On Institutions for Modular Coalgebraic Specifications

Alexander Kurz, Rolf Hennicker
Institut für Informatik
Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 München, Germany
e-mail: {kurz,hennicke}@informatik.uni-muenchen.de

May 1, 2000

Abstract

We present an algebraic extension of standard coalgebraic specification techniques for state-based systems which allows us to integrate constants and n-ary operations in a smooth way and which leads to institutions enabling the use of modular specification techniques. A sound and complete proof system for first-order observational properties of modular specifications is given. The framework of (Ω, Ξ)-structures that we present can be considered as the result of a transformation of concepts of observational logic as in [14] into the coalgebraic world. Moreover, 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.

1 Introduction

In this paper 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 [14] and from similar ideas of swinging data types (Padawitz [30]) and recent extensions of hidden algebra (see Diaconescu [8] and Goguen and Roşu [10]. 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 [32], Jacobs [17]).

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

$$\mathit{bal} : X \to \mathbb{Z}, \qquad \mathit{update} : X \times \mathbb{Z} \to X,$$

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 $\_.\mathit{bal}$, $\_.\mathit{update}(n).\mathit{bal}$, $\_.\mathit{update}(n_1).\mathit{update}(n_2).\mathit{bal}, \ldots$ 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

[1]This splitting is also the basic idea of the algebraic frameworks [30, 14, 8, 10], in contrast to hidden algebra [9] where all operations having a state sort (hidden sort) as argument are implicitly regarded as observers.

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 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 $\to$ 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 $\to$ int or $owner$: account $\to$ 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 $\to$ account (since these operations do not contribute to the definition of the observational equivalence).

The paper 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

[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.

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 $(\alpha : \Omega X \to X, \beta : X \to \Xi X)$ 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 greatest 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. the given observers.

For proving observable properties of a specification we present in section 7 a sound, complete and modular proof system.

2 Notation and Technical Preliminaries

Given a category $\mathcal{C}$ and two functors $\Omega, \Xi : \mathcal{C} \to \mathcal{C}$ and an object $X \in \mathcal{C}$, morphisms $\alpha : \Omega X \to X$ and $\beta : X \to \Xi X$ are called *algebras* and *coalgebras*,

[4] Algebra-coalgebra pairs are also considered in Malcolm [27], 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 [27] indeed force algebra-coalgebra pairs to be (Ω, Ξ)-structures.

respectively. An *algebra morphism* $f : \alpha \to \alpha'$ of algebras $\alpha : \Omega X \to X$, $\alpha' : \Omega X' \to X'$ is a morphism $f : X \to X'$ in $\mathcal{C}$ such that $f \circ \alpha = \alpha' \circ \Omega f$. Analogously, a *coalgebra morphism* $f : \beta \to \beta'$ of coalgebras $\beta : X \to \Xi X$, $\beta' : X' \to \Xi X'$ is a morphism $f : X \to X'$ 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 [27] 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 paper, $\mathcal{C}$ will always 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 $(X_i)_{1 \leq i \leq n}$ and an arrow $f : X \to Y$ is a family of functions $(f_i : X_i \to Y_i)_{1 \leq i \leq n}$. 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 nonempty source) have a right (left) inverse and are hence preserved by functors.

The notion of a Ξ-bisimulation is well-known but we will recall it here to point out the correspondence to the perhaps less well-known notion of Ω-congruence (see [36], [28]). Since we do not need bisimulations between two different coalgebras we give directly the specialised definition. 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 $xRy \Rightarrow \alpha(a, x) R \alpha(a, y)$.

A *final coalgebra* $\pi : Z \to \Xi Z$ is characterised up to isomorphism by the property that for all coalgebras $\beta : X \to \Xi X$ there is a unique coalgebra morphism $! : \beta \to \pi$. In the case that Ξ preserves *weak pullbacks*[6] this morphism ! is intimately related to the greatest Ξ-bisimulation on β because then ! identifies exactly the bisimilar elements of X. Categorically this property may be expressed by the following lemma (Rutten and Turi [35], Malcolm [27]).

[5] The notions of relation and subset w.r.t. $\mathbf{Set^n}$ are defined componentwise.

[6] Weak means that the arrow into the weak pullback may not be unique, see Rutten [36] and Gumm [12] for a discussion. In the context of this paper all interesting functors preserve weak pullbacks.

Lemma 2.1. *Let $\beta : X \to \Xi X$ be a Ξ-coalgebra, $\pi : Z \to \Xi Z$ a final Ξ-coalgebra, and suppose that Ξ preserves weak pullbacks. Then $R \subset X \times X$ is the greatest bisimulation on β iff the diagram below is a pullback in* $\mathbf{Set^n}$*:*

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

Finally, in section 4 we will need a lemma about idempotent monads.

Lemma 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. □

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[7] expressed by the coalgebraic part.

Concerning the functors Ω, Ξ, we require that they preserve weak pullbacks and that a final Ξ-coalgebra exists. The condition that Ω, Ξ preserve weak pullbacks is needed for the characterisation of (Ω, Ξ)-structures involving bisimulations as given by theorem 3.7. The condition that the terminal Ξ-coalgebra exists

[7]Recall that in the coalgebraic approach behavioural equivalence is greatest Ξ-bisimulation (in the case that Ξ preserves weak pullbacks).

is technically convenient, though not strictly needed: One could alternatively define the behaviour of an (Ω, Ξ)-structure (definition 3.3) by taking a quotient w.r.t. largest behavioural equivalence relation (which exists for coalgebras for any functor Ξ on $\mathbf{Set^n}$, see [23]).

3.1 Basic Definitions and Results

Definition 3.1 ((Ω, Ξ)-structures). Let Ω, Ξ be functors on $\mathbf{Set^n}$ and let $\pi : Z \to \Xi Z$ be the final Ξ-coalgebra. Then an algebra-coalgebra pair $(\alpha : \Omega X \to X, \beta : X \to \Xi X)$ is called an (Ω, Ξ)-structure (on X) 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):

$$\begin{tikzcd}
\Omega X \arrow[r, "\alpha"] \arrow[d, "\Omega !"'] & X \arrow[r, "\beta"] \arrow[d, "!"'] & \Xi X \arrow[d, "\Xi !"] \\
\Omega Z \arrow[r, dotted, "h"] & Z \arrow[r, "\pi"] & \Xi Z
\end{tikzcd}$$

Note that h is in general not uniquely determined. But it follows from proposition 3.4 below 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 3.7, is even a characterisation of (Ω, Ξ)-structures):

Proposition 3.2. *Let (α, β) be an (Ω, Ξ)-structure on X. The greatest Ξ-bisimulation on the coalgebra β is an Ω-congruence on the algebra α.*

Proof. The greatest bisimulation R on β is given by the pullback diagram of lemma 2.1. Hence $! \circ \pi_1 = ! \circ \pi_2$. Using $h \circ \Omega! = ! \circ \alpha$, it follows $! \circ (\alpha \circ \Omega\pi_1) = ! \circ (\alpha \circ \Omega\pi_2)$. Since R is a pullback there is a mapping (even a unique one) $\delta : \Omega R \to R$ making R into a Ω-congruence. $\square$

Consider an (Ω, Ξ)-structure (α, β) and the corresponding unique morphism $!$ into the final Ξ-coalgebra. Then the image of $!$ gives rise to an (Ω, Ξ)-structure that is—from the observational point of view—equivalent to (α, β) and in which all Ξ-bisimilar elements are identified. Such a structure is called a behaviour.[8]

Definition 3.3 (Behaviour). Let $\Omega X \xrightarrow{\alpha} X \xrightarrow{\beta} \Xi X$ be an algebra-coalgebra pair, $\pi : Z \to \Xi Z$ final in $\mathbf{Set}_\Xi$ and $! : \beta \to \pi$. Furthermore let $X \xrightarrow{e} \mathrm{Im}(!) \xrightarrow{m} Z$ be the unique factorisation of $!$ (as an arrow in $\mathbf{Set^n}$) through its image. Then any algebra-coalgebra pair $(\bar{\alpha}, \bar{\beta})$ on the image of $!$ such that the diagram below

[8] The notion of minimal realisation in Malcolm [27] is equivalent to our notion of behaviour.

commutes is called a behaviour of (α, β).

$$\begin{tikzcd}
\Omega X \arrow[r, "\alpha"] \arrow[d, "\Omega e"'] & X \arrow[r, "\beta"] \arrow[d, "e"'] & \Xi X \arrow[d, "\Xi e"] \\
\Omega\,\mathrm{Im}(!) \arrow[r, dotted, "\bar{\alpha}"] & \mathrm{Im}(!) \arrow[r, dotted, "\bar{\beta}"] \arrow[d, "m"'] & \Xi\,\mathrm{Im}(!) \arrow[d, "\Xi m"] \\
 & Z \arrow[r, "\pi"] & \Xi Z
\end{tikzcd}$$

Similarly, we call $\bar{\beta}$ the behaviour of β.

Since we know from Rutten [36] that a coalgebra morphism uniquely factors through its image it is clear that $\bar{\beta}$ always exists.[9] The important point about the existence of a behaviour of (α, β) is therefore the existence of $\bar{\alpha}$.

Proposition 3.4. *Let $\Omega X \overset{\alpha}{\to} X \overset{\beta}{\to} \Xi X$ be an algebra-coalgebra pair. Then its behaviour—if it exists—is uniquely determined.*

Proof. Uniqueness of $\bar{\beta}$ follows from e epi, uniqueness of $\bar{\alpha}$ from Ωe epi (which, in turn, is due to the fact that epis in $\mathbf{Set^n}$ are split). $\square$

Note also that any behaviour is its own behaviour. Together with the following characterisation of behaviours, this implies that all behaviours are (Ω, Ξ)-structures.

Theorem 3.5. *Let $\Omega X \overset{\alpha}{\to} X \overset{\beta}{\to} \Xi X$ be an algebra-coalgebra pair. Then (α, β) is an (Ω, Ξ)-structure iff its behaviour exists.*

Proof. Let us write $X \overset{e}{\to} \bar{X} \overset{m}{\to} Z$ for the factorisation of $! : \beta \to \pi$. Consider the following diagram:

$$\begin{tikzcd}
\Omega X \arrow[r, "\alpha"] \arrow[d, "\Omega e"'] & X \arrow[r, "\beta"] \arrow[d, "e"'] & \Xi X \arrow[d, "\Xi e"] \\
\Omega\bar{X} \arrow[r, "\bar{\alpha}"] \arrow[d, "\Omega m"'] & \bar{X} \arrow[r, "\bar{\beta}"] \arrow[d, "m"'] & \Xi\bar{X} \arrow[d, "\Xi m"] \\
\Omega Z \arrow[r, "h"] & Z \arrow[r, "\pi"] & \Xi Z
\end{tikzcd}$$

and let j be a left inverse of m (i.e. $j \circ m = \mathrm{id}_{\bar{X}}$). For the "only if" part define $\bar{\alpha} = j \circ h \circ \Omega m$ and for the "if" part let $h = m \circ \bar{\alpha} \circ \Omega j$. That the respective conditions are met in both cases is checked easily. $\square$

[9]The existence of $\bar{\beta}$ can also be guaranteed without the assumption that a final coalgebra exists by defining $\bar{\beta}$ as the quotient of the coalgebra β w.r.t. the largest behavioural equivalence on β, see [23].

The proposition and theorem above allow for the following definition:

Definition 3.6. 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.

Next, we prove the converse of proposition 3.2 and thereby give a second characterisation of (Ω, Ξ)-structures.[10]

Theorem 3.7. *Let $\Omega X \xrightarrow{\alpha} X \xrightarrow{\beta} \Xi X$ be an algebra-coalgebra pair. Then (α, β) is an (Ω, Ξ)-structure iff the greatest Ξ-bisimulation on β is an Ω-congruence on α.*

Proof. The "only if" part was proved as proposition 3.2. For the converse we show that the behaviour of (α, β) exists (see theorem 3.5). Let us write $X \xrightarrow{e} \bar{X} \xrightarrow{m} Z$ for the factorisation of $! : \beta \to \pi$. From Rutten [36] we know that there is an appropriate $\bar{\beta} : \bar{X} \to \Xi\bar{X}$. 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 be the greatest bisimulation on β and $\Omega R \xrightarrow{\delta} R \xrightarrow{\gamma} \Xi R$ the arrows making R into a congruence and a bisimulation. Consider the following three layered diagram:

$$\begin{tikzcd}
\Omega\bar{X} \arrow[d, "\Omega i"'] & \Omega X \arrow[l, "\Omega e"'] \arrow[d, "r"'] \arrow[dr, "id"] & \\
\Omega X \arrow[d, "\alpha"'] & \Omega R \arrow[l, "\Omega\pi_1"'] \arrow[r, "\Omega\pi_2"] \arrow[d, "\delta"'] & \Omega X \arrow[d, "\alpha"] \\
X \arrow[dr, "!"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & X \arrow[dl, "!"] \\
& Z &
\end{tikzcd}$$

Recall that as a greatest bisimulation R is a pullback. Therefore (Ω preserving weak pullbacks) ΩR is a weak pullback. Together with $\Omega! \circ \Omega i \circ \Omega e = \Omega! \circ \mathrm{id}_{\Omega X}$ 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 since it is a bisimulation. Now, going from the top to the bottom yields $! \circ \alpha \circ \Omega i \circ \Omega e = ! \circ \alpha$, and therefore (using $! = m \circ e$ and m mono) $\bar{\alpha} \circ \Omega e = e \circ \alpha$. $\square$

[10] Theorem 3.7 is closely related to the result of Rutten and Turi [35] saying (very roughly) that a final semantics has an equivalent initial semantics if bisimulation is congruence.

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 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 3.8 ($\mathbf{Str}^{\Omega}_{\Xi}, \mathbf{bStr}^{\Omega}_{\Xi}, \overline{\mathbf{Str}}^{\Omega}_{\Xi}$). $\mathbf{Str}^{\Omega}_{\Xi}$ 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[11] subcategory of $\mathbf{Str}^{\Omega}_{\Xi}$ defined by the behaviours 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.[12] It should be intuitively clear (for a proof see corollary 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 4.1 we will show that B 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.

3.2 Coinductive Definitions

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 $x.\mathit{update}(n).\mathit{bal} = x.\mathit{bal}+n$ a coinductive definition of the *update*-operation. We now want to justify this informal terminology by relating axioms like $x.\mathit{update}(n).\mathit{bal} = x.\mathit{bal}+n$ to the formal account of coinduction as presented in Rutten [36] or Jacobs and Rutten [15].

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

[11] 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 $\mathbf{Str}^{\Omega}_{\Xi}$.

[12] In the case of the special functors Ω, Ξ in section 6 the morphisms in $\mathbf{bStr}^{\Omega}_{\Xi}$ can be described explicitly as certain relations, see the observational homomorphisms in [14], 3.9.

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, 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.[13] 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.[14]

Definition 3.9 (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 3.3).

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 $! : \beta \to \pi$).

[13] 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).

[14] The idea behind the definition is the same as in definition 3.1.

The following proposition generalises the second point above to arbitrary coalgebras.

Proposition 3.10. *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. As in the proof of theorem 3.5 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 $\bar{\alpha} \circ \Omega e = e \circ \alpha$. □

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.

Proposition 3.11. *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 differs fundamentally from other approaches like Jacobs [16], Gumm [11], and Kurz [22] where specifications force coalgebras to avoid "bad" elements.

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 4.8, subsection 4.2.1, and corollary 4.22 which only are needed to continue with section 6. The categorical background can be found in Mac Lane [25] (monads), Jacobs [18] (fibred category theory) and Borceux [4] (both).

4.1 The Behaviour Monad

The operation B on $\mathbf{Str}^{\Omega}_{\Xi}$ mapping an (Ω, Ξ)-structure to its behaviour was defined in 3.6. To show that B is a monad (in particular, a functor) we need the following proposition (see also 3.6 for the definition of η_M):

Proposition 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 $M = \Omega X \stackrel{\alpha}{\to} X \stackrel{\beta}{\to} \Xi X$, $BM = \Omega \bar{X} \stackrel{\bar{\alpha}}{\to} \bar{X} \stackrel{\bar{\beta}}{\to} \Xi \bar{X}$, $BN = \Omega \bar{Y} \stackrel{\bar{\gamma}}{\to} \bar{Y} \stackrel{\bar{\delta}}{\to} \Xi \bar{Y}$. 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. □

Definition 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 3.6 and $(-)^{\#}$ is the uniquely defined operation on morphisms of $\mathbf{Str}^{\Omega}_{\Xi}$ described in the proposition above.

It follows from lemma 2.2 that B is indeed a monad and, moreover, idempotent:

Proposition 4.3. B *is an idempotent monad.*

The following property of idempotent monads will be useful.

Proposition 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 $\varphi_{MN}^{-1}(g) = g \circ \eta_M$. $\square$

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 and "if" below).

Theorem 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. 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 $\eta_M = \varphi_{MM}^{-1}(\mathrm{id}_{BM})$. We have to check (i) $f = f^{\#} \circ \eta_M$, (ii) $(\eta_M)^{\#} = \mathrm{id}_{BM}$, (iii) $(f^{\#} \circ g)^{\#} = f^{\#} \circ g^{\#}$, (iv) $Bl = (\eta_M \circ l)^{\#}$,[15] and to show that the η_M are split epi (which implies idempotency, see lemma 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. $\square$

Next, we show that $\overline{\mathbf{Str}}_{\Xi}^{\Omega}$ is isomorphic to the category of Eilenberg-Moore algebras of the behaviour monad B and that $\mathbf{bStr}_{\Xi}^{\Omega}$ is isomorphic to the Kleisli category.

Proposition 4.6. $\overline{\mathbf{Str}}_{\Xi}^{\Omega}$ *is isomorphic to the category $(\mathbf{Str}_{\Xi}^{\Omega})^{\mathsf{B}}$ of Eilenberg-Moore algebras for the behaviour monad B.*

Proof. We define an isomorphism $\varphi : \overline{\mathbf{Str}}_{\Xi}^{\Omega} \to (\mathbf{Str}_{\Xi}^{\Omega})^{\mathsf{B}}$. 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 $\theta : BM \to M \in (\mathbf{Str}_{\Xi}^{\Omega})^{\mathsf{B}}$. By definition of $(\mathbf{Str}_{\Xi}^{\Omega})^{\mathsf{B}}$ it holds $\theta \circ \eta_M = id_M$ and by idempotency of B we have that θ is iso (see [4], vol.2, proposition 4.2.3). Therefore, $\theta = \eta_M^{-1}$ which shows that φ is onto. On morphisms, φ is the identity. $\square$

[15](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).

Proposition 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: $(\mathbf{Str}^{\Omega}_{\Xi})_{\mathsf{B}}(M,N) = \mathbf{Str}^{\Omega}_{\Xi}(M,BN) \simeq \mathbf{Str}^{\Omega}_{\Xi}(BM,BN) = \mathbf{bStr}^{\Omega}_{\Xi}(M,N)$. □

The following is an immediate corollary of the two propositions above.

Corollary 4.8. *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 [25] 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 [14], theorem 3.8). One way to state the uniqueness property is the following:

Definition 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.[16]

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 [38]. 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 $\Sigma \in \mathbf{Sig}$ 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 $M' \in \mathsf{Mod}(\Sigma')$

$$\mathsf{Mod}(\sigma)(M') \models_\Sigma \varphi \quad \Leftrightarrow \quad M' \models_{\Sigma'} \mathsf{Sen}(\sigma)(\varphi).$$

[16] That is, for all $X, Y \in \mathcal{C}$ there is at most one morphism $X \to BY$ in $\mathcal{C}$.

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.[17] 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 4.2.2.

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 4.13 and corollary 4.22) and, concretely, by restricting signature morphisms according to the slogan 'no new observations on old sorts' (see definition 6.3).

4.2.1 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 4.10 (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 $\rho : \Omega V \to V\Omega'$, $\tau : V\Xi' \to \Xi V$. We write $\sigma = (V, \rho, \tau)$. 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 $\sigma_1 = (V_1, \rho_1, \tau_1)$ and $\sigma_2 = (V_2, \rho_2, \tau_2)$, then $\sigma_2 \circ \sigma_1 = (V_1 V_2, \rho_2 \circ \rho_1, \tau_1 \circ \tau_2)$.

Definition 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 signature morphism relating the function symbols.

[17]This notation is also suggested by the fibred perspective of section 4.2.2: there, $\models$ will indeed be a single relation, not a collection of relations indexed by signatures.

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 4.12 (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 $M' = \Omega' X \stackrel{\alpha'}{\to} X \stackrel{\beta'}{\to} \Xi' X$ be in $\mathsf{Mod}(\mathcal{C}', \Omega', \Xi')$. We then define the reduct of M' w.r.t. σ as $\sigma^*(M') =$

$$\Omega V X \xrightarrow{\rho_X} V\Omega' X \xrightarrow{V\alpha'} VX \xrightarrow{V\beta'} V\Xi' X \xrightarrow{\tau_X} \Xi V X$$

which is an algebra-coalgebra pair $(V\alpha' \circ \rho_X, \tau_X \circ V\beta')$ 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 4.13. 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 3.8 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 4.14 ($\mathsf{Str}, \mathsf{bStr}, \overline{\mathsf{Str}}$). 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 4.15. $\mathsf{Str}, \overline{\mathsf{Str}}, \mathsf{bStr}$ *are functors.*

Proof. In the case of Str we have to show that $\sigma^*(M)$ is an (Ω, Ξ)-structure for all $M \in \mathbf{Str}^{\Omega'}_{\Xi'}$. By theorem 3.5 it suffices 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.

4.2.2 Fibrations of (Ω, Ξ)-structures

The aim of this section is to extend the characterisations of the categories $\overline{\mathbf{Str}}^{\Omega}_{\Xi}$ (proposition 4.6) and $\mathbf{bStr}^{\Omega}_{\Xi}$ (proposition 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 [18] 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 $(I, M), I \in \mathcal{B}, M \in F(I)$.[18] 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 $(\sigma, f) \circ (\rho, g) = (\sigma \circ \rho, (F\rho)(f) \circ g)$. $p : \mathcal{E} \to \mathcal{B}$ is defined as the first projection.

[18] That is, the class of objects of $\mathcal{E}$ is obtained as the disjoint union of the objects of all $F(I)$, $I \in \mathcal{B}$.

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 $\mathbf{CAT}$. 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 $\mathbf{Str}$ gives us splits fibrations which correspond to the functors $\overline{\mathsf{Str}}$ and bStr.

Definition 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 4.9). 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).[19] 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 4.17. *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: $k^{\mathrm{id}}_M = \mathrm{id}_{BM}$, $\sigma^*(k^{\rho}_M) \circ k^{\sigma}_{\rho^* M} = k^{\rho\circ\sigma}_M$. 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 $\mathbf{Str}$ yield split fibred cate-

[19] By definition 4.9 morphisms into behaviours are unique. Hence k^{σ} is natural.

gories again. The following proposition is a slight variation on exercise 1.7.9 in Jacobs [18].

Proposition 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 4.6) $\sigma^{\#}(\eta^{-1}_{M}) = \sigma^*(\eta^{-1}_{M}) \circ k^{\sigma}_{M}$ 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 4.19. *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.*

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 4.17, 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 4.18, 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 4.6 and 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 4.21). 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 4.15 and corollary 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 $\sigma^* \rho^* = (\rho \circ \sigma)^*$ only up to natural isomorphism.[20] This could be

[20]Or even only up-to some weaker notion of equivalence, see [38].

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.

4.2.3 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 4.20 (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.

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 4.21. *Let $p : \mathcal{E} \to \mathcal{B}$ be a split fibration and B a fibred monad on p, let $\mathsf{Sen} : \mathcal{B} \to \mathbf{Set}$ 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

$$\begin{aligned} &\sigma^*(M) \models\!\!\!\models \varphi \ \Leftrightarrow \ B\sigma^*(M) \models \varphi \ \Leftrightarrow \ \sigma^*(BM) \models \varphi \ \Leftrightarrow \\ &BM \models \mathsf{Sen}(\sigma)(\varphi) \ \Leftrightarrow \ M \models\!\!\!\models \mathsf{Sen}(\sigma)(\varphi). \end{aligned}$$

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 4.22. *Consider the categories of (Ω, Ξ)-structures $\mathbf{Str}^{\Omega}_{\Xi}, \mathbf{bStr}^{\Omega}_{\Xi}, \overline{\mathbf{Str}}^{\Omega}_{\Xi}$ (definition 3.8) and the respective operations $\mathsf{Str}, \mathsf{bStr}, \overline{\mathsf{Str}} : \mathbf{Sig}^{\mathrm{op}} \to \mathbf{CAT}$ (definition 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 4.13),*

2. *$\models\!\!\models$ is behavioural w.r.t. $\models$ and B (definitions 4.20 and 3.6),*

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 4.15 and checking the satisfaction condition as in the proof of the theorem. $\square$

5 Example of Observational Specifications

This section gives an example of the kind of specifications that will be considered in the remainder of this paper.

Figure 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 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 $\mathsf{Account}$, respectively, a model of Person is an (Ω, Ξ)-structure

$$X \times \text{string} \to X \to \text{string} \times \text{string}$$

spec Person
  observers
    $\_.name$: person $\to$ string
    $\_.address$: person $\to$ string
  operations
    $\_.change\_address$: person, string $\to$ person

spec Account
  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 1: Bank Account Example

and a model of type Account 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 Account. Note that σ does not introduce new observers on old sorts: Account has a new algebraic operation *change_owner* involving Person but does not specify a new observer on Person. Consequently, including Person in Account does not change the observational equivalence of Person. Technically, this can be expressed by saying that the reduct functor of σ preserves behaviours (definition 4.13) which in turn (see corollary 4.22) will allow us to build an institution in the next section.

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 4.22 which in turn will give us an institution for modular observational specifications.

6.1 Signatures

For the remainder of the paper, 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 $w \in \{1, \ldots n\}^*$, that is $w = s_1 \ldots s_k$, let $X_w = X_{s_1} \times \ldots X_{s_k}$.

The algebraic functors Ω are then given—componentwise—as a sum of finite products:

$$(\Omega X)_s = \sum_{i \in I} C_i \times X_{w_i}, \quad w_i \in \{1, \ldots n\}^*$$

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 [21] 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 $\alpha(s) = [\alpha(s) \circ \mathrm{in}_1; \ldots]$ and $\beta(s) = \langle \pi_1 \circ \beta(s), \ldots \rangle$. 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. [9]), 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 $\mathrm{Th}(\mathcal{D})$ for the set of formulas valid in $\mathcal{D}$.

Definition 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 5 we have (considering Person as state sort 1, Account as state sort 2, and renaming the function symbols f_i, g_j, h_j):

$$Opns(\Omega')_1 = \{\mathit{change\_address}\}, \qquad Obs(\Xi')_1 = \{\mathit{name}, \mathit{address}\},$$
$$Opns(\Omega')_2 = \{\mathit{new}, \mathit{update}, \mathit{change\_owner}, \mathit{paycharge}\}, \quad Obs(\Xi')_2 = \{\mathit{bal}, \mathit{owner}, \mathit{undo}\}.$$

$Opns(\Sigma)$ may include further operations on strings and integers.

Terms of special importance are the contexts:

Definition 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 5 we have (considering Person as state sort 1, Account as state sort 2):

$$\begin{aligned} Cont(\Xi, \text{string})_1 &= \{\mathit{name}(z_1), \mathit{address}(z_1)\}, \\ Cont(\Xi, \text{int})_2 &= \{\mathit{bal}(\mathit{undo}^n(z_2)) : n \geq 0\}, \\ Cont(\Xi, \text{string})_2 &= \{\mathit{name}(\mathit{owner}(\mathit{undo}^n(z_2))), \mathit{address}(\mathit{owner}(\mathit{undo}^n(z_2))) : n \geq 0\}, \end{aligned}$$

where $\mathit{undo}^n(z_2)$ is the obvious abbreviation.

Next we define a category of signatures. It is a subcategory of the category defined in definition 4.11 and will replace it in the sequel, hence we keep the old name.

Definition 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 $V : \mathbf{Set}^{\mathbf{n}'} \to \mathbf{Set}^{\mathbf{n}}$ which is induced by an injection $\sigma_{\text{sorts}} : \{1, \ldots n\} \to \{1, \ldots n'\}$, second, mappings $\rho_s : \mathit{Opns}(\Omega)_s \to \mathit{Opns}(\Omega')_{\sigma_{\text{sorts}}(s)}$, $\tau_s : \mathit{Obs}(\Xi')_{\sigma_{\text{sorts}}(s)} \to \mathit{Obs}(\Xi)_s$, $1 \leq s \leq n$. Furthermore the following conditions have to be satisfied:

1. - For all $f : C \times X_{s_1} \times \ldots X_{s_k} \to X_s \in \mathit{Opns}(\Omega)_s$ it holds that $\rho_s(f) : C \times X_{\sigma_{\text{sorts}}(s_1)} \times \ldots X_{\sigma_{\text{sorts}}(s_k)} \to X_{\sigma_{\text{sorts}}(s)} \in \mathit{Opns}(\Omega')_{\sigma_{\text{sorts}}(s)}$,
   - 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 4.10. 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 4.22.

Next, we give an explicit description of the final coalgebras that can be found in Cîrstea [7].

Proposition 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 4.21.

Proposition 6.5. *Let $\sigma : (\Omega, \Xi) \to (\Omega', \Xi') \in \mathbf{Sig}$ as in definition 6.3, σ^* as in definition 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 $[Cont(\Xi', B)_s \to B] \simeq [Cont(\Xi, B)_s \to B]$ for all output sets B. With proposition 6.4 one obtains that $\sigma^*(\pi)$ is isomorphic to the final Ξ-coalgebra. $\square$

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 4.22.

Corollary 6.6. *With the definitions of this section the reduct functors σ^* of definition 4.12 preserve behaviours (see definition 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 6.5 and the factorisations are unique, it follows $B\sigma^*(M') \simeq \sigma^*(BM')$. $\square$

6.2 Behavioural Satisfaction

It is well-known that first-order logic gives rise to an institution. Using corollary 6.6 and behavioural satisfaction (definition 4.20) it now follows from corollary 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[21] first-order (Ω, Ξ)-formulas as usual from equations $t = r$ (with the terms $t, r \in Terms(\Omega, \Xi)$ 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

[21] 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.

of parameter sort as elements of $\mathcal{D}$. In particular, terms formed from observers of sort s (see 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 $v : \mathit{Var} \to X + \mathcal{D}$, 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 [31].

Definition 6.7 (Ξ-satisfaction). Let (α, β) an (Ω, Ξ)-structure on X, Var a many-sorted set of variables, $v : \mathit{Var} \to X + \mathcal{D}$ 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:

- $(\alpha, \beta), v \models\!\!\!\models t_1 = t_2$, where t_1, t_2 are terms of state sort s, iff there is a Ξ-bisimulation R on β such that $v^*(t_1) R_s v^*(t_2)$,
- $(\alpha, \beta), v \models\!\!\!\models t_1 = t_2$, 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 $M \models\!\!\!\models \Phi$ iff $M \models\!\!\!\models \varphi$ for all $\varphi \in \Phi$. Moreover, $\Phi \models\!\!\!\models \varphi$ iff for all (Ω, Ξ)-structures M: $M \models\!\!\!\models \Phi$ implies $M \models\!\!\!\models \varphi$.

The next proposition (which is the analogue of [3], 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 $\models\!\!\!\models$ but using set-theoretic equality instead of Ξ-bisimulation in the first clause of definition 6.7.

Proposition 6.8. *Let (α, β) an (Ω, Ξ)-structure and $\varphi \in \mathcal{L}(\Omega, \Xi)$. Then*

$$(\alpha, \beta) \models\!\!\!\models \varphi \text{ iff } (\bar{\alpha}, \bar{\beta}) \models \varphi.$$

Hence $\models\!\!\!\models$ is behavioural w.r.t. $\models$ in the sense of definition 4.20. We can now apply corollary 4.22:

Theorem 6.9. *Let Ω, Ξ be functors as in section 6.1, let* **Sig** *be as in definition 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 4.14), and let* Sen *be the functor mapping signatures to the appropriate set of first-order formulas. Then* $(\mathbf{Sig}, \mathsf{Sen}, \mathsf{Str}, \models\!\!\!\models)$, $(\mathbf{Sig}, \mathsf{Sen}, \mathsf{bStr}, \models\!\!\!\models)$ *are institutions.*

Proof. We use corollary 4.22 (**Sig** according to definition 6.3 is a subcategory of **Sig** according to definition 4.11; but corollary 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 4.22. (ii) is proposition 6.8 and (i) is corollary 6.6. □

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 [38].

Definition 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.,

$$\mathsf{Mod}(Sp) = \{(\alpha, \beta) \in \mathbf{Set}^{\Omega}_{\Xi} : (\alpha, \beta) \models\!\!\!\models Ax\}.$$

Definition 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 6.10:

- $\mathsf{Sig}\, Sp = (\Omega, \Xi)$
- $\mathsf{Mod}\, Sp = \{M \in \mathbf{Str}^{\Omega}_{\Xi} : M \models\!\!\!\models Ax\}$

union For specifications $Sp_1 = (\Omega, \Xi, Ax_1)$ and $Sp_2 = (\Omega, \Xi, Ax_2)$:

- $\mathsf{Sig}(Sp_1 \cup Sp_2) = (\Omega, \Xi)$
- $\mathsf{Mod}(Sp_1 \cup Sp_2) = \{M \in \mathbf{Str}^{\Omega}_{\Xi} : M \models\!\!\!\models Ax_1 \cup Ax_2\}$

translate For $Sp = (\Omega, \Xi, Ax)$ and $\sigma : (\Omega, \Xi) \to (\Omega', \Xi')$:

- $\mathsf{Sig}(\mathsf{translate}\, Sp\, \mathsf{by}\, \sigma) = (\Omega', \Xi')$
- $\mathsf{Mod}(\mathsf{translate}\, Sp\, \mathsf{by}\, \sigma) = \{M \in \mathbf{Str}^{\Omega'}_{\Xi'} : \sigma^* M \models\!\!\!\models Ax\}$

hide For $Sp = (\Omega', \Xi', Ax')$ and $\sigma : (\Omega, \Xi) \to (\Omega', \Xi')$:

- $\mathsf{Sig}(\mathsf{hide}\, Sp\, \mathsf{by}\, \sigma) = (\Omega, \Xi)$
- $\mathsf{Mod}(\mathsf{hide}\, Sp\, \mathsf{by}\, \sigma) = \{\sigma^* M' \in \mathbf{Str}^{\Omega}_{\Xi} : M' \models\!\!\!\models Ax'\}$

Example 6.12. Consider the specification that is given by extending the signature in section 5 by the axioms given in figure 2. Let $Sp_1 = (\Omega, \Xi, Ax)$ be the flat specification of Person as given by the signature (Ω, Ξ) of section 5 and by the set of axioms Ax for Person. Let $Sp_2 = (\Omega', \Xi', Ax')$ be the flat specification of Account as given by the signature (Ω', Ξ') of section 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 Account 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 3.2. It follows from proposition 3.11 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).

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 Account
  import Person
  observers ...
  operations ...
  axioms
    $\forall x \in \text{account},\ \forall s \in \text{string},\ \forall n \in \text{int},\ \forall p \in \text{person}:$
      $new(s).bal = 0,$
      $new(s).owner.name = s,\ \ new(s).owner.address = \epsilon,$
      $new(s).undo = new(s),$

      $x.update(n).bal = x.bal + n,$
      $x.update(n).owner = x.owner,$
      $x.update(n).undo = x,$

      $x.change\_owner(p).bal = x.bal,$
      $x.change\_owner(p).owner = p,$
      $x.change\_owner(p).undo = x,$

      $x.paycharge.bal = x.bal - 10,$
      $x.paycharge.owner = x.owner,$
      $x.paycharge.undo = x.$

Figure 2: Axioms for Person and Account

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 6.2).

The set of variables of parameter sort of a context c is denoted by $Var(c)$. We write $\forall Var(c)$ to denote quantification over all these variables in $Var(c)$. Next we formulate a coinductive proof principle for (Ω, Ξ)-logic which is expressed by the following axiom:

Definition 7.1 **(**CoInd_Ξ**).**

$$(\mathrm{CoInd}_\Xi)_s = \forall x, y \in X : \bigwedge_{c \in Cont(\Xi)_s} (\forall Var(c) : c[x] = c[y]) \;\Rightarrow\; x = y$$

and $\mathrm{CoInd}_\Xi = \{(\mathrm{CoInd}_\Xi)_s, 1 \leq s \leq n\}$ 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 $\mathsf{Account}$ example from section 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 7.2 **(**(Ω, Ξ)**-proof system).** Let Ω, Ξ be functors as in section 6, let $\mathcal{D}$ be a data algebra and $\mathrm{Th}(\mathcal{D})$ the set of infinitary first-order formulas satisfied by $\mathcal{D}$. We write $\Phi \vdash_\Xi \varphi$ iff $\Phi \cup \{\mathrm{CoInd}_\Xi\} \cup \mathrm{Th}(\mathcal{D}) \vdash \varphi$ where $\vdash$ denotes derivability w.r.t. a sound and complete proof system for infinitary first-order logic as given, for instance, in Keisler [20].

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., [26, 1]) this property is not assumed and therefore has first to be checked before the coinductive proof principle can be applied.

Theorem 7.3 (Soundness and Completeness).

$$\Phi \vdash_\Xi \varphi \;\Leftrightarrow\; \Phi \models\!\!\models \varphi.$$

Proof. (Sketch.) Soundness follows from the remarks above. The proof of completeness uses the completeness proof in [14] 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 [14] the data algebra is not fixed in advance but part of the specification. Now, using $\Phi \cup \mathrm{Th}(\mathcal{D})$ as a specification for observational algebras and observing that, according to Scott's theorem (see e.g. [20]), $\mathrm{Th}(\mathcal{D})$ 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 $\Phi \cup \mathrm{Th}(\mathcal{D})$ are in one-to-one correspondence to the (Ω, Ξ)-structures for Φ. Showing that this correspondence preserves and reflects validity finishes the proof. □

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 [2] 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 [6]. 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 7.4. Consider the example of the $\mathsf{Account}$ specification from figure 2 and suppose one wants to show that

$$\forall x \in \text{account} : x.\mathit{paycharge} = x.\mathit{update}(-10).$$

We can write the $\mathsf{Account}$-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 2. Note that the proof uses that we may deduce from $x.paycharge = x.update(-10)$ the equality $x.paycharge.t = x.update(-10).t$ 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.

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 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 [37]. Completeness uses Borzyszkowski's [5] completeness result for institutions. According to [5] it has to be checked that the institutions satisfy the amalgamation and interpolation properties. This can be done as in [13].

$\square$

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 paper is still less expressive than the one of [14]. 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 $isin : \text{bank} \times \text{account} \to \text{Bool}$ 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 paper is also more general than [14] because the development in chapters 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 6 and allow for functors describing non-deterministic coalgebras (involving + and/or powerset). Since the use of equational logic in this paper is essentially due to the fact that the special format of the functors Ξ allows to transform them into functors for algebras (see [21]) 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 [29, 34, 33, 19, 24, 22, 23]).

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. Many thanks also go to Dirk Pattinson for explaining some fine points of fibred category theory. The diagrams were drawn with Paul Taylor's macros.

References

[1] 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.

[2] M. Bidoit and R. Hennicker. Behavioural theories and the proof of behavioural properties. *Theoretical Computer Science*, 175:3–55, 1996.

[3] M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. *Science of Computer Programming*, 25:149–186, 1995.

[4] Francis Borceux. *Handbook of Categorical Algebra.* Cambridge University Press, 1994.

[5] 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.

[6] 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.

[7] 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.

[8] Răzvan Diaconescu. Behavioural coherence in object-oriented algebraic specification. Technical Report IS-RR-98-0017F, Japan Advanced Institute for Science and Technology, 1998.

[9] J. Goguen and G. Malcolm. A hidden agenda. Technical Report CS97-538, UCSD, 1997.

[10] 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.

[11] 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.

[12] H. Peter Gumm. Functors for coalgebras. 1998. Preprint.

[13] 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.

[14] R. Hennicker and M. Bidoit. Observational logic. In Armando Haeberer, editor, *Algebraic Methodology and Software Technology (AMAST'98)*, volume 1548 of *LNCS*. Springer, 1999.

[15] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62, 1997.

[16] 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.

[17] 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.

[18] Bart Jacobs. *Categorical Logic and Type Theory*. Elsevier, 1998.

[19] Bart Jacobs. The temporal logic of coalgebras via galois algebras. Technical Report CSI-R9906, Computing Science Institute Nijmegen, 1999.

[20] H. J. Keisler. *Model Theory for Infinitary Logic*. North-Holland, 1971.

[21] 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.

[22] 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. To appear.

[23] Alexander Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität München, 2000. Forthcoming.

[24] Alexander Kurz. Specifying coalgebras with modal logic. *Theoretical Computer Science*, 260(1-2), 2000. To appear.

[25] Saunders Mac Lane. *Category Theory for the Working Mathematician*. Springer, 1971.

[26] G. Malcolm and J. Goguen. Proving correctness of refinement and implementation. Technical Report PRG-114, Oxford University Computing Laboratory, 1994.

[27] Grant Malcolm. Behavioural equivalence, bisimulation, and minimal realisation. *LNCS*, 1130:359–378, 1996.

[28] Ernest G. Manes. *Algebraic Theories*. Springer, 1976.

[29] Lawrence Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.

[30] 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.

[31] Horst Reichel. *Initial computability, algebraic specifications, and partial algebras*. Oxford, Clarendon Press, 1987.

[32] Horst Reichel. An approach to object semantics based on terminal coalgebras. *Mathematical Structures in Computer Science*, 5(2):129–152, June 1995.

[33] 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.

[34] Martin Rößiger. From modal logic to terminal coalgebras. *Theoretical Computer Science*, 260(1-2), 2000. To appear.

[35] J. Rutten and D. Turi. Initial algebra and final coalgebra semantics for concurrency. Report CS-R9409, CWI, Amsterdam, 1994.

[36] J. J. M. M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*. To appear.

[37] D. Sannella and A. Tarlecki. Specifications in arbirary institutions. *Information and Computation*, 76:165–210, 1988.

[38] 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.