(Ω, Ξ)-Logic: On the Algebraic Extension of Coalgebraic Specifications

Rolf Hennicker

Institut für Informatik
Universität Augsburg
Universitätsstr. 14, D-86135 Augsburg
Germany

Alexander Kurz

Institut für Informatik
Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 München
Germany

Abstract

We present an 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, moreover, leads to a simplification of the coalgebraic structure of the models of a specification. The framework of (Ω, Ξ)-logic can be considered as the result of a translation of concepts of observational logic (cf. [9]) into the coalgebraic world. As a particular outcome we obtain the notion of an (Ω, Ξ)-structure and a sound and complete proof system for (first-order) observational properties of specifications.

1 Introduction

In this paper we propose a logical framework, called (Ω, Ξ)-logic, 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 [9] and from similar ideas of swinging data types (Padawitz [19]) and hidden algebra (Goguen and Malcolm [6]). 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 [21], Jacobs [13]).

The specific goals of our approach are to integrate constants and n-ary op-

This paper will be published in
Electronic Notes in Theoretical Computer Science, Volume 19
URL: `www.elsevier.nl/locate/entcs`

erations, [1] 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 and to provide a sound and complete proof system for the verification of observational properties.

The starting point of our study is a consideration of standard coalgebraic specification techniques in the case where a polynomial functor $\Xi : \mathbf{Set} \to \mathbf{Set}$ is used to represent the possible operations on a (non-observable) state space X. As a simple example let us consider the following usual operations on bank accounts

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

which are extracted from the functor

$$\Xi X = \mathbb{Z} \times X^{\mathbb{Z}}$$

as the projections of the transition function $\beta : X \to \mathbb{Z} \times X^{\mathbb{Z}}$ (whereby, for update, we use the fact that functions $X \to X^{\mathbb{Z}}$ correspond to functions $X \times \mathbb{Z} \to X$). According to the definition of Ξ *both* operations *bal* and *update* are used to define an indistinguishability relation for bank accounts (formally expressed by Ξ-bisimulation). Thereby two bank accounts a and b are indistinguishable (in the following also called observationally equivalent), if each of the observable experiments $\_.bal$, $\_.update(n).bal$, $\_.update(n_1).update(n_2).bal$, ... 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 terminal Ξ-coalgebra) and *is even not adequate* 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 even 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). To decide what should count as an observer and what as an operation is part of the task of the specifier. This is analogous to algebraic specifications where also a decision has to be made what operations are to be considered as constructors and what 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).

[1] Recently, n-ary operations have been integrated into the framework of extended hidden algebra, see Diaconescu [5] and Roşu and Goguen [22].

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 $x.update(n).bal = x.bal + n$.

The structure of the paper is the following. Section 2 contains notational conventions and recalls some technical preliminaries. Then, 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). Our morphism notion for (Ω, Ξ)-structures is chosen in such a way that it reflects the relationships between the observable behaviour of (Ω, Ξ)-structures.[2] Also, 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.

In section 4 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 Sp (i.e. formulas which are Ξ-satisfied by all models of Sp), we present (in section 5) a sound and complete proof system for (Ω, Ξ)-logic.

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*, 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 [17] we call a pair (α, β) of an algebra $\alpha : \Omega X \to X$ and a coalgebra

[2] Algebra-coalgebra pairs are also considered in Malcolm [17], 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 [17] indeed force algebra-coalgebra pairs to be (Ω, Ξ)-structures.

$\beta : X \to \Xi X$ 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 **Set** of sets and functions. Furthermore the functors Ω, Ξ are required to preserve weak pullbacks [3] and we assume that a final Ξ-coalgebra always exists.

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 [24], [18]). 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 $R \subset X \times X$ such that there is a function $\gamma : R \to \Xi R$ 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 a function $\delta : \Omega R \to R$ 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 $! : \beta \to \pi$. This morphism $!$ is intimately related to the greatest Ξ-bisimulation on β because $!$ identifies exactly the bisimilar elements of X. Categorically this property may be expressed by the following lemma (Rutten and Turi [23], Malcolm [17]).

Lemma 2.1 *Let $\beta : X \to \Xi X$ be a Ξ-coalgebra, $\pi : Z \to \Xi Z$ a final Ξ-coalgebra. Then $R \subset X \times X$ is the greatest bisimulation on β iff the diagram below is a pullback in* **Set***:*

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

[3] *Weak* means that the arrow into the weak pullback may not be unique, see Rutten [24] and Gumm [7] for a discussion.

It might be interesting to note that the proof of this lemma is the essential point where the requirement enters that the functor Ξ preserves weak pullbacks.

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 observational equivalence [4] expressed by the coalgebraic part.

Definition 3.1 ((Ω,Ξ)-structures) *Let Ω, Ξ be functors on* **Set** *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 a function $h : \Omega Z \to Z$ 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 3.6, 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. □

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. [5]

Definition 3.3 (Behaviour) *Let $\Omega X \xrightarrow{\alpha} X \xrightarrow{\beta} \Xi X$ be an algebra-coalgebra pair, $\pi : Z \to \Xi Z$ terminal in $\mathbf{Set}_\Xi$ and $! : \beta \to \pi$. Furthermore let $X \xrightarrow{e} \mathrm{Im}(!) \xrightarrow{m} Z$ be the unique factorisation of $!$ (as a function in* **Set***) through its*

[4] Recall that the notion of observational equivalence is formalised in the coalgebraic approach as the greatest Ξ-bisimulation.

[5] The notion of minimal realisation in Malcolm [17] is equivalent to our notion of behaviour.

image. Then any algebra-coalgebra pair $(\bar{\alpha}, \bar{\beta})$ on the image of ! such that the diagram below 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 [24] that a coalgebra morphism uniquely factors through its image it is clear that $\bar{\beta}$ always exists. 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 **Set** are split). □

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. □

We can now prove the converse of proposition 3.2 and thereby give a second characterisation of (Ω, Ξ)-structures. [6]

Theorem 3.6 *Let $\Omega X \stackrel{\alpha}{\to} X \stackrel{\beta}{\to} \Xi X$ 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 $X \stackrel{e}{\to} \bar{X} \stackrel{m}{\to} Z$ for the factorisation of $! : \beta \to \pi$. From Rutten [24] 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 $\Omega R \stackrel{\delta}{\to} R \stackrel{\gamma}{\to} \Xi R$ the functions making R into a congruence and a bisimulation. Consider the following three layered diagram:

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

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) $\bar{\alpha} \circ \Omega e = e \circ \alpha$. $\square$

In order to obtain a category of (Ω, Ξ)-structures we still need an appropriate notion of morphism. Of course, we could use the obvious notion of an algebra-coalgebra pair morphism (see section 2). Since this does not reflect the relationships between the *observable behaviour* of algebras, we have chosen a different definition which implies in particular that (Ω, Ξ)-structures are isomorphic iff they have the same behaviour:

Definition 3.7 **((Ω, Ξ)-morphisms, $\mathbf{Set}^{\Omega}_{\Xi}$)** *Let (α_1, β_1), (α_2, β_2) be (Ω, Ξ)-*

[6] Theorem 3.6 is closely related to the result of Rutten and Turi [23] saying (very roughly) that a final semantics has an equivalent initial semantics if bisimulation is congruence.

structures. An (Ω, Ξ)*-morphism* $f : (\alpha_1, \beta_1) \to (\alpha_2, \beta_2)$ *is a function* f *that is both an algebra morphism* $f : \bar{\alpha}_1 \to \bar{\alpha}_2$ *and a coalgebra morphism* $f : \bar{\beta}_1 \to \bar{\beta}_2$ *between the respective behaviours* $(\bar{\alpha}_1, \bar{\beta}_1)$ *and* $(\bar{\alpha}_2, \bar{\beta}_2)$*. The category of* (Ω, Ξ)*-structures together with their morphisms is called* $\mathbf{Set}^{\Omega}_{\Xi}$*.*

An interesting consequence of this definition is:

Proposition 3.8 $\mathbf{Set}^{\Omega}_{\Xi}$ *is equivalent to the full subcategory consisting of the behaviours in* $\mathbf{Set}^{\Omega}_{\Xi}$*.*

It follows that, from a categorical perspective, it is sufficient to work in the much simpler category of behaviours. Nevertheless, from the point of view of computer science, it is essential to have the larger category where the implementations live.

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 [24] or Jacobs and Rutten [11].

There, the coalgebra $f : X \to \Xi X$ is said to be a coinductive definition of the function $\alpha : X \to Z$ if $Z \xrightarrow{\pi} \Xi Z$ is the final coalgebra and α is the unique coalgebra morphism, see the left hand diagram below:

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

In our context, we want to define the algebraic operations $\alpha : \Omega X \to X$ on a coalgebra $\beta : X \to \Xi X$ coinductively. First, let β be the final coalgebra and consider the right hand diagram above. Then any function $f : \Omega X \to \Xi\Omega X$ provides a coinductive definition of algebraic operations $\alpha : \Omega X \to X$. To see what f has to be in our example (α as *update*) recall $\Omega X = X \times \mathbb{Z}$, $\Xi X = \mathbb{Z}$, $\Xi\alpha = \mathrm{id}_{\mathbb{Z}}$, $\beta = \mathit{bal}$. It is easy to see that $f(x, n) = x.\mathit{bal} + n$ 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. [7] 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. [8]

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

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 $\bar{\alpha} \circ \Omega e = e \circ \alpha$. $\square$

[7] 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).

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

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 X. That is, a coinductive definition imposes closure conditions on a coalgebra X, in other words, forces the coalgebra to contain enough "good" elements. In this respect our approach differs fundamentally from other approaches like Jacobs [12], Gumm [8], and Kurz [15] where specifications force coalgebras to avoid "bad" elements.

4 (Ω, Ξ)-logic

In this section we show how first-order logic can be used to specify (Ω, Ξ)-structures. The important point is that Ξ-bisimulation is used to interpret the equality symbol w.r.t. states.

For the remainder of the paper, we consider the case where Ω is a sum of finite products, more precisely,

$$\Omega X = \sum_{i \in I} C_i \times X^{a_i},$$

where the C_i are a finite number of arbitrary (but fixed) sets and the arities a_i range over the natural numbers. In particular, we allow binary (a_i-ary) operations on states.

Ξ is a functor of the kind

$$\Xi X = \prod_{j \in J_1} X^{A_j} \times \prod_{j \in J_2} B_j^{A_j},$$

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.

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 $\alpha = [\alpha \circ \mathrm{in}_1; \ldots]$ and $\beta = \langle \pi_1 \circ \beta, \ldots, \pi_n \circ \beta \rangle$ (where $J_1 = \{1, \ldots, m\}, J_2 = \{m+1, \ldots, n\}$). To be able to define the notion of (Ω, Ξ)-terms 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. [6]), 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 $\mathit{Opns}(\Sigma)$. It is required that $\mathit{Opns}(\Sigma)$ is disjoint from $\mathit{Opns}(\Omega) \cup \mathit{Obs}(\Xi)$ and that every element of a parameter set is denoted by some ground Σ-term. Given a logic based on the terms formed from $\mathit{Opns}(\Sigma)$ and variables, we will write $\mathrm{Th}(\mathcal{D})$ for the set of formulas valid in $\mathcal{D}$.

Definition 4.1 (*$\mathit{Opns}(\Omega)$*, *$\mathit{Obs}(\Xi)$*, (Ω, Ξ)-terms)
Let Ω, Ξ be functors as above. The set $\mathit{Opns}(\Omega)$ consists of typed function symbols $f_i : C_i \times X^{a_i} \to X$ for every $i \in I$. The set $\mathit{Obs}(\Xi)$ consists of typed function symbols $g_j : X \times A_j \to X, j \in J_1$ and $h_j : X \times A_j \to B_j, j \in J_2$. The set $\mathit{Terms}(\Omega, \Xi)$ of (Ω, Ξ)-terms is formed in the usual way using a countable set of variables Var *and the function symbols of $\mathit{Opns}(\Omega) \cup \mathit{Obs}(\Xi) \cup \mathit{Opns}(\Sigma)$.*

Using (Ω, Ξ)-terms we can define the set $\mathcal{L}(\Omega, \Xi)$ of many-sorted [9] first-order (Ω, Ξ)-formulas as usual from equations $t = r$ (with the terms $t, r \in \mathit{Terms}(\Omega, \Xi)$ of the same sort), the logical connectives $\neg, \wedge, \vee$ and the quantifiers $\forall, \exists$. In some cases we will also consider infinitary conjunctions and disjunctions over countable sets of formulas.

Given an (Ω, Ξ)-structure (α, β) on X and a valuation for the variables, we have the usual interpretation of terms of state sort as elements of X and of terms of parameter sort as elements of $\mathcal{D}$. In particular, terms formed from observers $g_j : X \times A_j \to X, j \in J_1$ and $h_j : X \times A_j \to B_j, j \in J_2$ are interpreted by using the isomorphisms

$$X \times A_j \to X \simeq X \to X^{A_j}, \quad X \times A_j \to B_j \simeq X \to B_j^{A_j}.$$

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 $\beta : X \to \prod_{j \in J_1} X^{A_j} \times \prod_{j \in J_2} B_j^{A_j}$. Therefore

$$\begin{aligned} v^*(g_j(t_1, t_2)) &= \pi_j \circ \beta(v^*(t_1))(v^*(t_2)) \in X, \\ v^*(h_j(t_1, t_2)) &= \pi_j \circ \beta(v^*(t_1))(v^*(t_2)) \in B_j. \end{aligned}$$

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 [20].

---

[9] One sort for each of X, A_j, B_j, C_i. The name X is used synonymously as a sort called *state sort* and also for the set of states.

Definition 4.2 (Ξ-satisfaction) *Let Ω, Ξ be functors as above, (α, β) an (Ω, Ξ)-structure on X,* Var *a set of variables, $v : \mathrm{Var} \to X + \mathcal{D}$ a valuation and $\varphi \in \mathcal{L}(\Omega, \Xi)$. Then $(\alpha, \beta), v \models_\Xi \varphi$ is defined by induction on the structure of φ:*

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

Another way to achieve that equality is interpreted as Ξ-bisimulation is to interpret the equality symbol as equality in the *behaviour* of a structure. The next proposition shows that both ways to define satisfaction are indeed equivalent. (It is the analogue of Bidoit et al. [4], theorem 3.11.) Thereby, we write $\models$ for the standard satisfaction relation that is defined as $\models_\Xi$ but using standard set-theoretic equality instead of Ξ-bisimulation in the first clause of definition 4.2.

Proposition 4.3 *Let Ω, Ξ be functors as above, (α, β) an (Ω, Ξ)-structure, and $\varphi \in \mathcal{L}(\Omega, \Xi)$. Then*

$$(\alpha, \beta) \models_\Xi \varphi \textit{ iff } (\bar{\alpha}, \bar{\beta}) \models \varphi,$$

Now, we introduce specifications and the class of models satisfying a given specification.

Definition 4.4 ((Ω, Ξ)-specification) *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 $\mathrm{Mod}(Sp)$ *of the (Ω, Ξ)-specification Sp consists of all (Ω, Ξ)-structures that Ξ-satisfy Ax, i.e.,*

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

Example 4.5 The following specification of bank accounts (taken from [9]) has additionally to the observer *bal* an observer *undo* which is intended to reconstruct the previous state of an account after having performed an action. Hence, by using *undo* one can potentially reveal more information (namely the account's history) than a single balance check would provide. Thus *undo* has indeed to be declared as an observer. (We may call *undo* an "indirect" observer because it only leads to visible output in combination with the "direct" observer *bal*.) In addition to the *update* operation the specification contains a constant *new* (representing the initial state of an account) and an operation *paycharge* which reduces the balance of an account by a constant monthly fee.

spec ACCOUNT

observers

$\_.bal$: account $\to$ int

$\_.undo$: account $\to$ account

operations

new : $\to$ account

$\_.update\_$: account, int $\to$ account

$\_.paycharge$: account $\to$ account

axioms

$\forall x \in$ account, $\forall n \in$ int :

$new.bal = 0,\ new.undo = new$

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

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

The above notation shows the concrete syntax of the specification. Its abstract syntax is given by the functor

$$\Xi X = \mathbb{Z} \times X$$

corresponding to the two observers $\langle bal, undo\rangle : X \to \mathbb{Z} \times X$ and by the functor

$$\Omega X = 1 + X \times \mathbb{Z} + X$$

corresponding to the operations $[new; update; paycharge] : 1{+}X{\times}\mathbb{Z}{+}X \to X$. A possible model of the specification ACCOUNT which satisfies the axioms even literally can be defined in terms of lists of integers. Another model which Ξ-satisfies the axioms (but not literally) can be constructed by using the well-known array with pointer realization of lists.

In the above specification the behaviour of the operations is specified by a complete case distinction w.r.t. the given observers. Moreover, it is not difficult to see that this specification is a coinductive definition in the sense of section 3. 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 only (Ω, Ξ)-structures are admissible models).

5 (Ω, Ξ)-proof system

In this section we give a sound and complete proof system for (Ω, Ξ)-logic. 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.

Definition 5.1 (Ξ-context) *The set $Cont(\Xi)$ of observable Ξ-contexts consists of the terms of output sort formed from the set of function symbols $Obs(\Xi)$, variables of parameter sort, and a special variable z of state sort. Substitution of a term t in the context c for the variable z is denoted by $c[t]$.*

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 variables in $Var(c)$. Next we formulate a coinductive proof principle for (Ω, Ξ)-logic which is expressed by the following axiom:

Definition 5.2 (CoInd_Ξ)

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

Whether the axiom is infinitary depends on the bisimulation defined by the coalgebra functor Ξ. In the ACCOUNT example from the last section 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 5.3 ((Ω, Ξ)-proof system) *Let Ω, Ξ be functors as above, 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 [14].*

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 Malcolm and Goguen [16], Bidoit and Hennicker [1]) this property is not assumed and therefore has first to be checked before the coinductive proof principle can be applied.

Theorem 5.4 (Soundness)

$$\Phi \vdash_\Xi \varphi \;\Rightarrow\; \Phi \models_\Xi \varphi.$$

Proof. Follows from the remarks above. □

Theorem 5.5 (Completeness) *Let $\mathcal{D}$ be a countable data algebra and $\mathrm{Th}(\mathcal{D})$ its theory w.r.t. infinitary first-order logic. Then*

$$\Phi \models_\Xi \varphi \;\Rightarrow\; \Phi \vdash_\Xi \varphi.$$

Proof. (Sketch.) The proof uses the completeness proof in [9] 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 [9] 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. [14]), $\mathrm{Th}(\mathcal{D})$ 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, see the example below. 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).

Example 5.6 Consider the example of the ACCOUNT specification from the last section and suppose one wants to show that

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

As we have seen above the axiom CoInd_Ξ becomes

$$\forall x, y \in \text{account} : (\bigwedge_{i \in \mathbb{N}} x.\mathit{undo}^i.\mathit{bal} = y.\mathit{undo}^i.\mathit{bal}) \;\Rightarrow\; x = y.$$

Instantiating x with $x.\mathit{paycharge}$ and y with $x.\mathit{update}(-10)$, we see that it is sufficient to prove the infinitary formula

$$\bigwedge_{i \in \mathbb{N}} x.\mathit{paycharge}.\mathit{undo}^i.\mathit{bal} = x.\mathit{update}(-10).\mathit{undo}^i.\mathit{bal},$$

which follows directly from the corresponding axioms.

6 Conclusion

(Ω, Ξ)-logic provides the foundations of a flexible specification technique for state-based systems which extends standard coalgebraic specifications by incorporating the basic ideas of observational logic. For simplicity we have only considered here (Ω, Ξ)-structures with a single-sorted state space. The extension to the many-sorted case should be straightforward. Important next steps of our approach are the construction of structured (Ω, Ξ)-specifications for modular descriptions of large systems (which is already included in observational logic) and the investigation of refinement relations between (Ω, Ξ)-specifications together with associated proof techniques. We are confident that for this purpose we can use results of [3] and [10].

Acknowledgements

We want to thank Pietro Cenciarelli for the many discussions initiating our work on the connection between algebraic and coalgebraic specifications and also Alexander Knapp for carefully reading previous drafts. The diagrams are drawn with Paul Taylor's macro package.

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, Berlin, 1994. Springer.

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

[3] M. Bidoit and R. Hennicker. Modular correctness proofs of behavioural implementations. *Acta Informatica*, 35:951–1005, 1998.

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

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

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

[7] H. Peter Gumm. Functors for coalgebras. *Algebra Universalis.* To appear.

[8] 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.

[9] R. Hennicker and M. Bidoit. Observational logic. In *Proc. of AMAST'98, 7th International Conference on Algebraic Methodology and Software Technology*, LNCS 1548, pages 263–277, 1999.

[10] Rolf Hennicker. Structured specifications with behavioural operators: Semantics, proof methods and applications. Habilitation thesis, Universität München, 1997.

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

[12] Bart Jacobs. Mongruences and cofree coalgebras. *Lecture Notes in Computer Science*, 936, 1995.

[13] 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*. Kluwer Acad. Publ., 1996.

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

[15] Alexander Kurz. A co-variety-theorem for modal logic. *Proceedings of Advances in Modal Logic, Uppsala*, pages 222–230, 1998. `http://www.informatik.uni-muenchen.de/~kurz`.

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

[17] Grant Malcolm. Behavioural equivalence, bisimulation, and minimal realisation. *Lecture Notes in Computer Science*, 1130:359–378, 1996.

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

[19] 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, Berlin, 1996. Springer.

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

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

[22] G. Roşu and J. Goguen. Hidden congruent deduction. *Proc. International Workshop on First Order Theorem Proving*, 1998. `http://www-cse.ucsd.edu/users/goguen/ps/cong.ps.gz`.

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

[24] Jan Rutten. Universal coalgebra: A theory of systems. Report CS R 9652, CWI, Amsterdam, 1996.