On the Duality between Observability and Reachability [*]

Michel Bidoit[1], Rolf Hennicker[2], and Alexander Kurz[3]

[1] Laboratoire Spécification et Vérification (LSV), CNRS & ENS de Cachan, France
[2] Institut für Informatik, Ludwig-Maximilians-Universität München, Germany
[3] Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands

Abstract. Observability and reachability are important concepts in formal software development. While observability concepts allow to specify the required observable behavior of a program or system, reachability concepts are used to describe the underlying data in terms of data type constructors. In this paper we show that there is a duality between observability and reachability, both from a methodological and from a formal point of view. In particular, we establish a correspondence between observer operations and data type constructors, observational algebras and constructor-based algebras, and observational and inductive properties of specifications. Our study is based on the observational logic institution [7] and on a novel treatment of reachability which introduces the institution of constructor-based logic. The duality between both concepts is formalised in a category-theoretic setting.

1 Introduction

An important role in software specification and program development is played by observability and reachability concepts which deal with different aspects of software systems. While observational approaches focus on the observable properties of a system, reachability notions describe the underlying data manipulated by the system. Both concepts are treated in a formal way in various algebraic specification frameworks.

Considering observability, one can distinguish two main approaches: The first one is based on an observational equivalence relation between algebras which is used to abstract from the (standard) model class of a specification; cf. e.g. [17]. The second approach relaxes the (standard) satisfaction relation so that all algebras are accepted as observational models of a specification which satisfy a given set of axioms up to observational equality of the elements of the algebra. (This idea was originally introduced by Reichel; cf. e.g. [16].) Thereby two elements are considered to be observationally equal if they cannot be distinguished by a set of observable experiments.

Concerning reachability, the standard approach is to introduce a set of data type constructors and to consider those algebras which are reachable w.r.t. the

[*] This work is partially supported by the ESPRIT Working Group 29432 CoFI and by the Bayer. Forschungsstiftung.

F. Honsell and M. Miculan (Eds.): FOSSACS 2001, LNCS 2030, pp. 72–87, 2001.

given constructors. Most algebraic specification languages incorporate features to express reachability like, for instance, the CASL language [15]. Since observability and reachability are used for different purposes both concepts look quite unrelated. It is the aim of this study to show that there is a methodological and even formal duality between the two concepts which we believe contributes to a clarification of specification methodologies and their semantic foundations.[1] The correspondence will be based on the following working hypothesis (in the spirit of Hoare [9]):

> The model class of a specification SP describes
> the class of all correct realizations of SP.

The underlying paradigm of the algebraic approach is to model programs by (many-sorted) algebras and to describe the properties of these algebras by logical axioms provided by some specification SP. Then a program is a correct realization if it is a model of SP. Based on these assumptions we will study algebraic frameworks for observability and for reachability and we will compare both concepts.

First, in Section 2, we give an overview of the observational logic institution [7] which we will use as the basis for formalising observability. Then, in Section 3, we discuss reachability and we introduce a new institution, called constructor-based logic, to express reachability issues in accordance with the above working hypothesis. For this purpose we introduce, in particular, the notions of a constructor-based algebra and the constructor-based satisfaction relation. Section 4 exhibits the syntactic and semantic correspondences between all notions used in observational logic and in constructor-based logic. In Section 5, we focus on the properties that are valid consequences of an observational or constructor-based specification. In each case these properties can be characterized by the standard first-order theory of a class of algebras which represent the "idealized" models (also called "black box view") of a specification. By comparing the black box views of observational and constructor-based specifications it turns out that fully abstract models correspond to reachable models.

The results obtained so far show a syntactic, semantic and methodological analogy between observational and constructor-based specifications. In Section 6, we show that this correspondence can even be characterized by a formal duality in a category-theoretic setting. Thereby the syntactic aspects of the observational and the constructor-based notions are expressed by appropriate (pairs of) functors and the semantic aspects are expressed by using algebra-coalgebra pairs; cf. also [14]. Finally, some concluding remarks are given in Section 7.

We assume that the reader is familiar with the basic notions of algebraic specifications (see e.g., [13]), like the notions of (many-sorted) *signature* $\Sigma = (S, OP)$ (where S is a set of *sorts* and OP is a set of *operation symbols* $op : s_1, \ldots, s_n \to s$), *(total)* Σ*-algebra* $A = ((A_s)_{s \in S}, (op^A)_{op \in OP})$, class $\mathrm{Alg}(\Sigma)$ of all Σ-algebras, Σ*-term algebra* $T_\Sigma(X)$ over a family of variables X and *interpretation* $I_\alpha : T_\Sigma(X) \to A$ w.r.t. a *valuation* $\alpha : X \to A$.

[1] In the context of automata theory a similar duality was already investigated by Arbib and Manes in [2].

2 The Observational Logic Institution

Observability concepts provide a means to specify the observable behaviour of software systems in an abstract, implementation independent way. They take into account our working hypothesis (of the Introduction) in the sense that any program which satisfies the observable behaviour described by a specification SP is considered as a correct realization of SP. Observability concepts are particularly suited to specify the observable properties of state-based systems since they allow us to abstract from concrete state representations and to consider any two states which cannot be distinguished by observable experiments as "observationally equal". A flexible approach to formalise observable experiments is suggested (in a similar way) e.g. in [7] and [6] where the operations of an algebraic signature are split into a set of "observer operations" for building observable experiments and the "other" operations which can be used to manipulate states. In this study we will use the observational logic institution to formalise observability. An overview of observational logic is given in the remainder of this section (for more details see [7]).

Definition 1 (Observational signature). *Let $\Sigma = (S, OP)$ be a signature and $S_{\mathrm{Obs}} \subseteq S$ be a set of* observable sorts*. An* observer *is a pair (op, i) where $op : s_1, \ldots, s_n \to s \in OP$, $1 \leq i \leq n$, and $s_i \notin S_{\mathrm{Obs}}$.*[2] *$(op, i)$ is a* direct observer *of s_i if $s \in S_{\mathrm{Obs}}$, otherwise it is an* indirect observer*. If $op : s_1 \to s$ is a unary observer we will simply write op instead of $(op, 1)$. An observational signature $\Sigma_{\mathrm{Obs}} = (\Sigma, S_{\mathrm{Obs}}, OP_{\mathrm{Obs}})$ consists of a signature $\Sigma = (S, OP)$, a set $S_{\mathrm{Obs}} \subseteq S$ of observable sorts and a set OP_{Obs} of observers (op, i) with $op \in OP$.*

Any observational signature determines a set of observable contexts which represent the observable experiments. In the following definition observable contexts are defined in a co-inductive style

Definition 2 (Observable context). *Let Σ_{Obs} be an observational signature, let $X = (X_s)_{s \in S}$ be a family of countable infinite sets X_s of variables of sort s and let $Z = (\{z_s\})_{s \in S \setminus S_{\mathrm{Obs}}}$ be a disjoint family of singleton sets (one for each non observable sort). For all $s \in S \setminus S_{\mathrm{Obs}}$ and $s' \in S_{\mathrm{Obs}}$ the set $\mathcal{C}(\Sigma_{\mathrm{Obs}})_{s \to s'}$ of* observable Σ_{Obs}-contexts *with "application sort" s and "observable result sort" s' is inductively defined as follows:*

1. *For each direct observer (op, i) with $op : s_1, \ldots, s_i, \ldots, s_n \to s'$ and pairwise disjoint variables $x_1 : s_1, \ldots, x_n : s_n$,*
   $op(x_1, \ldots, x_{i-1}, z_{s_i}, x_{i+1}, \ldots, x_n) \in \mathcal{C}(\Sigma_{\mathrm{Obs}})_{s_i \to s'}$,
2. *For each observable context $c \in \mathcal{C}(\Sigma_{\mathrm{Obs}})_{s \to s'}$, for each indirect observer (op, i) with $op : s_1, \ldots, s_i, \ldots, s_n \to s$, and for each pairwise disjoint variables $x_1 : s_1, \ldots, x_n : s_n$ not occurring in c,*
   $c[op(x_1, \ldots, x_{i-1}, z_{s_i}, x_{i+1}, \ldots, x_n)/z_s] \in \mathcal{C}(\Sigma_{\mathrm{Obs}})_{s_i \to s'}$
   where $c[op(x_1, \ldots, x_{i-1}, z_{s_i}, x_{i+1}, \ldots, x_n)/z_s]$ denotes the term obtained from c by substituting the term $op(x_1, \ldots, x_{i-1}, z_{s_i}, x_{i+1}, \ldots, x_n)$ for z_s.

[2] Non-observable sorts are also called "state-sorts".

The syntactic notion of an observable context induces, for any Σ-algebra A a semantic relation, called observational equality, which expresses indistinguishability of states w.r.t. the given observable contexts.

Definition 3 (Σ_{Obs}-equality). *Let Σ_{Obs} be an observational signature. For any Σ-algebra $A \in \mathrm{Alg}(\Sigma)$, the* observational Σ_{Obs}-equality *on A is denoted by $\approx_{\Sigma_{\mathrm{Obs}},A}$ and defined by:*
For all $s \in S$, two elements $a, b \in A_s$ are observationally equal w.r.t. Σ_{Obs}, i.e., $a \approx_{\Sigma_{\mathrm{Obs}},A} b$, if and only if

1. *$a = b$, if $s \in S_{\mathrm{Obs}}$,*
2. *for all observable sorts $s' \in S_{\mathrm{Obs}}$, for all observable contexts $c \in \mathcal{C}(\Sigma_{\mathrm{Obs}})_{s \to s'}$, and for all valuations $\alpha, \beta : X \cup \{z_s\} \to A$ with $\alpha(x) = \beta(x)$ if $x \in X$, $\alpha(z_s) = a$ and $\beta(z_s) = b$, we have $I_\alpha(c) = I_\beta(c)$, if $s \in S \setminus S_{\mathrm{Obs}}$.*

Note that only the observer operations are used to build observable contexts and hence to define the observational equality. As a consequence we require that the non-observer operations should not contribute to distinguish states. This requirement is guaranteed by observational algebras defined as follows.

Definition 4 (Observational algebra). *Let Σ_{Obs} be an observational signature. An* observational Σ_{Obs}-algebra *is a Σ-algebra A such that $\approx_{\Sigma_{\mathrm{Obs}},A}$ is a Σ-congruence on A. The class of all observational Σ_{Obs}-algebras is denoted by* $\mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$. [3]

In the next step we define an observational satisfaction relation for observational algebras and first-order Σ-formulas. The underlying idea of the observational satisfaction relation is to interpret the equality symbol $=$ occurring in a first-order formula φ not by the set-theoretic equality but by the observational equality of elements.

Definition 5 (Observational satisfaction relation). *The* observational satisfaction relation *between observational Σ_{Obs}-algebras and first-order Σ-formulas is denoted by $\models_{\Sigma_{\mathrm{Obs}}}$ and defined as follows:*
Let $A \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$.

1. *For any two terms $t, r \in T_\Sigma(X)_s$ of the same sort s and for any valuation $\alpha : X \to A$, $A, \alpha \models_{\Sigma_{\mathrm{Obs}}} t = r$ holds if $I_\alpha(t) \approx_{\Sigma_{\mathrm{Obs}},A} I_\alpha(r)$.*
2. *For any arbitrary Σ-formula φ and for any valuation $\alpha : X \to A$, $A, \alpha \models_{\Sigma_{\mathrm{Obs}}} \varphi$ is defined by induction over the structure of the formula in the usual way.*
3. *For any arbitrary Σ-formula φ, $A \models_{\Sigma_{\mathrm{Obs}}} \varphi$ holds if for all valuations $\alpha : X \to A$, $A, \alpha \models_{\Sigma_{\mathrm{Obs}}} \varphi$ holds.*

Definition 6 (Basic observational specification). *A* basic observational specification $\mathrm{SP}_{\mathrm{Obs}} = \langle \Sigma_{\mathrm{Obs}}, \mathrm{Ax} \rangle$ *consists of an observational signature $\Sigma_{\mathrm{Obs}} = (\Sigma, S_{\mathrm{Obs}}, OP_{\mathrm{Obs}})$ and a set* Ax *of Σ-sentences, called the axioms of* $\mathrm{SP}_{\mathrm{Obs}}$. *The semantics of* $\mathrm{SP}_{\mathrm{Obs}}$ *is given by its signature* $\mathrm{Sig}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}})$ *and by its class of models* $\mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}})$ *which are defined by:*

$$\mathrm{Sig}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}) \overset{\mathrm{def}}{=} \Sigma_{\mathrm{Obs}}, \mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}) \overset{\mathrm{def}}{=} \{A \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}}) \mid A \models_{\Sigma_{\mathrm{Obs}}} \mathrm{Ax}\}$$

[3] Observational morphisms are defined as relations; see [7].

The definitions stated above provide the basic ingredients for defining the *observational logic institution* which is detailed in [7]. Thereby it is particularly important to use an appropriate morphism notion for observational signatures which guarantees encapsulation of observable properties (formally expressed by the satisfaction condition of institutions; cf. [5]). The basic idea for achieving this is to require that no "new"observations are introduced for "old" sorts when composing systems via signature morphisms. Thus the observational logic institution provides a suitable framework for instantiating the institution-independent specification-building operators introduced in [18], hence for defining structured observational specifications.

3 The Constructor-Based Logic Institution

Reachability concepts are used to describe the underlying data manipulated by a program. For this purpose, a standard approach is to declare a distinguished subset OP_{Cons} of the operation symbols OP (of a signature $\Sigma = (S, OP)$) as constructor symbols and to restrict the admissible models of a specification to those algebras which are reachable w.r.t. the given constructors. Syntactically we will follow this approach which leads to the notion of a constructor-based signature (see Definition 7 below). However, from the semantic point of view we do not adopt the above interpretation which we believe is too restrictive w.r.t. our working hypothesis (of the Introduction). Let us illustrate our viewpoint by a simple example.

Let NAT be a standard specification of the natural numbers with signature $\Sigma_{NAT} = (\{nat\}, \{zero : \ \to nat, \ succ : nat \to nat, \ add : nat \times nat \to nat\})$ and with standard axioms. We declare *zero* and *succ* as constructor symbols. Then a Σ_{NAT}-algebra A is reachable w.r.t. the given constructors if any element of A is denotable by a term $succ^i(zero)$ with $i \geq 0$. Obviously the set $\mathbb{N}$ of the natural numbers (equipped with the usual operations) is a reachable algebra. But note that the set $\mathbb{Z}$ of the integers (equipped with the usual interpretations of *zero*, *succ* and *add*) is not reachable w.r.t. the given constructors and therefore is not an admissible (standard) model of NAT. Nevertheless the integers can obviously be used as an implementation of the natural numbers which just happens to contain the negative integers as junk elements. Hence, in order to satisfy our working hypothesis, the integers should be admitted as a model of NAT. As a consequence we are interested in a more flexible framework where the constructor symbols are still essential, in the sense that they determine the data of interest, but nevertheless non-reachable algebras can be accepted as models if they satisfy certain conditions which are formalised by our notion of constructor-based algebra (see Definition 10 below).

In this way we obtain a novel treatment of reachability in algebraic specifications which finally leads to the institution of constructor-based logic. All steps performed in this section are quite analogous to the development of the observational logic institution. The correspondence will be analysed in Section 4 and formalised in Section 6.

Definition 7 (Constructor-based signature). *A constructor-based signature $\Sigma_{\text{Cons}} = (\Sigma, S_{\text{Cons}}, OP_{\text{Cons}})$ consists of a signature $\Sigma = (S, OP)$, a set $S_{\text{Cons}} \subseteq S$ of* constrained sorts *and a set $OP_{\text{Cons}} \subseteq OP$ of* constructor symbols *such that, for any $op \in OP_{\text{Cons}}$ with arity $op : s_1, \ldots, s_n \to s$, $s \in S_{\text{Cons}}$. We assume also that for each constrained sort $s \in S_{\text{Cons}}$, there exists at least one constructor in OP_{Cons} with range s.*

Any constructor-based signature determines a set of constructor terms.

Definition 8 (Constructor term). *Let Σ_{Cons} be a constructor-based signature, and let $X = (X_s)_{s\in S}$ be a family of countable infinite sets X_s of variables of sort s. A* constructor term *is a term $t \in T_{\Sigma'}(X')$, where $\Sigma' = (S, OP_{\text{Cons}})$, and $X' = (X'_s)_{s\in S}$ with $X'_s = X_s$ if $s \in S \setminus S_{\text{Cons}}$ and $X'_s = \emptyset$ if $s \in S_{\text{Cons}}$. The set of constructor terms is denoted by $\mathcal{T}(\Sigma_{\text{Cons}})$.*

The syntactic notion of a constructor term induces, for any Σ-algebra A, the definition of a family of subsets of the carrier sets of A, called reachable part, which consists of those data which are relevant according to the given constructors.

Definition 9 (Reachable part). *Let Σ_{Cons} be a constructor-based signature. For any Σ-algebra $A \in \text{Alg}(\Sigma)$, the reachable part $\langle A\rangle_{\text{Cons}} = (\langle A\rangle_{\text{Cons},s})_{s\in S}$ of A is defined by:*
For each $s \in S$, $\langle A\rangle_{\text{Cons},s} = \{a \in A_s \mid$ there exists a term $t \in \mathcal{T}(\Sigma_{\text{Cons}})_s$ and a valuation $\alpha : X' \to A$ such that $I_\alpha(t) = a\}$. [4]

Note that only the constructor symbols are used to build constructor terms and hence to define the reachable part. Since the reachable part represents the data of interest we require that no further elements should be constructible by the non-constructor operations:

Definition 10 (Constructor-based algebra). *Let Σ_{Cons} be a constructor-based signature. A* constructor-based Σ_{Cons}-algebra *is a Σ-algebra A such that $\langle A\rangle_{\text{Cons}}$, equipped with the canonical restrictions of the operations op^A of A to the carrier sets of $\langle A\rangle_{\text{Cons}}$, is a Σ-subalgebra of A. The class of all constructor-based Σ_{Cons}-algebras is denoted by $\text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$.*

Definition 11 (Constructor-based morphism). *Let $A, B \in \text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$ be two constructor-based Σ_{Cons}-algebras. A* constructor-based Σ_{Cons}-morphism *$h : A \to B$ is a Σ-morphism between $\langle A\rangle_{\text{Cons}}$ and $\langle B\rangle_{\text{Cons}}$.*

For any constructor-based signature Σ_{Cons}, the class $\text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$ together with the constructor-based Σ_{Cons}-morphisms is a category.

[4] Note that for any non-constrained sort s, $\langle A\rangle_{\text{Cons},s} = A_s$.

The underlying idea of the constructor-based satisfaction relation is to restrict the valuations of variables to the generated values (i.e. to the elements of the reachable part) only.[5] Hence the following definition is quite similar to the definition of the standard satisfaction relation. The only difference concerns valuations: "$\alpha : X \to A$" is replaced by "$\alpha : X \to \langle A \rangle_{\mathrm{Cons}}$".

Definition 12 (Constructor-based satisfaction relation). *The* constructor-based satisfaction relation *between constructor-based Σ_{Cons}-algebras and first-order Σ-formulas is denoted by $\models_{\Sigma_{\mathrm{Cons}}}$ and defined as follows:*
Let $A \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$.

1. *For any two terms $t, r \in T_\Sigma(X)_s$ of the same sort s and for any valuation $\alpha : X \to \langle A \rangle_{\mathrm{Cons}}$, $A, \alpha \models_{\Sigma_{\mathrm{Cons}}} t = r$ holds if $I_\alpha(t) = I_\alpha(r)$.*
2. *For any arbitrary Σ-formula φ and for any valuation $\alpha : X \to \langle A \rangle_{\mathrm{Cons}}$, $A, \alpha \models_{\Sigma_{\mathrm{Cons}}} \varphi$ is defined by induction over the structure of the formula in the usual way.*
3. *For any arbitrary Σ-formula φ, $A \models_{\Sigma_{\mathrm{Cons}}} \varphi$ holds if for all valuations $\alpha : X \to \langle A \rangle_{\mathrm{Cons}}$, $A, \alpha \models_{\Sigma_{\mathrm{Cons}}} \varphi$ holds.*

As an example consider again the specification *NAT* and the integers which satisfy w.r.t. the constructor-based satisfaction relation the following Peano axiom

$$\mathbb{Z} \models_{\Sigma_{\mathrm{Cons}}} \forall x : nat.\ succ(x) \neq zero$$

Indeed this is true since the reachable part of $\mathbb{Z}$ w.r.t. the constructors *zero* and *succ* is just $\mathbb{N}$ and hence the universally quantified variable x is only interpreted in $\mathbb{N}$. Thus the integers will be an admissible model of *NAT* considered as a constructor-based specification w.r.t. the constructors *zero* and *succ*.

Definition 13 (Basic constructor-based specification). *A* basic constructor-based specification $\mathrm{SP}_{\mathrm{Cons}} = \langle \Sigma_{\mathrm{Cons}}, \mathrm{Ax} \rangle$ *consists of a constructor-based signature $\Sigma_{\mathrm{Cons}} = (\Sigma, S_{\mathrm{Cons}}, OP_{\mathrm{Cons}})$ and a set* Ax *of Σ-sentences, called the axioms of* $\mathrm{SP}_{\mathrm{Cons}}$*. The semantics of* $\mathrm{SP}_{\mathrm{Cons}}$ *is given by its signature* $\mathrm{Sig}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}})$ *and by its class of models* $\mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}})$ *which are defined by:*

$$\mathrm{Sig}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}}) \stackrel{\mathrm{def}}{=} \Sigma_{\mathrm{Cons}}$$

$$\mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}}) \stackrel{\mathrm{def}}{=} \{A \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}}) \mid A \models_{\Sigma_{\mathrm{Cons}}} \mathrm{Ax}\}$$

To obtain the constructor-based logic institution we still need an appropriate morphism notion for constructor-based signatures which guarantees encapsulation of properties w.r.t. the constructor-based satisfaction relation. The basic idea to achieve this is to require that no "new" constructors are introduced for "old" sorts when composing systems via signature morphisms which is formally captured by the following definition.

[5] This idea is related to the ultra-loose approach of [19] where the same effect is achieved by using formulas with relativized quantification.

Definition 14 (Constructor-based signature morphism). *Let* $\Sigma_{\text{Cons}} = (\Sigma, S_{\text{Cons}}, OP_{\text{Cons}})$ *and* $\Sigma'_{\text{Cons}} = (\Sigma', S'_{\text{Cons}}, OP'_{\text{Cons}})$ *be two constructor-based signatures with* $\Sigma = (S, OP)$ *and* $\Sigma' = (S', OP')$. *A* constructor-based signature morphism $\sigma_{\text{Cons}} : \Sigma_{\text{Cons}} \to \Sigma'_{\text{Cons}}$ *is a signature morphism* $\sigma : \Sigma \to \Sigma'$ *such that:*

1. *For all* $s \in S$, $s \in S_{\text{Cons}}$ *if and only if* $\sigma(s) \in S'_{\text{Cons}}$.
2. *If* $op \in OP_{\text{Cons}}$, *then* $\sigma(op) \in OP'_{\text{Cons}}$.
3. *If* $op' \in OP'_{\text{Cons}}$ *with* $op' : s'_1, \ldots, s'_n \to s'$ *and* $s' \in \sigma(S)$ *then there exists* $op \in OP$ *such that* $op \in OP_{\text{Cons}}$ *and* $op' = \sigma(op)$.

Constructor-based signatures together with constructor-based signature morphisms form a category which has pushouts. Moreover, for any constructor-based signature morphism $\sigma_{\text{Cons}} : \Sigma_{\text{Cons}} \to \Sigma'_{\text{Cons}}$, one can associate a constructor-based reduct functor $\_|_{\sigma_{\text{Cons}}} : \text{Alg}_{\text{Cons}}(\Sigma'_{\text{Cons}}) \to \text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$ in a straightforward way. One can also show that the constructor-based satisfaction condition holds, i.e., for any constructor-based signature morphism $\sigma_{\text{Cons}} : \Sigma_{\text{Cons}} \to \Sigma'_{\text{Cons}}$, constructor-based Σ'_{Cons}-algebra $A' \in \text{Alg}_{\text{Cons}}(\Sigma'_{\text{Cons}})$ and Σ-sentence φ:
$A' \models_{\Sigma'_{\text{Cons}}} \sigma(\varphi)$ if and only if $A'|_{\sigma_{\text{Cons}}} \models_{\Sigma_{\text{Cons}}} \varphi$.

This means that the definitions stated above provide the necessary ingredients for defining an institution (cf. [5]) which is called the *constructor-based logic institution.* As in the observational case this institution provides a suitable framework for instantiating the institution-independent specification-building operators introduced in [18], hence for defining structured constructor-based specifications.

4 A First Comparison

The observational logic institution and the constructor-based logic institution were developed step by step in a totally analogous way. Indeed there is a close correspondence between all notions of the observability and reachability concepts which is summarized in Table 1.

First, there is an obvious syntactic correspondence between an observational signature and a constructor-based signature which, on the one hand, leads to the notion of an observable context and, on the other hand, leads to the definition of a constructor term.

In any case the syntactic notions induce a semantic relation on any Σ-algebra A. In the observational case we obtain a binary relation $\approx_{\Sigma_{\text{Obs}},A}$, called observational equality, and in the constructor case we obtain a unary relation $\langle A\rangle_{\text{Cons}}$, called reachable part. Then we require that the operations of an algebra are compatible with the given relations. This means, in the observational case, that the observational equality is a Σ-congruence thus leading to the notion of an observational algebra. In the constructor case this means that the reachable part is a Σ-subalgebra thus leading to the notion of a constructor-based algebra.

In order to satisfy our working hypothesis we have relaxed the standard satisfaction relation such that, in the observational case, equality is considered

Table 1. Comparing Observability and Reachability

Observability	**Reachability**
observational signature $\Sigma_{\text{Obs}} = (\Sigma, S_{\text{Obs}}, OP_{\text{Obs}})$	*constructor-based signature* $\Sigma_{\text{Cons}} = (\Sigma, S_{\text{Cons}}, OP_{\text{Cons}})$
observable context	*constructor term*
observational equality $\approx_{\Sigma_{\text{Obs}},A} \subseteq A \times A$	*reachable part* $\langle A \rangle_{\text{Cons}} \subseteq A$
observational algebra $\approx_{\Sigma_{\text{Obs}},A}$ is a Σ-congruence	*constructor-based algebra* $\langle A \rangle_{\text{Cons}}$ is a Σ-subalgebra of A
observational satisfaction $A \models_{\Sigma_{\text{Obs}}} \phi$ interpret "=" by "$\approx_{\Sigma_{\text{Obs}},A}$"	*constructor-based satisfaction* $A \models_{\Sigma_{\text{Cons}}} \phi$ use valuations $\alpha : X \to \langle A \rangle_{\text{Cons}}$
observational specification $\text{SP}_{\text{Obs}} = \langle \Sigma_{\text{Obs}}, \text{Ax} \rangle$ $\text{Mod}_{\text{Obs}}(\text{SP}_{\text{Obs}}) \overset{\text{def}}{=}$ $\{A \in \text{Alg}_{\text{Obs}}(\Sigma_{\text{Obs}}) \mid A \models_{\Sigma_{\text{Obs}}} \text{Ax}\}$	*constructor-based specification* $\text{SP}_{\text{Cons}} = \langle \Sigma_{\text{Cons}}, \text{Ax} \rangle$ $\text{Mod}_{\text{Cons}}(\text{SP}_{\text{Cons}}) \overset{\text{def}}{=}$ $\{A \in \text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}}) \mid A \models_{\Sigma_{\text{Cons}}} \text{Ax}\}$
observational logic institution	*constructor-based logic institution*

as observational equality and, in the constructor case, variables are interpreted only by values of the reachable part. Then it is straightforward to introduce the notions of observational and constructor-based specifications whose semantics are defined according to the generalized satisfaction relations. Finally we have pointed out that both frameworks lead to an institution by using appropriate notions of signature morphisms.

It is still important to stress that there are also corresponding specification methods when writing observational and constructor-based specifications. In the observational case the idea is to specify the effect of each non-observer operation (in a co-inductive style) by a (complete) case distinction w.r.t. the given observers. A general schema for observer complete definitions is studied in [3]. As a standard example consider an observational specification of an alternating merge function `merge: stream x stream` → `stream` on streams with observers `head: stream` → `elem` and `tail: stream` → `stream`. Then the `merge` function is specified by the following complete case distinction w.r.t. the observers `head` and `tail`:

```
head(merge(s1, s2)) = head(s1)
tail(merge(s1, s2)) = merge(s2, tail(s1))
```

Analogously it is well-known that in the constructor case it is a standard technique to specify the non-constructor operations in an inductive-style by a (complete) case distinction w.r.t. the given constructors. In the categorical framework of algebras and co-algebras this analogy is described in [11].

5 Logical Consequences of Specifications: The Black Box View

So far we have emphasized the fact that the model class semantics of a specification should reflect all its correct realizations. According to our working hypothesis, a program P is a correct realization of SP_X if it determines a $\mathrm{Sig}_X(\mathrm{SP}_X)$-algebra which belongs to $\mathrm{Mod}_X(\mathrm{SP}_X)$.[6] In the following we will refer to $\mathrm{Mod}_X(\mathrm{SP}_X)$ as the *glass box semantics* of a specification since it reveals its correct realizations. Glass box semantics is appropriate from an implementor's point of view.

Of equal importance are the logical consequences of a given specification. In this section we focus on the properties φ that can be inferred from a given specification SP_X. This means that we are interested in statements $\mathrm{SP}_X \models_X \varphi$ which express that $\mathrm{Mod}_X(\mathrm{SP}_X) \models_X \varphi$ holds.

For this purpose it is convenient to *abstract* the models of a specification into "idealized" models, such that the consequences of the actual models of the specification of interest, in the chosen logic, are exactly the consequences of the idealized models, in *standard* first-order logic. Hence to any specification SP_X we will associate the class of its "idealized" models (which lie in the standard algebraic institution), and this class will be called the *black box semantics* of the specification. Black box semantics is appropriate from a client's point of view.

5.1 Black Box Semantics of Observational Specifications

Let Σ_{Obs} be an observational signature. Since for any Σ_{Obs}-algebra A, the observational equality $\approx_{\Sigma_{\mathrm{Obs}},A}$ is a Σ-congruence, we can construct its quotient $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ which identifies all elements of A which are indistinguishable "from the outside". $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ can be considered as the "black box view" of A and represents the "observable behaviour" of A w.r.t. Σ_{Obs}. $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ is *fully abstract* in the sense that the observational equality (w.r.t. Σ_{Obs}) on $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ coincides with the set-theoretic equality. By considering $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ just as a Σ-algebra we obtain (for any observational signature Σ_{Obs}) a functor from the category $\mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$ of observational algebras into the category $\mathrm{Alg}(\Sigma)$ of (standard) Σ-algebras which establishes a one to one correspondence between observational morphisms $h : A \to B$ and standard morphisms $k : A/{\approx_{\Sigma_{\mathrm{Obs}},A}} \to B/{\approx_{\Sigma_{\mathrm{Obs}},B}}$, i.e., this functor is full and faithful.

[6] We use the subscript $_X$ to denote the fact that we work either in the observational logic institution or in the constructor-based logic institution.

Theorem 1 (Behavior functor). *For any observational signature Σ_{Obs} with underlying standard signature Σ, the following defines a full and faithful functor $\mathcal{FA}_{\Sigma_{\mathrm{Obs}}} : \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}}) \to \mathrm{Alg}(\Sigma)$:*

1. *For each $A \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$, $\mathcal{FA}_{\Sigma_{\mathrm{Obs}}}(A) \overset{\mathrm{def}}{=} A/\approx_{\Sigma_{\mathrm{Obs}},A}$ and is called the* observational behavior *of A.*
2. *For each observational morphism $h : A \to B$, $\mathcal{FA}_{\Sigma_{\mathrm{Obs}}}(h) : A/\approx_{\Sigma_{\mathrm{Obs}},A} \to B/\approx_{\Sigma_{\mathrm{Obs}},B}$ is defined by $\mathcal{FA}_{\Sigma_{\mathrm{Obs}}}(h)([a]) = [b]$ if $a\ h\ b$.*

Definition 15 (Black box semantics). *Let* $\mathrm{SP}_{\mathrm{Obs}}$ *be an observational specification with signature* $\mathrm{Sig}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}) = \Sigma_{\mathrm{Obs}}$. *Its black box semantics is defined by* $[\![\mathrm{SP}_{\mathrm{Obs}}]\!] \overset{\mathrm{def}}{=} \mathcal{FA}_{\Sigma_{\mathrm{Obs}}}(\mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}))$.

It may be interesting to note that the black box semantics of an observational specification is exactly the class of its fully abstract models (viewed as ordinary algebras).

Fact 2 (Black box semantics relies on fully abstract models)
Let $\mathrm{SP}_{\mathrm{Obs}} = \langle \Sigma_{\mathrm{Obs}}, \mathrm{Ax} \rangle$ *be a basic observational specification. Then*
$[\![\mathrm{SP}_{\mathrm{Obs}}]\!] = \{\Sigma\text{−algebra } A \mid A \models \mathrm{Ax} \text{ and } A \text{ is fully abstract w.r.t. } \approx_{\Sigma_{\mathrm{Obs}},A}\}$.

Theorem 3 (Observational consequences). *Let Σ_{Obs} be an observational signature with underlying standard signature Σ, let φ be a Σ-formula, let A be a Σ_{Obs}-algebra, and let* $\mathrm{SP}_{\mathrm{Obs}}$ *be an observational specification with signature Σ_{Obs}.*

1. *$A \models_{\Sigma_{\mathrm{Obs}}} \varphi$ if and only if $\mathcal{FA}_{\Sigma_{\mathrm{Obs}}}(A) \models \varphi$.*
2. $\mathrm{SP}_{\mathrm{Obs}} \models_{\Sigma_{\mathrm{Obs}}} \varphi$ *if and only if* $[\![\mathrm{SP}_{\mathrm{Obs}}]\!] \models \varphi$.

This theorem shows the adequacy of the black box semantics in the observational case. The theorem is a variant of Theorem 3.11 in [4] and it is related to similar results in [10].

5.2 Black Box Semantics of Constructor-Based Specifications

Let Σ_{Cons} be a constructor-based signature. Since for any Σ_{Cons}-algebra A, the reachable part $\langle A \rangle_{\mathrm{Cons}}$ of A is a Σ-algebra, which by definition contains only those elements that are generated by the given constructors (for the constrained sorts), we can consider the reachable part $\langle A \rangle_{\mathrm{Cons}}$ of A as its black box view (abstracting away from all junk values that may lie in A). This restriction to the reachable sub-algebra provides (for any constructor-based signature Σ_{Cons}) a functor from the category $\mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$ of constructor-based algebras into the category $\mathrm{Alg}(\Sigma)$ of (standard) Σ-algebras which is full and faithful.

Theorem 4 (Restrict functor). *For any constructor-based signature Σ_{Cons} with underlying standard signature Σ, the following defines a full and faithful functor $\mathcal{R}_{\Sigma_{\mathrm{Cons}}} : \mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}}) \to \mathrm{Alg}(\Sigma)$:*

1. *For each* $A \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$, $\mathcal{R}_{\Sigma_{\mathrm{Cons}}}(A) \overset{\mathrm{def}}{=} \langle A \rangle_{\mathrm{Cons}}$ *and is called the* reachable part *of* A*.*
2. *For each constructor-based morphism* $h : A \to B$*,* $\mathcal{R}_{\Sigma_{\mathrm{Cons}}}(h) : \langle A \rangle_{\mathrm{Cons}} \to \langle B \rangle_{\mathrm{Cons}}$ *is defined by* $\mathcal{R}_{\Sigma_{\mathrm{Cons}}}(h)(a) = h(a)$*.*

Definition 16 (Black box semantics). *Let* $\mathrm{SP}_{\mathrm{Cons}}$ *be a constructor-based specification with signature* $\mathrm{Sig}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}}) = \Sigma_{\mathrm{Cons}}$*. Its black box semantics is defined by* $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] \overset{\mathrm{def}}{=} \mathcal{R}_{\Sigma_{\mathrm{Cons}}}(\mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}}))$*.*

Again, it may be interesting to note that the black box semantics of a constructor-based specification is exactly the class of its reachable models. Thereby an algebra is called reachable w.r.t. a set OP_{Cons} of constructor symbols if all elements of A are denotable by a constructor term.

Fact 5 (Black box semantics relies on reachable models)
Let $\mathrm{SP}_{\mathrm{Cons}} = \langle \Sigma_{\mathrm{Cons}}, \mathrm{Ax} \rangle$ *be a basic constructor-based specification. Then* $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] = \{\Sigma\text{−algebra } A \mid A \models \mathrm{Ax} \text{ and } A \text{ is reachable w.r.t. } OP_{\mathrm{Cons}}\}$*.*

Theorem 6 (Inductive consequences). *Let* Σ_{Cons} *be a constructor-based signature with underlying standard signature* Σ*, let* φ *be a* Σ*-formula, let* A *be a* Σ_{Cons}*-algebra, and let* $\mathrm{SP}_{\mathrm{Cons}}$ *be a constructor-based specification with signature* Σ_{Cons}*.*

1. $A \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *if and only if* $\mathcal{R}_{\Sigma_{\mathrm{Cons}}}(A) \models \varphi$*.*
2. $\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *if and only if* $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] \models \varphi$*.*

This theorem shows the adequacy of the black box semantics in the constructor-based case.

6 Formalising the Duality

In this section we establish a formal duality of the observability and reachability concepts considered in the previous sections. For this purpose we first need a precise notion of duality. This is provided by using category theory.

6.1 Categorical Duality

We briefly review categorical duality, for more details see e.g. [1]. A category $\mathcal{C}$ consists of a class of objects, also denoted by $\mathcal{C}$, and for all $A, B \in \mathcal{C}$ of a set of arrows (or morphisms) $\mathcal{C}(A, B)$. The *dual* (or opposite) category $\mathcal{C}^{\mathrm{op}}$ has the same objects and arrows $\mathcal{C}^{\mathrm{op}}(A, B) = \mathcal{C}(B, A)$. We write A^{op} and f^{op} for $A \in \mathcal{C}$ and $f \in \mathcal{C}(B, A)$ to indicate when we think of A as an object in $\mathcal{C}^{\mathrm{op}}$ and of f as an arrow in $\mathcal{C}^{\mathrm{op}}(A, B)$. Duality can now be formalised as follows: Let P be a property of objects or arrows in $\mathcal{C}$. We then say that

an object A (arrow f, respectively) in $\mathcal{C}$ has property co-P
iff A^{op} (f^{op}, respectively) has property P.

For example, an object A is co-initial in $\mathcal{C}$ (usually called terminal or final) iff A is initial in $\mathcal{C}^{\mathrm{op}}$; $C = A + B$ is a co-product iff C^{op} is the product $A^{\mathrm{op}} \times B^{\mathrm{op}}$.

The duality principle can also be extended to functors. The dual of a functor $F : \mathcal{C} \to \mathcal{D}$ is the functor $F^{\mathrm{op}} : \mathcal{C}^{\mathrm{op}} \to \mathcal{D}^{\mathrm{op}}$ which acts on objects and morphisms as F does. For instance, for an endofunctor F, the category of F-coalgebras is (isomorphic to) the dual of the category of F^{op}-algebras (cf. [11] for a study of algebras and co-algebras).

6.2 The Duality Principle for Observability and Reachability

We first give a categorical account of the signatures and models in the observational and in the constructor-based approach. The formal duality principle will then be an immediate consequence.

Motivated by the approach in [8], we represent a signature (over a base category $\mathcal{X}$) by two functors $\Omega, \Xi : \mathcal{X} \to \mathcal{X}$. A model is an algebra-coalgebra pair $\Omega X \to X \to \Xi X$ (cf. also [14]). We call Ω the *algebraic signature* and Ξ the *coalgebraic signature.*

Next, we introduce signatures for constructors and observers, given by functors $\mathcal{R}, \mathcal{O} : \mathcal{X} \to \mathcal{X}$, respectively. That is, for formalizing the observational signatures, we consider signatures $(\Omega, \Xi) = (\Omega, \Xi' \times \mathcal{O})$ and models

$$\begin{tikzcd} \Omega X \arrow[r, "\omega"] & X \arrow[r, "\langle \xi', o \rangle"] & (\Xi' \times \mathcal{O})X \end{tikzcd}$$

where ω, ξ' are operations and o are observers.

On the other hand, for formalizing constructor-based signatures, we consider signatures $(\Omega, \Xi) = (\Omega' + \mathcal{R}, \Xi)$ and models

$$\begin{tikzcd} (\Omega' + \mathcal{R})X \arrow[r, "{[\omega', \rho]}"] & X \arrow[r, "\xi"] & \Xi X \end{tikzcd}$$

where ω', ξ are operations and ρ are constructors.

Definition 17 (Observational models). *Let $\mathcal{X}$ be a category with binary products. An* ***observational signature*** *$(\Omega; \Xi', \mathcal{O})$ over $\mathcal{X}$ consists of functors $\Omega, \Xi', \mathcal{O} : \mathcal{X} \to \mathcal{X}$ such that a final $\mathcal{O}$-coalgebra $Z \xrightarrow{\zeta} \mathcal{O}Z$ exists. $\mathcal{O}$ is called the* ***observability constraint****. An algebra-coalgebra pair $(\omega, \langle \xi', o \rangle)$ for $(\Omega, \Xi' \times \mathcal{O})$ satisfies the observability constraint $\mathcal{O}$ and is called a* ***model*** *for $(\Omega; \Xi', \mathcal{O})$ iff there are dotted arrows such that the following diagram commutes*

$$\begin{tikzcd} \Omega X \arrow[r, "\omega"] \arrow[d, "\Omega !"'] & X \arrow[r, "\xi'"] \arrow[d, "!"'] & \Xi' X \arrow[d, "\Xi' !"] \\ \Omega Z \arrow[r, dotted] & Z \arrow[r, dotted] & \Xi' Z \end{tikzcd}$$

where $! : X \to Z$ is the unique coalgebra morphism from the observers $X \xrightarrow{o} \mathcal{O}X$ to the final $\mathcal{O}$-coalgebra. A ***morphism*** *between models is an arrow which is simultaneously a morphism for the algebra and the coalgebra part. The resulting* ***category of models*** *is denoted by $\mathsf{Mod}(\Omega; \Xi', \mathcal{O})$. A model is* ***fully abstract*** *iff $!$ is an embedding (injective).*

Remark 1. The diagram expresses in an abstract way that the model satisfies the condition for observational algebras of Definition 4. Indeed, for any observational signature Σ_{Obs} (in the sense of Section 2) whose observers have only one non-observable argument sort, appropriate polynomial functors $\Omega, \Xi', \mathcal{O}$ over the base category Set^n can be found such that the observational equivalence induced by $\mathcal{O}$ is a congruence w.r.t. Ω and Ξ' (and anyway w.r.t. $\mathcal{O}$). This is due to the fact that ! identifies precisely all observationally equivalent points.

Definition 18 (Constructor-based models). *Let $\mathcal{X}$ be a category with binary coproducts. A* ***constructor-based signature*** *$(\Omega', \mathcal{R}; \Xi)$ over $\mathcal{X}$ consists of functors $\Omega', \mathcal{R}, \Xi : \mathcal{X} \to \mathcal{X}$ such that an initial $\mathcal{R}$-algebra $\mathcal{R}I \xrightarrow{\iota} I$ exists. $\mathcal{R}$ is called the* ***reachability constraint****. An algebra-coalgebra pair $([\omega', \rho], \xi)$ satisfies the reachability constraint $\mathcal{R}$ and is called a* ***model*** *for $(\Omega', \mathcal{R}; \Xi)$ iff there are dotted arrows such that the following diagram commutes*

$$\begin{tikzcd}
\Omega' X \arrow[r, "\omega'"] & X \arrow[r, "\xi"] & \Xi X \\
\Omega' I \arrow[u, "\Omega' ?"] \arrow[r, dotted] & I \arrow[u, "?"] \arrow[r, dotted] & \Xi I \arrow[u, "\Xi ?"']
\end{tikzcd}$$

where $? : I \to X$ is the unique algebra morphism from the initial $\mathcal{R}$-algebra to the constructor-algebra $\mathcal{R}X \xrightarrow{\rho} X$. A ***morphism*** *between models is an arrow which is simultaneously a morphism for the algebra and the coalgebra part. The resulting* ***category of models*** *is denoted by $\mathsf{Mod}(\Omega', \mathcal{R}; \Xi)$. A model is* ***reachable*** *iff ? is a quotient (surjective).*

Remark 2. The diagram expresses in an abstract way that the model satisfies the condition for constructor-based algebras of Definition 10: Indeed, for any constructor-based signature Σ_{Cons} (in the sense of Section 3) appropriate polynomial functors $\Omega', \mathcal{R}, \Xi$ over the base category Set^n can be found such that the image of I under ? is the reachable part and whenever one of the operations ω', ξ takes all its arguments from the image of I under ? the results are in the image again (and this is anyway true for ρ).

Definitions 17 and 18 give rise to a **duality principle** for constructor-based and observational models which is stated formally by the following isomorphisms of categories:

$$\mathsf{Mod}(\Omega; \Xi', \mathcal{O})^{\mathrm{op}} \simeq \mathsf{Mod}(\Xi'^{\mathrm{op}}, \mathcal{O}^{\mathrm{op}}; \Omega^{\mathrm{op}}),$$

$$\mathsf{Mod}(\Omega', \mathcal{R}; \Xi)^{\mathrm{op}} \simeq \mathsf{Mod}(\Xi^{\mathrm{op}}; \Omega'^{\mathrm{op}}, \mathcal{R}^{\mathrm{op}}).$$

The isomorphisms map models $(\alpha, \beta)^{\mathrm{op}} = (\alpha, \beta)$ to $(\beta^{\mathrm{op}}, \alpha^{\mathrm{op}})$. In the following, we identify $(\alpha, \beta)^{\mathrm{op}}$ with $(\beta^{\mathrm{op}}, \alpha^{\mathrm{op}})$.

As a consequence of the duality principle we obtain, for example:

Theorem 7.

1. *An algebra-coalgebra pair M for $(\Omega, \Xi' \times \mathcal{O})$ satisfies the observability constraint $\mathcal{O}$ iff M^{op} satisfies the reachability constraint $\mathcal{O}^{op}$.*
2. *An algebra-coalgebra pair M for $(\Omega' + \mathcal{R}, \Xi)$ satisfies the reachability constraint $\mathcal{R}$ iff M^{op} satisfies the observability constraint $\mathcal{R}^{op}$.*
3. *A model M is reachable iff M^{op} is fully abstract.*

The first theorem similar to (3) is due to Kalman [12] and was proved for linear systems in control theory. Later, Arbib and Manes [2] brought to light the general principles underlying this duality by considering—essentially—systems (automata) as Ω-algebras for arbitrary functors Ω. Compared to [2] the main point of our formalization consists in the use of coalgebras to formalise our notion of observation and in the consideration of observability and reachability constraints.

7 Conclusion

In this paper we have studied and formalised the duality between observability and reachability concepts used in algebraic approaches to software development taking into account observability and reachability constraints. We hope that the exhibition of this duality contributes to a clarification of specification methodologies and of their semantic foundations.

As a particular outcome we have presented the novel institution of constructor-based logic. The formal dualisation of the categorical representation of observational logic in [8] gave us the intuition for the adequate notions of constructor logic which provide sufficient flexibility to describe the semantically correct realizations of a specification from the reachability point of view (in the same way as observational logic does from the observational point of view). In this paper we have focused on a comparison of the two concepts and *not* on their integration. The combination of the two concepts offers a promising perspective of future reserach. We believe that such an integration will be strongly related to (a generalization of) the notion of partial observational equivalence considered e.g. in [4] and [10].

References

1. J. Adámek, H. Herrlich, and G. Strecker. *Abstract and Concrete Categories.* John Wiley & Sons, 1990.
2. M.A. Arbib and E.G. Manes. Adjoint machines, state-behaviour machines, and duality. *Journ. of Pure and Applied Algebra*, 6:313–344, 1975.
3. M. Bidoit and R. Hennicker. Observer complete definitions are behaviourally coherent. In *OBJ/CafeOBJ/Maude at Formal Methods '99*, pages 83–94. THETA, 1999.
4. M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. *Science of Computer Programming*, 25:149–186, 1995.

5. J. Goguen and R. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39 (1):95–146, 1992.
6. 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.
7. R. Hennicker and M. Bidoit. Observational logic. In Armando Haeberer, editor, *Algebraic Methodology and Software Technology (AMAST'98)*, volume 1548 of *LNCS*. Springer, 1999.
8. R. Hennicker and A. Kurz. (Ω, Ξ)-logic: On the algebraic extension of coalgebraic specifications. In B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science (CMCS'99)*, volume 19 of *Electronic Notes in Theoretical Computer Science*, pages 195–211, 1999.
9. C.A.R. Hoare. Proofs of correctness of data representations. *Acta Informatica*, 1:271–281, 1972.
10. M. Hofmann and D.T. Sannella. On behavioural abstraction and behavioural satisfaction in higher-order logic. In *TAPSOFT '95*, volume 915 of *LNCS*, pages 247–261. Springer, 1995.
11. B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62, 1997.
12. R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in Mathematical System Theory.* McGraw-Hill, 1969.
13. J. Loeckx, H.-D. Ehrich, and M. Wolf. *Specification of Abstract Data Types.* Wiley and Teubner, 1996.
14. Grant Malcolm. Behavioural equivalence, bisimulation, and minimal realisation. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specification*, volume 1130 of *LNCS*, pages 359–378. Springer, 1996.
15. CoFI Task Group on Language Design. Casl - the cofi algebraic specification language - summary. `http://www.brics.dk/Projects/CoFI/Documents/CASL/Summary/`.
16. Horst Reichel. *Initial computability, algebraic specifications, and partial algebras.* Oxford, Clarendon Press, 1987.
17. D.T. Sannella and A. Tarlecki. On observational equivalence and algebraic specification. *Journal Comput. System Sci.*, 34:150–178, 1987.
18. D.T. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76:165–210, 1988.
19. M. Wirsing and M. Broy. A modular framework for specification and information. In F. Orejas J. Diaz, editor, *TAPSOFT '89*, volume 351 of *LNCS*, pages 42–73. Springer, 1989.