi)_s$, $1 \leq s \leq n$. Furthermore the following conditions have to be satisfied:

1. • For all $f : C \times X_{s_1} \times \ldots X_{s_k} \to X_s \in Opns(\Omega)_s$ it holds that $\varrho_s(f) : C \times X_{\sigma_{\text{sorts}}(s_1)} \times \ldots X_{\sigma_{\text{sorts}}(s_k)} \to X_{\sigma_{\text{sorts}}(s)} \in Opns(\Omega')_{\sigma_{\text{sorts}}(s)}$,

- for all $g : X_{\sigma_{\text{sorts}}(s)} \times A \to X_{\sigma_{\text{sorts}}(s')} \in \mathit{Obs}(\Xi')_{\sigma_{\text{sorts}}(s)}$ it holds that $\tau_s(g) : X_s \times A \to X_{s'} \in \mathit{Obs}(\Xi)_s$,
- for all $h : X_{\sigma_{\text{sorts}}(s)} \times A \to B \in \mathit{Obs}(\Xi')_{\sigma_{\text{sorts}}(s)}$ it holds that $\tau_s(h) : X_s \times A \to B \in \mathit{Obs}(\Xi)_s$.

2. The τ_s have to be bijections.

The first condition above just tells that the mapping between the function symbols has to respect the typing.

The second condition ensures that, extending a signature by a larger one, the possible observations on the states of the smaller signature are not changed. The slogan here is: *no new observations on old sorts*. This is essential to prove that reduct functors preserve behaviours which in turn is essential to obtain an institution.

The second condition seems to be rather restrictive. However, it is important to note that we still can introduce new observers in Ξ' as long as the observed sort is a new sort, i.e., is not in the image of σ_{sorts}. Also, we can introduce new operations on old sorts. For instance, building a bank from accounts one has to have some observer(s) for the new sort "bank" and also some new operations involving the old sort "account" (like, e.g., adding a new account to the bank).

Note also that indeed ϱ and τ define natural transformations as in definition 4.4.10. But the notion of a natural transformation would allow for more general signature morphisms. For example, one could consider to give up that in both signatures the same parameter sets have to appear. The difficulty here is to find a general condition guaranteeing the preservation of behaviours as needed for corollary 4.4.22.

Next, we give an explicit description of the final coalgebras that can be found in Cîrstea [25].

Proposition 4.6.4. *Let Ξ be a functor as above. Then the final Ξ-coalgebra $\pi : Z \to \Xi Z$ has carriers Z_s given by $\prod[\mathit{Cont}(\Xi, B)_s \to B]$ where $[\mathit{Cont}(\Xi, B)_s \to B]$ is the set of all functions from contexts of type B to B and the product is over all output sets B.*

Using the explicit description of signatures, signature morphisms, and final coalgebras we can prove the following proposition that will allow us to instantiate theorem 4.4.21.

Proposition 4.6.5. *Let $\sigma : (\Omega, \Xi) \to (\Omega', \Xi') \in$ **Sig** as in definition 4.6.3, σ^* as in definition 4.4.12 and π be the final Ξ'-coalgebra. Then $\sigma^*(\pi)$ is isomorphic to a subcoalgebra of the final Ξ-coalgebra.*

Proof. The conditions on signature morphisms guarantee that $[\mathit{Cont}(\Xi', B)_s \to B] \simeq [\mathit{Cont}(\Xi, B)_s \to B]$ for all output sets B. With proposition 4.6.4 one obtains that $\sigma^*(\pi)$ is isomorphic to the final Ξ-coalgebra. □

The proof above shows that with our choice of signature morphisms not only behaviours but also final coalgebras are preserved by reduct functors. This is more than we need to get an institution and suggests that generalisations of our notion of signature morphism are possible.

The following corollary is of central importance because it allows to apply corollary 4.4.22.

Corollary 4.6.6. *With the definitions of this section the reduct functors σ^* of definition 4.4.12 preserve behaviours (see definition 4.4.13).*

Proof. Let $\sigma : (\Omega, \Xi) \to (\Omega', \Xi') \in \mathbf{Sig}$, $M' \in \mathbf{Str}^{\Omega'}_{\Xi'}$, and $\pi \in \mathbf{Str}^{\Omega}_{\Xi}, \pi' \in \mathbf{Str}^{\Omega'}_{\Xi'}$ the final coalgebras. Let $M' \xrightarrow{e'} BM' \xrightarrow{m'} \pi'$ and $\sigma^*(M') \xrightarrow{e} B\sigma^*(M') \xrightarrow{m} \pi$ be unique epi-mono factorisations. Now consider $\sigma^*(M') \xrightarrow{\sigma^*(e')} \sigma^*(BM') \xrightarrow{\sigma^*(m')} \sigma^*(\pi') \xrightarrow{i} \pi$. Since i is mono by proposition 4.6.5 and the factorisations are unique, it follows $B\sigma^*(M') \simeq \sigma^*(BM')$. □

4.6.2 Behavioural Satisfaction

It is well-known that first-order logic gives rise to an institution. Using corollary 4.6.6 and behavioural satisfaction (definition 4.4.20) it now follows from corollary 4.4.22 that (Ω, Ξ)-structures with behavioural satisfaction give rise to institutions. The details are as follows.

Using (Ω, Ξ)-terms we can define the set $\mathcal{L}(\Omega, \Xi)$ of many-sorted[16] first-order (Ω, Ξ)-formulas as usual from equations $t = r$ (with the terms $t, r \in \mathit{Terms}(\Omega, \Xi)$ of the same sort), the logical connectives $\neg, \wedge, \vee$ and the quantifiers $\forall, \exists$. In some cases we will also consider infinitary conjunctions and disjunctions over countable sets of formulas.

Given an (Ω, Ξ)-structure (α, β) on X and a valuation for the variables, we have the usual interpretation of terms of state sort as elements of X and of terms of parameter sort as elements of $\mathcal{D}$. In particular, terms formed from observers of sort s (see 4.6.1) $g_j : X_s \times A_j \to X_{s_j}, j \in J_1$ and $h_j : X_s \times A_j \to B_j, j \in J_2$ are interpreted by using the isomorphisms

$$X_s \times A_j \to X_{s_j} \;\simeq\; X_s \to X_{s_j}^{A_j}, \qquad X_s \times A_j \to B_j \;\simeq\; X_s \to B_j^{A_j}.$$

To be more precise, given a valuation $v : \mathrm{Var} \to X + \mathcal{D}$, we define a mapping $v^* : \mathit{Terms}(\Omega, \Xi) \to X + \mathcal{D}$ as follows. The definition of $v^*(t)$ is obvious if t is a variable or a term with leading function symbol from $\mathit{Opns}(\Omega) \cup \mathit{Opns}(\Sigma)$. To see how function symbols from $\mathit{Obs}(\Xi)$ are interpreted recall that the sth component of $\beta : X \to \Xi X$ is $\beta_s : X_s \to \prod_{j \in J_1} X_{s_j}^{A_j} \times \prod_{j \in J_2} B_j^{A_j}$. Therefore (with $J_1 = \{1, \ldots m\}, J_2 = \{m+1, \ldots m'\}$)

$$\begin{aligned} v^*(g_j(t_1, t_2)) &= (\pi_j \circ \beta_s(v^*(t_1)))(v^*(t_2)) \in X_{s_j}, \\ v^*(h_j(t_1, t_2)) &= (\pi_j \circ \beta_s(v^*(t_1)))(v^*(t_2)) \in B_j. \end{aligned}$$

Next, we define the satisfaction relation. From the observational point of view two elements of an (Ω, Ξ)-structure are equal if they cannot be distinguished by observations determined by the coalgebra functor Ξ, i.e. if they are Ξ-bisimilar. This idea leads to our notion of Ξ-satisfaction of arbitrary first-order formulas where the equality symbol is interpreted by Ξ-bisimulation. This idea corresponds to the notion of observational satisfaction which originally goes back to Reichel [92].

Definition 4.6.7 (Ξ-satisfaction). Let (α, β) an (Ω, Ξ)-structure on X, Var a many-sorted set of variables, $v : \mathrm{Var} \to X + \mathcal{D}$ a valuation and $\varphi \in \mathcal{L}(\Omega, \Xi)$. Then $(\alpha, \beta), v \models\!\!\!\models \varphi$ is defined by induction on the structure of φ:

[16] One sort for each of X_s, A_j, B_j, C_i. As mentioned already: the sorts for X_s are called state sorts (denoted simply by s); the names A_j, B_j, C_i are used synonymously for the sorts and the sets.

- $(\alpha,\beta),v \models t_1 = t_2$, where t_1,t_2 are terms of state sort s, iff there is a Ξ-bisimulation R on β such that $v^*(t_1)R_s v^*(t_2)$,
- $(\alpha,\beta),v \models t_1 = t_2$, where t_1,t_2 are terms of parameter sort, iff $v^*(t_1) = v^*(t_2)$,
- for logical connectives and quantifiers as usual.

We use the following standard notation: Let M be an (Ω,Ξ)-structure, φ an (Ω,Ξ)-formula and Φ a set of (Ω,Ξ)-formulas. Then $M \models \Phi$ iff $M \models \varphi$ for all $\varphi \in \Phi$. Moreover, $\Phi \models \varphi$ iff for all (Ω,Ξ)-structures M: $M \models \Phi$ implies $M \models \varphi$.

The next proposition (which is the analogue of [17], theorem 3.11, where also a proof can be found) tells that the above satisfaction relation is already determined by the fact that the equality symbol is interpreted as *equality in the behaviour* of a structure. Recall that we write $\models$ for the standard first-order satisfaction relation that is defined like $\models$ but using set-theoretic equality instead of Ξ-bisimulation in the first clause of definition 4.6.7.

Proposition 4.6.8. *Let (α,β) an (Ω,Ξ)-structure and $\varphi \in \mathcal{L}(\Omega,\Xi)$. Then*

$$(\alpha,\beta) \models \varphi \textit{ iff } (\bar{\alpha},\bar{\beta}) \models \varphi.$$

Hence $\models$ is behavioural w.r.t. $\models$ in the sense of definition 4.4.20. We can now apply corollary 4.4.22:

Theorem 4.6.9. *Let Ω,Ξ be functors as in section 4.6.1, let* **Sig** *be as in definition 4.6.3, let $\mathbf{Str}^{\Omega}_{\Xi}, \mathbf{bStr}^{\Omega}_{\Xi}$ be the corresponding categories of (Ω,Ξ)-structures, and* $\mathsf{Str}, \mathsf{bStr} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}, \sigma \mapsto \sigma^*$ *the corresponding operations (definition 4.4.14), and let* Sen *be the functor mapping signatures to the appropriate set of first-order formulas. Then* $(\mathbf{Sig}, \mathsf{Sen}, \mathsf{Str}, \models)$, $(\mathbf{Sig}, \mathsf{Sen}, \mathsf{bStr}, \models)$ *are institutions.*

Proof. We use corollary 4.4.22 (**Sig** according to definition 4.6.3 is a subcategory of **Sig** according to definition 4.4.11; but corollary 4.4.22 is invariant under taking subcategories of **Sig**). That the behaviours together with standard satisfaction $\models$ form an institution is well known and yields condition (iii) of corollary 4.4.22. (ii) is proposition 4.6.8 and (i) is corollary 4.6.6. □

4.6.3 Specifications

We introduce flat and structured specifications. Flat specifications use the first-order logic defined above. Structured specifications use the specification-building operations which come along with every institution, see for example Tarlecki [113].

Definition 4.6.10 (Flat Specifications). An (Ω,Ξ)-specification *Sp* is a tuple (Ω,Ξ,Ax) where Ax is a set of formulas of $\mathcal{L}(\Omega,\Xi)$. The class of *models* $\mathsf{Mod}(Sp)$ of the (Ω,Ξ)-specification *Sp* consists of all (Ω,Ξ)-structures that Ξ-satisfy Ax, i.e.,

$$\mathsf{Mod}(Sp) = \{(\alpha,\beta) \in \mathbf{Set}^{\Omega}_{\Xi} : (\alpha,\beta) \models Ax\}.$$

Definition 4.6.11 (Structured Specifications). Structured specifications are given by the following specification-building operations which assign to every specification *Sp* a signature $\mathsf{Sig}\,Sp$ and a class of models $\mathsf{Mod}\,Sp$:

basic For the specifications $Sp = (\Omega, \Xi, Ax)$ of definition 4.6.10:

- $\mathsf{Sig}\, Sp = (\Omega, \Xi)$
- $\mathsf{Mod}\, Sp = \{M \in \mathbf{Str}^{\Omega}_{\Xi} : M \models Ax\}$

union For specifications $Sp_1 = (\Omega, \Xi, Ax_1)$ and $Sp_2 = (\Omega, \Xi, Ax_2)$:

- $\mathsf{Sig}(Sp_1 \cup Sp_2) = (\Omega, \Xi)$
- $\mathsf{Mod}(Sp_1 \cup Sp_2) = \{M \in \mathbf{Str}^{\Omega}_{\Xi} : M \models Ax_1 \cup Ax_2\}$

translate For $Sp = (\Omega, \Xi, Ax)$ and $\sigma : (\Omega, \Xi) \to (\Omega', \Xi')$:

- $\mathsf{Sig}(\mathsf{translate}\, Sp \,\mathsf{by}\, \sigma) = (\Omega', \Xi')$
- $\mathsf{Mod}(\mathsf{translate}\, Sp \,\mathsf{by}\, \sigma) = \{M \in \mathbf{Str}^{\Omega'}_{\Xi'} : \sigma^* M \models Ax\}$

hide For $Sp = (\Omega', \Xi', Ax')$ and $\sigma : (\Omega, \Xi) \to (\Omega', \Xi')$:

- $\mathsf{Sig}(\mathsf{hide}\, Sp \,\mathsf{by}\, \sigma) = (\Omega, \Xi)$
- $\mathsf{Mod}(\mathsf{hide}\, Sp \,\mathsf{by}\, \sigma) = \{\sigma^* M' \in \mathbf{Str}^{\Omega}_{\Xi} : M' \models Ax'\}$

Example 4.6.12. Consider the specification that is given by extending the signature in section 4.5 by the axioms given in figure 4.2. Let $Sp_1 = (\Omega, \Xi, Ax)$ be the flat specification of PERSON as given by the signature (Ω, Ξ) of section 4.5 and by the set of axioms Ax for PERSON. Let $Sp_2 = (\Omega', \Xi', Ax')$ be the flat specification of ACCOUNT2 as given by the signature (Ω', Ξ') of section 4.5 and by the set of axioms Ax' for ACCOUNT. An example of a structured specification is given by combining these two specifications. Let $\sigma : (\Omega, \Xi) \to (\Omega', \Xi')$ be the obvious signature morphism. Then the specification of ACCOUNT2 together with PERSON is given by

$$(\mathsf{translate}\, Sp_1 \,\mathsf{by}\, \sigma) \cup Sp_2.$$

In the above specification the behaviour of the operations is specified by a complete case distinction w.r.t. the given observers. It is not difficult to see that this specification is a coinductive definition in the sense of section 4.3.2. It follows from proposition 4.3.13 that this specification is consistent.

A more loose specification can be obtained, for instance, by removing the equations for the *paycharge* operation. Then the semantics of the specification is still restricted to those models where the interpretation of *paycharge* is compatible with the greatest Ξ-bisimulation (since all models are (Ω, Ξ)-structures).

4.7 Proof System

In this section we give a sound and complete proof system. We first treat flat and then structured specifications.

spec PERSON
  observers ...
  operations ...
  axioms
    $\forall x \in \text{person},\ \forall s \in \text{string}:$
      $x.change\_address(s).name = x.name,$
      $x.change\_address(s).address = s.$

spec ACCOUNT2
  import PERSON
  observers ...
  operations ...
  axioms
    $\forall x \in \text{account},\ \forall s \in \text{string},\ \forall n \in \text{int},\ \forall p \in \text{person}:$
      $new(s).bal = 0,$
      $new(s).owner.name = s, \quad new(s).owner.address = \epsilon,$
      $new(s).undo = new(s),$

      $x.update(n).bal = x.bal + n,$
      $x.update(n).owner = x.owner,$
      $x.update(n).undo = x,$

      $x.change\_owner(p).bal = x.bal,$
      $x.change\_owner(p).owner = p,$
      $x.change\_owner(p).undo = x,$

      $x.paycharge.bal = x.bal - 10,$
      $x.paycharge.owner = x.owner,$
      $x.paycharge.undo = x.$

Figure 4.2: Axioms for PERSON and ACCOUNT2

4.7.1 Proof System for Flat Specifications

First we give the proof system, then we discuss the implications of using infinitary logic (which is needed for the completeness result). Finally we give an example of a proof in our system.

Because of the compatibility of the algebraic operations with Ξ-bisimulation, every proof system sound for first-order logic is also sound for (Ω, Ξ)-logic. What remains to be done is to add axioms that capture Ξ-bisimulation. This is done in the usual way using contexts (see definition 4.6.2).

The set of variables of parameter sort of a context c is denoted by $\mathrm{Var}(c)$. We write $\forall\mathrm{Var}(c)$ to denote quantification over all these variables in $\mathrm{Var}(c)$. Next we formulate a coinductive proof principle for (Ω, Ξ)-logic which is expressed by the following axiom:

Definition 4.7.1 **(CoInd_Ξ).**

$$(\mathrm{CoInd}_\Xi)_s = \forall x, y \in X : \bigwedge_{c \in Cont(\Xi)_s} (\forall \mathrm{Var}(c) : c[x] = c[y]) \;\Rightarrow\; x = y$$

and $\mathrm{CoInd}_\Xi = \{(\mathrm{CoInd}_\Xi)_s, 1 \le s \le n\}$ where n is the number of state sorts.

Whether the axiom is infinitary depends on the bisimulation defined by the coalgebra functor Ξ. In the ACCOUNT2 example from section 4.5 it is infinitary, because—intuitively—observationally equivalent accounts have to have the same balance after an arbitrary number of *undo*-operations. If we omit *undo* from the specification, the axiom becomes finitary.

Definition 4.7.2 **((Ω, Ξ)-proof system).** Let Ω, Ξ be functors as in section 4.6, let $\mathcal{D}$ be a data algebra and $\mathsf{Th}(\mathcal{D})$ the set of infinitary first-order formulas satisfied by $\mathcal{D}$. We write $\Phi \vdash_\Xi \varphi$ iff $\Phi \cup \{\mathrm{CoInd}_\Xi\} \cup \mathsf{Th}(\mathcal{D}) \vdash \varphi$ where $\vdash$ denotes derivability w.r.t. a sound and complete proof system for infinitary first-order logic as given, for instance, in Keisler [69].

Obviously, the coinductive proof principle is sound, since our semantic objects are (Ω, Ξ)-structures whose operations are required to be compatible with the observational equivalence given by the greatest Ξ-bisimulation. In previous approaches in the literature (see, e.g., [79, 14]) this property is not assumed and therefore has first to be checked before the coinductive proof principle can be applied.

Theorem 4.7.3 (Soundness and Completeness).

$$\Phi \vdash_\Xi \varphi \;\Leftrightarrow\; \Phi \models \varphi.$$

Proof. (Sketch.) Soundness follows from the remarks above. The proof of completeness uses the completeness proof in [50] by showing that their models (called *observational algebras*) and (Ω, Ξ)-structures are in a one-to-one correspondence. The main difference between observational algebras and (Ω, Ξ)-structures is that in [50] the data algebra is not fixed in advance but part of the specification. Now, using $\Phi \cup \mathsf{Th}(\mathcal{D})$ as a specification for observational algebras and observing that, according to Scott's theorem (see e.g. [69]), $\mathsf{Th}(\mathcal{D})$ determines the data part up to isomorphism (since the data algebra is assumed to be countable, since the data signature Σ allows to denote every element of $\mathcal{D}$, and since the logic has infinitary disjunctions), it is not difficult to show that the observational algebras for $\Phi \cup \mathsf{Th}(\mathcal{D})$ are in one-to-one correspondence to the (Ω, Ξ)-structures for Φ. Showing that this correspondence preserves and reflects validity finishes the proof. $\square$

Let us discuss the use of infinitary logic. First note that if there are only direct observers there exist (up to α-equivalence) only finitely many observable contexts and hence CoInd_Ξ is finitary. In this case we can choose a formal (i.e. finitary) proof system and any available theorem prover for first-order logic can be used.

Second, if there are also indirect observers there may be infinitely many observable contexts and CoInd_Ξ becomes infinitary. In this case, the above completeness result is mainly of theoretical interest. However, it is important to note that the infinitary formulas CoInd_Ξ can still be very useful. In practical examples the infinitary premise of CoInd_Ξ can often be established by a simple inductive proof. Using a result of [15] it is even possible to encode the infinitary formulas CoInd_Ξ by finitary ones if one introduces auxiliary symbols and reachability constraints. Hence the problem of the non-completeness of finitary proof systems for (Ω, Ξ)-logic corresponds exactly to the non-completeness of finitary proof systems for inductively defined data types (in particular of arithmetic). A recent study of the incompleteness of behavioural logics can be found in Buss and Roşu [22]. Finally, let us note that in the case of flat specifications we could replace the use of infinitary formulas by giving the formulas $(\mathrm{CoInd}_\Xi)_s$ as infinitary rules.

Example 4.7.4. Consider the example of the ACCOUNT2 specification from figure 4.2 and suppose one wants to show that

$$\forall x \in \text{account} : x.\mathit{paycharge} = x.\mathit{update}(-10).$$

We can write the ACCOUNT-component of CoInd_Ξ as

$$\begin{array}{l} \forall x, y \in \text{account} : \\ \qquad \begin{array}{rcl} \bigwedge_{i\in\mathbb{N}} x.\mathit{undo}^i.\mathit{bal} = y.\mathit{undo}^i.\mathit{bal} & \wedge & \\ \bigwedge_{i\in\mathbb{N}} x.\mathit{undo}^i.\mathit{owner}.\mathit{name} = y.\mathit{undo}^i.\mathit{owner}.\mathit{name} & \wedge & \\ \bigwedge_{i\in\mathbb{N}} x.\mathit{undo}^i.\mathit{owner}.\mathit{address} = y.\mathit{undo}^i.\mathit{owner}.\mathit{address} & \Rightarrow & x = y. \end{array} \end{array}$$

Instantiating x with $x.\mathit{paycharge}$ and y with $x.\mathit{update}(-10)$, the premise of the implication above follows directly from the axioms in figure 4.2. Note that the proof uses that we may deduce from $x.\mathit{paycharge} = x.\mathit{update}(-10)$ the equality $x.\mathit{paycharge}.t = x.\mathit{update}(-10).t$ for a term t of appropriate type. This substitution is only sound because the operations *paycharge* and *update* are assumed to respect the observational equivalence defined by the coalgebraic signature.

4.7.2 Proof System for Structured Specifications

Corresponding to the specification-building operators there are the following proof rules defining the relation $\Vdash$:

basic From $Ax \vdash_{\Xi} \varphi$ derive $(\Omega, \Xi, Ax) \Vdash \varphi$

union From $Sp \Vdash \varphi$ derive $Sp \cup Sp' \Vdash \varphi$ and $Sp' \cup Sp \Vdash \varphi$

translate From $Sp \Vdash \varphi$ derive $\mathsf{translate}\, Sp\ \mathsf{by}\ \sigma \Vdash \mathsf{Sen}(\sigma)(\varphi)$

hide From $Sp' \Vdash \mathsf{Sen}(\sigma)(\varphi)$ derive $\mathsf{hide}\, Sp'\ \mathsf{by}\ \sigma \Vdash \varphi$

Soundness and completeness now follow from general results on institutions.

Theorem 4.7.5 (Soundness and Completeness). *The above proof system is sound and complete.*

Proof. (Sketch.) Soundness of this proof system follows from a general soundness result for institutions of Sannella and Tarlecki [110]. Completeness uses Borzyszkowski's [20] completeness result for institutions. According to [20] it has to be checked that the institutions satisfy the amalgamation and interpolation properties. This can be done as in [49].

□

4.8 Conclusion

(Ω, Ξ)-structures provide the foundations of a flexible specification technique for state-based systems which extends standard coalgebraic specifications by incorporating the basic ideas of observational logic. But there is one point where the approach of this chapter is still less expressive than the one of [50]. There, it is possible to treat observers with multiple arguments of state sort ("binary observers") as long as one argument is designated to be the observed one. This is a useful feature in specifications (consider e.g. "isin"-methods like $isin : \mathrm{bank} \times \mathrm{account} \to \mathrm{Bool}$ which should be an observer for the first argument but not for the second). Unfortunately it is not yet clear to us how to solve this problem in this setting.

On the other hand, the approach of this chapter is also more general than [50] because the development in sections 3 and 4 puts no restriction on the functors Ω, Ξ to be 'algebraic'. For example, it seems natural to drop the restrictions on Ξ introduced in section 4.6 and allow for functors describing non-deterministic coalgebras (involving + and/or powerset). Since the use of equational logic in this chapter is essentially due to the fact that the special format of the functors Ξ allows to transform them into functors for algebras (see 1.7) we would expect a combination of algebraic and coalgebraic (i.e. modal) logics to be useful in this setting (concerning coalgebras and modal logic see [87, 105, 104, 66, 67] and chapters 2 and 3).

In Bodoit et al. [16], the approach of this chapter has been dualised to give an account of reachability in algebraic specifications. This led to a new notion of constructor-based specifications as well as to the insight that observability and reachability in (co)algebraic specifications are dual concepts, a phenomenon which was discovered earlier in the context of automata theory, see Arbib and Manes [7].

Acknowledgements

We want to thank Pietro Cenciarelli for the many discussions initiating our work on the connection between algebraic and coalgebraic specifications. For valuable comments we are grateful to Ataru Nakagawa, Alexander Knapp and the anonymous referees of TCS. Many thanks also go to Dirk Pattinson for explaining some fine points of fibred category theory.

Appendix A

Categorical Notions

For the convenience of the reader we collect some definitions and results of category theory. The notions of functor, natural transformation, (co)limit and adjunction are assumed to be familiar. But we try to mention all notions needed in this thesis that may not be contained in some introduction to category theory for computer scientists. The material presented here is based on the textbooks by Mac Lane [78], Adámek, Herrlich, Strecker [4], and Borceux [19].

An excellent introduction to algebras and coalgebras, monads and comonads, and their applications to the semantics of programming languages can be found in Turi [119].

The material presented here is used in chapter 1. Chapter 2 needs only the section on factorisation structures for sinks.

A.1 Miscellaneous Remarks

If not stated otherwise, we assume a category $\mathcal{C}$ to consist of a *class* of objects and, for all objects $A, B \in \mathcal{C}$, to have a *set* of morphisms $\mathcal{C}(A, B)$. A category where the class of objects actually is a set is called **small**. A *diagram* in $\mathcal{C}$ is a functor $D : \mathcal{I} \to \mathcal{C}$. A diagram is called small iff $\mathcal{I}$ is a small category. (Co)limits of small diagrams are called small as well, though we sometimes omit small here. In this work, when we speak e.g. of a subcategory as being **closed under coproducts** we mean closed under small coproducts. In our context, the distinction between a '*large* set of objects/morphisms' (= class of objects/morphisms) and a '*small* set of objects/morphisms' (= set of objects/morphisms) is crucial in order to prove the existence of final coalgebras.

The **dual** (or opposite) $\mathcal{C}^{\mathrm{op}}$ of a category $\mathcal{C}$ has the same objects and morphisms $\mathcal{C}^{\mathrm{op}}(A, B) = \mathcal{C}(B, A)$. Identities are the same in both categories, composition is reversed: Let $f \in \mathcal{C}^{\mathrm{op}}(A, B)$, $g \in \mathcal{C}^{\mathrm{op}}(B, C)$ then $g \circ^{\mathrm{op}} f = f \circ g$. Though $\mathcal{C}^{\mathrm{op}}(A, B) = \mathcal{C}(B, A)$ it is sometimes convenient to write f^{op} to indicate when we think of $f \in \mathcal{C}(B, A)$ as a morphism in $\mathcal{C}^{\mathrm{op}}(A, B)$.

A **multiple pullback** (dual notion: multiple pushout) is the limit of a (possibly large) family of morphisms $(f_i : A_i \to A)_{i \in I}$. If $|I| = 2$ then it is a pullback. Multiple pullbacks are also called **generalised pullbacks**. Their import for universal coalgebra has been investigated by Gumm and Schröder (see [46] or [44, 42]).

An **intersection** of a class of monos $m_i : A_i \to B$ is a mono $f : A \to B$ such that

(1) f factors through all m_i and (2) all $f' : A' \to B$ which factor through all m_i factor through f. Equivalently, intersections can be defined via multiple pullbacks of large mono-diagrams. The intersection f gives the largest subobject contained in all m_i. The dual notion is a **cointersection**. In **Set**, $f : B \to A$ is a cointersection of epis $e_i : B \to A_i$ iff the kernel[1] of f is the equivalence relation generated by the union of kernels of the e_i. Since the monos with codomain A and the epis with domain A form categories which are—up to equivalence—posets, intersection is a (bounded) meet and cointersection is a (bounded) join.

As usual, **complete** means complete with respect to small diagrams (i.e. limits of small diagrams exist) and finitely complete means complete w.r.t. finite diagrams. **Strongly complete** means complete and closed under intersections where intersections are allowed to be large (i.e. taken over a *class* of monos). Complete and wellpowered implies strongly complete. (Dual notions: (strongly/finitely) cocomplete.)

For the purposes of this work it will also be fruitful to use the notion of a concrete category (see e.g. Manes [81] or Adámek, Herrlich, Strecker [4]). A **concrete category** $(\mathcal{C}, U)$ is just a faithful functor $U : \mathcal{C} \to \mathcal{X}$. The idea is that $\mathcal{C}$ is a category of structures (here usually coalgebras) over 'carriers' in $\mathcal{X}$. That U is faithful has as a consequence that a morphism $f : A \to B \in \mathcal{C}$ is determined by the corresponding morphism $Uf : UA \to UB$ in the base category $\mathcal{X}$. (It is often convenient to identify Uf and f.) Whereas the theory of chapter 2 can be developed concentrating only on properties of the **abstract category**[2] $\mathcal{C}$, the investigations in section 1.4 show that the characteristic flavour of coalgebras depends on properties of the functor U which cannot be described internally to $\mathcal{C}$.

For example, the next section discusses the difficulties to define the notions of quotient and subobject as internal to an abstract category $\mathcal{C}$. In concrete categories there exists satisfying concepts, namely those of final epis and initial monos which will be defined next.

Definition A.1.1 (initial morphism). Let $U : \mathcal{C} \to \mathcal{X}$ be faithful functor. $m : A \to B \in \mathcal{C}$ is initial if for all $C \in \mathcal{C}$ and all $g : UC \to UA \in \mathcal{X}$ it holds that $m \circ g \in \mathcal{C}$ implies $g \in \mathcal{C}$.[3]

And, dually, a morphism is final in $\mathcal{C}$ iff it is initial in $\mathcal{C}^{\mathrm{op}}$, to spell it out:

Definition A.1.2 (final morphism). Let $U : \mathcal{C} \to \mathcal{X}$ be faithful functor. $e : B \to A \in \mathcal{C}$ is final if for all $C \in \mathcal{C}$ and all $g : UA \to UC \in \mathcal{X}$ it holds that $g \circ e \in \mathcal{C}$ implies $g \in \mathcal{C}$.

Concerning the usual categories of (co)algebras over **Set** the surjective/injective morphisms are precisely the final epis/initial monos. In general, final and epi in the base (resp. initial and mono in the base) gives a good notion of quotient (resp. subobject) in concrete categories. For example, in the category of topological spaces a morphism is final epi iff it is surjective and the codomain is equipped with the quotient topology and a morphisms is initial mono iff it is injective and the domain is equipped with the subspace topology.

[1]The **kernel** of a function $f : B \to A \in$ **Set** is the equivalence relation $\sim \subset B \times B$ defined by $b \sim b'$ iff $f(b) = f(b')$. In categories not based on sets the concept of a kernel is replaced by that of a kernel pair, sometimes also simply denoted by kernel.

[2]We speak of $\mathcal{C}$ as an abstract category when we want to emphasise that the properties discussed do not depend on a 'forgetful' (i.e. faithful) functor $U : \mathcal{C} \to \mathcal{X}$ (for some 'base' category $\mathcal{X}$).

[3]Using faithfulness, by saying $m \circ g \in \mathcal{C}$ one means 'there is $f \in \mathcal{C}$ such that $Uf = Um \circ g$' and '$g \in \mathcal{C}$' means 'there is $g' \in \mathcal{C}$ such that $Ug' = g$'.

Concerning functors we write $\mathrm{Id}_{\mathcal{C}}$ for the identity functor (often dropping the subscript) and K_A for the constant functor $K_A : \mathbf{1} \to \mathcal{C}$ mapping the single object of the terminal category $\mathbf{1}$ to $A \in \mathcal{C}$. It is often convenient to abbreviate K_A by A.

For the applications of coalgebra the multiplicative and polynomial functors play an important role. **Multiplicative functors** on $\mathcal{C}$ are given by the following abstract syntax ($A \in \mathcal{C}$)

$$F ::= \mathrm{Id} \mid A \mid F \times F \mid F^A$$

and **polynomial functors** by

$$F ::= \mathrm{Id} \mid A \mid F \times F \mid F + F \mid F^A$$

Some authors allow polynomial functors to be also built from the (finite) powerset.

Given two functors $F : \mathcal{A} \to \mathcal{C}$, $G : \mathcal{B} \to \mathcal{C}$ the **comma category** $F{\downarrow}G$ has objects (A, h, B) with $A \in \mathcal{A}$, $B \in \mathcal{B}$, $h : FA \to GB$. A morphism $(A, h, B) \to (A', h', B')$ is a pair $(f : A \to A', g : B \to B')$ such that $h' \circ Ff = Gg \circ h$. Of interest for us are the special cases $U{\downarrow}K_X$, where $U : \mathcal{C} \to \mathcal{X}$ and $X \in \mathcal{X}$, and $\mathrm{Id}_{\mathcal{C}}{\downarrow}K_A$ and $K_A{\downarrow}\mathrm{Id}_{\mathcal{C}}$, where $A \in \mathcal{C}$ which we will describe explicitly.
$U{\downarrow}K_X$ is usually denoted as $U{\downarrow}X$ with objects (A, h), $h : UA \to X$, and morphisms $f : (A, h) \to (A', h')$ where $f : A \to A' \in \mathcal{C}$ and $h' \circ Uf = h$.
$\mathrm{Id}_{\mathcal{C}}{\downarrow}K_A$ is usually denoted as $\mathcal{C}{\downarrow}A$ (or $\mathcal{C}/A$) with objects $h : B \to A$, $B \in \mathcal{C}$, and morphisms $f : (h : B \to A) \to (h : B' \to A)$ where $f : B \to B' \in \mathcal{C}$ and $h' \circ f = h$.
$K_A{\downarrow}\mathrm{Id}_{\mathcal{C}}$ is usually denoted as $A{\downarrow}\mathcal{C}$ (or $A\backslash\mathcal{C}$) with objects $h : A \to B$, $B \in \mathcal{C}$, and morphisms $f : (h : A \to B) \to (h : A \to B')$ where $f : B \to B' \in \mathcal{C}$ and $f \circ h = h'$.

Speaking of morphisms with common domain $A \in \mathcal{C}$ (resp. codomain) as isomorphic, we refer to isomorphic as objects in $A{\downarrow}\mathcal{C}$ (resp. $\mathcal{C}{\downarrow}A$).

$\mathcal{C}$ is **wellpowered** iff every object $A \in \mathcal{C}$ has—up to isomorphism—only a set of monos with codomain A (i.e. $\mathcal{C}{\downarrow}A$ is small (up to iso)). $\mathcal{C}$ is called **cowellpowered** iff every object $A \in \mathcal{C}$ has—up to isomorphism—only a set of epis with domain A.

A.2 Classes of Morphisms

Given a faithful functor $U : \mathcal{C} \to \mathbf{Set}$, one may want to say that $f \in \mathcal{C}$ is a surjective morphism iff Uf is surjective (= epi in **Set**). This implies that f is epi in $\mathcal{C}$ (because of U faithful) but the converse does not hold in general: f epi in $\mathcal{C}$ does not imply Uf epi in **Set**. E.g., in the categories of monoids the injection of the natural numbers[4] into the integers is epi but not surjective. A consequence of this observation is that, in general, one cannot simply identify quotients and epis. Similarly, subobjects and monos need not coincide (for an algebraic example see Manes [81], chapter 2, (1.47), for a coalgebraic example see example 1.3.1 which is due to Gumm and Schröder [42]). Therefore, several strengthenings of the notions of epi and mono have been invented. For our work are most important the

[4] With 0 as neutral element and addition as composition of the monoid.

extremal, strong, and regular monos/epis (we only give the definitions for monos, the ones for epis being dual).

A mono m is **extremal** iff $m = f \circ e$ for some f and some epi e implies that e is iso.

A mono is **strong** iff for all epis e and all f, g with $m \circ f = g \circ e$ there is a unique d (called the diagonal fill-in) such that

$$\begin{tikzcd} \bullet \arrow[r, "e"] \arrow[d, "f"'] & \bullet \arrow[d, "g"] \arrow[dl, dotted, "d"] \\ \bullet \arrow[r, "m"] & \bullet \end{tikzcd}$$

commutes.

A mono is **regular** iff it is an equaliser.

A mono is **split** (also called a section) iff it has a left-inverse, i.e. there is f such that $f \circ m = \mathrm{id}$. In that case f is a split epi (also called a retraction).

We write $\mathit{Mono}(\mathcal{C})$, $\mathit{ExtrMono}(\mathcal{C})$, $\mathit{StrongMono}(\mathcal{C})$, $\mathit{RegMono}(\mathcal{C})$, $\mathit{SplitMono}(\mathcal{C})$ for the classes of monos, extremal monos, strong monos, regular monos, and split monos of $\mathcal{C}$, respectively. A similar notation is used for classes of epis. If the category $\mathcal{C}$ is understood from the context (or can be just an arbitrary one) we write just *Mono*, etc.

Some facts on these classes of morphisms:

Split implies regular implies strong implies extremal.

f is extremal mono and epi iff f is iso.[5]

The classes differ with respect to closure properties. *Mono*, *StrongMono*, and *SplitMono* are closed under composition (and *ExtrMono* and *RegMono* in general not). *Mono*, *StrongMono*, and *RegMono* are closed under pullbacks (and *ExtrMono* and *SplitMono* in general not). Only *Mono* and *StrongMono* are always closed under intersections. All but *RegMono* are always closed under left-cancellation[6].

In **Set** the different classes of monos (and epis) mentioned above coincide, with the exception that a mono with empty domain and nonempty codomain is not split.

Here is a proposition stating further relationships between these classes of morphisms.

Proposition A.2.1.

1. *If $\mathcal{C}$ has pushouts (or if $\mathcal{C}$ has equalisers and intersections) then $\mathit{ExtrMono} = \mathit{StrongMono}$. ([4], 14C)*

[5] In general, epi and mono does not imply iso: consider again the inclusion of the monoid of natural numbers into the monoid of integers.

[6] M is closed under left-cancellation iff $f \circ g \in M \;\Rightarrow\; g \in M$.

2. *If (Epi, M) is a factorisation system and $M \subset Mono$ then $M = ExtrMono = StrongMono$. ([4], 14C)*

3. *If $\mathcal{C}$ has equalisers and pushouts (or if $\mathcal{C}$ is strongly complete) and $RegMono$ is closed under composition then $ExtrMono = RegMono$. ([4], 12B)*

4. *$Epi = StrongEpi$ implies $Mono = ExtrMono$. Dually, $Mono = StrongMono$ implies $Epi = ExtrEpi$.*

5. *If $\mathcal{C}$ has $(Epi, StrongMono)$-factorisations and weak pullbacks then $Epi = ExtrEpi$ implies $Mono = StrongMono$.*

We conclude with a useful proposition linking pullbacks and monos (or, dually, pushouts and epis). It will be used at many places in this work without further mentioning.

Proposition A.2.2. *The diagram*

$$\begin{tikzcd} \bullet \arrow[r, "f"] \arrow[d, "f"'] & \bullet \arrow[d, "g"] \\ \bullet \arrow[r, "g"] & \bullet \end{tikzcd}$$

1. *is a weak pullback iff g is mono and f is split epi,*

2. *is a pullback iff g is mono and f iso.*

This simple proposition has an interesting corollary for the theory of coalgebras:

Corollary A.2.3. *Let $\mathcal{X}$ be a category with weak pullbacks and Ω a functor on $\mathcal{X}$. Then Ω preserves monos if it preserves weak pullbacks.*

Remark. It would be enough to require $\mathcal{X}$ to have and Ω to preserve weak kernel pairs of monos.

The proposition above also holds if we replace g by a mono source. For future reference we will state this explicitly (in the dualised version).

Proposition A.2.4. *Let (s_i) be a sink. Then (Q, f, f) in the diagram below*

$$\begin{tikzcd} \bullet \arrow[r, "(s_i)"] \arrow[d, "(s_i)"'] & \bullet \arrow[d, "f"] \\ \bullet \arrow[r, "f"] & Q \end{tikzcd}$$

1. *is a weak colimit iff the sink (s_i) is epi and f is split mono,*

2. *is a colimit iff the sink (s_i) is epi and f iso.*

A.3 Creation of (Co)limits

Intuitively, the fact that a forgetful functor $U : \mathcal{C} \to \mathcal{X}$ creates colimits means that colimits in $\mathcal{C}$ can be calculated as colimits in $\mathcal{X}$ (note that, in general, the base category $\mathcal{X}$ is simpler than the category $\mathcal{C}$).

A **weak colimit**, and a weak colimiting cocone, are defined like colimit (and colimiting cocone) but the morphism from the colimiting cocone to another cocone need not be unique.

Given a functor $U : \mathcal{C} \to \mathcal{X}$ and a diagram $D : I \to \mathcal{C}$ one says that

- U **creates colimits** if for every colimiting cocone $(c_i : UDi \to X)_{i \in I}$ of UD in $\mathcal{X}$ there are unique morphisms $d_i : Di \to C$ in $\mathcal{C}$ such that $Ud_i = c_i$, and if, moreover, $(d_i : Di \to C)_{i\in I}$ is a colimiting cocone for D.
- U **preserves (weak) colimits** if for every (weak) colimiting cocone $(d_i : Di \to C)_{i\in I}$ of D in $\mathcal{C}$ it holds that $(Ud_i : UDi \to UC)_{i\in I}$ is a (weak) colimiting cocone of UD in $\mathcal{X}$.
- U **reflects colimits** if morphisms $(d_i : Di \to C)_{i\in I}$ in $\mathcal{C}$ are a colimit of D in $\mathcal{C}$ whenever $(Ud_i : UDi \to UC)_{i\in I}$ is a colimit if UD in $\mathcal{X}$.

Note that in the definition of creation the uniqueness requirement is only w.r.t. $Ud_i = c_i$ (meaning that colimits in $\mathcal{C}$ are calculated as colimits in $\mathcal{X}$).

The notions of creating, preserving, reflecting (weak) limits are defined dually.

A functor $U : \mathcal{C} \to \mathcal{X}$ **weakly preserves** a colimit if it maps colimiting cocones to weak colimiting cocones. If a category has a type of colimit then U preserves weak colimits of this type iff it preserves colimits of this type weakly.

If U creates colimits then it preserves and reflects them.

A standard result for coalgebras is the following (see, e.g., Barr [12]). Because of its importance we sketch a proof.

Proposition A.3.1. *Let Ω be an endofunctor on $\mathcal{X}$. Then $U : \mathcal{X}_\Omega \to \mathcal{X}$ creates colimits. Also, U creates each type of limit which is preserved by Ω.*

Proof. Let $D : I \to \mathcal{X}_\Omega$ be a diagram and $(c_i : UDi \to X)_{i\in I}$ be a colimiting cocone of UD in $\mathcal{X}$. First, we have to show that X can be equipped uniquely with a structure $\xi : X \to \Omega X$ such that the c_i are morphisms $Di \to (X, \xi)$ in $\mathcal{X}_\Omega$. Existence and uniqueness of ξ follow because X is a colimit. Second, we have to check that (X, ξ) is a colimit. Let $f_i : Di \to A$ be a cocone for D. Uf_i is a cocone in $\mathcal{X}$ hence there is a unique $h : UX \to UA$ such that $h \circ c_i = Uf_i$. h is a morphism $(X, \xi) \to A$ since the sink $(c_i)_{i\in I}$ is epi. h is uniquely determined as a morphisms in $\mathcal{X}_\Omega$ since U is faithful.

The second statement is proved analogously. To show existence and uniqueness of ξ use that ΩX is a limit. To show that the mediating morphism h is indeed a coalgebra morphism use that the source $(\Omega c_i)_{i\in I}$ is mono since $(c_i)_{i\in I}$ is mono and mono sources are preserved by Ω since Ω preserves limits (see [4] 13.5(2)). $\square$

Remark. The proof also shows that, in case that Ω preserves a type of limits weakly, U lifts limiting cones to just cones. As a corollary, if Ω additionally preserves small mono sources then U lifts limits to weak limits.

Remark. The proposition and remark above make no assumption that the (co)limits be small.

An essential corollary is the following (Rutten [109]4.7.1).

Corollary A.3.2 (U preserves epis). *Let Ω be an endofunctor on $\mathcal{X}$. Then $U : \mathcal{X}_\Omega \to \mathcal{X}$ preserves and reflects epis.*

Proof. U reflects epis because it is faithful. To show that U preserves epis use the characterisation of epis in proposition A.2.4 and that U preserves colimits. $\square$

Remark. In the case of $\mathcal{X} = \mathbf{Set}$ the proposition states that a coalgebra morphisms is epi iff it is surjective.

Remark. The same proof shows that $U : \mathcal{X}_\Omega \to \mathcal{X}$ preserves and reflects epi sinks.

Corollary A.3.3. *Let Ω be a weak pullback preserving endofunctor on $\mathcal{X}$. Then $U : \mathcal{X}_\Omega \to \mathcal{X}$ creates kernel pairs of morphisms which are mono in $\mathcal{X}$.*

Proof. Ω preserves kernel pairs of monos in $\mathcal{X}$ (see corollary A.2.3), hence U creates them. $\square$

Note that it does not follow from the proposition that U preserves monos in $\mathcal{X}_\Omega$. The following proposition is Rutten [109]4.7.2 though the proof given there does not work because the argument only shows that U preserves kernels of morphisms which are mono in the base and not that U preserves kernels of morphisms which are mono as coalgebra morphisms. The proof given below is due to Gumm and Schröder [40].

Proposition A.3.4. *Let Ω be a weak pullback preserving endofunctor on a category $\mathcal{X}$ with pullbacks. Then $U : \mathcal{X}_\Omega \to \mathcal{X}$ preserves and reflects monos.*

Proof. U reflects monos because U is faithful. To show that U preserves monos let $g \in \mathcal{X}_\Omega$ be mono. Consider the kernel pair (R, p, q) of Ug. Since Ω preserves weak pullbacks, there is a coalgebra structure on R that makes p, q into morphism in $\mathcal{X}_\Omega$. Since g mono, it follows $p = q$, hence Ug mono (see proposition A.2.2). $\square$

Remark. It follows that, in the case of $\mathcal{X} = \mathbf{Set}$ and Ω preserving weak pullbacks, coalgebra morphisms that are mono are also injective.

Another corollary concerns the creation of equalisers. Recall that a *cokernel pair* (p, q) of a morphism f is a pushout

$$\begin{tikzcd} \bullet \arrow[r, "f"] \arrow[d, "f"'] & \bullet \arrow[d, "p"] \\ \bullet \arrow[r, "q"] & \bullet \end{tikzcd}$$

(And, dually, a *kernel pair* is a pullback.) Also recall that m is a **split equaliser** of p, q iff there are morphisms s, t such that $s \circ m = \mathrm{id}$, $t \circ p = \mathrm{id}$, and $t \circ q = m \circ s$. Split equalisers are preserved by all functors, that is, they are *absolute equalisers.* A functor $U : \mathcal{C} \to \mathcal{X}$ is said to **create split equalisers** if U creates those equalisers which are split in $\mathcal{X}$ (they are *not* required to be split as equalisers in $\mathcal{C}$).

Corollary A.3.5. *Let Ω be an endofunctor on $\mathcal{X}$ and suppose that in $\mathcal{X}$ equalisers of cokernel pairs are split. Then $U : \mathcal{X}_\Omega \to \mathcal{X}$ creates equalisers of cokernel pairs.*

Proof. Let (p, q) be a cokernel pair of $f \in \mathcal{X}_\Omega$. Since U preserves colimits, (p, q) is a cokernel pair of f in $\mathcal{X}$. Let m be the equaliser of p, q in $\mathcal{X}$. Since the equaliser is split, it is preserved by Ω and hence created by U. □

Remark. Let Ω be an endofunctor on **Set**. Then $U : \mathbf{Set}_\Omega \to \mathbf{Set}$ creates equalisers of cokernel pairs. — Proof: In **Set**, equalisers of cokernel pairs are split if the domain of the equaliser is non-empty. The case of equalisers with empty domain and non-empty codomain (they are not split) has to be checked separately.

A.4 Factorisation Systems

We have already hinted at the problem of defining quotients and subobjects in categories. One possible solution is to axiomatise these notions by introducing factorisation systems. This section is based on Adámek, Herrlich, Strecker [4], chapter 14.

Definition A.4.1. Let E, M be classes of morphisms in $\mathcal{C}$. (E, M) is called a **factorisation system for** $\mathcal{C}$ and $\mathcal{C}$ is called an (E, M)-category iff

1. E, M are closed under isomorphism.

2. $\mathcal{C}$ **has** (E, M)**-factorisations**, i.e. every morphism f in $\mathcal{C}$ has a factorisation $f = m \circ e$ for some $m \in M$ and $e \in E$. We call m the **image** of f and e the **kernel** of f.

3. $\mathcal{C}$ has **the unique** (E, M)**-diagonalisation property**, i.e. whenever the square

$$\begin{tikzcd} A \arrow[r, "e"] \arrow[d, "f"'] & B \arrow[d, "g"] \arrow[dl, dotted, "d"] \\ C \arrow[r, "m"'] & D \end{tikzcd}$$

commutes for $m \in M$, $e \in E$, then there is a unique diagonal d making the triangles commute.

Remark. The definition does not require $M \subset Mono$ or $E \subset Epi$ though this will be the case in all cases of interest for our work.

In an (E, M)-category the morphisms in E are called **quotients**, or **E-quotients**, and the morphisms in M are called **subobjects**, or **M-subobjects**. The For a morphism $f \in \mathcal{C}$ factoring as $f = m \circ e$, $m \in M$, $e \in E$, we call m the **image** of f.

A full subcategory $\mathcal{B} \subset \mathcal{C}$ is **closed under quotients** or **closed under quotients images of morphisms** iff for all $B \in \mathcal{B}$ and all $e : B \to C \in E$ also $C \in \mathcal{B}$.

We also adapt the notions of wellpowered and cowellpowered to factorisation systems. An (E, M)-category $\mathcal{C}$ is called **M-wellpowered** iff for every object $A \in \mathcal{C}$ there is—up to isomorphism—only a set of subobjects of A. And $\mathcal{C}$ is called **E-cowellpowered** iff there is—up to isomorphism—only a set of quotients of A.

The definition of factorisation systems given above has the advantage that it emphasises the property of unique diagonalisation which is the main ingredient in many proofs using factorisation systems. The following definition gives an alternative characterisation.

Proposition A.4.2. *(E, M) is a factorisation system iff*

1. *$i \in E \cap M$ if i iso .*
2. *E, M are closed under composition.*
3. *(E, M)-factorisations are unique up to isomorphism.*[7]

The next proposition collects further facts on factorisation systems.

Proposition A.4.3. *Let (E, M) be a factorisation system.*

1. *$i \in E \cap M$ iff i iso.*
2. *Let $e, e' \in E$, $m, m' \in M$. If the diagrams (they are each others dual)*

$$\begin{tikzcd} \bullet \arrow[r, "e"] \arrow[d, "\mathrm{id}"'] & \bullet \arrow[d, "m"] \arrow[dl, "d"'] \\ \bullet \arrow[r, "f"'] & \bullet \end{tikzcd} \qquad\qquad \begin{tikzcd} \bullet \arrow[r, "f'"] \arrow[d, "e'"'] & \bullet \arrow[d, "\mathrm{id}"] \arrow[dl, "d'"'] \\ \bullet \arrow[r, "m'"'] & \bullet \end{tikzcd}$$

   commute then e is iso and $f \in M$ as well as m' is iso and $f' \in E$.

3. *If $e \in E$ and e split mono, then e is iso. If $m \in M$ and m split epi then m is iso.*
4. *M determines E (and vice versa): Let $e \in \mathcal{C}$ and consider the diagram in definition A.4.1(3). If for all $m \in M$ and all $f, g \in \mathcal{C}$ such that the square commutes there is a diagonal d making the triangles commute then $e \in E$.*

[7]That is, for $m \circ e = m' \circ e'$ being two factorisations, there is a unique iso i such that $i \circ e = e'$ and $m' \circ i = m$.

5. *M is stable under pullbacks and closed under products.*[8] *$M \cap Mono$ is closed under intersections. Dually, E is stable under pushouts and closed under coproducts. $E \cap Epi$ is closed under cointersections, [4]14.15.*

6. *If $E = Epi$ and $M \subset Mono$ then $M = ExtrMono = StrongMono$, [4]14C(f).*

7. *$M \subset Mono$ implies $ExtrEpi \subset E$, [4]14.10.*

8. *$M \subset Mono$ if $SplitEpi \subset E$ and $\mathcal{C}$ has binary coproducts, [4]14.11.*

The next proposition shows how to obtain $(Epi, RegMono)$-factorisations in case that the category has pushouts and equalisers.

Proposition A.4.4. *Let $(Epi, RegMono)$ be a factorisation system for a category $\mathcal{C}$ that has pushouts and equalisers. Then a factorisation $f = m \circ e$, $m \in RegMono$, $e \in Epi$, of $f \in \mathcal{C}$ can be obtained by taking the cokernel pair p, q of f and letting m the be equaliser of p, q. e is then the unique morphism making*

$$\begin{tikzcd}
\bullet \arrow[rr, "f"] \arrow[dr, "e"'] & & \bullet \arrow[r, shift left, "p"] \arrow[r, shift right, "q"'] & \bullet \\
& \bullet \arrow[ur, "m"'] & &
\end{tikzcd}$$

commute.

Proof. It has to be shown that e is epi. Let $x, y \in \mathcal{C}$ such that $x \circ e = y \circ e$ and let m' be the equaliser of x, y. There is h such that $m' \circ h = e$. Since p, q is the cokernel of f and $p \circ m = q \circ m$ and $p \circ m \circ m' = q \circ m \circ m'$ and $f = m \circ m' \circ h$ it follows that p, q is the cokernel pair of m as well as of $m' \circ m$. Since $(Epi, RegMono)$ is a factorisation system, regular monos are closed under composition, hence $m' \circ m$ is an equaliser. Since an equaliser is the equaliser of its cokernel pair it follows that m' is iso, hence $x = y$. $\square$

Existence of Factorisation Systems

For a proposition relating the different classes of morphisms see proposition A.2.1. For proofs of the following proposition see [4], 14.17, 14.19, 14.22.

Proposition A.4.5. *Let $\mathcal{C}$ be a category.*

1. *$(Epi, ExtrMono)$ is a factorisation system, if $\mathcal{C}$ is finitely cocomplete and has cointersections, or, if $\mathcal{C}$ has equalisers and intersections.*

2. *If $\mathcal{C}$ has equalisers and pushouts then $(Epi, RegMono)$ is a factorisation system iff $RegMono$ is closed under composition.*

[8] Closure under products: For $m_i : A_i \to B_i \in M$ also $\prod m_i : \prod A_i \to \prod B_i \in M$.

Factorisation Systems in Subcategories

In contrast to the notions surjective and injective, the notions epi and mono are not local: they depend on the whole category and change their meaning when we restrict our attention to subcategories.

We still have the following: Let (E, M) be a factorisation system for $\mathcal{C}$ and $\mathcal{B} \subset \mathcal{C}$ a full subcategory that is closed under quotients (or subobjects). Then $(E \cap \mathcal{B}, M \cap \mathcal{B})$ is a factorisation system for $\mathcal{B}$. — But note that, in general, $E \cap \mathcal{B}$, $M \cap \mathcal{B}$ do not inherit from E and M properties like being the class of epis or strong monos.

It is a special property of $(\mathit{Epi}, \mathit{RegMono})$ factorisation systems that they behave nicely w.r.t. taking certain subcategories.

Proposition A.4.6. *Let $(\mathit{Epi}(\mathcal{C}), \mathit{RegMono}(\mathcal{C}))$ be a factorisation system for $\mathcal{C}$ and $\mathcal{B} \subset \mathcal{C}$ a full subcategory closed under quotients.Then $(\mathit{Epi}(\mathcal{B}), \mathit{RegMono}(\mathcal{B}))$ is a factorisation structure for $\mathcal{B}$.*

Proof. Every morphisms $f \in \mathcal{B}$ factors as $f = m \circ e$ with $e \in \mathit{Epi}(\mathcal{C}) \cap \mathcal{B}$, $m \in \mathit{RegMono}(\mathcal{C}) \cap \mathcal{B}$ since $\mathcal{B}$ is closed under quotients and full. Clearly, $\mathit{Epi}(\mathcal{C}) \cap \mathcal{B} \subset \mathit{Epi}(\mathcal{B})$. But we have to show that $m \in \mathit{RegMono}(\mathcal{B})$. There are morphisms $f_1, f_2 \in \mathcal{C}$ such that m is their equaliser. Consider factorisations $f_1 = m_1 \circ e_1$ and $f_2 = m_2 \circ e_2$. There is an iso d such that

$$
\begin{tikzcd}
 & \bullet \arrow[dd, "d"] \arrow[dr, "m_1"] & \\
\bullet \arrow[ur, "e_1"] \arrow[dr, "e_2"'] & & \bullet \\
 & \bullet \arrow[ur, "m_2"'] &
\end{tikzcd}
$$

commutes. Since $\mathcal{B}$ is closed under quotients, it holds $d \circ e_1, e_2 \in \mathcal{B}$. Since m_2 is mono, m is an equaliser of $d \circ e_1$ and e_2, hence $m \in \mathit{RegMono}(\mathcal{B})$. Unique diagonalisation is due to the regularity of the monos. □

Remark. As the proof shows $\mathit{Epi}(\mathcal{B}) \supset \mathit{Epi}(\mathcal{C}) \cap \mathcal{B}$, $\mathit{RegMono}(\mathcal{B}) = \mathit{RegMono}(\mathcal{C}) \cap \mathcal{B}$. The inclusion is proper in general.

A.5 Factorisation Structures for Sinks

Factorisation systems are used to deal abstractly with the notions of subobject and quotient. Factorisation structures for sinks are used to generalise unions of subobjects. (And factorisation structures for sources can be used for intersections of quotients.) This section is based on Adámek, Herrlich, Strecker [4], chapter 15.

A **sink** $\mathsf{s} = (A, (s_i)_{i \in I})$ in a category $\mathcal{C}$ consists of an object $A \in \mathcal{C}$ and a family of morphisms $s_i : A_i \to A \in \mathcal{C}$ with common codomain A and indexed by a class I.

Remark. The dual notion is that of a **source**. It will be mentioned only occasionally.

Remark. I may be empty (that is the reason why the codomain A had to be made explicit in the definition of a sink). If the domain and the indexing class of the sink is clear from the context we often denote the sink simply by (s_i).

We can adapt to sinks most of the terminology and definitions for morphisms. The collection (A_i) is called the **domain** of s. Composition $f \circ \mathsf{s}$ of a morphism f and a sink s and composition $\mathsf{s} \circ (\mathsf{t}_i)$ of a sink $\mathsf{s} = (A, (s_i : A_i \to A)_{i \in I})$ with a family of sinks $\mathsf{t}_i = (A_i, (t_{ij} : A_{ij} \to A_i)_{j \in J_i})$ are defined in the obvious way. The definitions of **epi sink**, or **extremal epi sink**, or **strong epi sink** follow the respective definitions for morphisms. An important example of extremal epi sinks are the colimiting cocones.

Definition A.5.1 (factorisation structure for sinks). Let $\mathcal{E}$ be a collection[9] of sinks and M a class of morphisms in $\mathcal{C}$. $(\mathcal{E}, M)$ is called a factorisation structure for $\mathcal{C}$ and $\mathcal{C}$ is called an $(\mathcal{E}, M)$-category iff

1. $\mathcal{E}, M$ are closed under isomorphisms.
2. $\mathcal{C}$ **has** $(\mathcal{E}, M)$**-factorisation** of sinks, i.e. every sink s in $\mathcal{C}$ has a factorisation $\mathsf{s} = m \circ \mathsf{e}$ for some $m \in M$ and $\mathsf{e} \in \mathcal{E}$.
3. $\mathcal{C}$ has **the unique** $(\mathcal{E}, M)$**-diagonalisation property**, i.e. whenever for $m \in M$, $(e_i) \in \mathcal{E}$, $f \in \mathcal{C}$, and a sink (s_i) in $\mathcal{C}$, the square

$$\begin{tikzcd} A \arrow[r, "e_i"] \arrow[d, "s_i"'] & B \arrow[d, "f"] \arrow[dl, dotted, "d"'] \\ C \arrow[r, "m"'] & D \end{tikzcd}$$

commutes for all $i \in I$ then there is a unique diagonal d making the triangles commute.

Remark. Every factorisation structure $(\mathcal{E}, M)$ for sinks induces a factorisation system (E, M) by letting E be the class of sinks consisting of single morphisms. Hence notions like $\mathcal{E}$-quotients, $\mathcal{E}$-cowellpowered, etc denote E-quotients, E-cowellpowered, etc.

Remark. A subcategory $\mathcal{B} \subset \mathcal{E}$ is **closed under $\mathcal{E}$-sinks** iff for all sinks $(e_i : B_i \to C)$ with domain in $\mathcal{B}$ (i.e. $B_i \in \mathcal{B}$) also $C \in \mathcal{B}$.

Remark. It is probably worth spelling out what conditions 2 and 3 mean for empty sinks.

2. For every $C \in \mathcal{C}$ there are $(A, \{\}) \in \mathcal{E}$ and $m : A \to C \in M$.
3. For every sink $(B, \{\}) \in \mathcal{E}$, every $m : C \to D \in M$ and every $f : B \to D$ there is a unique $d : B \to C$ such that $m \circ d = f$.

In particular, $\mathcal{E}$ and M are not empty for every non-empty $(\mathcal{E}, M)$-category. Moreover, the empty sinks in $\mathcal{E}$ are precisely the empty sinks that are projective w.r.t. all $m \in M$.

[9]Since every sink may be indexed by a class there may be more than class-many sinks in $\mathcal{E}$.

Example A.5.2. **Set** is an $(EpiSink, Mono)$-category. Given a sink (s_i) and a factorisation $(s_i) = m \circ (e_i)$, (the image of) m is the union of the images of the s_i. The empty set is the only empty sink.

Some important facts on factorisation structures are collected in the next proposition (see [4] for proofs).

Proposition A.5.3. *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category.*

1. *$m \in M$ only if m mono.*
2. *$(\mathcal{E}, M)$-factorisations are unique up to isomorphism.*
3. *$\mathcal{E}, M$ are closed under composition.*
4. *s extremal-epi sink only if $\mathsf{s} \in \mathcal{E}$.*
5. *$m \in M$ and $m \in \mathcal{E}$ only if m iso.*
6. *A sink belongs to $\mathcal{E}$ iff every M-morphism through which it factors is an iso.*
7. *$f \circ g \in M$ and f mono only if $g \in M$.*
8. *M is stable under pullbacks.*
9. *M is closed under intersections.*

In general, sinks in $\mathcal{E}$ need not be epi. The following proposition (see [4]15.7) characterises the factorisation structures for which sinks in $\mathcal{E}$ are epi.

Proposition A.5.4. *Let $\mathcal{C}$ be an $(\mathcal{E}, M)$-category. Then sinks in $\mathcal{E}$ are epi iff $\mathcal{C}$ has equalisers and $RegMono(\mathcal{C}) \subset M$.*

Proof. "only if": Let $f, g : A \to B$ in $\mathcal{C}$ and consider the sink $(s_i : A_i \to A)$ of all s_i with $f \circ s_i = g \circ s_i$. Now, factoring (s_i) as $m \circ (e_i)$, m is the equaliser of f, g: $f \circ m = g \circ m$ because (e_i) is epi; the universal property follows from definition of (s_i) and m mono. That an arbitrary equaliser m is in M can be seen by factoring m as $m' \circ e'$: m' is an equaliser (use e' epi and m' mono), hence e' is iso, hence m is M.
"if": Let $(e_i) \in \mathcal{E}$ and $f, g \in \mathcal{C}$ with $f \circ (e_i) = g \circ (e_i)$. Let h be the equaliser of f, g. Since h is an equaliser, (e_i) factors through h, and since $h \in M$, it follows from proposition A.5.3(6) that h is iso, hence $f = g$. ☐

Remark. The use made in "only if" of factorisation structures for sinks to construct limits is a typical example for the use of sinks and also appears in section 1.6 to obtain limits in categories of coalgebras and in [71] to show the existence of cartesian liftings in cofibrations of coalgebras.

Existence of Factorisation Structures

Under assumptions that are usually met in the context of coalgebras a factorisation system (E, M) can be extended uniquely to a factorisation structure for sinks $(\mathcal{E}, M)$ (for a full proof of the following proposition see [4], 15.21):

Proposition A.5.5. *Let (E, M) be a factorisation system for a category $\mathcal{C}$ that has coproducts and is M-wellpowered. Moreover, assume that the morphisms in M are mono. Then there is a unique collection $\mathcal{E}$ of sinks such that $(\mathcal{E}, M)$ is a factorisation structure for sinks.*

Proof. We sketch the part of the proof showing how $(\mathcal{E}, M)$-factorisations are obtained from (E, M)-factorisations. Let $(s_i : A_i \to A)_{i \in I}$ be a sink in $\mathcal{C}$ and let $A_i \xrightarrow{e_i} A_i' \xrightarrow{m_i} A$ be an (E, M)-factorisation of each s_i. Since $\mathcal{C}$ is M-wellpowered there are $J \subset I$, $e_i' : A_i \to \Sigma_{j \in J} A_j$, $g : \Sigma_{j \in J} A_j \to A$ such that $(s_i) = g \circ (e_i')$. Now, (E, M)-factoring g as $m \circ e$ shows that (s_i) has an $(\mathcal{E}, M)$-factorisation $m \circ (e \circ (e_i'))$. □

Remark. The empty sinks $(B, \{\})$ in $\mathcal{E}$ are obtained by factoring the morphisms $0 \to A$ (where 0 the initial object). It follows that in the case of $\mathcal{C} = \mathbf{Set}$ the only empty sink in $(\mathit{EpiSink}, \mathit{Mono})$ is $(\{\}, \{\} \to \{\})$.

Also not needed in this thesis it might be interesting to note further facts on the existence of factorisation structures for sinks (see [4], 15.10, 15.24, 15.25).

Proposition A.5.6. *Let $\mathcal{C}$ be a category. Then:*

- *If in $\mathcal{C}$ every sink has an $(\mathit{EpiSink}, \mathit{Mono})$-factorisation then $\mathcal{C}$ is an $(\mathit{EpiSink}, \mathit{ExtrMono})$-category.*
- *If $\mathcal{C}$ is an $(\mathcal{E}, M)$-category and (E', M') is a factorisation system for $\mathcal{C}$ with $M' \subset M$ then (E', M') can be uniquely extended to a factorisation structure for sinks $(\mathcal{E}', M')$.*
- *If $\mathcal{C}$ is strongly cocomplete and ExtrMono-wellpowered then*
  - *$\mathcal{C}$ is an $(\mathit{EpiSink}, \mathit{ExtrMono})$-category.*
  - *if regular monos are closed under composition, then $\mathcal{C}$ is an $(\mathit{EpiSink}, \mathit{RegMono})$-category.*

A.6 Adjoint Functor Theorems

Adjoint functor theorems are useful in the context of coalgebras because they provide a general way to show the existence of final coalgebras (see Barr [12]). We state the general and special adjoint functor theorems (abbreviated GAFT and SAFT, respectively) and sketch the proofs because they show how terminal coalgebras are obtained as quotients of certain colimits.

A category $\mathcal{C}$ has **a set of generators** $\mathcal{G}$ iff for all $C \in \mathcal{C}$ the sink consisting of all morphisms $G \to A$, $G \in \mathcal{G}$ is epi.

A set $\mathcal{A}$ of objects of $\mathcal{C}$ is called **a weakly terminal set of objects** iff for all $C \in \mathcal{C}$ there is a morphism $C \to A$ for some $A \in \mathcal{A}$.

Recall that a functor $U : \mathcal{C} \to \mathcal{X}$ has a right adjoint iff for all $X \in \mathcal{X}$ the category[10] $U{\downarrow}X$ has a terminal object.

Theorem A.6.1 (GAFT). *Let $\mathcal{C}$ be cocomplete and $U : \mathcal{C} \to \mathcal{X}$ preserve colimits. Moreover, suppose that for all $X \in \mathcal{X}$ there is a weakly terminal set of objects for $U{\downarrow}X$. Then U has a right adjoint.*

Proof. (Sketch.) In view of the above characterisation of the right adjoint we just have to show that a cocomplete category with a weakly terminal set of objects has a terminal object ($U{\downarrow}X$ is cocomplete because $\mathcal{C}$ is).

Fix $X \in \mathcal{X}$ and let $\mathcal{A}$ be a weakly terminal set of objects. Let A be the coproduct of all objects in $\mathcal{A}$ and let $e : A \to T$ be the colimit of the diagram consisting of all endomorphisms of A. Then T is the terminal object of $U{\downarrow}X$. □

Remark. The famous solution set condition is just the statement that for all $X \in \mathcal{X}$ there is a weakly terminal set of objects for $U{\downarrow}X$.

Easier to apply is often the following theorem.

Theorem A.6.2 (SAFT). *Let $\mathcal{C}$ be cocomplete and cowellpowered and let $U : \mathcal{C} \to \mathcal{X}$ preserve colimits. Moreover suppose that $\mathcal{C}$ has a set of generators. Then U has a right adjoint.*

Proof. In order to use the GAFT one has to show that for all $X \in \mathcal{X}$ there is a weakly terminal set of objects for $U{\downarrow}X$. Here we only do the case $X = 1$ (assuming that $\mathcal{X}$ has a terminal object 1). This is sufficient to see how the terminal coalgebras are obtained.

Let $\mathcal{G}$ be the set of generators and $A = \Sigma_{G \in \mathcal{G}} G$ the coproduct of all generators. Let $\mathcal{S} = \{S \mid \exists e : A \to S, e \text{ epi}\}$. We show that $\mathcal{S}$ is weakly terminal for $U{\downarrow}1 \simeq \mathcal{C}$ ($\mathcal{S}$ is—up to isomorphism—a set because $\mathcal{C}$ is cowellpowered).

Let $C \in \mathcal{C}$ and consider an epi sink $\mathsf{e} = (e_i : G_i \to C)_{i \in I}$, $G_i \in \mathcal{G}$. Let $B = \Sigma_{i \in I} G_i$. Since B is a coproduct the epi sink e factors through B as $\mathsf{e} = c \circ \mathsf{s}$ for some morphism $c : B \to C$. Note that c is epi since e is. Similarly, the sink i consisting of the inclusions $G_i \hookrightarrow A$, $i \in I$, factors through B as $\mathsf{i} = a \circ \mathsf{t}$ for some $a : B \to A$. Now consider the pushout

$$\begin{tikzcd} B \arrow[r, "a"] \arrow[d, "c"'] & A \arrow[d, "c'"] \\ C \arrow[r] & S \end{tikzcd}$$

Since c is epi, also c' is, hence $S \in \mathcal{S}$ which shows that $\mathcal{S}$ is weakly terminal. □

Remark. Note that the case $X = 1$ does not require U to preserve colimits.

[10] The category $U{\downarrow}X$ is defined as follows. Objects are pairs $(C, f : UC \to X)$ and morphisms $g : (C_1, f_1 : UC_1 \to X) \to (C_2, f_2 : UC_2 \to X)$ are morphisms $g : C_1 \to C_2 \in \mathcal{C}$ such that $f_2 \circ Ug = f_1$.

Remark. The results in chapter 2 rely on factorisation structures for sinks. It would therefore be interesting to have version of SAFT where we could replace cowellpowered by $\mathcal{E}$-cowellpowered and $\mathcal{C}$ having a set of generators by $\mathcal{C}$ being bounded (definition 1.5.3). Unfortunately, the straight forward adaption of the proof above does not work: If we do not know that the sink e is epi we can not conclude from a factorisation $\mathsf{e} = c \circ \mathsf{s}$ and $\mathsf{e} \in \mathcal{E}$ that also $c \in \mathcal{E}$ (which is used in an essential way in the pushout diagram above).

A.7 Comonads

A comonad $\mathsf{S} = (S, \varepsilon, \delta)$ on a category $\mathcal{X}$ is given by an endofunctor S on $\mathcal{X}$ and two natural transformations $\varepsilon : S \to \mathrm{Id}_{\mathcal{C}}$, $\delta : S \to S^2$ making the following diagrams commute for all $X \in \mathcal{X}$:

$$\begin{tikzcd}
SX & S^2X \arrow[l, "\varepsilon_{SX}"'] \arrow[r, "S\varepsilon_X"] & SX \\
& SX \arrow[ul, "\mathrm{id}_{SX}"] \arrow[u, "\delta_X"] \arrow[ur, "\mathrm{id}_{SX}"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
S^3X & S^2X \arrow[l, "S\delta_X"'] \\
S^2X \arrow[u, "\delta_{SX}"] & SX \arrow[l, "\delta_X"] \arrow[u, "\delta_X"']
\end{tikzcd}$$

A coalgebra (X, ξ) for a comonad S consists of an object $X \in \mathcal{X}$ and an arrow $\xi : X \to SX$ in $\mathcal{X}$ satisfying:

$$\begin{tikzcd}
X \arrow[r, "\xi"] \arrow[dr, "\mathrm{id}_X"'] & SX \arrow[d, "\varepsilon_X"] \\
& X
\end{tikzcd}
\qquad
\begin{tikzcd}
SX \arrow[r, "\delta_X"] & S^2X \\
X \arrow[u, "\xi"] \arrow[r, "\xi"'] & SX \arrow[u, "S\xi"']
\end{tikzcd}$$

A coalgebra morphism between (X, ξ) and (Y, ν) is an arrow $f : X \to Y \in \mathcal{X}$ such that:

$$\begin{tikzcd}
Y \arrow[r, "\nu"] & SY \\
X \arrow[u, "f"] \arrow[r, "\xi"'] & SX \arrow[u, "Sf"']
\end{tikzcd}$$

$\delta_X : SX \to S^2X$ is the cofree coalgebra over X with associated colouring $\varepsilon_X : SX \to X$.

The coalgebras for a comonad S form a category $\mathcal{X}_{\mathsf{S}}$, the category of (Eilenberg-Moore) coalgebras for the comonad. Let us write $U_{\mathsf{S}} : \mathcal{X}_{\mathsf{S}} \to \mathcal{X}$ for the corresponding forgetful functor. A functor $U : \mathcal{C} \to \mathcal{X}$ is called **comonadic** iff there is a comonad S on $\mathcal{X}$ such that U and U_{S} are concretely isomorphic, i.e., there is an iso $K : \mathcal{C} \to \mathcal{X}_{\mathsf{S}}$ with $U = U_{\mathsf{S}}K$. The following theorem characterises the comonadic functors.

Proposition A.7.1 (Beck's Theorem). *A functor $U : \mathcal{C} \to \mathcal{X}$ is comonadic iff it has a right adjoint and creates split equalisers.*

A.8 Locally Presentable and Accessible Categories

Both for categories of algebras and coalgebras it is an important property if—in a sense that has to be made precise—every object in the category is the union (or more generally some kind of colimit) of a small number of objects which are small themselves in a certain sense.

For coalgebras we found most useful the notions discussed in section 1.5. For algebras many formalisations of this idea exist, a comparison of a number of them can be found in Jürjens [68]. We recall below the definition of the one needed in this thesis, namely accessibility. For more information see, e.g., Adámek and Rosický [5] or Borceux [19], vol.2. Accessible categories of coalgebras are investigated in Power and Watanabe [91].

Let $\mathcal{C}$ be a category. An object $A \in \mathcal{C}$ is **κ-presentable** if the functor $\mathcal{C}(A, -)$ preserves κ-filtered colimits. The category $\mathcal{C}$ is called **κ-accessible** if it has κ-filtered colimits and if there is a full subcategory $\mathcal{S} \hookrightarrow \mathcal{C}$ consisting of κ-presentable objects such that every object in $A \in \mathcal{C}$ is a κ-filtered colimit of a diagram in $\mathcal{S}$. A category is called **accessible** if it is accessible for some regular cardinal κ. A category is called **locally κ-presentable** if it is κ-accessible and cocomplete, it is called locally presentable if it is locally κ-presentable for some regular cardinal κ.

Appendix B

Basic Notions of Modal Logic

The modal logics considered in this thesis are extensions of propositional logic. In this appendix we review some basic notions.[1]

The modal language considered here is supposed to be a language for transition systems where transitions may take labels from a set I: A modal language $\mathcal{L}$ consists of propositional connectives, a set of atomic propositions (also called propositional variables) $\mathbf{P}$ and a set of unary modal operators that we write as $\{[i] : i \in I\}$, I being the set of labels.

Given a modal language with atomic propositions $\mathbf{P}$ and modal operators $\{[i] : i \in I\}$, a *Kripke frame* $(W, \mathcal{R})$ for this language is given by a set of worlds W (also called states or points) and a family of relations $\mathcal{R} = (R_i)_{i \in I}, R_i \subset W \times W$. wR_iv should be read as "there is a transition labelled i from state w to state v". A *Kripke model* $(W, \mathcal{R}, V)$ for the given language consists of a Kripke frame $(W, \mathcal{R})$ and a mapping, called valuation, $V : \mathbf{P} \to \mathcal{P}(W)$ that assigns to every atomic proposition a set of worlds. The natural notion of a morphism of Kripke models is that of a functional bisimulation, traditionally called *p-morphism* . A p-morphism $f : (W, \mathcal{R}, V) \to (W', \mathcal{R}', V')$ is a function $f : W \to W'$ satisfying (i) $wR_iv \;\Rightarrow\; f(w)R'_if(v)$, (ii) $f(w)R'_iv' \;\Rightarrow\; \exists v : wR_iv \;\&\; f(v) = v'$ and (iii) $w \in V(p) \;\Leftrightarrow\; f(w) \in V'(p)$. To define p-morphisms for Kripke frames forget condition (iii).

The semantics of modal logic in terms of transition systems is defined as follows. Given a formula φ of the language and a Kripke model for the language $M = (W, \mathcal{R}, V)$ and a world $w \in W$ of the model, the relation $M, w \models \varphi$ is defined for the propositional connectives as to be expected and for atomic propositions and modal operators as follows:

$$\begin{array}{lll} M, w \models p & \text{iff} & p \in \mathbf{P} \text{ and } w \in V(p) \\ M, w \models [i]\varphi & \text{iff} & \forall v : wR_iv \;\Rightarrow\; M, v \models \varphi \end{array}$$

As usual, $M \models \varphi$ is defined by quantifying over all worlds, and $\models \varphi$ iff $M \models \varphi$ for all M. Similarly, given a frame $T = (W, \mathcal{R})$ and $w \in W$, then $T, w \models \varphi$ iff $(W, \mathcal{R}, V), w \models \varphi$ for all valuations $V : \mathbf{P} \to \mathcal{P}(W)$. $T \models \varphi$ iff $T, w \models \varphi$ for all $w \in W$. For Γ a set of formulas and φ a formula, the *consequence relation* $\Gamma \models \varphi$ is to be understood in its global sense, that is, $\Gamma \models \varphi \;\Leftrightarrow\; \forall M(M \models \Gamma \;\Rightarrow\; M \models \varphi)$. The *theory* of a world w in a model (or frame) S is $\mathsf{Th}(S, w) = \{\varphi : S, w \models \varphi\}$ and $\mathsf{Th}(S) = \{\varphi : S \models \varphi\}$. Two models or frames are called *logically equivalent* (or sometimes *equivalent* for short) iff they have the same theory.

[1] For more details see e.g. Blackburn, de Rijke, Venema [18] or Goldblatt [36].

p-Morphisms between Kripke models preserve and reflect satisfaction of formulas in every world of a model. Moreover, if $f : M \to M'$ a (surjective) p-morphism then $M \models \varphi$ if (and only if) $M' \models \varphi$. p-Morphisms between Kripke frames preserve satisfaction of formulas in every world of a frame. Moreover, if $f : (W, \mathcal{R}) \to (W', \mathcal{R}')$ a surjective p-morphism then $(W, \mathcal{R}) \models \varphi$ only if $(W', \mathcal{R}') \models \varphi$.

Given a modal language $\mathcal{L}$, the modal logic determined by all Kripke models for $\mathcal{L}$ is named $\mathbf{K}_{\mathcal{L}}$, i.e., $\mathbf{K}_{\mathcal{L}} = \{\varphi : M \models \varphi \text{ for all models } M \text{ for } \mathcal{L}\}$. $\mathbf{K}_{\mathcal{L}}$ has a strongly complete axiomatisation given by the axioms and rules below. To simplify notation we use $\Box$ as syntactic variable for the modal operators $[i]$.

(taut)	all propositional tautologies	
(dist)	$\Box(\varphi \to \psi) \to \Box\varphi \to \Box\psi$	for all $\Box$ of $\mathcal{L}$
(mp)	from $\varphi, \varphi \to \psi$ derive ψ	
(nec)	from φ derive $\Box\varphi$	for all $\Box$ of $\mathcal{L}$

Strictly speaking, (dist) is not an axiom but an *axiom scheme*, i.e. all instantiations of (dist) with formulas of $\mathcal{L}$ substituted for φ, ψ are axioms. Following a common abuse of language we will often refer to axiom schemes as axioms.

For Φ a set of formulas and φ a formula, $\Phi \vdash \varphi$ means that there is a finite derivation of φ using only the axioms and rules and the formulas in Φ. The above calculus is sound and strongly complete, that is, $\Phi \vdash \varphi \Leftrightarrow \Phi \models \varphi$.

Appendix C

Coalgebraic Logic

First the definition of coalgebraic logic from Moss [87] is given. For a detailed discussion and results, the reader is referred to the original paper. We need the category **SET** of classes and set-continuous functions. The functors Ω are on **SET** and have to be set-based, standard and to preserve weak pullbacks. (A functor Ω on **Set** is extended to a functor on **SET** by defining $\Omega K = \bigcup\{\Omega X : X \subset K, X \text{ a set}\}$ for classes K.)

Definition C.0.1 (coalgebraic logic, syntax). The Ω-language $\mathcal{CL}_\Omega$ is defined to be the least class satisfying:

$$\begin{array}{lcl} \Phi \subset \mathcal{CL}_\Omega,\ \Phi \text{ a set} & \Longrightarrow & \bigwedge \Phi \in \mathcal{CL}_\Omega \\ \varphi \in \mathcal{CL}_\Omega & \Longrightarrow & \neg\varphi \in \mathcal{CL}_\Omega \\ \varphi \in \Omega(\mathcal{CL}_\Omega) & \Longrightarrow & \varphi \in \mathcal{CL}_\Omega \end{array}$$

Due to the first clause $\bigwedge\{\}$, denoted by *true*, is in $\mathcal{CL}_\Omega$ and $\mathcal{CL}_\Omega$ is a proper class. The last clause uses the fact that Ω is a functor on **SET** and can also be applied to classes of formulas. The second clause is not a proper part of $\mathcal{CL}_\Omega$ as defined in [87] but it is shown there that negation may be added.

Definition C.0.2 (coalgebraic logic, semantics). Given a coalgebra M the semantics is given by the least relation $\models_\Omega \subset UM \times \mathcal{CL}_\Omega$ such that (let $x \in UM$):

$$\begin{array}{lcl} x \models_\Omega \varphi \text{ for all } \varphi \in \Phi, \Phi \subset \mathcal{CL}_\Omega,\ \Phi \text{ a set} & \Rightarrow & x \models_\Omega \bigwedge \Phi \\ x \not\models_\Omega \varphi & \Rightarrow & x \models_\Omega \neg\varphi \\ \text{there is } w \in \Omega(\models_\Omega) \text{ s.t. } \Omega\pi_1(w) = f(x), \Omega\pi_2(w) = \varphi & \Rightarrow & x \models_\Omega \varphi \end{array}$$

where π_1, π_2 denote the projections from the product $UM \times \mathcal{CL}_\Omega$ to its components.

Note that the last clause makes use of $\models_\Omega$ being in **SET** and applies Ω to it.

For an example take again $\Omega = C \times \mathcal{P}$. A typical formula of $\mathcal{CL}_\Omega$ is of the kind (c, Φ) where $c \in C$ a colour and $\Phi \subset \mathcal{CL}_\Omega$ a set of formulas. Satisfiability for such formulas is described by the next proposition.

Proposition C.0.3. *Let $\Omega = C \times \mathcal{P}$, $c \in C$, $\Phi \subset \mathcal{CL}_\Omega$, $M = (UM, f)$ a Ω-coalgebra, and $x \in UM$. Then*

$$x \models_\Omega (c, \Phi) \;\Leftrightarrow\; \begin{cases} \forall y \in \pi_2 \circ f(x) : \exists \varphi \in \Phi : y \models_\Omega \varphi & \text{and} \\ \forall \varphi \in \Phi : \exists y \in \pi_2 \circ f(x) : y \models_\Omega \varphi & \text{and} \\ c = \pi_1 \circ f(x) \end{cases}$$

Proof. See [87]. $\square$

Using the modal operators $\Box, \Diamond$, the first two clauses of the right-hand side of the above equivalence may be written as: $x \models \Box \bigvee \Phi \wedge \bigwedge \Diamond \Phi$, where $\Diamond\Phi = \{\Diamond\varphi : \varphi \in \Phi\}$.

Bibliography

[1] Peter Aczel. *Non-Well-Founded Sets.* Center for the Study of Language and Information, Stanford University, 1988.

[2] Peter Aczel and Nax Mendler. A final coalgebra theorem. In D. H. Pitt et al, editor, *Category Theory and Computer Science*, volume 389 of *LNCS*, pages 357–365. Springer, 1989.

[3] J. Adámek. bounded=accessible. Categories Mailing List, March 2000. Available at `http://www.mta.ca/~cat-dist/categories.html`.

[4] J. Adámek, H. Herrlich, and G. Strecker. *Abstract and Concrete Categories.* John Wiley & Sons, 1990.

[5] J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*, volume 189 of *London Mathematical Society Lecture Notes Series.* Camebridge University Press, 1994.

[6] H. Andréka and I. Nemeti. A general axiomatizability theorem formulated in terms of cone-injective subcategories. In B. Csakany, E. Fried, and E. Schmidt, editors, *Universal Algebra*, pages 13–35. North Holland, 1982.

[7] M.A. Arbib and E.G. Manes. Adjoint machines, state-behaviour machines, and duality. *Journ. of Pure and Applied Algebra*, 6:313–344, 1975.

[8] S. Awodey and J. Hughes. The coalgebraic dual of Birkhoff's variety theorem. October 2000. `http://www.contrib.andrew.cmu.edu/user/jesse/papers/CoBirkhoff.ps.gz`.

[9] Alexandru Baltag. *STS: A Structural Theory of Sets.* PhD thesis, Indiana University, 1998.

[10] Alexandru Baltag. A logic for coalgebraic simulation. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 41–60, 2000.

[11] B. Banaschewski and H. Herrlich. Subcategories defined by implications. *Houston Journal of Mathematics*, 2(2):149–171, 1976.

[12] M. Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114(2):299–315, June 1993.

[13] J. Barwise and L. Moss. *Vicious Circles.* Center for the Study of Language and Information, Stanford University, 1996.

[14] M. Bidoit and R. Hennicker. Proving behavioural theorems with standard first-order logic. In G. Levi and M. Rodriguez-Artalejo, editors, *Proc. Algebraic and Logic Programming, 4th International Conference, ALP '94, Madrid, September 1994*, volume 850 of *LNCS*, pages 41–58. Springer, 1994.

[15] M. Bidoit and R. Hennicker. Behavioural theories and the proof of behavioural properties. *Theoretical Computer Science*, 175:3–55, 1996.

[16] M. Bidoit, R. Hennicker, and A. Kurz. On observability and reachability. 2000. `http://www.cwi.nl/~kurz/obsreach.ps.gz`.

[17] M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. *Science of Computer Programming*, 25:149–186, 1995.

[18] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* To appear. See `http://www.mlbook.org`.

[19] Francis Borceux. *Handbook of Categorical Algebra.* Cambridge University Press, 1994.

[20] Tomasz Borzyszkowski. Completeness of a logical system for structured specifications. In F. Parisi-Presicce, editor, *Recent Trends in Algebraic Development Techniques, WADT 97*, volume 1376 of *LNCS*, pages 107–121. Springer, 1998.

[21] S. Burris and H.P. Sankappanavar. *A Course in Universal Algebra.* Springer, 1981.

[22] S. Buss and G. Roşu. Incompleteness of behavioural logic. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 61–80, 2000.

[23] A. Carboni, G. Kelly, and R. Wood. A 2-categorical approach to change of base and geometric morphisms I. Technical Report 90-1, Department of Pure Mathematics, University of Sydney, 1990.

[24] P. Cenciarelli, A. Knapp, B. Reus, and M. Wirsing. An Event-Based Structural Operational Semantics of Multi-Threaded Java. In Jim Alves-Foss, editor, *Formal Syntax and Semantics of Java*, volume 1523 of *LNCS*, pages 157–200. Springer, 1999.

[25] Corina Cîrstea. Coalgebraic semantics for hidden algebra: parameterized objects and inheritance. In F. Parisi-Presicce, editor, *Recent Trends in Algebraic Development Techniques, WADT 97*, volume 1376 of *LNCS*. Springer, 1998.

[26] Corina Cîrstea. A coequational approach to specifying behaviours. In B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'99)*, volume 19 of *Electronic Notes in Theoretical Computer Science*, pages 173–194, 1999.

[27] Corina Cîrstea. An algebra-coalgebra framework for system specification. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 81–112, 2000.

[28] A. Corradini, M. Große-Rhode, and R. Heckel. Structured transition system as lax coalgebras. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 23–42, 1998.

[29] Andrea Corradini. A complete calculus for equational deduction in coalgebraic specification. In F. Parisi-Presicce, editor, *Recent Trends in Algebraic Development Techniques, WADT 97*, volume 1376 of *LNCS*. Springer, 1998.

[30] G. Denker and P. Hartel. TROLL—An Object Oriented Formal Method for Distributed Information System Design: Syntax and Pragmatics (TROLL Version 3.0). Technical Report Informatik Berichte 97-03, Technische Universität Braunschweig, 1997.

[31] G. Denker, J. Ramos, C. Caleiro, and A. Sernadas. A linear temporal logic approach to objects with transactions. In Michael Johnson, editor, *Algebraic Methodology and Software Technology (AMAST'97)*, volume 1349 of *LNCS*, pages 170–184. Springer, 1997.

[32] Răzvan Diaconescu. Behavioural coherence in object-oriented algebraic specification. Technical Report IS-RR-98-0017F, Japan Advanced Institute for Science and Technology, 1998.

[33] H.-D. Ehrich, C. Caleiro, A. Sernadas, and G. Denker. Logics for Specifying Concurrent Information Systems. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 167–198. Kluwer, 1998.

[34] J. Goguen and G. Malcolm. A hidden agenda. Technical Report CS97-538, UCSD, 1997.

[35] J. Goguen and G. Roşu. Hiding more of hidden algebra. In J.M. Wing, J. Woodcock, and J. Davies, editors, *Formal Methods (FM'99)*, volume 1709 of *LNCS*, pages 1704–1719. Springer, 1999.

[36] Robert Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 1992. Second Edition.

[37] Robert Goldblatt. A framework for infinitary modal logic. In *Mathematics of Modality*, chapter 9. Center for the Study of Language and Information, Stanford University, 1993.

[38] Robert Goldblatt. Saturation and the Hennessy-Milner property. In A. Ponse et.al., editor, *Modal Logic and Process Algebra*, volume 53 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 1995.

[39] Robert Goldblatt. What is the coalgebraic analogue of Birkhoff's variety theorem? *Theoretical Computer Science*, To appear.

[40] H. P. Gumm and T. Schröder. Products of coalgebras. *Algebra Universalis*. Accepted for publication.

[41] H. P. Gumm and T. Schröder. Covarieties and complete covarieties. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, 1998.

[42] H. P. Gumm and T. Schröder. Coalgebraic structure from weak limit preserving functors. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 113–134, 2000.

[43] H. P. Gumm and T. Schröder. Coalgebras of bounded type. Technical Report 25, FG Informatik, Philipps-Universitt Marburg, 2000. `http://www.mathematik.uni-marburg.de/~gumm/Papers/Bounded.ps`.

[44] H. Peter Gumm. Functors for coalgebras. *Algebra Universalis*. To appear.

[45] H. Peter Gumm. Equational and implicational classes of co-algebras. Extended abstract. *RelMiCS'4. The 4th International Seminar on Relational Methods in Logic, Algebra and Computer Science, Warsaw*, 1998.

[46] H. Peter Gumm. Elements of the general theory of coalgebras. LUATCS'99, 1999.

[47] Peter Hartel. *Konzeptionelle Modellierung von Informationssystemen als verteilte Objektsysteme*. Reihe DISDBIS. infix-Verlag, Sankt Augustin, 1997.

[48] M. Hennessy and R. Milner. Algebraic laws for nondeterminism. *Journal of the Association for Computing Machinery*, 32:137–161, 1985.

[49] R. Hennicker and M. Bidoit. Observational logic (long version). Technical Report LSV-98-6, LSV, Ecole Normale Supérieure de Cachan, June 1998. www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-1998-6.rr.ps.

[50] R. Hennicker and M. Bidoit. Observational logic. In Armando Haeberer, editor, *Algebraic Methodology and Software Technology (AMAST'98)*, volume 1548 of *LNCS*. Springer, 1999.

[51] R. Hennicker and A. Kurz. (Ω, Ξ)-logic: On the algebraic extension of coalgbraic specifications. In B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'99)*, volume 19 of *Electronic Notes in Theoretical Computer Science*, pages 195–211, 1999.

[52] U. Hensel, M. Huisman, B. Jacobs, and H. Tews. Reasoning about classes in object-oriented languages: Logical models and tools. In Ch. Hankin (ed), editor, *European Symposium on Programming*, volume 1381 of *LNCS*, pages 105–121. Springer, 1998.

[53] U. Hensel and H. Reichel. Defining equations in terminal coalgebras. In E. Astesiano, G. Reggio, and A.Tarlecki, editors, *Recent Trends in Data Type Specification (COMPASS/ADT 1995)*, volume 906 of *LNCS*, pages 307–318. Springer, 1995.

[54] H. Herrlich and G. Strecker. *Category Theory.* Allyn and Bacon, 1973.

[55] Marco Hollenberg. Hennessy-Milner classes and process algebra. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, volume 53 of *CSLI Lecture Notes.* Center for the Study of Language and Information, Stanford University, 1995.

[56] Jesse Hughes. The coalgebraic analogue of Birkhoff's variety theorem. Bulletin of Symbolic Logic, June 2000. Abstract for talk delivered at 1999–2000 Winter Meeting of the ASL, Washington, DC.

[57] B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors. *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11. Electronic Notes in Theoretical Computer Science, 1998.

[58] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62, 1997.

[59] B. Jacobs and J. Rutten, editors. *Coalgebraic Methods in Computer Science*, volume 19. Electronic Notes in Theoretical Computer Science, 1999.

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

[61] Bart Jacobs. Mongruences and cofree coalgebras. In V. S. Alagar and M. Nivat, editors, *Algebraic Methods and Software Technology (AMAST'95)*, volume 936 of *LNCS*, pages 245–260. Springer, 1995.

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

[63] Bart Jacobs. *Categorical Logic and Type Theory.* Elsevier, 1998.

[64] Bart Jacobs. Coalgebraic reasoning about classes in object-oriented languages. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 235–246, 1998.

[65] Bart Jacobs. Coalgebras in specification and verification for object-oriented languages. *Newsletter 3 of the Dutch Association for Theoretical Computer Science (NVTI)*, pages 15–27, 1999.

[66] Bart Jacobs. The temporal logic of coalgebras via galois algebras. Technical Report CSI-R9906, Computing Science Institute Nijmegen, 1999.

[67] Bart Jacobs. Towards a duality result in coalgebraic modal logic. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 163–198, 2000.

[68] Jan Jürjens. Kleine Objekte in Kategorien von Algebren. Master's thesis, Universität Bremen, 1998.

[69] H. J. Keisler. *Model Theory for Infinitary Logic.* North-Holland, 1971.

[70] Fred Kröger. *Temporal Logic of Programs.* Springer, 1987.

[71] A. Kurz and D. Pattinson. Notes on coalgebras, co-fibrations and concurrency. In H. Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 199–233, 2000.

[72] A. Kurz and R.Hennicker. On institutions for modular coalgebraic specifications. *Theoretical Computer Science.* Accepted for publication.

[73] Alexander Kurz. Specifying coalgebras with modal logic. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 57–72, 1998.

[74] Alexander Kurz. Modal rules are co-implications. February 1999. Revised May 2000. `http://www.informatik.uni-muenchen.de/~kurz/`.

[75] Alexander Kurz. A co-variety-theorem for modal logic. In *Proceedings of Advances in Modal Logic 2, Uppsala, 1998.* Center for the Study of Language and Information, Stanford University, 2000.

[76] Alexander Kurz. Modal logic is dual to equational logic. January 2000. `http://www.informatik.uni-muenchen.de/~kurz/public_html/dual.ps.gz`.

[77] Alexander Kurz. Specifying coalgebras with modal logic. *Theoretical Computer Science*, 260(1-2), 2000. To appear.

[78] Saunders Mac Lane. *Category Theory for the Working Mathematician.* Springer, 1971.

[79] G. Malcolm and J. Goguen. Proving correctness of refinement and implementation. Technical Report PRG-114, Oxford University Computing Laboratory, 1994.

[80] Grant Malcolm. Behavioural equivalence, bisimulation, and minimal realisation. *LNCS*, 1130:359–378, 1996.

[81] Ernest G. Manes. *Algebraic Theories.* Springer, 1976.

[82] D. Martin and R. Solovay. Internal Cohen extensions. *Annals of Mathematical Logic*, 2:143–178, 1970.

[83] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[84] Robin Milner. *Communication and Concurrency.* International series in computer science. Prentice Hall, 1989.

[85] Luís Monteiro. Semantic domains based on sets with families of equivalences. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, 1998.

[86] Luís Monteiro. Observation systems. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, 2000.

[87] Lawrence Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.

[88] Peter Padawitz. Swinging data types: syntax, semantics, and theory. In O.-J. Dahl M. Haveraaen, O. Owe, editor, *Recent Trends in Data Type Specification*, volume 1130 of *LNCS*, pages 409–435. Springer, 1996.

[89] D. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer, 1981.

[90] Dirk Pattinson. Semantical principles in the modal logic of coalgebras. In *Proceedings 18th International Symposium on Theoretical Aspects of Computer Science (STACS 2001)*, Lecture Notes in Computer Science, Berlin. Springer. To appear.

[91] J. Power and H. Watanabe. An axiomatics for categories of coalgebras. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, 1998.

[92] Horst Reichel. *Initial computability, algebraic specifications, and partial algebras.* Oxford, Clarendon Press, 1987.

[93] Horst Reichel. An approach to object semantics based on terminal co-algebras. *Mathematical Structures in Computer Science*, 5(2):129–152, June 1995.

[94] Horst Reichel. Dialgebraic logics. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11. Electronic Notes in Theoretical Computer Science, 1998.

[95] Horst Reichel. Nested sketches. Technical Report ECS-LFCS-98-401, University of Edinburgh, 1998.

[96] Horst Reichel. A uniform model theory for the specifcation of data and process types. 1999. Submitted to Proceedings of WADT99.

[97] Horst Reichel, editor. *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, 2000.

[98] Grigore Roşu. A Birkhoff-like axiomatisability result for hidden algebra and coalgebra. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 179–196, 1998.

[99] Grigore Roşu. Personal Communication, August 1999.

[100] Grigore Roşu. Equational axiomatizability for coalgebra. *Theoretical Computer Science*, 260/1-2, 2000.

[101] Grigore Roşu. *Hidden Logic*. PhD thesis, University of California at San Diego, 2000.

[102] Martin Rößiger. From modal logic to terminal coalgebras. Report MATH-AL-3-1998, University of Dresden, 1998.

[103] Martin Rößiger. Languages for coalgebras on datafunctors. In J. Rutten B. Jacobs, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 19 of *Electronic Notes in Theoretical Computer Science*, pages 55–76, 1999.

[104] Martin Rößiger. Coalgebras and modal logic. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 299–320, 2000.

[105] Martin Rößiger. From modal logic to terminal coalgebras. *Theoretical Computer Science*, 260(1-2), 2000. To appear.

[106] J. Rutten and D. Turi. Initial algebra and final coalgebra semantics for concurrency. Report CS-R9409, CWI, Amsterdam, 1994.

[107] J. J. M. M. Rutten. A structural co-induction theorem. Technical Report CS R 9346, CWI, Amsterdam, 1993.

[108] J. J. M. M. Rutten. Universal coalgebra: A theory of systems. Technical Report CS R 9652, CWI, Amsterdam, 1996.

[109] J. J. M. M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[110] D. Sannella and A. Tarlecki. Specifications in arbitrary institutions. *Information and Computation*, 76:165–210, 1988.

[111] Krister Segerberg. An essay in classical modal logic. *Filosofiska Studier 13*, 1971.

[112] Krister Segerberg. A model existence theorem in infinitary propositional modal logic. *Journal of Philosophical Logic*, 23:337–367, 1994.

[113] Andrzej Tarlecki. Institutions: An abstract framework for formal specifications. In E. Astesiano, H.-J. Kreowski, and B. Krieg-Brückner, editors, *Algebraic Foundations of Systems Specification*, chapter 4. Springer, 1999.

[114] Paul Taylor. *Practical Foundations of Mathematics.* Cambridge University Press, 1999.

[115] Hendrik Tews. Coalgebras for binary methods. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 321–347, 2000.

[116] S.K. Thomason. Categories of frames for modal logics. *Journal of Symbolic Logic*, 37:439–442, 1975.

[117] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. of the LICS'97*, pages 280–305, 1997.

[118] D. Turi and J. Rutten. On the foundations of final coalgebra semantics: non-well-founded sets, partial orders, metric spaces. *Mathematical Structures in Computer Science*, 8, 1998.

[119] Daniele Turi. *Functorial Operational Semantics and its Denotational Dual.* PhD thesis, Free University, Amsterdam, 1996.

[120] Tarmo Uustalu. Combining object-oriented and logic paradigms: A modal logic programming approach. In O. Lehrmann Madsen, editor, *Proceedings 6th European Conf. on Object-Oriented Programming, ECOOP'92*, volume 615 of *LNCS*, pages 98–113. Springer, 1992.

[121] Johan van Benthem. *Modal Correspondence Theory.* PhD thesis, University of Amsterdam, 1976.

[122] Johan van Benthem. *Modal Logic and Classical Logic.* Bibliopolis, Naples, 1983.

[123] Johan van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2. Reidel, Dordrecht, 1984.

[124] Wolfgang Wechler. *Universal Algebra for Computer Scientists*, volume 25 of *EATCS Monographs on Theoretical Computer Science.* Springer, 1992.

[125] Uwe Wolter. On corelations, cokernels, and coequations. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 347–366, 2000.

[126] James Worrell. PhD thesis. Forthcoming.

[127] James Worrell. Toposes of coalgebras and hidden algebra. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'98)*, volume 11 of *Electronic Notes in Theoretical Computer Science*, 1998.

[128] James Worrell. Coinduction for recursive data types: partial order, metric spaces and ω-categories. In Horst Reichel, editor, *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 367–386, 2000.

Index