Observational Logic, Constructor-Based Logic, and their Duality [⋆]

Michel Bidoit [a], Rolf Hennicker [b], and Alexander Kurz [c]

[a] *Laboratoire Spécification et Vérification (LSV), CNRS & ENS de Cachan, France*

[b] *Institut für Informatik, Ludwig-Maximilians-Universität München, Germany*

[c] *Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands*

Abstract

Observability and reachability are important concepts for formal software development. While observability concepts are used to specify the required observable behavior of a program or system, reachability concepts are used to describe the underlying data in terms of datatype constructors. In this paper we first reconsider the observational logic institution which provides a logical framework for dealing with observability. Then we develop in a completely analogous way the constructor-based logic institution which formalizes a novel treatment of reachability. Both institutions are tailored to capture the semantically correct realizations of a specification from either the observational or the reachability point of view. We show that there is a methodological and even formal duality between both frameworks. In particular, we establish a correspondence between observer operations and datatype constructors, observational and constructor-based algebras, fully abstract and reachable algebras, and observational and inductive consequences of specifications. The formal duality between the observability and reachability concepts is established in a category-theoretic setting.

Key words: Algebraic specification, observability, reachability, duality, institution

[⋆] This work was partially supported by the ESPRIT Working Group 29432 CoFI, by the Bayer. Forschungsstiftung, and by the German DFG-project InopSys. Preliminary results of this study have been published in [9], and preliminary results about the observational logic institution have been published in [18].

Email addresses: `bidoit@lsv.ens-cachan.fr` (Michel Bidoit),
`hennicke@informatik.uni-muenchen.de` (Rolf Hennicker),
`Alexander.Kurz@cwi.nl` (Alexander Kurz).

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 are used to describe the underlying data manipulated by the system. Since observability and reachability are used for different purposes, both concepts may seem unrelated. In this study we show that there is a methodological and even formal duality between the two concepts. We believe that investigating this duality 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 [22]):

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. Using these assumptions we will study algebraic frameworks for observability and for reachability (which both form an institution), we will analyze the analogy between the two institutions and, finally, we will develop a categorical representation of our observability and reachability notions (in terms of algebras and coalgebras defined w.r.t. appropriate functors), which leads to a formal duality principle between the two concepts.

1.1 Observability

Observability concepts provide means to specify the observable behavior of software systems in an abstract, implementation independent way. They take into account our working hypothesis from above in the sense that any program which satisfies the observable behavior prescribed by a specification SP is considered as a correct realization of SP.

One can distinguish two main approaches to observability. [2] The first one is based on an observational equivalence relation between algebras which is used

[1] In the context of automata theory, a similar duality was already investigated by Arbib and Manes in [3].
[2] The relationships between the two approaches have been intensively studied in [10].

to abstract from the (standard) model class of a specification, see, e.g., [36]. The second approach relaxes the (standard) satisfaction relation so that the observational models of a specification are all algebras which satisfy the given set of axioms up to observational equality of the elements of the algebra. (This idea was originally introduced by Reichel, see, e.g., [34].) Thereby two elements are considered to be observationally equal if they cannot be distinguished by a set of observable experiments.

In this work we will follow the second approach. A flexible framework to formalize observable experiments is suggested (in a similar way) e.g. in [18], [16] and [32] 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, for instance, to manipulate (non-visible) states of a system. In this study we will use the observational logic institution (introduced in [18]), where the non-observer operations are required to respect the observational equality (induced by the observer operations) which is formally captured by our notion of an observational algebra. The observational semantics of a specification SP consists of all observational algebras which satisfy observationally (i.e. up to observational equality) the axioms of SP.

To study observational consequences of a specification SP, we also consider its (observational) "black box semantics" which consists of the fully abstract models of SP. The axiomatization of full abstractness leads to proof principles for verifying observational consequences of a specification.

1.2 Reachability

Reachability concepts provide means to specify generation principles for datatypes. The standard approach to reachability is to introduce a set of datatype constructors and to admit as models of a specification only those algebras which are reachable w.r.t. the given constructors. Most algebraic specification languages incorporate features to express reachability like, for instance, the Casl language [4].

Syntactically, we will follow these approaches where the operations of an algebraic signature are split into a set of "constructor operations" for generating the relevant data and the "other" operations which perform computations. From the semantic point of view, however, we do not adopt the standard interpretation which is too restrictive w.r.t. our working hypothesis from above, since many examples show that a correct realization of a specification may contain non-reachable (junk) elements which are simply not relevant for computations. It is only important that the non-constructor operations, when applied to reachable data, cannot produce values which lie outside the constructor-generated part of the algebra. This property is captured by our notion of constructor-based algebra. The constructor-based semantics of a specification SP consists of all constructor-based algebras which satisfy up to junk elements the axioms of SP. Hence we use, analogously to the observational approach, a relaxed satisfaction relation (called constructor-based satisfaction), which interprets variables of a formula only by values in the constructor-generated part of an algebra. Using these notions we develop a novel institution, called the constructor-based logic institution, for the treatment of reachability.

To study inductive consequences of a constructor-based specification SP, we consider its (constructor-based) "black box semantics" which consists of the reachable models of SP. The axiomatization of reachability leads to proof principles like finitary and infinitary induction for verifying inductive consequences of a specification.

1.3 Duality Principle

It is obvious that the notions and results of the observational and constructor-based logic institutions (like observer and constructor operation, observational equality and constructor-generated part, observational and constructor-based algebra, observational and constructor-based satisfaction, fully abstract and reachable algebra etc.) are developed in a completely analogous way. This leads to the question whether there is a formalization of the analogy between the two concepts. We will show that indeed a formal duality principle can be established if we express the central notions of the observational and constructor-based logics in a category-theoretic setting. Thereby the syntactic aspects of the observational and the constructor-based notions are expressed by appropriate functors and the semantic notions of observational and constructor-based algebras and their properties are represented by dual constructions on algebras and coalgebras defined w.r.t. these functors.

1.4 Organization of this Work

First, in Section 2, we reconsider the observational logic institution [18] which is used as the basis for formalizing observability. Then, in Section 3, we discuss reachability and we introduce the constructor-based logic institution. 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 black box views and on proof systems for observational and constructor-based specifications which lead to a further comparison between observability and reachability. In Section 6, we develop the formal duality principle for our observability and reachability concepts. Finally, some concluding remarks are given in Section 7.

1.5 Algebraic Preliminaries

We assume that the reader is familiar with the basic notions of algebraic specifications (see, e.g., [31,5]), 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* $op : s_1, \ldots, s_n \rightarrow s$), *signature morphism* $\sigma : \Sigma \rightarrow \Sigma'$, *(total)* Σ*-algebra* $A = ((A_s)_{s \in S}, (op^A)_{op \in OP})$, Σ*-term algebra* $T_\Sigma(X)$ over a family $X = (X_s)_{s \in S}$ of sets X_s of variables of sort s and *interpretation* $I_\alpha : T_\Sigma(X) \rightarrow A$ w.r.t. a *valuation* $\alpha : X \rightarrow A$. The class of all Σ-algebras is denoted by $\mathrm{Alg}(\Sigma)$. Together with Σ-morphisms this class forms a category which, for simplicity, is also denoted by $\mathrm{Alg}(\Sigma)$. For any signature morphism $\sigma : \Sigma \rightarrow \Sigma'$, the *reduct functor* $\_|_\sigma : \mathrm{Alg}(\Sigma') \rightarrow \mathrm{Alg}(\Sigma)$ is defined as usual. The reduct of a relation $R' \subseteq A' \times B'$ w.r.t. $\sigma : \Sigma \rightarrow \Sigma'$ is denoted by $R'|_\sigma$ where $R'|_\sigma \subseteq A'|_\sigma \times B'|_\sigma$ is defined by $(R'|_\sigma)_s \overset{\mathrm{def}}{=} R'_{\sigma(s)}$ for all $s \in S$.

2 The Observational Logic Institution

In this study we will use the observational logic institution introduced in [18] to formalize observability. In the remainder of this section we reconsider this institution (with a modified definition of observational signature and observable context) and we will provide all the necessary proofs, in particular that observational logic satisfies the satisfaction condition of institutions.[3]

First, we introduce the notion of an observational signature which is a standard algebraic signature together with a distinguished set of observer operations. An n-ary operation $op : s_1, \ldots, s_n \rightarrow s$ with several non-observable argument sorts may also be used as an observer. In this case op is equipped with a "position number" $1 \leq i \leq n$ which indicates the argument sort of the observed elements (also called "states").

Definition 1 (Observational signature) *An* observer *is a pair* (obs, i) *where obs is an operation symbol* $obs : s_1, \ldots, s_i, \ldots, s_n \rightarrow s$ *with* $1 \leq i \leq n$*. The distinguished argument sort* s_i *of obs is called a* state sort *(or* hidden

[3] Up to now proofs for the observational logic framework have only been given in a technical report [17]. The proofs provided here are more elegant and, moreover, we will see that a completely analogous reasoning can be used to prove corresponding facts for the constructor-based logic institution in Section 3.

sort*). If $obs : s_1 \to s$ is a unary observer we will simply write obs instead of* $(obs, 1)$.

An observational signature $\Sigma_{\mathrm{Obs}} = (\Sigma, OP_{\mathrm{Obs}})$ *consists of a signature* $\Sigma = (S, OP)$ *and a set* OP_{Obs} *of observers* (obs, i) *with* $obs \in OP$.

The set $S_{\mathrm{State}} \subseteq S$ *of* state sorts *(or* hidden sorts*, w.r.t.* OP_{Obs}*) consists of all sorts* s_i *such that there exists at least one observer* (obs, i) *in* OP_{Obs}, $obs : s_1, \dots, s_i, \dots, s_n \to s$. *The set* $S_{\mathrm{Obs}} \subseteq S$ *of* observable sorts *consists of all sorts which are not a state sort, i.e.* $S_{\mathrm{Obs}} = S \setminus S_{\mathrm{State}}$.

An observer $(obs, i) \in OP_{\mathrm{Obs}}$ *with profile* $obs : s_1, \dots, s_i, \dots, s_n \to s$ *is called a* direct observer *of* s_i *if* $s \in S_{\mathrm{Obs}}$*, otherwise it is an* indirect observer.

We implicitly assume in the following that whenever we consider an observational signature Σ_{Obs}, then $\Sigma_{\mathrm{Obs}} = (\Sigma, OP_{\mathrm{Obs}})$ with $\Sigma = (S, OP)$ and similarly for Σ'_{Obs} etc.

Note that in the above definition the state sorts and the observable sorts are uniquely determined by the given observers. This is different from [18] (and other previous approaches) where the set of observable sorts was explicitly declared as part of an observational signature. We believe that the new definition is closer to our intuition since, indeed, declaring an observer (obs, i) with $obs : s_1, \dots, s_i, \dots, s_n \to s$ means simultaneously that s_i is not directly visible, i.e., is a state sort. In particular, if $OP_{\mathrm{Obs}} = \emptyset$, then there is no state sort, i.e. all sorts are observable. This corresponds also to the constructor-based case where, if no constructors are provided, there is no constrained sort, i.e. all sorts are loose (see Section 3). Moreover, we will see in Section 6 that in the coalgebraic setting, observers are expressed by functors which, by definition, simultaneously determine state sorts and observable sorts.

For example, an observational signature for streams of booleans could be obtained from the following standard signature $\Sigma_{\mathrm{STREAM}} = (\{bool, stream\}, \{head : stream \to bool,\ tail : stream \to stream,\ merge : stream \times stream \to stream,\ rev : stream \to stream\})$ by choosing *head* and *tail* as observers. Hence *stream* is a state sort and *bool* is an observable sort.[4]

Any observational signature determines a set of observable contexts which represent the observable experiments. Observable contexts are built by observer operations only. They have a state sort as “application sort” (since they are used to observe states) and an observable result sort. The following definition shows how observable contexts are constructed in a coinductive style starting from direct observers. This is syntactically different from [18] (and other previous work) where observable contexts were defined in an inductive

[4] Usual operations on booleans are omitted.

style starting from "trivial" contexts consisting only of a single variable z_s. We do not adopt this approach anymore since the coinductive style is more adequate w.r.t. observability. First, it leads directly to a coinductive specification method (see Section 4) and, secondly, it leads to a coinduction scheme for performing proofs of observational properties as discussed at the end of Section 5.1.

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

(1) For each direct observer (obs, i) with $obs : s_1, \ldots, s_i, \ldots, s_n \to s'$ and pairwise different variables $x_1{:}s_1, \ldots, x_n{:}s_n$,
$obs(x_1, \ldots, x_{i-1}, z_{s_i}, x_{i+1}, \ldots, x_n) \in \mathcal{C}(\Sigma_{\text{Obs}})_{s_i \to s'}$.

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

The set of all observable contexts is denoted by $\mathcal{C}(\Sigma_{\text{Obs}})$. We implicitly assume in the following that for any state sort $s \in S_{\text{State}}$ there exists an observable context with application sort s.

The syntactic notion of 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 $A \in \text{Alg}(\Sigma)$, the* observational Σ_{Obs}-equality *on A is denoted by $\approx_{\Sigma_{\text{Obs}},A}$ and defined as follows.*
For all $s \in S$, two elements $a, b \in A_s$ are observationally equal w.r.t. Σ_{Obs}, i.e., $a \approx_{\Sigma_{\text{Obs}},A} b$, if and only if

***Case** $s \in S_{\text{Obs}}$: $a = b$*
***Case** $s \in S_{\text{State}}$: for all observable sorts $s' \in S_{\text{Obs}}$, for all observable contexts $c \in \mathcal{C}(\Sigma_{\text{Obs}})_{s \to s'}$, 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 $I_\alpha(c) = I_\beta(c)$.*

Definition 4 (Fully-abstract algebra) *Let Σ_{Obs} be an observational signature. A Σ-algebra A is called* fully abstract *(w.r.t. Σ_{Obs}) if the observational Σ_{Obs}-equality $\approx_{\Sigma_{\text{Obs}},A}$ on A coincides with the set-theoretic equality.*

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 fulfilled by observational algebras defined as follows.

Definition 5 (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* $\mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$*.*

Since for any observational 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 is a Σ-algebra that 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 behavior" of A w.r.t. Σ_{Obs}. $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ is *fully abstract* since the observational equality (w.r.t. Σ_{Obs}) on $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ coincides with the set-theoretic equality.

Definition 6 (Observational black box view) *Let A be an observational Σ_{Obs}-algebra. The quotient algebra $A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$ is called the* (observational) black box view *of A.*

To obtain a category of observational algebras we define the following observational morphism notion which is a generalization of standard Σ-morphisms reflecting the relationships between the observable behaviors of algebras.

Definition 7 (Observational morphism) *Let $A, B \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$ be two observational Σ_{Obs}-algebras. An* observational Σ_{Obs}-morphism *$h : A \to B$ is an S-sorted family $(h_s)_{s\in S}$ of relations $h_s \subseteq A_s \times B_s$ with the following properties, for all $s \in S$:*

(1) For all $a \in A_s$, there exists $b \in B_s$ such that $a\ h_s\ b$.
(2) For all $a \in A_s, b, b' \in B_s$, if $a\ h_s\ b$, then ($a\ h_s\ b'$ if and only if $b \approx_{\Sigma_{\mathrm{Obs}},B} b'$).
(3) For all $a, a' \in A_s, b \in B_s$, if $a\ h_s\ b$ and $a \approx_{\Sigma_{\mathrm{Obs}},A} a'$, then $a'\ h_s\ b$.
(4) For all $op : s_1, \dots, s_n \to s \in OP$ and $a_i \in A_{s_i}, b_i \in B_{s_i}$, if $a_i\ h_{s_i}\ b_i$ for $i = 1, \dots, n$, then $op^A(a_1, \dots, a_n)\ h_s\ op^B(b_1, \dots, b_n)$.

The following lemma shows that there is a one to one correspondence between observational morphisms $h : A \to B$ and standard morphisms $k : A/{\approx_{\Sigma_{\mathrm{Obs}},A}} \to B/{\approx_{\Sigma_{\mathrm{Obs}},B}}$ between the observational black box views of A and B.[5]

[5] Hence observational morphisms could have been defined also directly as standard morphisms between the black box views of two observational algebras A and B. We prefer, however, an explicit definition on the carriers of A and B and to distinguish clearly between the category of observational algebras and the one of standard

Lemma 8 *Let* $A, B \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$ *be two observational* Σ_{Obs}*-algebras and* $h : A \to B$ *be an observational* Σ_{Obs}*-morphism. Then* $h/{\approx_{\Sigma_{\mathrm{Obs}}}} : A/{\approx_{\Sigma_{\mathrm{Obs}},A}} \to B/{\approx_{\Sigma_{\mathrm{Obs}},B}}$*, defined by* $h/{\approx_{\Sigma_{\mathrm{Obs}}}}([a]) = [b]$ *if* $a\ h\ b$*, is a* Σ*-morphism. Moreover, for each* Σ*-morphism* $k : A/{\approx_{\Sigma_{\mathrm{Obs}},A}} \to B/{\approx_{\Sigma_{\mathrm{Obs}},B}}$ *, there exists a unique* Σ_{Obs}*-morphism* $h : A \to B$ *such that* $h/{\approx_{\Sigma_{\mathrm{Obs}}}} = k$*.*

Proof. The properties of observational morphisms imply that $h/{\approx_{\Sigma_{\mathrm{Obs}}}}$ is a well-defined Σ-morphism. For proving the second part of the lemma assume that $k : A/{\approx_{\Sigma_{\mathrm{Obs}},A}} \to B/{\approx_{\Sigma_{\mathrm{Obs}},B}}$ is a Σ-morphism. Then k induces a family of relations $h_s \subseteq A_s \times B_s$ such that for all $a \in A_s$, $b \in B_s$ we have $a\ h_s\ b$ if and only if $k_s([a]) = [b]$. It is straightforward to show that h is indeed an observational Σ_{Obs}-morphism between A and B such that $h/{\approx_{\Sigma_{\mathrm{Obs}}}} = k$. For proving the uniqueness of h let $h' : A \to B$ be an observational Σ_{Obs}-morphism with $h'/{\approx_{\Sigma_{\mathrm{Obs}}}} = k$. Then, for any $a \in A_s$, $b \in B_s$, $a\ h_s\ b$ iff $k_s([a]) = [b]$ iff $h'/{\approx_{\Sigma_{\mathrm{Obs}}}}([a]) = [b]$ iff $a\ h'_s\ b$. □

Definition 9 (Category of observational algebras) *For any observational signature* Σ_{Obs}*, the class* $\mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$ *together with the observational* Σ_{Obs}*-morphisms defines a category which, by abuse of notation, will also be denoted by* $\mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$*. The composition of observational* Σ_{Obs}*-morphisms is the usual composition of relations and for each* $A \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$*, the identity* $id_A : A \to A$ *is the observational equality* $\approx_{\Sigma_{\mathrm{Obs}},A}$*.*[6]

Using the observational black box construction of Definition 6, one can relate, for any observational signature Σ_{Obs}, the category $\mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$ of observational Σ_{Obs}-algebras and the category $\mathrm{Alg}(\Sigma)$ of (standard) Σ-algebras by a functor which associates to any observational algebra its black box view. According to Lemma 8 this functor establishes a one to one correspondence between observational and standard morphisms, i.e., it is full and faithful.

Definition 10 (Observational black box functor) *For any observational signature* Σ_{Obs}*,* $\mathcal{BB}_{\Sigma_{\mathrm{Obs}}} : \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}}) \to \mathrm{Alg}(\Sigma)$ *is the full and faithful functor defined by:*

(1) For each $A \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$*,* $\mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(A) \overset{\mathrm{def}}{=} A/{\approx_{\Sigma_{\mathrm{Obs}},A}}$*.*
(2) For each observational Σ_{Obs}*-morphism* $h : A \to B$*,* $\mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(h) \overset{\mathrm{def}}{=} h/{\approx_{\Sigma_{\mathrm{Obs}}}}$ *where* $h/{\approx_{\Sigma_{\mathrm{Obs}}}} : A/{\approx_{\Sigma_{\mathrm{Obs}},A}} \to B/{\approx_{\Sigma_{\mathrm{Obs}},B}}$ *is defined in Lemma 8.*

In the next step we define an observational satisfaction relation between observational algebras and first-order Σ-formulas. The underlying idea of this 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. Hence the following definition is quite similar to the def-

algebras.

[6] It is easy to prove that all properties of a category are indeed satisfied.

inition of the standard satisfaction relation. The only difference concerns *(1)* where "$I_\alpha(t) = I_\alpha(r)$" is replaced by "$I_\alpha(t) \approx_{\Sigma_{\mathrm{Obs}},A} I_\alpha(r)$".

Definition 11 (Observational satisfaction relation) *The* observational satisfaction relation *between 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 φ 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.

The notation $A \models_{\Sigma_{\mathrm{Obs}}} \varphi$ is extended in the usual way to classes of observational algebras and sets of formulas. The next theorem shows that the observational black box functor is compatible with the observational and standard satisfaction relations.

Theorem 12 *Let Σ_{Obs} be an observational signature with underlying standard signature Σ, let φ be a Σ-formula and let A be a Σ_{Obs}-algebra. Then:*
$A \models_{\Sigma_{\mathrm{Obs}}} \varphi$ if and only if $\mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(A) \models_\Sigma \varphi$.[7]

This theorem is a generalization of Theorem 3.11 in [10]. The proof is done by induction on the form of the formula φ (along the lines of the proof of Theorem 3.11 in [10]). Similar results are provided in [23] and in [33].

Definition 13 (Basic observational specification) *A* basic observational specification $\mathrm{SP}_{\mathrm{Obs}} = \langle \Sigma_{\mathrm{Obs}}, \mathrm{Ax} \rangle$ *consists of an observational signature* $\Sigma_{\mathrm{Obs}} = (\Sigma, OP_{\mathrm{Obs}})$ *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:*

$$\mathrm{Sig}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}) \overset{\mathrm{def}}{=} \Sigma_{\mathrm{Obs}}$$

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

In the following, $\mathrm{SP}_{\mathrm{Obs}} \models_{\Sigma_{\mathrm{Obs}}} \varphi$ means $\mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}) \models_{\Sigma_{\mathrm{Obs}}} \varphi$.

The definitions stated above provide the basic ingredients for defining the *observational logic institution.* Thereby it is particularly important to use an appropriate morphism notion for observational signatures which guarantees

[7] When it is clear from the context we often write $\models$ instead of $\models_\Sigma$ to denote the standard satisfaction relation.

encapsulation of properties with respect to the observational satisfaction relation (formally expressed by the satisfaction condition of institutions, see [14]). To ensure that the satisfaction condition holds, the crucial idea is to require that observers are preserved (formally expressed by condition *(1)* below) and that no "new" observer can be introduced for "old" sorts via a signature morphism (formally expressed by condition *(2)* below). Then the set of observable contexts for observing "old" sorts remains unchanged (up to renaming) and so does the observational equality. This fact is formally stated in Lemma 16 below.

Definition 14 (Observational signature morphism) *Given two observational signatures $\Sigma_{\mathrm{Obs}} = (\Sigma, OP_{\mathrm{Obs}})$ and $\Sigma'_{\mathrm{Obs}} = (\Sigma', OP'_{\mathrm{Obs}})$ with $\Sigma = (S, OP)$ and $\Sigma' = (S', OP')$, an* observational signature morphism $\sigma_{\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma'_{\mathrm{Obs}}$ *is a signature morphism $\sigma : \Sigma \to \Sigma'$ such that:*

(1) If $(obs, i) \in OP_{\mathrm{Obs}}$, then $(\sigma(obs), i) \in OP'_{\mathrm{Obs}}$.
(2) If $(obs', i) \in OP'_{\mathrm{Obs}}$ with $obs' : s'_1, \ldots, s'_i, \ldots, s'_n \to s'$ and $s'_i \in \sigma(S)$, then there exists $obs \in OP$ such that $(obs, i) \in OP_{\mathrm{Obs}}$ and $obs' = \sigma(obs)$.

Note that this definition implies that for all sorts s in S, $s \in S_{\mathrm{State}}$ if and only if $\sigma(s) \in S'_{\mathrm{State}}$ and $s \in S_{\mathrm{Obs}}$ if and only if $\sigma(s) \in S'_{\mathrm{Obs}}$.

We implicitly assume in the following that whenever we consider an observational signature morphism $\sigma_{\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma'_{\mathrm{Obs}}$, then the underlying signature morphism is $\sigma : \Sigma \to \Sigma'$.

Lemma 15 *Observational signatures together with observational signature morphisms form a category which has pushouts.*

Proof. Obviously the properties of a category are satisfied. To show the existence of pushouts let $\sigma_{1,\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma_{1,\mathrm{Obs}}$ and $\sigma_{2,\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma_{2,\mathrm{Obs}}$ be observational signature morphisms with underlying signature morphisms $\sigma_1 : \Sigma \to \Sigma_1$ and $\sigma_2 : \Sigma \to \Sigma_2$. It is well-known that in the category of algebraic signatures there exists a pushout as shown in the following diagram.

$$\begin{tikzcd}
\Sigma \arrow[r, "\sigma_1"] \arrow[d, "\sigma_2"'] & \Sigma_1 \arrow[d, "\sigma'_1"] \\
\Sigma_2 \arrow[r, "\sigma'_2"] & \Sigma'
\end{tikzcd}$$

Now let $OP'_{\mathrm{Obs}} = \{(\sigma'_1(op_1), i) \mid (op_1, i) \in OP_{1,\mathrm{Obs}}\} \cup \{(\sigma'_2(op_2), i) \mid (op_2, i) \in OP_{2,\mathrm{Obs}}\}$ and $\Sigma'_{\mathrm{Obs}} = (\Sigma', OP'_{\mathrm{Obs}})$. It is straightforward to prove that σ'_1 and σ'_2 give rise to observational signature morphisms $\sigma'_{1,\mathrm{Obs}}$ and $\sigma'_{2,\mathrm{Obs}}$ such that the following diagram is a pushout in the category of observational signature morphisms.

$$\begin{tikzcd}
\Sigma_{\mathrm{Obs}} \arrow[r, "\sigma_{1,\mathrm{Obs}}"] \arrow[d, "\sigma_{2,\mathrm{Obs}}"'] & \Sigma_{1,\mathrm{Obs}} \arrow[d, "\sigma'_{1,\mathrm{Obs}}"] \\
\Sigma_{2,\mathrm{Obs}} \arrow[r, "\sigma'_{2,\mathrm{Obs}}"] & \Sigma'_{\mathrm{Obs}}
\end{tikzcd}$$

□

The next lemma provides the basis for defining the observational reduct functor and for proving the (observational) satisfaction condition. It says that observational equalities are compatible with reducts along observational signature morphisms.

Lemma 16 *For any observational signature morphism $\sigma_{\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma'_{\mathrm{Obs}}$ and observational Σ'_{Obs}-algebra $A' \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma'_{\mathrm{Obs}})$, we have $(\approx_{\Sigma'_{\mathrm{Obs}},A'})|_\sigma = \approx_{\Sigma_{\mathrm{Obs}},(A'|_\sigma)}$. Thereby $(\approx_{\Sigma'_{\mathrm{Obs}},A'})|_\sigma$ is the reduct of the observational Σ'_{Obs}-equality on A' along σ (see Section 1.5) and $\approx_{\Sigma_{\mathrm{Obs}},(A'|_\sigma)}$ is the observational Σ_{Obs}-equality on the reduct $A'|_\sigma$.*

Proof. Let $s \in S$ and $a, b \in (A'|_\sigma)_s$. Then $a, b \in A'_{\sigma(s)}$ and $a\ (\approx_{\Sigma'_{\mathrm{Obs}},A'})|_\sigma\ b$ iff $a \approx_{\Sigma'_{\mathrm{Obs}},A'} b$. Hence it is sufficient to prove $a \approx_{\Sigma'_{\mathrm{Obs}},A'} b$ iff $a \approx_{\Sigma_{\mathrm{Obs}},(A'|_\sigma)} b$. If $s \in S_{\mathrm{Obs}}$ then $\sigma(s) \in S'_{\mathrm{Obs}}$ and conversely. Hence, in this case, $a \approx_{\Sigma'_{\mathrm{Obs}},A'} b$ iff $a = b$ iff $a \approx_{\Sigma_{\mathrm{Obs}},(A'|_\sigma)} b$. If $s \in S_{\mathrm{State}}$ then $\sigma(s) \in S'_{\mathrm{State}}$ and conversely. In this case, the conditions *(1)* and *(2)* of Definition 14 imply that for any observable context $c' \in \mathcal{C}(\Sigma'_{\mathrm{Obs}})$ with application sort $\sigma(s)$ one can construct a corresponding observable context $c \in \mathcal{C}(\Sigma_{\mathrm{Obs}})$ with application sort s and vice versa. Hence we can conclude $a \approx_{\Sigma'_{\mathrm{Obs}},A'} b$ iff $a \approx_{\Sigma_{\mathrm{Obs}},(A'|_\sigma)} b$. □

As a first obvious consequence of Lemma 16 we obtain the following fact which allows us to define the observational reduct functor in Definition 18.

Corollary 17 *For any observational signature morphism $\sigma_{\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma'_{\mathrm{Obs}}$ and for any observational Σ'_{Obs}-algebra $A' \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma'_{\mathrm{Obs}})$, $A'|_\sigma \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$. Moreover, for any observational Σ'_{Obs}-morphism $h' : A' \to B'$ the reduct $h'|_\sigma : A'|_\sigma \to B'|_\sigma$ is an observational Σ_{Obs}-morphism.*

Definition 18 (Observational reduct functor) *For any observational signature morphism $\sigma_{\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma'_{\mathrm{Obs}}$, the functor $\_|_{\sigma_{\mathrm{Obs}}} : \mathrm{Alg}_{\mathrm{Obs}}(\Sigma'_{\mathrm{Obs}}) \to \mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$ is defined as follows.*

(1) For each $A' \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma'_{\mathrm{Obs}})$, $A'|_{\sigma_{\mathrm{Obs}}} \stackrel{\mathrm{def}}{=} A'|_\sigma$.
(2) For each observational Σ'_{Obs}-morphism $h' : A' \to B'$, $h'|_{\sigma_{\mathrm{Obs}}} \stackrel{\mathrm{def}}{=} h'|_\sigma$.

As a second consequence of Lemma 16, we obtain that the (observational) black box functor commutes with the reduct functor. This important fact shows again the adequacy of the notion of observational signature morphisms.

Corollary 19 *For any observational signature morphism* $\sigma_{\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma'_{\mathrm{Obs}}$ *and for any observational* Σ'_{Obs}*-algebra* $A' \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma'_{\mathrm{Obs}})$,
$\mathcal{BB}_{\Sigma'_{\mathrm{Obs}}}(A')|_{\sigma} \ = \ \mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(A'|_{\sigma_{\mathrm{Obs}}})$.

The last corollary and Theorem 12 are the essential facts that are needed to prove the (observational) satisfaction condition.

Theorem 20 (Observational satisfaction condition) *For any observational signature morphism* $\sigma_{\mathrm{Obs}} : \Sigma_{\mathrm{Obs}} \to \Sigma'_{\mathrm{Obs}}$*, observational* Σ'_{Obs}*-algebra* $A' \in \mathrm{Alg}_{\mathrm{Obs}}(\Sigma'_{\mathrm{Obs}})$ *and* Σ*-sentence* φ*:*
$A' \models_{\Sigma'_{\mathrm{Obs}}} \sigma(\varphi)$ *if and only if* $A'|_{\sigma_{\mathrm{Obs}}} \models_{\Sigma_{\mathrm{Obs}}} \varphi$.

Proof. $A' \models_{\Sigma'_{\mathrm{Obs}}} \sigma(\varphi)$ iff, by Theorem 12, $\mathcal{BB}_{\Sigma'_{\mathrm{Obs}}}(A') \models_{\Sigma'} \sigma(\varphi)$ iff (since the satisfaction condition holds in the institution of standard many-sorted first-order logic) $\mathcal{BB}_{\Sigma'_{\mathrm{Obs}}}(A')|_{\sigma} \models_{\Sigma} \varphi$ iff, by Corollary 19, $\mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(A'|_{\sigma_{\mathrm{Obs}}}) \models_{\Sigma} \varphi$ iff, by Theorem 12, $A'|_{\sigma_{\mathrm{Obs}}} \models_{\Sigma_{\mathrm{Obs}}} \varphi$. □

We have now defined all ingredients that constitute the *observational logic institution*. The category of signatures is the category of observational signatures and observational signature morphisms, for each observational signature $\Sigma_{\mathrm{Obs}} = (\Sigma, \mathit{OP}_{\mathrm{Obs}})$, the sentences are finitary first-order Σ-sentences, the model functor assigns to each observational signature Σ_{Obs} the category $\mathrm{Alg}_{\mathrm{Obs}}(\Sigma_{\mathrm{Obs}})$ of observational Σ_{Obs}-algebras and Σ_{Obs}-morphisms, and the satisfaction relation is the observational satisfaction relation.

The following remark discusses briefly some properties and further aspects of the observational logic institution.

Remark 21

(1) Observational logic satisfies the amalgamation property as defined, for instance, in [39]. This can be proved by applying the construction of amalgamations for standard algebras to observational algebras. That the amalgamated union of two observational algebras is again an observational algebra is a consequence of Lemma 16.

(2) If we allowed infinitary Σ*-sentences (with countably infinite conjunctions and disjunctions) and restricted to injective signature morphisms then the interpolation property would be satisfied as well.*[8] *The proof of this fact is given in [17]. It relies on the infinitary axiomatization of full abstractness presented in Section 5.1 and on Corollary 44 and Theorem 45.*

[8] For the definition of the interpolation property see, e.g., [39].

(3) On top of the observational logic institution, structured observational specifications can be defined by applying the institution-independent specification-building operators introduced in [37] and similarly in [6]. Since the observational logic institution satisfies the amalgamation property, one can compute, following the construction in [6], for each structured observational specification, a normal form which consists (in general) of a basic observational specification restricted to an export signature.

(4) From the above theorems we can conclude that the functors $\mathcal{BB}_{\Sigma_{\mathrm{Obs}}}$ associated to observational signatures Σ_{Obs} can be extended to an institution encoding (in the sense of [39]) which maps the institution of observational logic to the institution of standard first-order logic. A concrete discussion on how this institution encoding works is outside the scope of this paper.

3 The Constructor-Based Logic Institution

Reachability concepts are used to describe the underlying data manipulated by a program. For this purpose a distinguished subset OP_{Cons} of the operation symbols OP (of a signature $\Sigma = (S, OP)$) is declared as a set of constructor symbols which leads to our notion of a constructor-based signature (see Definition 22 below). As already discussed in Section 1.2 the standard semantic approach to reachability is to restrict the admissible models of a specification to those algebras which are reachable w.r.t. the given constructors. We believe that this interpretation 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 $\Sigma_{\mathrm{NAT}} = (\{nat\}, \{zero : \to nat, succ : nat \to nat, add : nat \times nat \to nat\})$ 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, where negative integers are just junk elements, since they are not used as representations for natural numbers. 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 their subsets of constructor-generated elements are closed under the non-constructor operations. This condition is formalized by our notion of constructor-based algebra in Definition 26 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 correspondences will be analyzed in Section 4 and formalized in Section 6.

Definition 22 (Constructor-based signature) *A* constructor *is an operation symbol $cons : s_1, \ldots, s_n \to s$ with $n \geq 0$. The result sort s of $cons$ is called a* constrained sort*.*

A constructor-based signature *$\Sigma_{\text{Cons}} = (\Sigma, OP_{\text{Cons}})$ consists of a signature $\Sigma = (S, OP)$ and a set $OP_{\text{Cons}} \subseteq OP$ of constructors.*

The set $S_{\text{Cons}} \subseteq S$ of constrained sorts *(w.r.t. OP_{Cons}) consists of all sorts s such that there exists at least one constructor in OP_{Cons} with range s. The set $S_{\text{Loose}} \subseteq S$ of* loose *sorts consists of all sorts which are not a constrained sort, i.e. $S_{\text{Loose}} = S \setminus S_{\text{Cons}}$.*

We implicitly assume in the following that whenever we consider a constructor-based signature Σ_{Cons}, then $\Sigma_{\text{Cons}} = (\Sigma, OP_{\text{Cons}})$ with $\Sigma = (S, OP)$ and similarly for Σ'_{Cons} etc.

Note that in the above definition, the constrained sorts and the loose sorts are uniquely determined by the given constructors. Indeed, declaring a constructor $cons : s_1, \ldots, s_n \to s$ means simultaneously that s is constrained. In particular, if $OP_{\text{Cons}} = \emptyset$, then there is no constrained sort, i.e., all sorts are loose.

For example a constructor-based signature for the natural numbers is obtained from Σ_{NAT} (cf. above) by choosing *zero* and *succ* as constructors.

Any constructor-based signature determines a set of constructor terms. The following definition shows how constructor terms are inductively constructed starting from constants. The interpretation of a constructor term denotes always a value of a constrained sort. [9]

Definition 23 (Constructor term) *Let be given a constructor-based signature Σ_{Cons}, and let $X = (X_s)_{s \in S}$ be a family of countably infinite sets X_s of variables of sort s. For all $s \in S_{\text{Cons}}$, the set $\mathcal{T}(\Sigma_{\text{Cons}})_s$ of* constructor terms *with "constrained result sort" s is inductively defined as follows:*

[9] This would not be the case if we used another definition where single variable terms x_s with $s \in S_{\text{Loose}}$ would be included in the set of constructor terms. Moreover, the definition given here points out clearly the analogy with the definition of observable contexts in Definition 2.

(1) Each constant $cons :\to s \in OP_{\text{Cons}}$ *belongs to* $\mathcal{T}(\Sigma_{\text{Cons}})_s$ *.*

(2) For each constructor $cons : s_1, \ldots, s_n \to s \in OP_{\text{Cons}}$ *with* $n \geq 1$ *and terms* $t_1, \ldots, t_n$ *such that* t_i *is a variable* $x_i{:}s_i$ *if* $s_i \in S_{\text{Loose}}$ *and* $t_i \in \mathcal{T}(\Sigma_{\text{Cons}})_{s_i}$ *if* $s_i \in S_{\text{Cons}}$, $cons(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma_{\text{Cons}})_s$*.*

The set of all constructor terms is denoted by $\mathcal{T}(\Sigma_{\text{Cons}})$*. We implicitly assume in the following that for any constrained sort* $s \in S_{\text{Cons}}$ *there exists a constructor term of sort* s*.*

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 the Σ_{Cons}-generated part, which intuitively consists of those data which are relevant according to the given constructors.

Definition 24 (Σ_{Cons}-generated part) *Let be given a constructor-based signature* Σ_{Cons}*. For any* Σ*-algebra* $A \in \mathrm{Alg}(\Sigma)$*, the* Σ_{Cons}-generated part *of* A*, denoted by* $Gen_{\Sigma_{\text{Cons}}}(A) = (Gen_{\Sigma_{\text{Cons}}}(A)_s)_{s \in S}$*, is defined by:*

Case $s \in S_{\text{Loose}}$***:*** $Gen_{\Sigma_{\text{Cons}}}(A)_s = A_s$

Case $s \in S_{\text{Cons}}$***:*** $Gen_{\Sigma_{\text{Cons}}}(A)_s = \{a \in A_s \mid \text{there exists a term } t \in \mathcal{T}(\Sigma_{\text{Cons}})_s \text{ and a valuation } \alpha : X \to A \text{ such that } I_\alpha(t) = a\}$.

Definition 25 (Reachable algebra) *Let* Σ_{Cons} *be a constructor-based signature. A* Σ*-algebra* A *is called* reachable *(w.r.t.* Σ_{Cons}*) if its carrier sets coincide with the carrier sets of its* Σ_{Cons}*-generated part.*

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

Definition 26 (Constructor-based algebra) *Let* Σ_{Cons} *be a constructor-based signature. A* constructor-based Σ_{Cons}-algebra *is a* Σ*-algebra* A *such that* $Gen_{\Sigma_{\text{Cons}}}(A)$*, equipped with the canonical restrictions of the operations* op^A *of* A *to the carrier sets of* $Gen_{\Sigma_{\text{Cons}}}(A)$*, is a* Σ*-subalgebra of* A*. The class of all constructor-based* Σ_{Cons}*-algebras is denoted by* $\mathrm{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$*.*

Since for any Σ_{Cons}-algebra A, the Σ_{Cons}-generated part $Gen_{\Sigma_{\text{Cons}}}(A)$ of A is a Σ-algebra which contains only those elements that are generated by the given constructors, we can consider the Σ_{Cons}-generated part $Gen_{\Sigma_{\text{Cons}}}(A)$ as the (constructor-based) "black box view" of A (abstracting away from all junk values that may lie in A). Obviously, $Gen_{\Sigma_{\text{Cons}}}(A)$ is *reachable* w.r.t. Σ_{Cons}.

Definition 27 (Constructor-based black box view) *Let* A *be a constructor-based* Σ_{Cons}*-algebra. The* Σ_{Cons}*-generated part* $Gen_{\Sigma_{\text{Cons}}}(A)$ *(considered as a subalgebra of* A*) is called the* (constructor-based) black box view *of* A*.*

For instance, the black box view of the integers $\mathbb{Z}$ w.r.t. the constructors *zero* and *succ* corresponds to the natural numbers.

To obtain a category of constructor-based algebras, we define the following morphism notion which reflects the relationships between the Σ_{Cons}-generated parts of algebras.

Definition 28 (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 an S-sorted family $(h_s)_{s \in S}$ of partial mappings $h_s : A_s \to B_s$ with the following properties, for all $s \in S$:*

(1) The definition domain of h_s is $Gen_{\Sigma_{\text{Cons}}}(A)_s$.
(2) $h_s(Gen_{\Sigma_{\text{Cons}}}(A)_s) \subseteq Gen_{\Sigma_{\text{Cons}}}(B)_s$.
(3) For all $op : s_1, \ldots, s_n \to s \in OP$ and $a_i \in Gen_{\Sigma_{\text{Cons}}}(A)_{s_i}$, $h_s(op^A(a_1, \ldots, a_n)) \;=\; op^B(h_{s_1}(a_1), \ldots, h_{s_n}(a_n))$.

Obviously, there is a one to one correspondence between constructor-based morphisms $h : A \to B$ and standard morphisms $k : Gen_{\Sigma_{\text{Cons}}}(A) \to Gen_{\Sigma_{\text{Cons}}}(B)$. [10] For instance, the integers are isomorphic to the natural numbers w.r.t. the constructors *zero* and *succ*.

Lemma 29 *Let $A, B \in \text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$ be two constructor-based Σ_{Cons}-algebras and $h : A \to B$ be a constructor-based Σ_{Cons}-morphism. Then the restriction $h|_{Gen_{\Sigma_{\text{Cons}}}(A)} : Gen_{\Sigma_{\text{Cons}}}(A) \to Gen_{\Sigma_{\text{Cons}}}(B)$ is a Σ-morphism. Moreover, for each Σ-morphism $k : Gen_{\Sigma_{\text{Cons}}}(A) \to Gen_{\Sigma_{\text{Cons}}}(B)$, there exists a unique Σ_{Cons}-morphism $h : A \to B$ such that $h|_{Gen_{\Sigma_{\text{Cons}}}(A)} = k$.*

Definition 30 (Category of constructor-based algebras) *For any constructor-based signature Σ_{Cons}, the class $\text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$ together with the constructor-based Σ_{Cons}-morphisms defines a category which, by abuse of notation, will also be denoted by $\text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$.*

Using the constructor-based black box construction of Definition 27, one can relate, for any constructor-based signature Σ_{Cons}, the category $\text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}})$ of constructor-based Σ_{Cons}-algebras and the category $\text{Alg}(\Sigma)$ of (standard) Σ-algebras by a functor which associates to any constructor-based algebra its black box view. According to Lemma 29, this functor is full and faithful.

Definition 31 (Constructor-based black box functor) *For any constructor-based signature Σ_{Cons}, $\mathcal{BB}_{\Sigma_{\text{Cons}}} : \text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}}) \to \text{Alg}(\Sigma)$ is the full and faithful functor defined by:*

[10] Similarly to the observational case, constructor-based morphisms could have been defined also directly as standard morphisms between the (constructor-based) black box views of two constructor-based algebras A and B.

(1) *For each* $A \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$, $\mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(A) \overset{\mathrm{def}}{=} \mathit{Gen}_{\Sigma_{\mathrm{Cons}}}(A)$.

(2) *For each constructor-based* Σ_{Cons}*-morphism* $h : A \to B$, $\mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(h) \overset{\mathrm{def}}{=} h|_{\mathit{Gen}_{\Sigma_{\mathrm{Cons}}}(A)}$.

In the next step we define a constructor-based satisfaction relation between constructor-based algebras and first-order Σ-formulas. The underlying idea of this satisfaction relation is to restrict the valuations of variables to the generated values (i.e. to the elements of the Σ_{Cons}-generated part) only.[11] 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 "$\alpha : X \to \mathit{Gen}_{\Sigma_{\mathrm{Cons}}}(A)$".

Definition 32 (Constructor-based satisfaction relation) *The* constructor-based satisfaction relation *between* 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* $\alpha : X \to \mathit{Gen}_{\Sigma_{\mathrm{Cons}}}(A)$, $A, \alpha \models_{\Sigma_{\mathrm{Cons}}} t = r$ *holds if* $I_\alpha(t) = I_\alpha(r)$.

(2) *For any arbitrary* Σ*-formula* φ *and for any valuation* $\alpha : X \to \mathit{Gen}_{\Sigma_{\mathrm{Cons}}}(A)$, $A, \alpha \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *is defined by induction over the structure of the formula* φ *in the usual way. In particular,* $A, \alpha \models_{\Sigma_{\mathrm{Cons}}} \forall x{:}s.\ \varphi$ *holds if for all* $a \in (\mathit{Gen}_{\Sigma_{\mathrm{Cons}}}(A))_s$, $A, \alpha' \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *where* $\alpha'(x) = a$ *and* $\alpha'(y) = \alpha(y)$ *for* $y \neq x$.

(3) *For any arbitrary* Σ*-formula* φ, $A \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *holds if for all valuations* $\alpha : X \to \mathit{Gen}_{\Sigma_{\mathrm{Cons}}}(A)$, $A, \alpha \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *holds.*

The notation $A \models_{\Sigma_{\mathrm{Cons}}} \varphi$ is extended in the usual way to classes of constructor-based algebras and sets of formulas.

As an example consider again the specification NAT and the integers which satisfy w.r.t. the constructor-based satisfaction relation the third Peano axiom, i.e., $\mathbb{Z} \models_{\Sigma_{\mathrm{Cons}}} \forall x{:}nat.\ succ(x) \neq zero$. Indeed this is true since the Σ_{Cons}-generated 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 $\mathbb{N}$.

The next theorem shows that the constructor-based black box functor is compatible with the constructor-based and standard satisfaction relations.

Theorem 33 *Let* Σ_{Cons} *be a constructor-based signature with underlying standard signature* Σ*, let* φ *be a* Σ*-formula and let* A *be a* Σ_{Cons}*-algebra. Then:* $A \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *if and only if* $\mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(A) \models_\Sigma \varphi$.

The proof of this theorem is straightforward by induction on the form of the

[11] This idea is related to the ultra-loose approach of [40], where the same effect is achieved by using formulas with relativized quantification.

formula φ.

Definition 34 (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* $\Sigma_{\mathrm{Cons}} = (\Sigma, OP_{\mathrm{Cons}})$ *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:*

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

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

In the following, $\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ means $\mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}}) \models_{\Sigma_{\mathrm{Cons}}} \varphi$.

For instance, according to the constructor-based satisfaction relation, the integers are an admissible model of NAT considered as a constructor-based specification with constructors *zero* and *succ*.

The definitions stated above provide the basic ingredients for defining the *constructor-based logic institution.* As in the observational case it is again particularly important to use an appropriate morphism notion for constructor-based signatures which guarantees encapsulation of properties with respect to the constructor-based satisfaction relation. To ensure that the satisfaction condition of institutions holds, the crucial idea is quite similar to the observational case. We require that constructors are preserved (formally expressed by condition *(1)* below) and that no “new” constructor can be introduced for “old” sorts via a signature morphism (formally expressed by condition *(2)* below). Then the set of constructor terms for constructing elements of “old” sorts remains unchanged (up to renaming) and so does the Σ_{Cons}-generated part. This fact is formally stated in Lemma 37 below.

Definition 35 (Constructor-based signature morphism) *Given two constructor-based signatures* $\Sigma_{\mathrm{Cons}} = (\Sigma, OP_{\mathrm{Cons}})$ *and* $\Sigma'_{\mathrm{Cons}} = (\Sigma', OP'_{\mathrm{Cons}})$ *with* $\Sigma = (S, OP)$ *and* $\Sigma' = (S', OP')$, *a* constructor-based signature morphism $\sigma_{\mathrm{Cons}} : \Sigma_{\mathrm{Cons}} \to \Sigma'_{\mathrm{Cons}}$ *is a signature morphism* $\sigma : \Sigma \to \Sigma'$ *such that:*

(1) If $cons \in OP_{\mathrm{Cons}}$, *then* $\sigma(cons) \in OP'_{\mathrm{Cons}}$.
(2) If $cons' \in OP'_{\mathrm{Cons}}$ *with* $cons' : s'_1, \ldots, s'_n \to s'$ *and* $s' \in \sigma(S)$, *then there exists* $cons \in OP_{\mathrm{Cons}}$ *such that* $cons' = \sigma(cons)$.

This definition implies that for all sorts s in S, $s \in S_{\mathrm{Cons}}$ if and only if $\sigma(s) \in S'_{\mathrm{Cons}}$ and $s \in S_{\mathrm{Loose}}$ if and only if $\sigma(s) \in S'_{\mathrm{Loose}}$.

We implicitly assume in the following that whenever we consider a constructor-based signature morphism $\sigma_{\mathrm{Cons}} : \Sigma_{\mathrm{Cons}} \to \Sigma'_{\mathrm{Cons}}$, then the underlying signature morphism is $\sigma : \Sigma \to \Sigma'$.

Lemma 36 *Constructor-based signatures together with constructor-based signature morphisms form a category which has pushouts.*

Proof. The proof is performed in the same way as the proof of Lemma 15 by replacing observational signatures by constructor-based signatures and observers by constructors. □

To justify that our constructor-based approach indeed yields an institution the order of arguments is completely analogous to the one used in Section 2 for the observational logic institution. First, we need the following lemma which provides the basis for defining the constructor-based reduct functor and for proving the (constructor-based) satisfaction condition. It says that constructor generated parts are compatible with reducts along constructor-based signature morphisms.

Lemma 37 *For any constructor-based signature morphism* $\sigma_{\mathrm{Cons}} : \Sigma_{\mathrm{Cons}} \to \Sigma'_{\mathrm{Cons}}$ *and for any constructor-based* Σ'_{Cons}*-algebra* $A' \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma'_{\mathrm{Cons}})$, $Gen_{\Sigma'_{\mathrm{Cons}}}(A')|_\sigma = Gen_{\Sigma_{\mathrm{Cons}}}(A'|_\sigma)$.

Proof. If $s \in S_{\mathrm{Loose}}$ then $\sigma(s) \in S'_{\mathrm{Loose}}$ and conversely. Hence, in this case, $(Gen_{\Sigma'_{\mathrm{Cons}}}(A')|_\sigma)_s = Gen_{\Sigma'_{\mathrm{Cons}}}(A')_{\sigma(s)} = A'_{\sigma(s)} = (A'|_\sigma)_s = Gen_{\Sigma_{\mathrm{Cons}}}(A'|_\sigma)_s$. If $s \in S_{\mathrm{Cons}}$ then $\sigma(s) \in S'_{\mathrm{Cons}}$ and conversely. In this case, the conditions *(1)* and *(2)* of Definition 35 imply that for any constructor term $t' \in \mathcal{T}(\Sigma'_{\mathrm{Cons}})_{\sigma(s)}$, one can construct a corresponding constructor term $t \in \mathcal{T}(\Sigma_{\mathrm{Cons}})_s$ and vice versa. Hence we can conclude that $(Gen_{\Sigma'_{\mathrm{Cons}}}(A')|_\sigma)_s = Gen_{\Sigma'_{\mathrm{Cons}}}(A')_{\sigma(s)} = Gen_{\Sigma_{\mathrm{Cons}}}(A'|_\sigma)_s$. □

As a first obvious consequence of Lemma 37 we obtain the following fact which allows us to define the constructor-based reduct functor in Definition 39.

Corollary 38 *For any constructor-based signature morphism* $\sigma_{\mathrm{Cons}} : \Sigma_{\mathrm{Cons}} \to \Sigma'_{\mathrm{Cons}}$ *and for any constructor-based* Σ'_{Cons}*-algebra* $A' \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma'_{\mathrm{Cons}})$, $A'|_\sigma \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$. *Moreover, for any constructor-based* Σ'_{Cons}*-morphism* $h' : A' \to B'$ *the reduct* $h'|_\sigma : A'|_\sigma \to B'|_\sigma$ *is a constructor-based* Σ_{Cons}*-morphism.*

Definition 39 (Constructor-based reduct functor) *For any constructor-based signature morphism* $\sigma_{\mathrm{Cons}} : \Sigma_{\mathrm{Cons}} \to \Sigma'_{\mathrm{Cons}}$*, the following defines a functor* $\_\_|_{\sigma_{\mathrm{Cons}}} : \mathrm{Alg}_{\mathrm{Cons}}(\Sigma'_{\mathrm{Cons}}) \to \mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$*:*

(1) *For each* $A' \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma'_{\mathrm{Cons}})$, $A'|_{\sigma_{\mathrm{Cons}}} \overset{\mathrm{def}}{=} A'|_\sigma$.
(2) *For each constructor-based* Σ'_{Cons}*-morphism* $h' : A' \to B'$, $h'|_{\sigma_{\mathrm{Cons}}} \overset{\mathrm{def}}{=} h'|_\sigma$.

As a second consequence of Lemma 37 we obtain that the (constructor-based) black box functor commutes with the reduct functor.

Corollary 40 *For any constructor-based signature morphism* $\sigma_{\mathrm{Cons}} : \Sigma_{\mathrm{Cons}} \to$

Σ'_{Cons} *and for any constructor-based* Σ'_{Cons}*-algebra* $A' \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma'_{\mathrm{Cons}})$, $\mathcal{BB}_{\Sigma'_{\mathrm{Cons}}}(A')|_\sigma \;=\; \mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(A'|_{\sigma_{\mathrm{Cons}}})$.

The last corollary and Theorem 33 are the essential facts that are needed to prove the (constructor-based) satisfaction condition.

Theorem 41 (Constructor-based satisfaction condition) *For any constructor-based signature morphism* $\sigma_{\mathrm{Cons}} : \Sigma_{\mathrm{Cons}} \to \Sigma'_{\mathrm{Cons}}$*, constructor-based* Σ'_{Cons}*-algebra* $A' \in \mathrm{Alg}_{\mathrm{Cons}}(\Sigma'_{\mathrm{Cons}})$ *and* Σ*-sentence* φ*:*
$A' \models_{\Sigma'_{\mathrm{Cons}}} \sigma(\varphi)$ *if and only if* $A'|_{\sigma_{\mathrm{Cons}}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$.

Proof. $A' \models_{\Sigma'_{\mathrm{Cons}}} \sigma(\varphi)$ iff, by Theorem 33, $\mathcal{BB}_{\Sigma'_{\mathrm{Cons}}}(A') \models_{\Sigma'} \sigma(\varphi)$ iff (since the satisfaction condition holds in the standard first-order logic institution) $\mathcal{BB}_{\Sigma'_{\mathrm{Cons}}}(A')|_\sigma \models_\Sigma \varphi$ iff, by Corollary 40, $\mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(A'|_{\sigma_{\mathrm{Cons}}}) \models_\Sigma \varphi$ iff, by Theorem 33, $A'|_{\sigma_{\mathrm{Cons}}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$. [12] □

We have now introduced all ingredients that constitute the *constructor-based logic institution.* The category of signatures is the category of constructor-based signatures and constructor-based signature morphisms, for each constructor-based signature $\Sigma_{\mathrm{Cons}} = (\Sigma, OP_{\mathrm{Cons}})$ the sentences are finitary first-order Σ-sentences, the model functor assigns to each constructor-based signature Σ_{Cons} the category $\mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$ of Σ_{Cons}-algebras and Σ_{Cons}-morphisms, and the satisfaction relation is the constructor-based satisfaction relation.

As in the observational case, the following remark discusses briefly some properties and further aspects of the constructor-based logic institution.

Remark 42

(1) Constructor-based logic satisfies the amalgamation property. This can again be proved by applying the construction of amalgamations for standard algebras. That the amalgamated union of two constructor-based algebras is a constructor-based algebra is a consequence of Lemma 37.

(2) If we allowed infinitary Σ*-sentences and restricted to injective signature morphisms then the interpolation property would be satisfied as well. The proof of this fact relies on the infinitary axiomatization of reachability presented in Section 5.2 and on Corollary 52 and Theorem 53.*

(3) Of course, we can also build structured constructor-based specifications by using the specification-building operators of [37] or [6] and one can compute normal forms according to [6].

(4) The functors $\mathcal{BB}_{\Sigma_{\mathrm{Obs}}}$ *associated to constructor-based signatures* Σ_{Obs} *can be extended to an institution encoding (see [39]) which maps the institution of constructor-based logic to the institution of standard first-order*

[12] Note that this proof is totally analogous to the proof of Theorem 20 for the observational satisfaction condition.

logic. A concrete discussion on how this institution encoding works is outside the scope of this paper.

4 A First Comparison

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

Observability	**Reachability**
observational signature $\Sigma_{\text{Obs}} = (\Sigma, OP_{\text{Obs}})$	*constructor-based signature* $\Sigma_{\text{Cons}} = (\Sigma, OP_{\text{Cons}})$
state sorts S_{State} *and* *observable sorts* S_{Obs}	*constrained sorts* S_{Cons} *and* *loose sorts* S_{Loose}
observable contexts $\mathcal{C}(\Sigma_{\text{Obs}})$	*constructor terms* $\mathcal{T}(\Sigma_{\text{Cons}})$
observational Σ_{Obs}*-equality* $\approx_{\Sigma_{\text{Obs}},A} \subseteq A \times A$	Σ_{Cons}*-generated part* $Gen_{\Sigma_{\text{Cons}}}(A) \subseteq A$
fully abstract algebra	*reachable algebra*
observational algebra $\approx_{\Sigma_{\text{Obs}},A}$ is a Σ-congruence	*constructor-based algebra* $Gen_{\Sigma_{\text{Cons}}}(A)$ is a Σ-subalgebra of A
observational black box functor $\mathcal{BB}_{\Sigma_{\text{Obs}}} : \text{Alg}_{\text{Obs}}(\Sigma_{\text{Obs}}) \to \text{Alg}(\Sigma)$	*constructor-based black box functor* $\mathcal{BB}_{\Sigma_{\text{Cons}}} : \text{Alg}_{\text{Cons}}(\Sigma_{\text{Cons}}) \to \text{Alg}(\Sigma)$
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 $\alpha : X \to Gen_{\Sigma_{\text{Cons}}}(A)$
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*

Table 1
Comparing Observability and Reachability

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 observable contexts and, on the other hand, leads to the definition of constructor terms.

In both cases, the syntactic notions induce a semantic relation on any Σ-algebra A. In the observational case we obtain a binary relation $\approx_{\Sigma_{\mathrm{Obs}},A}$, called observational equality, and in the constructor case we obtain a unary relation $Gen_{\Sigma_{\mathrm{Cons}}}(A)$, called Σ_{Cons}-generated 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 Σ_{Cons}-generated part is a Σ-subalgebra thus leading to the notion of a constructor-based algebra. In each case we can construct a black box functor which, in the observational approach, identifies indistinguishable elements of an algebra and, in the constructor-based approach, abstracts from junk values.

In order to satisfy our working hypothesis of the Introduction, we have relaxed the standard satisfaction relation such that, in the observational case, equality is considered as observational equality and, in the constructor case, variables are interpreted only by values of the constructor generated 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 shown 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 coinductive style) by a (complete) case distinction w.r.t. the given observers. A general schema for observer complete definitions is studied in [7]. As a standard example, consider again streams of booleans with observers $head : stream \to bool$ and $tail : stream \to stream$, and consider an observational specification of an alternating merge function $merge : stream \times stream \to stream$ and of a reverse function $rev : stream \to stream$ that reverses each bit of the stream. Both functions are specified by the following complete case distinctions w.r.t. the observers $head$ and $tail$ as follows.

$head(merge(s1, s2)) = head(s1)$
$tail(merge(s1, s2)) = merge(s2, tail(s1))$
$head(rev(s)) = not(head(s))$
$tail(rev(s)) = rev(tail(s))$

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 coalgebras this analogy is described in [24].

5 Logical Consequences of Specifications and Corresponding Proof Systems

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 $\mathrm{Mod}_X(\mathrm{SP}_X)$.[13] 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 $\mathrm{SP}_X \models_{\Sigma_X} \varphi$ which express that $\mathrm{Mod}_X(\mathrm{SP}_X) \models_{\Sigma_X} \varphi$ holds, and in corresponding proof systems.

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.

Formally, the black box semantics of a specification SP_X will be defined as the class $\mathcal{BB}_{\Sigma_X}(\mathrm{Mod}_X(\mathrm{SP}_X))$ obtained by applying the black box functors (of Definitions 10 and 31) to the model class of the given specification.

5.1 Black Box Semantics and Proof Systems for Observational Specifications

Definition 43 (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* $[\![\mathrm{SP}_{\mathrm{Obs}}]\!] \overset{\mathrm{def}}{=} \mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(\mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}))$.

As a consequence of Theorem 12 we obtain the following fact.

Corollary 44 (Observational consequences) *Let* $\mathrm{SP}_{\mathrm{Obs}}$ *be an observational specification with signature* Σ_{Obs} *and let* φ *be a* Σ*-formula. Then:*

[13] 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.

$\mathrm{SP}_{\mathrm{Obs}} \models_{\Sigma_{\mathrm{Obs}}} \varphi$ *if and only if* $[\![\mathrm{SP}_{\mathrm{Obs}}]\!] \models \varphi$.

This fact shows the adequacy of the black box semantics in the observational case. In this case the black box semantics can be characterized as follows.

Theorem 45 (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 we have:* $[\![\mathrm{SP}_{\mathrm{Obs}}]\!] = \{\Sigma\text{−algebra } A \mid A \models \mathrm{Ax} \text{ and } A \text{ is fully abstract w.r.t. } \approx_{\Sigma_{\mathrm{Obs}},A}\}$.

Proof. Let A be a Σ-algebra, where Σ is the standard signature underlying Σ_{Obs}.
$\subseteq$: Assume $A \in [\![\mathrm{SP}_{\mathrm{Obs}}]\!]$. Then $A = \mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(B)$ for some $B \in \mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}})$. Hence A is fully abstract and, since $B \models_{\Sigma_{\mathrm{Obs}}} \mathrm{Ax}$, by Theorem 12, $A \models \mathrm{Ax}$.
$\supseteq$: Assume $A \models \mathrm{Ax}$ and A is fully abstract. Then obviously $A \models_{\Sigma_{\mathrm{Obs}}} \mathrm{Ax}$ as well, and A can be considered as a Σ_{Obs}-algebra, hence $A \in \mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}})$. Since A is fully abstract, $A = \mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(A)$, hence $A \in [\![\mathrm{SP}_{\mathrm{Obs}}]\!]$. □

We have shown in Corollary 44 how to relate the observational consequences of an observational specification to the consequences in standard first-order logic of the black box semantics of the given specification. The next step is to find an adequate axiomatization of the black box semantics in order to be able to define sound and complete proof systems. According to Theorem 45 this amounts to finding an axiomatic characterization of full abstractness. The next definition provides the required axiomatization which, however, can only be stated by using *infinitary* first-order formulas.

Definition 46 (Fully abstract axiom) *Let* Σ_{Obs} *be an observational signature with underlying signature* Σ. *The* fully abstract axiom *associated to* Σ_{Obs} *is the sentence* $\mathrm{FA}(\Sigma_{\mathrm{Obs}})$ *defined by:*

$$\mathrm{FA}(\Sigma_{\mathrm{Obs}}) \overset{\mathrm{def}}{=} \bigwedge_{s \in S_{\mathrm{State}}} \mathrm{FA}(\Sigma_{\mathrm{Obs}})_s$$

where for each state sort $s \in S_{\mathrm{State}}$, $\mathrm{FA}(\Sigma_{\mathrm{Obs}})_s$ *is defined by:*

$$\mathrm{FA}(\Sigma_{\mathrm{Obs}})_s \overset{\mathrm{def}}{=} \forall x, y{:}s. \left(\bigwedge_{s' \in S_{\mathrm{Obs}},\ c \in \mathcal{C}(\Sigma_{\mathrm{Obs}})_{s \to s'}} \forall \mathrm{Var}(c).\ c[x] = c[y] \right) \Rightarrow x = y\,.$$ [14]

Lemma 47 *Let* Σ_{Obs} *be an observational signature with underlying signature* Σ. *A* Σ*-algebra* A *is fully abstract w.r.t.* Σ_{Obs} *if and only if* $A \models \mathrm{FA}(\Sigma_{\mathrm{Obs}})$.

Now let Π_{IFOLEq} be a sound and complete proof system for infinitary first-order logic with equality (see [26]). From Π_{IFOLEq} we obtain a sound and complete proof system for observational logic by adding to it, as an extra axiom, $\mathrm{FA}(\Sigma_{\mathrm{Obs}})$.

[14] $\forall \mathrm{Var}(c)$ is an abbreviation for $\forall x_1{:}s_1 \ldots \forall x_n{:}s_n$, where $x_1, \ldots, x_n$ are the variables (of sort $s_1, \ldots, s_n$) of the context c, apart from its context variable z_s.

Theorem 48 *For any observational signature* Σ_{Obs}*, let* $\Pi_{\text{Obs}} \overset{\text{def}}{=} \Pi_{\textbf{IFOLEq}} \cup \text{FA}(\Sigma_{\text{Obs}})$*. Then for any basic observational specification* $\text{SP}_{\text{Obs}} = \langle \Sigma_{\text{Obs}}, \text{Ax} \rangle$ *and any* Σ*-formula* φ*, we have:*
$\text{SP}_{\text{Obs}} \models_{\Sigma_{\text{Obs}}} \varphi$ *if and only if* $\text{Ax} \vdash_{\Pi_{\text{Obs}}} \varphi$*.*

Proof. $\text{SP}_{\text{Obs}} \models_{\Sigma_{\text{Obs}}} \varphi$ iff, by Corollary 44, $[\![\text{SP}_{\text{Obs}}]\!] \models \varphi$ iff, by Theorem 45, $\{\Sigma\text{−algebra } A \mid A \models \text{Ax and } A \text{ is fully abstract w.r.t. } \approx_{\Sigma_{\text{Obs}},A}\} \models \varphi$ iff, by Lemma 47, $\text{Ax} \cup \text{FA}(\Sigma_{\text{Obs}}) \models \varphi$ iff, by soundness and completeness of $\Pi_{\textbf{IFOLEq}}$, $\text{Ax} \cup \text{FA}(\Sigma_{\text{Obs}}) \vdash_{\Pi_{\textbf{IFOLEq}}} \varphi$ iff, by definition of Π_{Obs}, $\text{Ax} \vdash_{\Pi_{\text{Obs}}} \varphi$. □

The difficulty with the above proof system is that it uses infinitary formulas (and also infinitary proof rules of $\Pi_{\textbf{IFOLEq}}$). An alternative is to restrict to finitary formulas and to use only a particular set of infinitary proof rules (see the discussion in [6]). The idea now is, instead of "capturing" full abstractness by the infinitary axiom $\text{FA}(\Sigma_{\text{Obs}})$, to "capture" it by specialized infinitary proof rules called infinitary coinduction. These infinitary rules are necessary to ensure completeness. A further step will then be to implement (in a theorem prover) these infinitary rules by finite (but incomplete) coinduction schemes, as discussed at the end of this section.

Definition 49 (Infinitary coinduction) *Let* Σ_{Obs} *be an observational signature with underlying signature* Σ*. The* infinitary coinduction rule $\text{iCI}(\Sigma_{\text{Obs}})$ *associated to* Σ_{Obs} *is defined by* $\text{iCI}(\Sigma_{\text{Obs}}) \overset{\text{def}}{=} \{\text{iCI}(\Sigma_{\text{Obs}})_s \mid s \in S_{\text{State}}\}$ *where for each state sort* $s \in S_{\text{State}}$*,* $\text{iCI}(\Sigma_{\text{Obs}})_s$ *is defined by:*

$$\text{iCI}(\Sigma_{\text{Obs}})_s \qquad \frac{\varphi \Rightarrow \forall \text{Var}(c).\ c[x] = c[y] \qquad \begin{array}{l} \text{for all observable sorts } s' \in S_{\text{Obs}} \\ \text{and all contexts } c \in \mathcal{C}(\Sigma_{\text{Obs}})_{s \to s'} \end{array}}{\varphi \Rightarrow x = y}$$

where φ *denotes an arbitrary first-order* Σ*-formula.*

Now let $\Pi_{\textbf{FOLEq}}$ be a sound and complete proof system for finitary first-order logic with equality. From the finitary proof system $\Pi_{\textbf{FOLEq}}$ we obtain a sound and complete (semi-formal) proof system for observational logic by adding to it the extra infinitary proof rules $\text{iCI}(\Sigma_{\text{Obs}})$.

Theorem 50 *For any observational signature* Σ_{Obs}*, let* $\Pi^2_{\text{Obs}} \overset{\text{def}}{=} \Pi_{\textbf{FOLEq}} \cup \text{iCI}(\Sigma_{\text{Obs}})$*. Then for any basic observational specification* $\text{SP}_{\text{Obs}} = \langle \Sigma_{\text{Obs}}, \text{Ax} \rangle$ *and any* Σ*-formula* φ*, we have:*
$\text{SP}_{\text{Obs}} \models_{\Sigma_{\text{Obs}}} \varphi$ *if and only if* $\text{Ax} \vdash_{\Pi^2_{\text{Obs}}} \varphi$*.*

Proof. Again, as in the proof of Theorem 48, $\text{SP}_{\text{Obs}} \models_{\Sigma_{\text{Obs}}} \varphi$ iff $\text{Ax} \cup \text{FA}(\Sigma_{\text{Obs}}) \models \varphi$. Hence, it is sufficient to show that the latter is equivalent to $\text{Ax} \vdash_{\Pi^2_{\text{Obs}}} \varphi$.

The soundness, i.e., $\mathrm{Ax} \vdash_{\Pi^2_{\mathrm{Obs}}} \varphi$ implies $\mathrm{Ax} \cup \mathrm{FA}(\Sigma_{\mathrm{Obs}}) \models \varphi$, is obvious and can be proved by induction on the length of the derivation. The completeness, i.e., $\mathrm{Ax} \cup \mathrm{FA}(\Sigma_{\mathrm{Obs}}) \models \varphi$ implies $\mathrm{Ax} \vdash_{\Pi^2_{\mathrm{Obs}}} \varphi$, has been shown in [20] for the case where *all* operations with non-observable arguments are observers. The completeness proof given in [20] relies on the omitting types theorem (see [12]). A generalization of this proof to an arbitrary set of observers is straightforward. □

A last step is then to implement (in a theorem prover) the above infinitary rules by finite (but incomplete) adequate coinduction schemes. In practice, for proving the infinitely many hypotheses $\varphi \Rightarrow \forall \mathrm{Var}(c).\ c[x] = c[y]$ of the rule $\mathrm{iCI}(\Sigma_{\mathrm{Obs}})_s$, one would use a coinduction scheme according to the coinductive definition of the contexts $\mathcal{C}(\Sigma_{\mathrm{Obs}})_{s \to s'}$ (see Definition 2).

For instance, to prove that $\forall s{:}stream.\ rev(rev(s)) = s$ is an observational consequence of the observational specification of streams, one would have to prove:
$\forall s{:}stream.\ head(rev(rev(s))) = head(s)$ and
$(\forall s{:}stream.\ c[rev(rev(s))] = c[s]) \Rightarrow$
$(\forall s{:}stream.\ c[tail(rev(rev(s)))] = c[tail(s)])$
where c denotes an arbitrary observable context.

Indeed both proof obligations can easily be discharged due to the coinductive definition of the operation *rev*.

5.2 Black Box Semantics and Proof Systems for Constructor-Based Specifications

Definition 51 (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* $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] \overset{\mathrm{def}}{=} \mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(\mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}}))$.

As a consequence of Theorem 33 we obtain the following fact.

Corollary 52 (Inductive consequences) *Let* $\mathrm{SP}_{\mathrm{Cons}}$ *be a constructor-based specification with signature* Σ_{Cons} *and let* φ *be a* Σ*-formula. Then:*
$\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *if and only if* $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] \models \varphi$.

This fact shows the adequacy of the black box semantics in the constructor-based case. Again, we can provide also in this case a characterization of the black box semantics.

Theorem 53 (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:*

$[\![\mathrm{SP}_{\mathrm{Cons}}]\!] = \{\Sigma\text{−algebra } A \mid A \models \mathrm{Ax} \text{ and } A \text{ is reachable w.r.t. } \Sigma_{\mathrm{Cons}}\}$.

Proof. Let A be a Σ-algebra, where Σ is the standard signature underlying Σ_{Cons}.
$\subseteq$: Assume $A \in [\![\mathrm{SP}_{\mathrm{Cons}}]\!]$. Then $A = \mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(B)$ for some $B \in \mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}})$. Hence A is reachable and, since $B \models_{\Sigma_{\mathrm{Cons}}} \mathrm{Ax}$, by Theorem 33, $A \models \mathrm{Ax}$.
$\supseteq$: Assume $A \models \mathrm{Ax}$ and A is reachable. Then obviously $A \models_{\Sigma_{\mathrm{Cons}}} \mathrm{Ax}$ as well, and A can be considered as a Σ_{Cons}-algebra, hence $A \in \mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}})$. Since A is reachable, $A = \mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(A)$, hence $A \in [\![\mathrm{SP}_{\mathrm{Cons}}]\!]$. □

We have shown in Corollary 52 how to relate the inductive consequences of a constructor-based specification to the consequences in standard first-order logic of the black box semantics of the given specification. Again, the next step is to find an adequate axiomatization of the black box semantics in order to be able to define sound and complete proof systems. According to Theorem 53 this amounts to finding an axiomatic characterization of reachability which is provided in the next definition (again using *infinitary* first-order formulas).

Definition 54 (Reachability axiom) *Let Σ_{Cons} be a constructor-based signature with underlying signature Σ. The* reachability axiom *associated to Σ_{Cons} is the sentence $\mathrm{REACH}(\Sigma_{\mathrm{Cons}})$ defined by:*
$$\mathrm{REACH}(\Sigma_{\mathrm{Cons}}) \stackrel{\mathrm{def}}{=} \bigwedge_{s \in S_{\mathrm{Cons}}} \mathrm{REACH}(\Sigma_{\mathrm{Cons}})_s$$
where for each constrained sort $s \in S_{\mathrm{Cons}}$, $\mathrm{REACH}(\Sigma_{\mathrm{Cons}})_s$ is defined by:
$$\mathrm{REACH}(\Sigma_{\mathrm{Cons}})_s \stackrel{\mathrm{def}}{=} \forall x{:}s. \bigvee_{t \in \mathcal{T}(\Sigma_{\mathrm{Cons}})_s} \exists \mathrm{Var}(t).\ x = t\,.^{15}$$

Lemma 55 *Let Σ_{Cons} be a constructor-based signature with underlying signature Σ. A Σ-algebra A is reachable w.r.t. Σ_{Cons} if and only if $A \models \mathrm{REACH}(\Sigma_{\mathrm{Cons}})$.*

To obtain a sound and complete proof system for constructor-based logic we can now add to the proof system Π_{IFOLEq} for infinitary first-order logic the extra axiom $\mathrm{REACH}(\Sigma_{\mathrm{Cons}})$.

Theorem 56 *For any constructor-based signature Σ_{Cons}, let $\Pi_{\mathrm{Cons}} \stackrel{\mathrm{def}}{=} \Pi_{\mathrm{IFOLEq}} \cup \mathrm{REACH}(\Sigma_{\mathrm{Cons}})$. Then for any basic constructor-based specification $\mathrm{SP}_{\mathrm{Cons}} = \langle \Sigma_{\mathrm{Cons}}, \mathrm{Ax}\rangle$ and any Σ-formula φ, we have:*
$\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ *if and only if* $\mathrm{Ax} \vdash_{\Pi_{\mathrm{Cons}}} \varphi$.

Proof. $\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ iff, by Corollary 52, $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] \models \varphi$ iff, by Theorem 53, $\{\Sigma\text{−algebra } A \mid A \models \mathrm{Ax} \text{ and } A \text{ is reachable w.r.t. } \Sigma_{\mathrm{Cons}}\} \models \varphi$ iff, by Lemma 55, $\mathrm{Ax} \cup \mathrm{REACH}(\Sigma_{\mathrm{Cons}}) \models \varphi$ iff, by soundness and completeness of Π_{IFOLEq}, $\mathrm{Ax} \cup \mathrm{REACH}(\Sigma_{\mathrm{Cons}}) \vdash_{\Pi_{\mathrm{IFOLEq}}} \varphi$ iff, by definition of Π_{Cons},

[15] $\exists \mathrm{Var}(t)$ is an abbreviation for $\exists x_1{:}s_1. \dots \exists x_n{:}s_n$, where $x_1, \dots, x_n$ are the variables (of sort $s_1, \dots, s_n$) of the constructor term t.

$\mathrm{Ax} \vdash_{\Pi_{\mathrm{Cons}}} \varphi$. □

The above proof system uses again infinitary formulas. To restrict to finitary formulas and to use only a particular set of infinitary proof rules the idea is now, instead of expressing reachability by the infinitary axiom $\mathrm{REACH}(\Sigma_{\mathrm{Cons}})$, to "capture" it by infinitary induction rules (which are necessary to ensure completeness).

Definition 57 (Infinitary induction) *Let Σ_{Cons} be a constructor-based signature with underlying signature Σ. The* infinitary induction rule $\mathrm{iI}(\Sigma_{\mathrm{Cons}})$ *associated to Σ_{Cons} is defined by $\mathrm{iI}(\Sigma_{\mathrm{Cons}}) \overset{\mathrm{def}}{=} \{\mathrm{iI}(\Sigma_{\mathrm{Cons}})_s \mid s \in S_{\mathrm{Cons}}\}$ where for each constrained sort $s \in S_{\mathrm{Cons}}$, $\mathrm{iI}(\Sigma_{\mathrm{Cons}})_s$ is defined by:*

$$\mathrm{iI}(\Sigma_{\mathrm{Cons}})_s \qquad \frac{\varphi[t/x] \text{ for all constructor terms } t \in \mathcal{T}(\Sigma_{\mathrm{Cons}})_s}{\forall x{:}s.\ \varphi}$$

where φ denotes an arbitrary first-order Σ-formula (with at least one free variable x of sort s).

From the finitary proof system Π_{FOLEq} for first-order logic we obtain a sound and complete (semi-formal) proof system for constructor-based logic by adding to it the extra infinitary proof rules $\mathrm{iI}(\Sigma_{\mathrm{Cons}})$.

Theorem 58 *For any constructor-based signature Σ_{Cons}, let $\Pi^2_{\mathrm{Cons}} \overset{\mathrm{def}}{=} \Pi_{\mathrm{FOLEq}} \cup \mathrm{iI}(\Sigma_{\mathrm{Cons}})$. Then for any basic constructor-based specification $\mathrm{SP}_{\mathrm{Cons}} = \langle \Sigma_{\mathrm{Cons}}, \mathrm{Ax}\rangle$ and any Σ-formula φ, we have:*
$\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ if and only if $\mathrm{Ax} \vdash_{\Pi^2_{\mathrm{Cons}}} \varphi$.

Proof. Again, as in the proof of Theorem 56, $\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ iff $\mathrm{Ax} \cup \mathrm{REACH}(\Sigma_{\mathrm{Cons}}) \models \varphi$. The latter is equivalent to $\mathrm{SP}_{reach} \models \varphi$ where

$$\mathrm{SP}_{reach} \overset{\mathrm{def}}{=} \textit{reach } \langle \Sigma, \mathrm{Ax}\rangle \textit{ with } OP_{\mathrm{Cons}}$$

according to the definition of specifications with reachability operators in [21]. For those specifications it has been shown in [21] (Corollary 3.18) that our proof system with the infinitary induction rules is sound and complete. □

In practice, for proving the infinitely many hypotheses $\varphi[t/x]$ of the rule $\mathrm{iI}(\Sigma_{\mathrm{Cons}})_s$, one would use an induction scheme like structural induction with respect to the constructor terms $\mathcal{T}(\Sigma_{\mathrm{Cons}})_s$. For instance, to prove a property $\forall x{:}nat.\ \varphi$ on natural numbers, it is enough to prove $\varphi[zero/x]$ and $\forall x{:}nat.\ \varphi \Rightarrow \varphi[succ(x)/x]$.

5.3 A Further Comparison

Taking into account the results of Sections 5.1 and 5.2, Table 1 of Section 4 can now be extended as shown in Table 2 below.

Observability	**Reachability**
black box semantics $[\![\mathrm{SP}_{\mathrm{Obs}}]\!] \overset{\mathrm{def}}{=} \mathcal{BB}_{\Sigma_{\mathrm{Obs}}}(\mathrm{Mod}_{\mathrm{Obs}}(\mathrm{SP}_{\mathrm{Obs}}))$	*black box semantics* $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] \overset{\mathrm{def}}{=} \mathcal{BB}_{\Sigma_{\mathrm{Cons}}}(\mathrm{Mod}_{\mathrm{Cons}}(\mathrm{SP}_{\mathrm{Cons}}))$
observational consequences $\mathrm{SP}_{\mathrm{Obs}} \models_{\Sigma_{\mathrm{Obs}}} \varphi$ iff $[\![\mathrm{SP}_{\mathrm{Obs}}]\!] \models \varphi$	*inductive consequences* $\mathrm{SP}_{\mathrm{Cons}} \models_{\Sigma_{\mathrm{Cons}}} \varphi$ iff $[\![\mathrm{SP}_{\mathrm{Cons}}]\!] \models \varphi$
black box semantics relies on fully abstract algebras	*black box semantics relies on reachable algebras*
fully abstract axiom $\mathrm{FA}(\Sigma_{\mathrm{Obs}})$	*reachability axiom* $\mathrm{REACH}(\Sigma_{\mathrm{Cons}})$
infinitary proof system Π_{Obs}	*infinitary proof system* Π_{Cons}
infinitary coinduction rules $\mathrm{iCI}(\Sigma_{\mathrm{Obs}})$	*infinitary induction rules* $\mathrm{iI}(\Sigma_{\mathrm{Cons}})$
semi-formal proof system Π^2_{Obs}	*semi-formal proof system* Π^2_{Cons}
coinduction proof scheme	*induction proof scheme*

Table 2
Comparing Observability and Reachability (cont.)

6 Formalizing 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 which is provided by category theory.

6.1 Categorical Duality

We briefly review categorical duality, for more details see, e.g., [30,1]. 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 formalized 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^{op} is initial in $\mathcal{C}^{\mathrm{op}}$; a morphism $f \in \mathcal{C}(A, B)$ is co-mono (usually called epi) iff f^{op} is mono; $C = A + B$ is a co-product (disjoint union in the case of sets) 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. And a functor F is left adjoint to G iff F^{op} is right adjoint to G^{op}.

The notions of quotient/embedding and kernel/image can be recognized as duals with the help of factorization systems. A *factorization system* $(\mathcal{E}, \mathcal{M})$ for $\mathcal{C}$ consists of classes $\mathcal{E}, \mathcal{M}$ of arrows of $\mathcal{C}$ satisfying (1) both $\mathcal{E}$ and $\mathcal{M}$ contain all isomorphisms and are closed under composition, (2) every arrow f in $\mathcal{C}$ has a factorization $f = m \circ e$ with $e \in \mathcal{E}$, $m \in \mathcal{M}$, and (3) this factorization is essentially unique. [16] We call the arrows in $\mathcal{E}$ and $\mathcal{M}$ *quotients* and *embeddings*, respectively, and, given a factorization $f = m \circ e$, we call e the *kernel* of f and m the *image* of f. Note that $(\mathcal{M}, \mathcal{E})$ is a factorization system for $\mathcal{C}^{\mathrm{op}}$. [17]

6.2 Algebras and Coalgebras

The categorical description of signatures, observational algebras, and constructor-based algebras relies on the notions of functor, coalgebra for a functor, and algebra for a functor, respectively.

For the remainder of Section 6 we assume a category $\mathcal{X}$ with a factorization system called the base category. $\mathcal{X}$ will be the category of the carriers of our models, usually Set (single-sorted) or Set^S (S-sorted). We first recall the definition of *algebra* and *coalgebra for a functor* (cf. [24] for more information). Let $\Omega, \Xi : \mathcal{X} \to \mathcal{X}$ be functors. Then an Ω-algebra is an arrow $\omega : \Omega X \to X$ in $\mathcal{X}$, a Ξ-coalgebra is an arrow $\xi : X \to \Xi X$ in $\mathcal{X}$. An arrow $f : X \to Y$ in $\mathcal{X}$ is an Ω-algebra morphism $f : \omega \to \omega'$ if the left-hand diagram below commutes and a Ξ-coalgebra morphism $f : \xi \to \xi'$ if the right-hand diagram

[16] That is, if $f = m \circ e = m' \circ e'$ are two $(\mathcal{E}, \mathcal{M})$-factorizations then there is a unique isomorphism h such that $f = m' \circ h \circ e$.

[17] See [1] for more information on factorization systems and e.g. [38] for a typical application to algebraic specifications.

below commutes.

$$\begin{tikzcd} \Omega X \arrow[r, "\omega"] \arrow[d, "\Omega f"'] & X \arrow[d, "f"] & X \arrow[r, "\xi"] \arrow[d, "f"'] & \Xi X \arrow[d, "\Xi f"] \\ \Omega Y \arrow[r, "\omega'"] & Y & Y \arrow[r, "\xi'"] & \Xi Y \end{tikzcd} \qquad (1)$$

Together with their respective morphisms Ω-algebras form a category $\mathrm{Alg}(\Omega)$ and Ξ-coalgebras a category $\mathrm{Coalg}(\Xi)$. Coalgebras are dual to algebras, that is, $\mathrm{Coalg}(\Xi)^{\mathrm{op}} \simeq \mathrm{Alg}(\Xi^{\mathrm{op}})$. Note that the functors Ω, Ξ play the role of signatures as explained in the following remarks.

Remark 59 *The concept of an Ω-algebra includes algebras in the usual sense. For instance, with $\mathcal{X} = \mathsf{Set}$ and $\Omega X = 1 + X + X \times X$, 1 denoting a one-element set, an algebra $[f_0, f_1, f_2] : 1 + X + X \times X \longrightarrow X$ is given by a constant $f_0 : 1 \to X$, a unary operation $f_1 : X \to X$, and a binary operation $f_2 : X \times X \to X$. Generally, for a single-sorted signature with a set OP of operation symbols f with arities $ar(f) \in \mathbb{N}$ we let $\Omega X = \coprod_{f \in OP} X^{ar(f)}$. For an S-sorted signature (S, OP), the functor $\Omega : \mathsf{Set}^S \to \mathsf{Set}^S$ has components, for each $s \in S$,*

$$(\Omega X)_s = \coprod_{op: s_1, \ldots, s_n \to s} X_{s_1} \times \ldots \times X_{s_n},$$

where X denotes an element of Set^S with components X_t, $t \in S$, and op ranges over all operation symbols in OP with result sort s. Finally, let us mention that it is natural to incorporate given parameter sets into the functors. For example, to describe lists over a given set of elements D we can use the single-sorted functor $\Omega X = 1 + D \times X$ giving rise to algebras $[nil, cons] : 1 + D \times X \to X$.

Remark 60 *The concept of a Ξ-coalgebra includes algebras with operations having precisely one argument of a state sort. For instance, fixing two sets O and I, an automaton with output $o : X \to O$ and transition function $\delta : X \times I \to X$ can be considered as a coalgebra $\langle o, \delta \rangle : X \longrightarrow O \times X^I$ for the functor $\Xi : \mathsf{Set} \to \mathsf{Set}$ given by $\Xi X = O \times X^I$. Generally, let (S, OP) be a signature with the properties that (i) the sorts are divided into two disjoint parts $S = S_{\mathrm{State}} \cup S_{\mathrm{Param}}$ called state sorts and parameter sorts and that (ii) an operation $op : s_1, \ldots, s_n \to s$ is in OP only if precisely one of the argument sorts s_i is in S_{State}. Then the functor $\Xi : \mathsf{Set}^{S_{\mathrm{State}}} \to \mathsf{Set}^{S_{\mathrm{State}}}$ has components, for each $s \in S_{\mathrm{State}}$,*

$$(\Xi X)_s = \prod_{op: s_1, \ldots, s_{i-1}, s, s_{i+1}, \ldots, s_n \to s'} Y_{s'}{}^{P_{s_1} \times \ldots \times P_{s_{i-1}} \times P_{s_{i+1}} \times \ldots \times P_{s_n}}$$

where X denotes an element of $\mathsf{Set}^{S_{\mathrm{State}}}$ with components X_t, $t \in S_{\mathrm{State}}$, and op ranges over all operation symbols in OP that have an argument of sort s, and P_t denotes the set interpreting the parameter sort $t \in S_{\mathrm{Param}}$, and $Y_{s'}$ is $X_{s'}$ for $s' \in S_{\mathrm{State}}$ and $P_{s'}$ for $s' \in S_{\mathrm{Param}}$. Finally, let us mention that the *functors Ξ described above have been characterized in [29] as those functors that, making the dependency on the parameters explicit, have a left adjoint. The relationship of coalgebras and hidden algebra [15] is discussed e.g. in [13] and [35].*

6.3 The Duality Principle for Observability and Reachability

The essence of our categorical description of observational and constructor-based signatures and models is the following. In the case of observability, a set of observer symbols is represented by a functor $\mathcal{O} : \mathcal{X} \to \mathcal{X}$, each $X \in \mathcal{X}$ is considered as an interpretation of the state sorts and each coalgebra $X \xrightarrow{o} \mathcal{O}X$ is considered as an interpretation of the observer operations. In the reachability case, a set of constructor symbols is represented by a functor $\mathcal{R} : \mathcal{X} \to \mathcal{X}$, each $X \in \mathcal{X}$ is considered as an interpretation of the constrained sorts and each algebra $\mathcal{R}X \xrightarrow{\rho} X$ is considered as an interpretation of the constructors.

An observational signature as defined in Section 2 specifies observer and non-observer operations. As described above the observers are represented by a functor $\mathcal{O} : \mathcal{X} \to \mathcal{X}$ and their interpretation is modeled as a coalgebra $X \xrightarrow{o} \mathcal{O}X$. In the categorical framework, the non-observer operations may be interpreted as algebras $\omega : \Omega X \to X$ or as coalgebras $\xi : X \to \Xi X$ depending on their type (as discussed in Remark 63 below). Hence, in general, an observational signature is represented by one functor $\mathcal{O} : \mathcal{X} \to \mathcal{X}$ corresponding to the observers and by two functors $\Omega, \Xi : \mathcal{X} \to \mathcal{X}$ corresponding to the non-observer operations.

Definition 61 (Observational signature) *An* observational signature *$(\Omega; \mathcal{O}, \Xi)$ over $\mathcal{X}$ consists of functors $\Omega, \mathcal{O}, \Xi : \mathcal{X} \to \mathcal{X}$ such that a final $\mathcal{O}$-coalgebra $\zeta : Z \to \mathcal{O}Z$ exists.*[18]

A model for the observational signature $(\Omega; \mathcal{O}, \Xi)$ is a triple (ω, o, ξ) with $\omega \in \mathrm{Alg}(\Omega)$, $o \in \mathrm{Coalg}(\mathcal{O})$, $\xi \in \mathrm{Coalg}(\Xi)$. A morphism $f : (\omega, o, \xi) \to (\omega', o', \xi')$ is an arrow f that is, at the same time, an algebra-morphism $\omega \to \omega'$, a coalgebra-morphism $o \to o'$, and a coalgebra-morphism $\xi \to \xi'$ (compare the diagrams (1)). The category of $(\Omega; \mathcal{O}, \Xi)$-models is denoted by $\mathsf{Mod}(\Omega; \mathcal{O}, \Xi)$.

Example 62 *The observational signature for streams can be represented by*

[18] Final coalgebras allow a convenient definition of observational equality (Definition 64), but it is possible to use weaker conditions which still guarantee a well-behaved notion of observational equality. For example, it is enough to require that $\mathcal{X}$ has cointersections (see [27], Section 1.2.3, for details), a condition which is satisfied by Set^S.

the following functors:

- $\Omega X = X + X \times X$ *corresponding to the operations* $[rev, merge] : X + X \times X \to X$,
- $\mathcal{O}X = \mathbb{B} \times X$ *corresponding to the observers* $\langle head, tail \rangle : X \to \mathbb{B} \times X$,

and, assuming a derived observer $nth : X \times \mathbb{N} \to \mathbb{B}$ *to determine the n-th successor of* x,

- $\Xi X = \mathbb{B}^{\mathbb{N}}$ *corresponding to the operation* $X \to \mathbb{B}^{\mathbb{N}}$ *obtained by currying nth.*

Remark 63 *In contrast to the definition of an observational signature in Section 2, Definition 61 does not allow observers with more than one argument of a state sort. More precisely, note first that* $X \in \mathcal{X}$ *interprets the state sorts and that observable sorts are interpreted by given parameters. Then, with* $\mathcal{X} = \mathsf{Set}^{S_{\text{State}}}$, *only operations of the following type can be modeled: observer operations of type (1)* $A \times X_s \to Y$ *and non-observer operations of type (2a)* $A \times X_{s_1} \times \ldots \times X_{s_n} \to X_s$ *and of type (2b)* $A \times X_s \to B$, *where* X_s, $X_{s_1}, \ldots X_{s_n}$ *denote the interpretations of state sorts, A denotes a product of interpretations of observable sorts, B denotes the interpretation of an observable sort and Y denotes the interpretation of an arbitrary sort. Operations of type (1) are considered coalgebraically* $X_s \to Y^A$ *and determine the functor* $\mathcal{O}$ *(see Remark 60 taking* $\mathcal{O}$ *for* Ξ*), operations of type (2a) determine* Ω *(see Remark 59), and operations of type (2b) are modeled coalgebraically via* Ξ *(see Remark 60). Operations of type (2b) can be considered as derived observers.*

The following provides a categorical definition of observational equality by means of coalgebras.

Definition 64 (Observational equality) *Given* $M = (\omega, o, \xi)$ *in* $\mathsf{Mod}(\Omega; \mathcal{O}, \Xi)$, *the* observational equality *of M is the kernel of* $! : X \to Z$ *where* $!$ *is the morphism to the final* $\mathcal{O}$*-coalgebra* $\zeta : Z \to \mathcal{O}Z$; *see the diagram below.*

$$\begin{tikzcd}
X \arrow[r, "o"] \arrow[d, "!"'] & \mathcal{O}X \arrow[d, "\mathcal{O}!"] \\
Z \arrow[r, "\zeta"] & \mathcal{O}Z
\end{tikzcd} \tag{2}$$

Remark 65 *In case of* $\mathcal{X} = \mathsf{Set}^S$, *writing* $(X_s)_{s \in S} \in \mathsf{Set}^S$ *for the carrier of M and* $(!_s)_{s \in S}$ *for* $!$, *we say that* $x, y \in X_s$ *are* observationally equal, *denoted by* $x \approx_M y$, *iff* $!_s(x) = !_s(y)$. *Indeed this definition is adequate since the notion of observational equality considered in Section 2 coincides with the equivalence relation defined by the unique morphism into the final coalgebra; see e.g. [13], Corollary 11.*

The next definition characterizes those models whose non-observer operations do not contribute to distinguish states (in the sense of observational algebras in Section 2). It generalizes the definition of an $(\Omega, \mathcal{O})$-structure in [19,28] in that an additional Ξ-part (for derived observers) is taken into account.

Definition 66 (Observational models) *$(\omega, o, \xi) \in \mathsf{Mod}(\Omega; \mathcal{O}, \Xi)$ is called an* observational model *for the observational signature $(\Omega; \mathcal{O}, \Xi)$ if there are dotted arrows such that the following diagrams commute*

$$\begin{tikzcd} \Omega X \arrow[r, "\omega"] \arrow[d, "\Omega !"'] & X \arrow[d, "!"'] & & X \arrow[r, "\xi"] \arrow[d, "!"] & \Xi X \arrow[d, "\Xi !"] \\ \Omega Z \arrow[r, dotted] & Z & & Z \arrow[r, dotted] & \Xi Z \end{tikzcd} \tag{3}$$

where $!$ is the unique coalgebra morphism $! : o \to \zeta$ into the final $\mathcal{O}$-coalgebra $\zeta : Z \to \mathcal{O}Z$; cf. diagram (2). The full subcategory of observational models is denoted by $\mathsf{Mod}_{\mathsf{Obs}}(\Omega; \mathcal{O}, \Xi)$. A model is fully abstract *if $! : o \to \zeta$ is an embedding (i.e. injective in case $\mathcal{X} = \mathsf{Set}^S$).*

Remark 67

(1) The diagrams express in an abstract way the condition for observational algebras of Definition 5. Indeed, assuming $\mathcal{X} = \mathsf{Set}^S$, both diagrams state that ω and ξ do not allow to distinguish observationally equal states. More precisely, observational equality (as in Remark 65) is a congruence for Ω-operations iff the dotted arrow in the left-hand diagram of (3) exists (see [19,28]) and, moreover, observational equality is a Ξ-bisimulation iff the dotted arrow in the right-hand diagram of (3) exists.[19]

(2) Another way to explain Definition 66 is the following. Let $M = (\omega, o, \xi) \in \mathsf{Mod}_{\mathsf{Obs}}(\Omega; \mathcal{O}, \Xi)$ with carrier $X \in \mathsf{Set}^S$ and denote by $e : X \to \bar{X}$ the quotient of X w.r.t. observational equality. Then there is a unique $\bar{M} \in \mathsf{Mod}_{\mathsf{Obs}}(\Omega; \mathcal{O}, \Xi)$ with carrier $\bar{X}$ such that e is a morphism $M \to \bar{M}$. That is, in $\mathsf{Mod}_{\mathsf{Obs}}(\Omega; \mathcal{O}, \Xi)$ fully-abstract quotient models exist.[20]

(3) Morphisms of $\mathsf{Mod}_{\mathsf{Obs}}(\Omega; \mathcal{O}, \Xi)$ are inherited from $\mathsf{Mod}(\Omega; \mathcal{O}, \Xi)$. Corollary 78 below describes how to obtain from $\mathsf{Mod}_{\mathsf{Obs}}(\Omega; \mathcal{O}, \Xi)$ a category (called $\mathcal{C}_B$ there) with observational morphisms as in Definition 7.

We now give a dual treatment of reachability.

Definition 68 (Constructor-based signature) *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*

[19] Two states are Ξ-*bisimilar* iff they can be identified by some Ξ-coalgebra morphism (for example, observational equality is $\mathcal{O}$-bisimilarity).

[20] A proof that the existence of fully-abstract quotient models is indeed equivalent to the condition expressed by the diagrams (3) is analogous to [28], Theorem 3.5. This proof generalizes from $\mathcal{X} = \mathsf{Set}^S$ to categories $\mathcal{X}$ with factorization systems if we assume that Ω preserves quotients and Ξ preserves embeddings.

$\mathcal{R}$-algebra $\iota : \mathcal{R}I \to I$ exists.

A model for the signature $(\Omega, \mathcal{R}; \Xi)$ is a triple (ω, ρ, ξ) with $\omega \in \mathrm{Alg}(\Omega)$, $\rho \in \mathrm{Alg}(\mathcal{R})$, $\xi \in \mathrm{Coalg}(\Xi)$. A morphism $f : (\omega, \rho, \xi) \to (\omega', \rho', \xi')$ is an arrow f that is, at the same time, an algebra-morphism $\omega \to \omega'$, an algebra-morphism $\rho \to \rho'$, and a coalgebra-morphism $\xi \to \xi'$. The category of models is denoted by $\mathsf{Mod}(\Omega, \mathcal{R}; \Xi)$.

Example 69 *The constructor-based signature for natural numbers can be represented by the following functors:*

- $\Omega X = X \times X$ *corresponding to the operation* $add : X \times X \to X$*,*
- $\mathcal{R}X = 1 + X$ *corresponding to the constructors* $[zero, succ] : 1 + X \to X$*,*

and, assuming an additional operation $iszero : X \to \mathbb{B}$*,*

- $\Xi X = \mathbb{B}$ *corresponding to the operation* $X \to \mathbb{B}$*.*

Remark 70 *According to Definition 68, the constrained sorts S_{Cons} in the sense of Section 3 are modeled by choosing $\mathcal{X} = \mathsf{Set}^{S_{\mathrm{Cons}}}$ and the loose sorts are interpreted by given parameters.*

The following provides a categorical definition of a constructor-generated part (in the sense of Section 3) by means of algebras.

Definition 71 (Generated part) *Given $M = (\omega, \rho, \xi) \in \mathsf{Mod}(\Omega, \mathcal{R}; \Xi)$, the* generated part *of M is the image of $? : I \to X$ where $?$ is the morphism from the initial $\mathcal{R}$-algebra $\iota : \mathcal{R}I \to I$ as depicted below.*

$$\begin{tikzcd}
X & \mathcal{R}X \arrow[l, "\rho"'] \\
I \arrow[u, "?"] & \mathcal{R}I \arrow[l, "\iota"'] \arrow[u, "\mathcal{R}?"']
\end{tikzcd} \tag{4}$$

Remark 72 *Instantiating the definition with $\mathcal{X} = \mathsf{Set}^S$ and writing $I = (I_s)_{s \in S} \in \mathsf{Set}^S$ and $? = (?_s)_{s \in S}$, the sets $?_s(I_s)$ contain all elements of M of sort s that can be constructed according to ρ.*

The next definition characterizes those models whose non-constructor operations preserve the generated part (in the sense of constructor-based algebras in Section 3). It is the formal dual of Definition 66.

Definition 73 (Constructor-based models) *$(\omega, \rho, \xi) \in \mathsf{Mod}(\Omega, \mathcal{R}; \Xi)$ is called a* constructor-based model *for the signature $(\Omega, \mathcal{R}; \Xi)$ if there are dotted* *arrows such that the following diagrams commute*

$$\begin{tikzcd}
\Xi X & X \arrow[l, "\xi"'] & & X & \Omega X \arrow[l, "\omega"'] \\
\Xi I \arrow[u, "\Xi ?"] & I \arrow[l, dotted] \arrow[u, "?"'] & & I \arrow[u, "?"'] & \Omega I \arrow[l, dotted] \arrow[u, "\Omega ?"']
\end{tikzcd} \tag{5}$$

where ? is the unique algebra-morphism $? : \iota \to \rho$ *from the initial* $\mathcal{R}$*-algebra* $\iota : \mathcal{R}I \to I$*; see diagram (4). The full subcategory of constructor-based models is denoted by* $\mathsf{Mod}_{\mathsf{Cons}}(\Omega, \mathcal{R}; \Xi)$*. A model is* reachable *if* $? : \iota \to \rho$ *is a quotient (i.e. surjective in case of* $\mathcal{X} = \mathsf{Set}^S$*).*

Remark 74

(1) The diagrams express in an abstract way the condition for constructor-based algebras of Definition 26. Indeed, assuming $\mathcal{X} = \mathsf{Set}^S$*, both diagrams state that the image of ? is closed under operations* ω *and* ξ*.*

(2) Another way to explain Definition 73 is the following. Let $M = (\omega, \rho, \xi) \in \mathsf{Mod}_{\mathsf{Cons}}(\Omega, \mathcal{R}; \Xi)$ *with carrier* $X \in \mathsf{Set}^S$ *and generated part* $m : \breve{X} \hookrightarrow X$*. Then there is a unique* $\breve{M} \in \mathsf{Mod}_{\mathsf{Cons}}(\Omega, \mathcal{R}; \Xi)$ *with carrier* $\breve{X}$ *such that* m *is a morphism* $\breve{M} \to M$*. That is, in* $\mathsf{Mod}_{\mathsf{Cons}}(\Omega, \mathcal{R}; \Xi)$ *reachable submodels exist.* [21]

(3) Morphisms of $\mathsf{Mod}_{\mathsf{Cons}}(\Omega, \mathcal{R}; \Xi)$ *are inherited from* $\mathsf{Mod}(\Omega, \mathcal{R}; \Xi)$*. Corollary 82 describes how to obtain from* $\mathsf{Mod}_{\mathsf{Cons}}(\Omega, \mathcal{R}; \Xi)$ *a category (called* $\mathcal{C}_R$ *there) with constructor-based morphisms as in Definition 28.*

Definitions 66 and 73 give rise to a **duality principle** for constructor-based and observational models which is stated formally by the following isomorphisms of categories:

$$(\mathsf{Mod}_{\mathsf{Obs}}(\Omega; \mathcal{O}, \Xi))^{\mathrm{op}} \cong \mathsf{Mod}_{\mathsf{Cons}}(\Xi^{\mathrm{op}}, \mathcal{O}^{\mathrm{op}}; \Omega^{\mathrm{op}}),$$

$$(\mathsf{Mod}_{\mathsf{Cons}}(\Omega, \mathcal{R}; \Xi))^{\mathrm{op}} \cong \mathsf{Mod}_{\mathsf{Obs}}(\Xi^{\mathrm{op}}; \mathcal{R}^{\mathrm{op}}, \Omega^{\mathrm{op}}).$$

The two isomorphisms map models $(\omega, f, \xi)^{\mathrm{op}}$ (with $f = o$ and $f = \rho$, respectively) to $(\xi^{\mathrm{op}}, f^{\mathrm{op}}, \omega^{\mathrm{op}})$. In the following theorem, we identify $(\omega, f, \xi)^{\mathrm{op}}$ with $(\xi^{\mathrm{op}}, f^{\mathrm{op}}, \omega^{\mathrm{op}})$.

As a consequence of the duality principle we obtain:

Theorem 75

(1) A model $M \in \mathsf{Mod}(\Omega; \mathcal{O}, \Xi)$ *is an observational model iff* M^{op} *is a constructor-based model.*

[21] A proof that the existence of reachable submodels is equivalent to the condition expressed by the diagrams (5) is dual to [28], Theorem 3.5.

(2) A model $M \in \mathsf{Mod}(\Omega, \mathcal{R}; \Xi)$ is a constructor-based model iff M^{op} is an observational model.
(3) A model M is reachable iff M^{op} is fully abstract.
(4) A model M is fully abstract iff M^{op} is reachable.

The first theorem similar to part 3 and 4 of Theorem 75 is due to Kalman [25] and was proved for linear systems in control theory. Later, Arbib and Manes (see [2] and [3]) brought to light the general principles underlying this duality by considering—essentially—systems as Ω-algebras for functors Ω. Compared to [3] the main point of our formalization consists in the use of coalgebras to formalize the notion of observational equality and in the consideration of observability and reachability constraints as expressed by the diagrams (3) and (5) which formalize in a category-theoretic way the conditions for observational and constructor-based algebras.

6.4 The Duality of Behavior and Restrict Functors

We show that much of the structure unveiled in Sections 2 and 3 can be derived from a simple abstract description of the respective black box semantics.

Definition 76 (Behavior functor) *Let $B : \mathcal{C} \to \mathcal{C}$ be an operation on the objects of a category $\mathcal{C}$. Assume that there is a family η of epimorphisms $\eta_M : M \to BM$, $M \in \mathcal{C}$, and an operation $(\cdot)^\sharp$ mapping morphisms $f : M \to BN$ to "lifted" morphisms $f^\sharp$ such that the diagram*

$$\begin{tikzcd}
BM \arrow[r, "f^\sharp"] & BN \\
M \arrow[u, "\eta_M"] \arrow[ur, "f"'] &
\end{tikzcd}$$

commutes. Then $(B, \eta, (\cdot)^\sharp)$, or sometimes B itself, is called a behavior functor. We denote by $\mathcal{C}^B$ the full subcategory of $\mathcal{C}$ consisting of objects isomorphic to some BM, $M \in \mathcal{C}$.

We call BM the behavior of M and $\mathcal{C}^B$ the category of behaviors. Intuitively, η_M is the quotient map from M onto its behavior. The existence of the lifting expresses that f cannot distinguish elements that are identified by η_M, that is, f preserves observational equality.

The reader not familiar with monads [30] can skip the next proposition and continue with its corollary and the following example.

Proposition 77 *A behavior functor $(B, \eta, (\cdot)^\sharp)$ is a monad whose multiplication is an isomorphism.*

Proof. It is easy to verify that $(B, \eta, (\cdot)^\sharp)$ satisfies the conditions of a Kleisli-triple and that the multiplication-morphisms $\mu_M = (id_{BM})^\sharp$, $M \in \mathcal{C}$, are isomorphisms (details can be found in [28]). □

The fact that B is a monad with isomorphic multiplication determines the structure described in the following corollary.

Corollary 78 *First, defining $Bf = (\eta_N \circ f)^\sharp$ for $f : M \to N$ in $\mathcal{C}$, B is indeed a functor. Second, there is a category $\mathcal{C}_B$ that has the same objects as $\mathcal{C}$ and morphisms $\mathcal{C}_B(M, N) = \mathcal{C}(M, BN)$. The identity on $M \in \mathcal{C}_B$ is η_M and composition of $f : L \to BM$, $g : M \to BN$ is given by $g^\sharp \circ f$. Third, we obtain the following relationships*

$$\begin{tikzcd}
 & & \mathcal{C}^B \arrow[dll, hook, bend right, "I'"'] \\
\mathcal{C} \arrow[loop left, "B"] \arrow[urr, bend right, "B'"'] \arrow[urr, phantom, "\top"] \arrow[drr, bend right, "H"'] \arrow[drr, phantom, "\top"] & & \\
 & & \mathcal{C}_B \arrow[ull, bend right, "G"'] \arrow[uu, "B''"']
\end{tikzcd}$$

where B', B'', and G map an object to its behavior, I' is the inclusion of behaviors, and H is the identity on objects, all satisfying $I'B' = B = GH$, $B''H = B'$, $I'B'' = G$. Moreover, behavior is left adjoint to inclusion ($B' \dashv I'$) and B'' is an equivalence of categories.

Proof. It follows from B being a monad: B is functorial, $\mathcal{C}_B$ is a category, the equations, the adjunctions, and B'' is full and faithful. Since the multiplication is iso, the category of algebras for the monad B is indeed $\mathcal{C}^B$, and I' is full and faithful and every object in $\mathcal{C}^B$ is isomorphic to an object in the image of B'' (compare [11], Vol.2, Proposition 4.2.3). □

Intuitively, $\mathcal{C}$ consists of all possible realizations of a specification whereas $\mathcal{C}^B$ only contains the black box views. $\mathcal{C}_B$ combines both aspects. The models are the same as in $\mathcal{C}$ but the morphisms incorporate the black box view, $\mathcal{C}_B(M, N) = \mathcal{C}^B(BM, BN)$.

Example 79 *Let Σ_{Obs} be an observational signature as in Section 2. Denote by $\mathcal{C}$ the category of observational algebras with standard algebra-morphisms and let B be the operation that maps an observational algebra to its black box view (given by the quotient w.r.t. observational equality). Then $\mathcal{C}_B$ is the category $\text{Alg}_{\text{Obs}}(\Sigma_{\text{Obs}})$ of observational algebras (with observational morphisms as in Definition 7). $\mathcal{C}^B$ is the full subcategory of $\mathcal{C}$ consisting of the fully abstract algebras. The observational black box functor $\mathcal{BB}_{\Sigma_{\text{Obs}}}$ is given by $\mathcal{C}_B \xrightarrow{B''} \mathcal{C}^B \hookrightarrow \text{Alg}(\Sigma)$. It is full and faithful since B'' is full and faithful.*

The relationship between behaviors and the different categories of models has been studied in [28]. We now dualize our results to describe restrict functors.

Definition 80 (Restrict functor) *Let $R : \mathcal{C} \to \mathcal{C}$ be an operation on the objects of a category $\mathcal{C}$. Assume that there is a family ε of monomorphisms $\varepsilon_M : RM \to M$, $M \in \mathcal{C}$, and an operation $(\cdot)^\sharp$ mapping morphisms $f : RN \to M$ to "lifted" morphisms $f^\sharp$ such that the diagram*

$$\begin{tikzcd}
RM \arrow[d, "\varepsilon_M"'] & RN \arrow[l, "f^\sharp"'] \arrow[ld, "f"] \\
M &
\end{tikzcd}$$

commutes. Then $(R, \varepsilon, (\cdot)^\sharp)$, or sometimes R itself, is called a restrict functor. We denote by $\mathcal{C}^R$ the full subcategory of $\mathcal{C}$ consisting of objects isomorphic to some RM, $M \in \mathcal{C}$.

We call RM the generated part of M. Intuitively, ε_M is the inclusion from the generated part RM into M. The existence of the lifting expresses that morphisms f preserve the generated part.

Proposition 81 *A restrict functor $(R, \varepsilon, (\cdot)^\sharp)$ is a comonad whose comultiplication is an isomorphism.*

Corollary 82 *First, defining $Rf = (f \circ \varepsilon_N)^\sharp$ for $f : N \to M$ in $\mathcal{C}$, R is indeed a functor. Second, there is a category $\mathcal{C}_R$ that has the same objects as $\mathcal{C}$ and morphisms $\mathcal{C}_R(N, M) = \mathcal{C}(RN, M)$. The identity on $M \in \mathcal{C}_R$ is ε_M and composition of $f : RM \to L$, $g : RN \to M$ is given by $f \circ g^\sharp$. Third, we obtain the following relationships*

$$\begin{tikzcd}
 & \mathcal{C}^R \arrow[ld, hook, bend right, "I'"'] \\
\mathcal{C} \arrow[loop left, "R"] \arrow[ru, bend right, "R'"'] \arrow[ru, phantom, "\perp"] \arrow[rd, bend right, "H"'] \arrow[rd, phantom, "\perp"] & \\
 & \mathcal{C}_R \arrow[lu, bend right, "G"'] \arrow[uu, bend right, "R''"']
\end{tikzcd}$$

where R', R'', and G map an object to its generated part, I' is the inclusion of generated parts, and H is the identity on objects, all satisfying $I'R' = R = GH$, $R''H = R'$, $I'R'' = G$. Moreover, restriction to generated parts is right adjoint to inclusion ($I' \dashv R'$) and R'' is an equivalence of categories.

Example 83 *Let Σ_{Cons} be a constructor-based signature as in Section 3. Denote by $\mathcal{C}$ the category of constructor-based algebras with standard algebra-morphisms and let R be the operation that maps a constructor-based algebra to its black box view (given by the generated part). Then $\mathcal{C}_R$ is the category $\mathrm{Alg}_{\mathrm{Cons}}(\Sigma_{\mathrm{Cons}})$ of constructor-based algebras (with constructor-based mor-*

phisms as in Definition 28). $\mathcal{C}^R$ is the full subcategory of $\mathcal{C}$ consisting of the reachable algebras. The constructor-based black box functor $\mathcal{BB}_{\Sigma_{\mathrm{Cons}}}$ is given by $\mathcal{C}_R \xrightarrow{R''} \mathcal{C}^R \hookrightarrow \mathrm{Alg}(\Sigma)$. It is full and faithful since R'' is full and faithful.

6.5 On the Usefulness of the Duality Principle

In contrast to Kalman [25], in our duality principle the models M and M^{op} live in different categories. In particular, if M is a model over the base category Set, M^{op} is a model over $\mathsf{Set}^{\mathrm{op}}$, i.e. over complete atomic Boolean algebras. Though Arbib and Manes [3] use this to deal with 'Boolean machines', complete atomic Boolean algebras are certainly of limited usefulness as a base category. Nevertheless, it is worthwhile to formalize the duality underlying reachability and observability in algebraic specifications for at least three reasons:

(1) As long as we prove something about e.g. reachability for models over Set using only properties shared by Set as well as $\mathsf{Set}^{\mathrm{op}}$, we immediately obtain a dual result about observability for models over Set.

(2) The formal duality expressed by the diagrams in Definitions 66 and 73 emphasizes the adequacy of the concepts introduced for observational and constructor-based logic. Moreover, having these diagrams in mind is a good heuristic means to support informal reasoning about reachability and observability. For instance, the notion of a constructor-based algebra originated from the question what would it mean to dualize the diagram in Definition 66.

(3) Since the categorical setting forced us to abstract from syntactic details, we were able to give a simple description of the models of coalgebraic specifications satisfying observability constraints (see [28]). Using the duality, we also obtain a simple categorical description of the models of algebraic specifications satisfying reachability constraints. Furthermore, since the coalgebraic signature functors Ξ, $\mathcal{O}$ can be used to describe partial functions and non-determinism, the approach of this section provides a perspective to incorporate these features into observational logic and constructor-based logic.

7 Conclusion

In this paper we have studied and formalized the duality between observability and reachability concepts used in algebraic approaches to software development. Our study is based on a loose semantics taking into account that the model class of a specification SP should describe the correct realizations of SP.

As a particular outcome, we have presented the novel institution of constructor-based logic. The formal dualization of the categorical representation of observational logic in [19] gave us the intuition to find the adequate notions of constructor-based 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).

This work focuses on a comparison of the two concepts and *not* on their integration. In the meanwhile our approaches to observability and reachability have been integrated in the so-called COL-institution (Constructor-based Observational Logic) introduced in [8]. The (more general) observational equality relation used in this integrated approach takes into account also the constructor-generated elements and hence is strongly related to the notion of partial observational equality considered e.g. in [10] and [23].

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. Foundations of system theory: decomposable systems. *Automatica*, 10:285–302, 1974.

[3] M.A. Arbib and E.G. Manes. Adjoint machines, state-behaviour machines, and duality. *Journal of Pure and Applied Algebra*, 6:313–344, 1975.

[4] E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P.D. Mosses, D. Sannella, and A. Tarlecki. CASL: The Common Algebraic Specification Language. *Theoretical Computer Science*, 286(2):153–196, 2002.

[5] E. Astesiano, H.-J. Kreowski, and B. Krieg-Brückner, editors. *Algebraic Foundations of Systems Specification.* Springer, 1999.

[6] M. Bidoit, M.V. Cengarle, and R. Hennicker. Proof systems for structured specifications and their refinements. In *[5]*, pages 385–433, 1999.

[7] M. Bidoit and R. Hennicker. Observer complete definitions are behaviourally coherent. In *Proc. OBJ/CafeOBJ/Maude Workshop at Formal Methods '99*, pages 83–94. THETA, 1999.

[8] M. Bidoit and R. Hennicker. On the integration of observability and reachability concepts. In M. Nielsen and U. Engberg, editors, *Proc. 5th Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS'02), Grenoble, France*, volume 2303 of *LNCS*, pages 21–36. Springer, 2002.

[9] M. Bidoit, R. Hennicker, and A. Kurz. On the duality between observability and reachability. In F. Honsell and M. Miculan, editors, *Proc. 4th Int. Conf.*

Foundations of Software Science and Computation Structures (FOSSACS'01), Genova, Italy, volume 2030 of *LNCS*, pages 72–87. Springer, 2001.

[10] M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. *Science of Computer Programming*, 25(2–3):149–186, 1995.

[11] F. Borceux. *Handbook of Categorical Algebra.* Cambridge University Press, 1994.

[12] C.C. Chang and H.J. Keisler. *Model Theory.* North-Holland, Amsterdam, 3rd edition, 1990.

[13] C. Cîrstea. Coalgebraic semantics for hidden algebra: parameterized objects and inheritance. In F. Parisi-Presicce, editor, *Recent Trends in Algebraic Development Techniques, WADT'97*, volume 1376 of *LNCS*, pages 174–189. Springer, 1998.

[14] J. Goguen and R. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the ACM*, 39 (1):95–146, 1992.

[15] J. Goguen and G. Malcolm. A Hidden Agenda. *Theoretical Computer Science*, 245(1):55–101, 2000.

[16] J. Goguen and G. Roşu. Hiding more of hidden algebra. In J.M. Wing, J. Woodcock, and J. Davies, editors, *Proc. Formal Methods (FM'99)*, volume 1709 of *LNCS*, pages 1704–1719. Springer, 1999.

[17] R. Hennicker and M. Bidoit. Observational logic (long version). Technical Report LSV-98-6, LSV, Ecole Normale Supérieure de Cachan, June 1998. Available at `www.lsv.ens-cachan.fr/Publis/RAPPORTS\_LSV/rr-lsv-1998-6.rr.ps`.

[18] R. Hennicker and M. Bidoit. Observational logic. In Armando Haeberer, editor, *Proc. 7th Int. Conf. Algebraic Methodology and Software Technology (AMAST'98), Amazonia, Brazil*, volume 1548 of *LNCS*, pages 263–277. Springer, 1999.

[19] R. Hennicker and A. Kurz. (Ω, Ξ)-Logic: On the algebraic extension of coalgebraic specifications. In B. Jacobs and J. Rutten, editors, *Proc. Coalgebraic Methods in Computer Science (CMCS'99)*, volume 19 of *Electronic Notes in Theoretical Computer Science*, pages 195–211, 1999.

[20] R. Hennicker and M. Wirsing. Behavioural specifications. In H. Schwichtenberg, editor, *Proof and Computation, International Summer School Marktoberdorf 1993*, volume 139 of *NATO ASI Series F*, pages 193–230. Springer, 1995.

[21] R. Hennicker, M. Wirsing, and M. Bidoit. Proof systems for structured specifications with observability operators. *Theoretical Computer Science*, 173(2):393–443, 1997.

[22] C.A.R. Hoare. Proofs of correctness of data representations. *Acta Informatica*, 1:271–281, 1972.

[23] M. Hofmann and D. Sannella. On behavioural abstraction and behavioural satisfaction in higher-order logic. In P.D. Mosses, M. Nielsen, and M.I. Schwartzbach, editors, *Proc. 6th Int. Joint Conf. Theory and Practice of Software Development (TAPSOFT'95), Aarhus, Denmark*, volume 915 of *LNCS*, pages 247–261. Springer, 1995.

[24] B. Jacobs and J. Rutten. A Tutorial on (Co)Algebras and (Co)Induction. *EATCS Bulletin*, 62:222–259, 1997.

[25] R.E. Kalman, P.L. Falb, and M.A. Arbib. *Topics in Mathematical System Theory.* McGraw-Hill, 1969.

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

[27] A. Kurz. *Logics for Coalgebras and Applications to Computer Science.* PhD thesis, Ludwig-Maximilians-Universität München, 2000. Available at `http://www.informatik.uni-muenchen.de/~kurz`.

[28] A. Kurz and R. Hennicker. On Institutions for Modular Coalgebraic Specifications. *Theoretical Computer Science*, 280(1–2):69–103, 2002.

[29] A. Kurz and D. Pattinson. Coalgebras and modal logics for parameterised endofunctors. Technical Report SEN-R0040, CWI, 2000. Available at `http://www.cwi.nl/~kurz`.

[30] S. Mac Lane. *Category Theory for the Working Mathematician.* Springer, 1971.

[31] J. Loeckx, H.-D. Ehrich, and M. Wolf. *Specification of Abstract Data Types.* Wiley and Teubner, 1996.

[32] P. Padawitz. Swinging types = functions + relations + transition systems. *Theoretical Computer Science*, 243(1–2):93–165, 2000.

[33] E. Poll and J. Zwanenburg. A logic for abstract data types as existential types. In J.-Y. Girard, editor, *Proc. 4th Int. Conf. Typed Lambda Calculi and Applications (TLCA'99)*, volume 1581 of *LNCS*, pages 310–324. Springer, 1999.

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

[35] G. Roşu. *Hidden Logic.* PhD thesis, University of California at San Diego, 2000.

[36] D. Sannella and A. Tarlecki. On observational equivalence and algebraic specification. *Journal of Computer and System Sciences*, 34:150–178, 1987.

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

[38] A. Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoretical Computer Science*, 37:269–304, 1986.

[39] A. Tarlecki. Towards heterogeneous specifications. In D. Gabbay and M. van Rijke, editors, *Proc. Int. Conf. Frontiers of Combining Systems (FroCos'98), Amsterdam*, pages 337–360. Research Studies Press, 2000.

[40] M. Wirsing and M. Broy. A modular framework for specification and information. In J. Diaz and F. Orejas, editors, *Proc. TAPSOFT'89*, volume 351 of *LNCS*, pages 42–73. Springer, 1989.