On the Integration of Observability and Reachability Concepts *

Michel Bidoit [1] and Rolf Hennicker [2]

[1] Laboratoire Spécification et Vérification (LSV), CNRS & ENS de Cachan, France
[2] Institut für Informatik, Ludwig-Maximilians-Universität München, Germany

Abstract. This paper focuses on the integration of reachability and observability concepts within an algebraic, institution-based framework. We develop the essential notions that are needed to construct an institution which takes into account both the generation- and observation-oriented aspects of software systems. Thereby the underlying paradigm is that the semantics of a specification should be as loose as possible to capture all its correct realizations. We also consider the so-called "idealized models" of a specification which are useful to study the behavioral properties a user can observe when he/she is experimenting with the system. Finally, we present sound and complete proof systems that allow us to derive behavioral properties from the axioms of a given specification.

1 Introduction

Reachability and observability concepts are both equally important in system specifications. Reachability concepts focus on the specification of generation principles usually presented by a set of constructors. Most algebraic specification languages incorporate features to express reachability like, for instance, the CASL language [1]. Observability concepts are used to specify the desired observable properties of a program or software system (see, e.g., [17, 18, 15, 16, 8]). Particular institutions which formalize the syntactic and semantic aspects of observability were introduced in [10] (hidden algebra) and in [11] (observational logic). In [5] we have shown that by dualization of observational logic one obtains a novel treatment of reachability, called the constructor-based logic institution. Both frameworks capture, *either* from the observability *or* from the reachability point of view, the idea that the model class of a specification SP should describe all correct realizations of SP. In many examples, however, both aspects have to be considered simultaneously. The aim of this paper is therefore to integrate our novel treatment of reachability *and* observational logic in a common, powerful institution, called the constructor-based observational logic institution.

Of course, we are aware that many approaches in the literature already cover in some way reachability and/or observability. However, most of them either are not based on a loose semantics (like [16]) or are too restrictive w.r.t. the interpretation of reachability in the sense that only reachable models are admitted.

* This work is partially supported by the German DFG-project InopSys and by the German BMBF-project GLOWA-Danube.

Thus standard implementations which simply contain junk (like the realization of the natural numbers by the integers) are ruled out from the models of a specification. The ultra-loose approach of [20], the notion of behavioral specification w.r.t. a partial observational equality in [6, 12] and the hidden algebra approach are closely related to our framework. The main difference to [20] is that no explicit notion of observer or constructor operation is used there while in our approach they are the basic ingredients of a signature which lead to a specification methodology and to an institution tailored to observability and reachability. The partial observational equality of [6] does not take into account a distinguished set of observer and constructor operations which in our case facilitates proofs and leads to a powerful notion of signature morphism. The main difference to the presentation of hidden algebra in [10] is that there the reachable values are given by a fixed data universe while in our approach constructors can be defined for arbitrary sorts and hence also for hidden state sorts which we believe is important to deal with reachable states.

We assume that the reader is familiar with the basic notions of algebraic specifications (see, e.g., [14, 2]), like the notions of (many-sorted) *signature* $\Sigma = (S, OP)$ (where S is a set of *sorts* and OP is a set of *operation symbols* $op : s_1, \ldots, s_n \to s$), *(total)* Σ*-algebra* $A = ((A_s)_{s \in S}, (op^A)_{op \in OP})$, class $\mathrm{Alg}(\Sigma)$ of all Σ-algebras, Σ*-term algebra* $T_\Sigma(X)$ over a family of variables X and *interpretation* $I_\alpha : T_\Sigma(X) \to A$ w.r.t. a *valuation* $\alpha : X \to A$. We implicitly assume throughout this paper that the carrier sets of an algebra are not empty.

2 The Constructor-based Observational Logic Institution

In this section we develop, step by step, the syntactic and semantic notions which lead to the constructor-based observational logic institution, called COL for short. We start by considering so-called COL-signatures which provide the syntactic basis to integrate reachability and observability concepts in algebraic system specifications. Technically, a COL-signature consists of a standard algebraic signature together with a distinguished set of constructor operations and a distinguished set of observer operations. Intuitively, the constructors determine those elements which are of interest from the user's point of view while the observers determine a set of observable experiments that a user can perform to examine hidden states. Thus we can abstract from junk elements and also from concrete state representations whereby two states are considered to be "observationally equal" if they cannot be distinguished by observable experiments.

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

A COL-signature $\Sigma_{\mathrm{COL}} = (\Sigma, OP_{\mathrm{Cons}}, OP_{\mathrm{Obs}})$ *consists of a signature* $\Sigma = (S, OP)$*, a set* $OP_{\mathrm{Cons}} \subseteq OP$ *of constructors and a set* OP_{Obs} *of observers* (obs, i) *with* $obs \in OP$*.*

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 constrained, i.e.* $S_{\text{Loose}} = S \setminus S_{\text{Cons}}$.

The set $S_{\text{State}} \subseteq S$ *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}, $obs : s_1, \ldots, s_i, \ldots, s_n \to s$*. The set* $S_{\text{Obs}} \subseteq S$ *of* observable sorts *consists of all sorts which are not a state sort, i.e.* $S_{\text{Obs}} = S \setminus S_{\text{State}}$.

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

Note that in many examples state sorts are also constrained sorts which allows us to deal with reachable states. We implicitly assume in the following that whenever we consider a COL-signature Σ_{COL}, then $\Sigma_{\text{COL}} = (\Sigma, OP_{\text{Cons}}, OP_{\text{Obs}})$ with $\Sigma = (S, OP)$ and similarly for Σ'_{COL} etc.

Example 1. As a running example we consider the following COL-signature $\Sigma_{\text{COL}} = (\Sigma, OP_{\text{Cons}}, OP_{\text{Obs}})$ for containers of natural numbers where:

$\Sigma = (S, OP)$, $S = \{\ bool,\ nat,\ container\ \}$

$$\begin{aligned} OP = \{\ & true : \to bool,\ false : \to bool, \\ & 0 : \to nat,\ succ : nat \to nat,\ add : nat \times nat \to nat, \\ & empty : \to container,\ insert : container \times nat \to container, \\ & remove : container \times nat \to container,\ isin : container \times nat \to bool\ \} \end{aligned}$$

$OP_{\text{Cons}} = \{\ true,\ false,\ 0,\ succ,\ empty,\ insert\ \}$

$OP_{\text{Obs}} = \{\ (isin, 1)\ \}$

Hence, in this example, all sorts are constrained, *container* is the only state sort and the observable sorts are *bool* and *nat*. $\diamond$

Any set OP_{Cons} of constructor symbols (and hence any COL-signature) determines a set of constructor terms.

Definition 2 (Constructor term). *Let* Σ_{COL} *be a COL-signature, 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* $\mathcal{T}(\Sigma_{\text{COL}})_s$ *of* constructor terms *with "constrained result sort"* s *is inductively defined as follows:*

1. *Each constant* $cons : \to s \in OP_{\text{Cons}}$ *belongs to* $\mathcal{T}(\Sigma_{\text{COL}})_s$.
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* $t_i \in \mathcal{T}(\Sigma_{\text{COL}})_{s_i}$ *if* $s_i \in S_{\text{Cons}}$, $cons(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma_{\text{COL}})_s$.

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

Note that only constructor symbols and variables of loose sorts are used to build constructor terms. In particular, if all sorts are constrained, i.e., $S_{\text{Cons}} = S$, the constructor terms are exactly the (S, OP_{Cons})-ground terms which are built by the constructor symbols. This is the case, for instance, in the above example.

The syntactic notion of a constructor term induces, for any Σ-algebra A, the definition of a family of subsets of the carrier sets of A, called reachable part, which consists of all elements which are reachable (from the loose sorts, if any) with respect to the given constructors. In the following considerations the reachable part plays a crucial role since it represents those elements which are of interest from the user's point of view.

Definition 3 (Reachable part). *Let Σ_{COL} be a COL-signature. For any Σ-algebra $A \in \mathrm{Alg}(\Sigma)$, the* reachable part *(w.r.t. OP_{Cons}) $\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) = (\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_s)_{s \in S}$ of A is defined by:*

1. $\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_s = A_s$, *if* $s \in S_{\mathrm{Loose}}$.
2. $\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_s = \{a \in A_s \mid \text{there exists a term } t \in \mathcal{T}(\Sigma_{\mathrm{COL}})_s \text{ and a valuation } \alpha : X \to A \text{ such that } I_\alpha(t) = a\}$, *if* $s \in S_{\mathrm{Cons}}$.

Definition 4 (Reachable algebra). *Let Σ_{COL} be a COL-signature. A Σ-algebra A is called* reachable *(w.r.t. OP_{Cons}) if it coincides with its reachable part.*

Example 2. Consider the signature Σ_{COL} of Example 1 and the following Σ-algebra A with carriers:
$A_{bool} = \{T, F\}$, $A_{nat} = \mathbb{Z}$ (set of the integers),
$A_{container} = \mathbb{Z}^* \times \mathbb{Z}^*$ (pairs of finite lists of integers)
and with operations:
$true^A = T$, $false^A = F$, $0^A = 0$, $succ^A(a) = a + 1$, $add^A(a, b) = a + b$,
$empty^A = (<>, <>)$,
$insert^A((< a_1, \ldots, a_n >, < b_1, \ldots, b_m >), a) =$
$\quad (< a, a_1, \ldots, a_n >, < b_1, \ldots, b_m >)$ if $a \neq a_i$ for $i = 1, \ldots, n$,
$insert^A((s, t), a) = (s, t)$ otherwise,
$remove^A((< a_1, \ldots, a_n >, < b_1, \ldots, b_m >), a) =$
$\quad (< a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n >, < a, b_1, \ldots, b_m >)$ if $a_i = a$ and $a_j \neq a$ for $j = 1, \ldots, i - 1$,
$remove^A((s, t), a) = (s, t)$ otherwise,
$isin^A((< a_1, \ldots, a_n >, t), a) = F$ if $a \neq a_i$ for $i = 1, \ldots, n$,
$isin^A((s, t), a) = T$ otherwise.
The above Σ-algebra A can be considered as an implementation of containers of natural numbers whereby the natural numbers are implemented by the integers and containers are implemented by two finite lists s and t such that s stores the elements which are actually in the container and t is a "trash" which stores those elements that have been removed from the container. The *remove* operation is defined in an efficient way: only one occurrence of a given element is deleted from the actual elements of a container. This is sufficient since the *insert* operation only stores an element if it does not already belong to the actual elements of a container. The reachable part $\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)$ of A consists of the following sets:
$\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_{bool} = \{T, F\}$,
$\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_{nat} = \mathbb{N}$ (set of the natural numbers),
$\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_{container} = \{(s, <>) \mid s \in \mathbb{N}^* \text{ and each element of } s \text{ occurs only once in } s\}$. ◇

Let us now focus on the set OP_{Obs} of observers declared by a COL-signature Σ_{COL}. The observers determine a set of observable contexts which represent the observable experiments. In contrast to the inductive definition of constructor terms, observable contexts are defined in a coinductive style.

Definition 5 (Observable context). *Let Σ_{COL} be a COL-signature, let $X = (X_s)_{s \in S}$ be a family of countably infinite sets X_s of variables of sort s and let $Z = (\{z_s\})_{s \in S_{\mathrm{State}}}$ be a disjoint family of singleton sets (one for each state sort). For all $s \in S_{\mathrm{State}}$ and $s' \in S_{\mathrm{Obs}}$ the set $\mathcal{C}(\Sigma_{\mathrm{COL}})_{s \to s'}$ of* observable Σ_{COL}-contexts *with "application sort" s and "observable result sort" s' is inductively defined as follows:*

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

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

Note that only the observer operations are used to build observable contexts For instance, the context $isin(z_{container}, x)$ is (up to renaming of the variable x) the only observable context in the container example.

The syntactic notion of an observable context will be used to define, for any Σ-algebra A, a semantic relation, called observational equality, which expresses indistinguishability of states. As already pointed out, the observable contexts represent observable experiments which can be applied to examine states. Then two states are observationally equal if they cannot be distinguished by these experiments.

If there is no constructor symbol, this intuitive idea can easily be formalized as done in the observational logic framework, see [11]. However, if we integrate observability and reachability concepts, we have to be careful with respect to the role of constructors in observable experiments. For instance, in the container example, the observable context $isin(z_{container}, x)$ represents a set of observable experiments on containers which depend on the actual values of the variable x of sort nat. Since nat is a constrained sort, from the user's point of view the only relevant values are representable by a constructor term (and hence belong to the reachable part). This leads to the following definition of the observational equality which depends, in contrast to the pure observational approach in [11], not only on the observers but also on the chosen constructors.

Definition 6 (Observational equality). *Let Σ_{COL} be a COL-signature. For any Σ-algebra $A \in \mathrm{Alg}(\Sigma)$, the* observational Σ_{COL}-equality *on A is denoted* *by $\approx_{\Sigma_{\mathrm{COL}},A}$ and defined as follows. For all $s \in S$, two elements $a, b \in A_s$ are observationally equal w.r.t. Σ_{COL}, i.e., $a \approx_{\Sigma_{\mathrm{COL}},A} b$, if and only if*

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

Definition 7 (Fully-abstract algebra). *Let Σ_{COL} be a COL-signature. A Σ-algebra A is called* fully abstract *(w.r.t. Σ_{COL}) if the observational Σ_{COL}-equality $\approx_{\Sigma_{\mathrm{COL}},A}$ on A coincides with the set-theoretic equality.*

Example 3. Consider the signature Σ_{COL} of Example 1 and the algebra of containers defined in Example 2 where a container is represented by a pair (s, t) of finite lists of integers. Two containers $(s1, t1)$ and $(s2, t2)$ are observationally equal, $(s1, t1) \approx_{\Sigma_{\mathrm{COL}},A} (s2, t2)$, if for all natural numbers n, $isin^A((s1, t1), n) = isin^A((s2, t2), n)$ holds. By definition of $isin^A$, this means that the same natural numbers occur in $s1$ and in $s2$. Thus the observational equality abstracts not only from the ordering and multiple occurrences of elements, but also from the occurrences of negative integers and from the content of each "trash" $t1$ and $t2$. This expresses exactly our intuition according to the given constructors and observers. For instance, the following container representations are observationally equal: $(< 1, 2 >, <>) \approx_{\Sigma_{\mathrm{COL}},A} (< 2, -7, 2, -3, 1 >, < 6, -4 >)$. $\diamond$

Up to now the syntactic notion of a COL-signature Σ_{COL} has lead to the semantic concepts of a reachable part (determined by the constructors) and of an observational equality (determined by the observers but with an impact of the constructors) which both have been defined for an arbitrary algebra over the underlying signature Σ. As we will see in the following discussion, the constructors and the observers induce also certain constraints on algebras which lead to the notion of a COL-algebra.

In traditional approaches to reachability, constructor symbols are used to restrict the admissible models of a specification to those algebras which are reachable with respect to the given constructors (i.e. to reachable algebras, see Definition 4). We do not adopt this interpretation since, as many examples show, it is too restrictive if the semantics of a specification is expected to capture all correct realizations. For instance, the container algebra of Example 2 is not reachable w.r.t. the given constructors but should be usable as a correct realization of containers. As a consequence, we are interested in a more flexible framework where the constructor symbols are still essential, but nevertheless non-reachable algebras can be accepted as models if they satisfy certain conditions. Since the reachable part represents the elements of interest, one could simply require that no further elements should be constructible by the non-constructor operations. Indeed, if we are working in a pure constructor-based framework, this condition fits perfectly to our intuition (see [5], Section 3). However, if we deal simultaneously with observability, this requirement is still too strong because from the user's point of view it doesn't matter if a non-constructor operation yields an element *outside* the reachable part as long as this element is observationally equal to some other element *inside* the reachable part. Formally, this condition is expressed by the following reachability constraint.

Definition 8 (Reachability constraint). *Let Σ_{COL} be a COL-signature. For any Σ-algebra $A \in \mathrm{Alg}(\Sigma)$, $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma}$ denotes the smallest Σ-subalgebra of A which includes the reachable part $\mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)$.* [1]

A Σ-algebra A satisfies the reachability constraint induced by Σ_{COL}, *if for any $a \in \langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma}$ there exists $b \in \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)$ such that $a \approx_{\Sigma_{\mathrm{COL}},A} b$.*

Example 4. Let A be the container algebra of Example 2. It is obvious that the reachable part of A is not closed under the operation $remove^A$. For instance, $remove^A((<1,2>,<>),1) = (<2>,<1>) \notin \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_{container}$. In fact, we have $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma,\mathrm{container}} = \{(s,t) \mid s,t \in \mathbb{N}^*$ and each element of s occurs only once in $s\}$. However, any element $(s,t) \in \langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma,\mathrm{container}}$ is observationally equal to $(s,<>)$ (see Example 3) which is an element of the reachable part. Considering the sort *nat*, the reachable elements (which are just the natural numbers) are preserved under add^A, i.e. $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma,\mathrm{nat}} = \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A)_{nat} = \mathbb{N}$. Thus A satisfies the reachability constraint induced by Σ_{COL}. $\diamond$

Let us now discuss the constraints on a Σ-algebra A that are induced by the observers OP_{Obs} of a COL-signature Σ_{COL}. Since the declaration of observers determines a particular observational equality on any Σ-algebra A, the (interpretations of the) non-observer operations should respect this observational equality, i.e. a non-observer operation should not contribute to distinguish states. For this purpose one could simply require that the observational equality is a Σ-congruence on A. Indeed, if we are working in a pure observational framework, this condition fits perfectly to our intuition (see [11]). However, if we deal simultaneously with reachability, this requirement is too strong because computations performed by a user can only lead to elements in the Σ-subalgebra $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma}$. As a consequence, it is sufficient to require the congruence property on this subalgebra which is expressed by the following observability constraint.

Definition 9 (Observability constraint). *Let Σ_{COL} be a COL-signature. A Σ-algebra A* satisfies the observability constraint induced by Σ_{COL}, *if $\approx_{\Sigma_{\mathrm{COL}},A}$ is a Σ-congruence on $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma}$.*

Example 5. The container algebra A of Example 2 satisfies the observability constraint of the given COL-signature for containers. Note, however, that $\approx_{\Sigma_{\mathrm{COL}},A}$ is only a Σ-congruence on $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma}$ but not on the whole algebra A since $remove^A$ does not respect the observational equality for *all* elements of A. Consider, for instance, the element $(<1,1>,<>) \notin \langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma,\mathrm{container}}$. $(<1,1>,<>) \approx_{\Sigma_{\mathrm{COL}},A} (<1>,<>)$ but since $remove^A((<1,1>,<>),1) = (<1>,<1>)$, it is *not* observationally equal to $remove^A((<1>,<>),1) = (<>,<1>)$. $\diamond$

---

[1] Indeed $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_{\Sigma}$ is the Σ-subalgebra of A generated by the operations OP over the carrier sets A_s with loose sort s.

Definition 10 (COL-algebra). *Let Σ_{COL} be a COL-signature. A* Σ_{COL}-algebra *(also called* COL-algebra*) is a Σ-algebra A which satisfies the reachability and the observability constraints induced by Σ_{COL}. The class of all Σ_{COL}-algebras is denoted by* $\text{Alg}_{\text{COL}}(\Sigma_{\text{COL}})$*.*

Note that the satisfaction of the two constraints is essential for defining the black box semantics of COL-specifications considered in Section 3 and hence for guaranteeing the soundness of the proof systems in Section 4. In particular the instantiation rule and the congruence rule of the equational calculus would not be sound w.r.t. the COL-satisfaction relation defined below without assuming both the reachability and observability constraints. The following notion of COL-morphism is a generalization of standard Σ-homomorphisms.

Definition 11 (COL-morphism). *Let $A, B \in \text{Alg}_{\text{COL}}(\Sigma_{\text{COL}})$ be two Σ_{COL}-algebras. A* Σ_{COL}-morphism *(also called* COL-morphism*) $h : A \to B$ is an S-sorted family $(h_s)_{s \in S}$ of relations $h_s \subseteq \langle \mathcal{R}_{\Sigma_{\text{COL}}}(A) \rangle_{\Sigma,s} \times \langle \mathcal{R}_{\Sigma_{\text{COL}}}(B) \rangle_{\Sigma,s}$ with the following properties, for all $s \in S$:*

1. *For all $a \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(A) \rangle_{\Sigma,s}$, there exists $b \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(B) \rangle_{\Sigma,s}$ such that $a \; h_s \; b$.*
2. *For all $a \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(A) \rangle_{\Sigma,s}$, $b, b' \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(B) \rangle_{\Sigma,s}$, if $a \; h_s \; b$, then ($a \; h_s \; b'$ if and only if $b \approx_{\Sigma_{\text{COL}},B} b'$).*
3. *For all $a, a' \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(A) \rangle_{\Sigma,s}$, $b \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(B) \rangle_{\Sigma,s}$, if $a \; h_s \; b$ and $a \approx_{\Sigma_{\text{COL}},A} a'$, then $a' \; h_s \; b$.*
4. *For all $op : s_1, \ldots, s_n \to s \in OP$, $a_i \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(A) \rangle_{\Sigma,s_i}$, $b_i \in \langle \mathcal{R}_{\Sigma_{\text{COL}}}(B) \rangle_{\Sigma,s_i}$, if $a_i \; h_{s_i} \; b_i$ for $i = 1, \ldots, n$, then $op^A(a_1, \ldots, a_n) \; h_s \; op^B(b_1, \ldots, b_n)$.*

For any COL-signature Σ_{COL}, the class $\text{Alg}_{\text{COL}}(\Sigma_{\text{COL}})$ together with the Σ_{COL}-morphisms is a category which, by abuse of notation, will also be denoted by $\text{Alg}_{\text{COL}}(\Sigma_{\text{COL}})$.

In the next step we generalize the standard satisfaction relation of first-order logic by abstracting with respect to reachability and observability. First, from the reachability point of view, the valuations of variables are restricted to the elements of the reachable part only.[2] From the observability point of view, the idea is to interpret the equality symbol $=$ occurring in a first-order formula φ not by the set-theoretic equality but by the observational equality of elements.

Definition 12 (COL-satisfaction relation). *The* COL-satisfaction relation *between Σ-algebras and first-order Σ-formulas (with variables in X) is denoted by $\models_{\Sigma_{\text{COL}}}$ and defined as follows. Let $A \in \text{Alg}(\Sigma)$.*

1. *For any two terms $t, r \in T_\Sigma(X)_s$ of the same sort s and for any valuation $\alpha : X \to \mathcal{R}_{\Sigma_{\text{COL}}}(A)$, $A, \alpha \models_{\Sigma_{\text{COL}}} t = r$ holds if $I_\alpha(t) \approx_{\Sigma_{\text{COL}},A} I_\alpha(r)$.*
2. *For any arbitrary Σ-formula φ and for any valuation $\alpha : X \to \mathcal{R}_{\Sigma_{\text{COL}}}(A)$, $A, \alpha \models_{\Sigma_{\text{COL}}} \varphi$ is defined by induction over the structure of the formula φ in the usual way. In particular, $A, \alpha \models_{\Sigma_{\text{COL}}} \forall x{:}s.\, \varphi$ if for all valuations $\beta : X \to \mathcal{R}_{\Sigma_{\text{COL}}}(A)$ with $\beta(y) = \alpha(y)$ for all $y \neq x$, $A, \beta \models_{\Sigma_{\text{COL}}} \varphi$.*

[2] This idea is related to the ultra-loose approach of [20] where the same effect is achieved by using formulas with relativized quantification.

3. *For any arbitrary Σ-formula φ, $A \models_{\Sigma_{\text{COL}}} \varphi$ holds if for all valuations $\alpha : X \to \mathcal{R}_{\Sigma_{\text{COL}}}(A)$, $A, \alpha \models_{\Sigma_{\text{COL}}} \varphi$ holds.*

The notation $A \models_{\Sigma_{\text{COL}}} \varphi$ is extended in the usual way to classes of algebras and sets of formulas. Note that the COL-satisfaction relation is defined for arbitrary Σ-algebras though it will only be used in this paper for COL-algebras. Note also that for COL-algebras, the COL-satisfaction relation would be the same if we would have used valuations "$\alpha : X \to \langle \mathcal{R}_{\Sigma_{\text{COL}}}(A)\rangle_{\Sigma}$" instead of "$\alpha : X \to \mathcal{R}_{\Sigma_{\text{COL}}}(A)$" in the above definition.

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

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

In the following, $\mathrm{SP}_{\text{COL}} \models_{\Sigma_{\text{COL}}} \varphi$ means $\mathrm{Mod}_{\text{COL}}(\mathrm{SP}_{\text{COL}}) \models_{\Sigma_{\text{COL}}} \varphi$. According to the flexible satisfaction relation, the model class of a COL-specification SP_{COL} describes all algebras which can be considered as correct realizations of SP_{COL}.

Example 6. The following specification extends the COL-signature of Example 1 by appropriate axioms for containers of natural numbers.[3]

spec CONTAINER =
  sorts *bool*, *nat*, *container*
  ops $true, false : bool$;
    $0 : nat$; $succ : nat \to nat$; $add : nat \times nat \to nat$;
    $empty : container$; $insert : container \times nat \to container$;
    $remove : container \times nat \to container$;
    $isin : container \times nat \to bool$;
  constructors *true*, *false*, *0*, *succ*, *empty*, *insert*
  observer (*isin*, *1*)
  axioms
  $\forall x, y : nat;\ c : container$
  %% standard axioms for booleans and natural numbers, plus
  - $isin(empty, x) = false$ (1)
  - $isin(insert(c, x), x) = true$ (2)
  - $x \neq y \Rightarrow isin(insert(c, y), x) = isin(c, x)$ (3)
  - $remove(empty, x) = empty$ (4)
  - $remove(insert(c, x), x) = remove(c, x)$ (5)
  - $x \neq y \Rightarrow remove(insert(c, y), x) = insert(remove(c, x), y)$ (6)

end

[3] We use here a syntactic sugar similar to the one of CASL.

It is important to note that the declaration of constructors and observers leads to corresponding specification methods. As usual, non-constructor operations can be defined by a complete case distinction w.r.t. the given constructors. For instance, the axioms (1) - (3) define the non-constructor *isin* by a complete case analysis w.r.t. *empty* and *insert* and, similarly, *remove* is specified by a constructor complete definition according to the axioms (4) - (6).

On the other hand, also the observers give rise to a specification method whereby the observable effect of the non-observer operations can be defined by a complete case distinction w.r.t. the given observers. For instance, axiom (1) can be considered as an observer complete definition of *empty* and axioms (2) and (3) can be considered as an observer complete definition of *insert* (see [3] for a general schema of observer complete definitions). Thus the axioms (1) - (3) can be seen from both sides, from the observational or from the reachability point of view, the result is the same.

However, this is not the case for the axioms (4) - (6) that specify *remove* (which is neither a constructor nor an observer). In this case we have chosen a constructor style, but we can ask whether we couldn't use just as well an observer style with the same semantic result. Indeed it is simple to provide an observer complete definition of *remove* by the following two formulas:

- $isin(remove(c, x), x) = false$ (7)
- $x \neq y \Rightarrow isin(remove(c, x), y) = isin(c, y)$ (8)

Obviously, with a standard interpretation, the formulas (7) and (8) are quite different from the axioms (4) - (6). However, in the COL framework developed in this paper it turns out that indeed the axioms (4) - (6) could be replaced by the formulas (7) and (8) without changing the semantics of the container specification. A formal proof of this fact (using the proof systems developed in Section 4) is provided in [4].

Let us still point out that the container algebra A of Example 2 is a model of CONTAINER. Thereby it is essential that the COL-satisfaction relation interprets the equality symbol by the observational equality. Otherwise, axiom (5) would not be satisfied by A. For instance, if we interpret c by the empty container $(<>, <>)$ and x by 1, we have $remove^A((<>, <>), 1) = (<>, <>)$ and $remove^A(insert^A((<>, <>), 1), 1) = remove^A((< 1 >, <>), 1) = (<>, < 1 >)$ where the results $(<>, <>)$ and $(<>, < 1 >)$ are not the same but are observationally equal.

On the other hand, if we would use (7) and (8) for specifying *remove* then it is essential that the COL-satisfaction relation interprets variables by values in the reachable part. Otherwise, axiom (7) would not be satisfied by the container algebra A. For instance, if we would interpret c by the non reachable container $(< 1, 1 >, <>)$ and x by 1, we would obtain:
$isin^A(remove^A((< 1, 1 >, <>), 1), 1) = isin^A((< 1 >, < 1 >), 1) = true.$ $\diamond$

The definitions stated above provide the basic ingredients for defining the *constructor-based observational logic institution.* Thereby it is particularly important to use an appropriate morphism notion for COL-signatures which guarantees encapsulation of properties with respect to the COL-satisfaction relation (formally expressed by the satisfaction condition of institutions, see [9]). To ensure that the satisfaction condition holds, the crucial idea is to require that neither "new" constructors nor "new" observers are introduced for "old" sorts when composing systems via signature morphisms. This requirement is formally captured by the following definition.

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

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

As a consequence of the definition, for all $s \in S$, the following holds:
$s \in S_{\mathrm{Cons}}$ if and only if $\sigma(s) \in S'_{\mathrm{Cons}}$, $s \in S_{\mathrm{Loose}}$ if and only if $\sigma(s) \in S'_{\mathrm{Loose}}$, $s \in S_{\mathrm{State}}$ if and only if $\sigma(s) \in S'_{\mathrm{State}}$, $s \in S_{\mathrm{Obs}}$ if and only if $\sigma(s) \in S'_{\mathrm{Obs}}$.

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

Thus we obtain the *constructor-based observational logic institution* in a straightforward way. This institution provides also a suitable framework for instantiating the institution-independent specification-building operators introduced in [19] and hence for defining structured COL-specifications (which will not be detailed here).

3 Logical Consequences of Specifications: The Black Box View

So far we have emphasized the fact that the model class $\mathrm{Mod}_{\mathrm{COL}}(\mathrm{SP}_{\mathrm{COL}})$ of a COL-specification $\mathrm{SP}_{\mathrm{COL}}$ reflects all its correct realizations. In the following we will refer to $\mathrm{Mod}_{\mathrm{COL}}(\mathrm{SP}_{\mathrm{COL}})$ as the *glass box semantics* of the specification $\mathrm{SP}_{\mathrm{COL}}$. 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 $\mathrm{SP}_{\mathrm{COL}}$. This means that we are interested in statements of the form $\mathrm{SP}_{\mathrm{COL}} \models_{\Sigma_{\mathrm{COL}}} \varphi$.

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 a COL-specification are exactly the consequences of its idealized models, in *standard* first-order logic. Hence to any specification $\mathrm{SP_{COL}}$ 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.

Let Σ_{COL} be a COL-signature and A be a Σ_{COL}-algebra. As pointed out in the previous section, $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma$ represents the only elements a user can compute (over the loose carrier sets) by invoking operations. Hence, in a first step, we can *restrict* to these elements. Since the observational Σ_{COL}-equality $\approx_{\Sigma_{\mathrm{COL}},A}$ on A is a Σ-congruence on $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma$, we can then construct the quotient $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A}$ which *identifies* all elements of $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma$ which are indistinguishable "from the outside". $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A}$ can be considered as the *black box view* of A and represents the "observable behavior" of A. Note that $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A}$ is *fully abstract* since the observational equality (w.r.t. Σ_{COL}) on $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A}$ coincides with the set-theoretic equality. Moreover, since A satisfies by assumption the reachability constraint induced by Σ_{COL}, any element in $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma$ is observationally equal to a reachable element (w.r.t. Σ_{COL}), and therefore $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A}$ is also a *reachable algebra*. By considering the Σ_{COL}-algebra $\langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A}$ just as a Σ-algebra, we obtain (for any signature Σ_{COL}) a functor from the category $\mathrm{Alg_{COL}}(\Sigma_{\mathrm{COL}})$ of Σ_{COL}-algebras into the category $\mathrm{Alg}(\Sigma)$ of (standard) Σ-algebras.

Theorem 1 (Behavior functor). *For any COL-signature $\Sigma_{\mathrm{COL}} = (\Sigma, OP_{\mathrm{Cons}}, OP_{\mathrm{Obs}})$, the following defines a full and faithful functor* $\mathcal{RI}_{\Sigma_{\mathrm{COL}}} : \mathrm{Alg_{COL}}(\Sigma_{\mathrm{COL}}) \to \mathrm{Alg}(\Sigma)$. [4]

1. *For each $A \in \mathrm{Alg_{COL}}(\Sigma_{\mathrm{COL}})$, $\mathcal{RI}_{\Sigma_{\mathrm{COL}}}(A) \stackrel{\mathrm{def}}{=} \langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A}$ and is called the* observational behavior *of A.*
2. *For each COL-morphism $h : A \to B$,* $\mathcal{RI}_{\Sigma_{\mathrm{COL}}}(h) : \langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(A) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},A} \to \langle \mathcal{R}_{\Sigma_{\mathrm{COL}}}(B) \rangle_\Sigma / \approx_{\Sigma_{\mathrm{COL}},B}$ *is defined by $\mathcal{RI}_{\Sigma_{\mathrm{COL}}}(h)([a]) = [b]$ if $a\ h\ b$.*

Definition 15 (Black box semantics). *Let $\mathrm{SP_{COL}}$ be a COL-specification with signature $\mathrm{Sig_{COL}}(\mathrm{SP_{COL}}) = \Sigma_{\mathrm{COL}}$. Its* black box semantics *is defined by* $[\![\mathrm{SP_{COL}}]\!] \stackrel{\mathrm{def}}{=} \mathcal{RI}_{\Sigma_{\mathrm{COL}}}(\mathrm{Mod_{COL}}(\mathrm{SP_{COL}}))$.

Theorem 2 (Behavioral consequences). *Let $\Sigma_{\mathrm{COL}} = (\Sigma, OP_{\mathrm{Cons}}, OP_{\mathrm{Obs}})$ be a COL-signature, let φ be a Σ-formula, let A be a Σ_{COL}-algebra, and let $\mathrm{SP_{COL}}$ be a COL-specification with signature Σ_{COL}.*

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

[4] The notation $\mathcal{RI}_{\Sigma_{\mathrm{COL}}}$ is chosen to emphasize the intuitive relationship to the "restrict-identify" steps used in various algebraic implementation concepts.

This theorem shows the adequacy of the black box semantics. The proof of the theorem is straightforward by induction on the structure of the formulas. The next theorem provides a characterization of the black box semantics of a COL-specification.

Theorem 3 (Black box semantics relies on reachable fully abstract models). *Let* $\mathrm{SP_{COL}} = \langle \Sigma_{\mathrm{COL}}, \mathrm{Ax} \rangle$ *be a basic COL-specification with signature* $\Sigma_{\mathrm{COL}} = (\Sigma, OP_{\mathrm{Cons}}, OP_{\mathrm{Obs}})$.
$[\![\mathrm{SP_{COL}}]\!] = \{\Sigma\text{−algebra } A \mid A \models \mathrm{Ax} \text{ and } A \text{ is both reachable and fully abstract w.r.t. } \Sigma_{\mathrm{COL}}\}$.

For a proof see [4]. For instance, the black box semantics of the container specification given in Example 6 is (up to isomorphism) the algebra of finite sets of natural numbers.

4 Proof Systems for Proving Consequences of Specifications

In the previous section we have shown how to relate the behavioral consequences of a COL-specification to the consequences in standard first-order logic of the black box semantics of the given specification. The next step is to find adequate axiomatizations of the black box semantics in order to be able to define sound and complete proof systems. According to Theorems 2 and 3, this amounts to find an axiomatic characterization of reachability and full abstractness. The next definitions provide the required axiomatizations which, in general, can only be stated by using *infinitary* first-order formulas.

Definition 16 (Reachability axiom). *Let* Σ_{COL} *be a COL-signature with underlying signature* Σ. *The* reachability axiom *associated to* Σ_{COL} *is the sentence* $\mathrm{REACH}(\Sigma_{\mathrm{COL}})$ *defined by:*

$$\mathrm{REACH}(\Sigma_{\mathrm{COL}}) \stackrel{\mathrm{def}}{=} \bigwedge_{s \in S_{\mathrm{Cons}}} \mathrm{REACH}(\Sigma_{\mathrm{COL}})_s$$

where for each constrained sort $s \in S_{\mathrm{Cons}}$, $\mathrm{REACH}(\Sigma_{\mathrm{COL}})_s$ *is defined by:*

$$\mathrm{REACH}(\Sigma_{\mathrm{COL}})_s \stackrel{\mathrm{def}}{=} \forall x{:}s. \bigvee_{t \in \mathcal{T}(\Sigma_{\mathrm{COL}})_s} \exists \mathrm{Var}(t).\ x = t\,.^5$$

Definition 17 (Fully abstract axiom). *Let* Σ_{COL} *be a COL-signature with underlying signature* Σ. *The* fully abstract axiom *associated to* Σ_{COL} *is the sentence* $\mathrm{FA}(\Sigma_{\mathrm{COL}})$ *defined by:*

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

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

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

$$\mathrm{FA}(\Sigma_{\mathrm{COL}})_s \stackrel{\mathrm{def}}{=} \forall x, y{:}s. \left(\bigwedge_{s' \in S_{\mathrm{Obs}}, c \in \mathcal{C}(\Sigma_{\mathrm{COL}})_{s \to s'}} \forall \mathrm{Var}(c).\ c[x] = c[y] \right) \Rightarrow x = y\,.^{6}$$

Note that in special cases the above axioms may reduce to finitary ones. For instance, in the container example, the fully abstract axioms reads:
$\forall c1, c2{:}\mathit{container}.\ (\forall x{:}\mathit{nat}.\ \mathit{isin}(c1, x) = \mathit{isin}(c2, x)) \Rightarrow c1 = c2\,.$

Proposition 1. *Let* Σ_{COL} *be a COL-signature with underlying signature* Σ*. A* Σ*-algebra* A *is both reachable and fully abstract w.r.t.* Σ_{COL} *if and only if* $A \models \mathrm{REACH}(\Sigma_{\mathrm{COL}}) \wedge \mathrm{FA}(\Sigma_{\mathrm{COL}})$*.*

Now let Π_{IFOLEq} be a sound and complete proof system for infinitary first-order logic with equality (see [13]). We obtain a sound and complete proof system Π_{COL} for COL by adding to Π_{IFOLEq}, as an extra axiom, $\mathrm{REACH}(\Sigma_{\mathrm{COL}}) \wedge \mathrm{FA}(\Sigma_{\mathrm{COL}})$ (see [4] for details). The difficulty with the proof system Π_{COL} is that, in general, it uses infinitary formulas (and also infinitary proof rules of Π_{IFOLEq}). An alternative is to restrict to finitary formulas and to use only a particular set of infinitary proof rules (see the discussion in [2, Chapter 11]). The idea now is, instead of "capturing" reachability (full abstractness, respectively) by the infinitary axiom $\mathrm{REACH}(\Sigma_{\mathrm{COL}})$ ($\mathrm{FA}(\Sigma_{\mathrm{COL}})$, respectively), to "capture" it by specialized infinitary proof rules called infinitary induction (infinitary coinduction, respectively). These infinitary rules are necessary to ensure completeness.

Definition 18 (Infinitary induction). *Let* Σ_{COL} *be a COL-signature with underlying signature* Σ*. The* infinitary induction rule $\mathrm{iI}(\Sigma_{\mathrm{COL}})$ *associated to* Σ_{COL} *is defined by:*
$\mathrm{iI}(\Sigma_{\mathrm{COL}}) \stackrel{\mathrm{def}}{=} \{\mathrm{iI}(\Sigma_{\mathrm{COL}})_s \mid s \in S_{\mathrm{Cons}}\}$
where for each constrained sort $s \in S_{\mathrm{Cons}}$*,* $\mathrm{iI}(\Sigma_{\mathrm{COL}})_s$ *is defined by:*

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

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

Definition 19 (Infinitary coinduction). *Let* Σ_{COL} *be a COL-signature with underlying signature* Σ*. The* infinitary coinduction rule $\mathrm{iCI}(\Sigma_{\mathrm{COL}})$ *associated to* Σ_{COL} *is defined by:*
$\mathrm{iCI}(\Sigma_{\mathrm{COL}}) \stackrel{\mathrm{def}}{=} \{\mathrm{iCI}(\Sigma_{\mathrm{COL}})_s \mid s \in S_{\mathrm{State}}\}$
where for each state sort $s \in S_{\mathrm{State}}$*,* $\mathrm{iCI}(\Sigma_{\mathrm{COL}})_s$ *is defined by:*

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

[6] $\forall \mathrm{Var}(c)$ 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.

where φ denotes an arbitrary Σ-formula.

Now let Π_{FOLEq} be a sound and complete proof system for finitary first-order logic with equality. We obtain a sound and complete (semi-formal) proof system $\Pi^{\star}_{\mathrm{COL}}$ for COL by adding to the finitary proof system Π_{FOLEq} the extra infinitary proof rules $\mathrm{iI}(\Sigma_{\mathrm{COL}})$ and $\mathrm{iCI}(\Sigma_{\mathrm{COL}})$.

Theorem 4. *For any COL-signature Σ_{COL}, let*

$$\Pi^{\star}_{\mathrm{COL}} \stackrel{\mathrm{def}}{=} \Pi_{\mathrm{FOLEq}} \cup \mathrm{iI}(\Sigma_{\mathrm{COL}}) \cup \mathrm{iCI}(\Sigma_{\mathrm{COL}}).$$

Then for any basic COL-specification $\mathrm{SP}_{\mathrm{COL}} = \langle \Sigma_{\mathrm{COL}}, \mathrm{Ax}\rangle$ and any Σ-formula φ, we have:
$\mathrm{SP}_{\mathrm{COL}} \models_{\Sigma_{\mathrm{COL}}} \varphi$ *if and only if* $\mathrm{Ax} \vdash_{\Pi^{\star}_{\mathrm{COL}}} \varphi$.

In practice, for proving the infinitely many hypotheses $\varphi[t/x]$ of the rule $\mathrm{iI}(\Sigma_{\mathrm{COL}})_s$, one would use an induction scheme like structural induction with respect to the constructor terms $\mathcal{T}(\Sigma_{\mathrm{COL}})_s$. Similarly, to prove the infinitely many hypotheses $\varphi \Rightarrow \forall \mathrm{Var}(c).\ c[x] = c[y]$ of the rule $\mathrm{iCI}(\Sigma_{\mathrm{COL}})_s$, one would use a coinduction scheme according to the coinductive definition of the contexts $\mathcal{C}(\Sigma_{\mathrm{COL}})_{s \to s'}$ provided in Definition 5.

5 Conclusion

We have seen that the integration of the observational and the constructor-based logics presented in [11] and [5] leads to a powerful formalism which integrates observability and reachability concepts in a common institution. An important aspect, which has not been worked out here, concerns structuring mechanisms for specifications and structured proof systems which can be defined on top of the given institution by applying the institution-independent specification-building operators of [19] and the proof rules for structured specifications of [7]. Future research concerns the investigation of refinement notions for COL-specifications and corresponding proof methods.

References

[1] 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*, 2002. To appear.

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

[3] M. Bidoit and R. Hennicker. Observer complete definitions are behaviourally coherent. In *Proc. OBJ/CafeOBJ/Maude Workshop at FM'99*, pages 83–94. THETA, 1999. `http://www.lsv.ens-cachan.fr/Publis/PAPERS/CafeOBJ.ps`.

[4] M. Bidoit and R. Hennicker. On the integration of observability and reachability concepts. Research Report LSV-02-2, 2002. Long version of this paper. `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-2.rr.ps`.

[5] M. Bidoit, R. Hennicker, and A. Kurz. On the duality between observability and reachability. In *Proc. FOSSACS'01, LNCS 2030*, pages 72–87. Springer, 2001. Long version: `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2001-7.rr.ps`.

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

[7] T. Borzyszkowski. Completeness of a logical system for structured specifications. In *Recent Trends in Algebraic Development Techniques, LNCS 1376*, pages 107–121. Springer, 1998.

[8] R. Diaconescu and K. Futatsugi. *CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification*. AMAST Series in Computing 6. World Scientific, 1998.

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

[10] J. Goguen and G. Roşu. Hiding more of hidden algebra. In *Proc. FM'99, LNCS 1709*, pages 1704–1719. Springer, 1999.

[11] R. Hennicker and M. Bidoit. Observational logic. In *Proc. AMAST'98, LNCS 1548*, pages 263–277. Springer, 1999.

[12] M. Hofmann and D. Sannella. On behavioural abstraction and behavioural satisfaction in higher-order logic. *Theoretical Computer Science*, 167:3–45, 1996.

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

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

[15] M.P. Nivela and F. Orejas. Initial behaviour semantics for algebraic specifications. In *Recent Trends in Data Type Specification, LNCS 332*, pages 184–207. Springer, 1988.

[16] P. Padawitz. Swinging data types: syntax, semantics, and theory. In *Recent Trends in Data Type Specification, LNCS 1130*, pages 409–435. Springer, 1996.

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

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

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

[20] M. Wirsing and M. Broy. A modular framework for specification and information. In *Proc. TAPSOFT'89, LNCS 351*, pages 42–73. Springer, 1989.