Observational Logic

Rolf Hennicker*, Michel Bidoit**

*Institut für Informatik, Ludwig-Maximilians-Universität München
Oettingenstr. 67, D-80538 München, GERMANY
**Laboratoire Spécification et Vérification, CNRS & Ecole Normale Supérieure de Cachan
61, Avenue du Président Wilson, 94235 Cachan Cedex, FRANCE

Abstract. We present an institution of observational logic suited for state-based systems specifications. The institution is based on the notion of an observational signature (which incorporates the declaration of a distinguished set of observers) and on observational algebras whose operations are required to be compatible with the indistinguishability relation determined by the given observers. In particular, we introduce a homomorphism concept for observational algebras which adequately expresses observational relationships between algebras. Then we consider a flexible notion of observational signature morphism which guarantees the satisfaction condition of institutions w.r.t. observational satisfaction of arbitrary first-order sentences. From the proof theoretical point of view we construct a sound and complete proof system for the observational consequence relation. Then we consider structured observational specifications and we provide a sound and complete proof system for such specifications by using a general, institution-independent result of [6].

1 Introduction

In this paper we study a logical framework for the specification of the observable behaviour of software systems which is particularly suited for state-based systems but may also be used for specifying infinite data and behavioural properties of abstract data types. Formally, we introduce an institution of observational logic and we study proof methods for first-order observational properties of structured specifications built over this institution.

Although our approach is novel, it is influenced by previous behavioural approaches, in particular of [4, 11, 16, 21, 23]. The important difference to [4] is that in the present approach we use a built-in observational semantics which previously led to problems w.r.t. the encapsulation of observational properties of parts of a system specification (cf. [1, 15]). In the hidden sorted algebra approach (cf. e.g. [11]) encapsulation is achieved at the cost of a rather restrictive notion of signature morphism which recently was generalized in [8] (see the discussion in Section 5). Another popular formalism which deals with a state-based view of systems is provided in the framework of coalgebras (cf. e.g. [23, 16]). These approaches, however, have problems to deal with n-ary operations working on several non-observable (hidden) argument sorts which frequently occur in practice (see also Section 5). Moreover, coalgebraic approaches are based on terminal semantics while we are interested in a loose semantics in order to obtain sufficient freedom for the choice of implementations.

The starting point of our approach is a methodological consideration: A well-known method for specifying abstract data types (or, more concretely, for writing functional programs) is to determine first a set of constructor symbols which describe how the elements of the data type are constructed and then to define functions which can be applied to the data (usually by case distinction over the given constructors). An analogous method can be used for specifying state-based systems. First, a set of observer symbols is declared which determines an indistinguishability relation (also called observational equality) for the non-observable elements (i.e. for the states). Then the operations are specified, usually by describing their effects w.r.t. the given observers.

Formally, these considerations lead to our notion of an observational signature which contains a distinguished set of observer symbols. A similar idea was presented in [21] (and recently in [9] and [8]).[1] Based on the notion of an observational signature we define observational algebras as those structures whose operations are compatible with the observational equality (determined by the observers of the signature). In this way we obtain, in Section 3, a category of observational algebras with a notion of observational homomorphism that is suited to express observational relationships between algebras. Moreover, we establish a full and faithful functor from the category of observational algebras to the category of standard algebras which is compatible with the observational satisfaction relation defined in Section 4.

In Section 5, we introduce the institution of observational logic. It turns out that our general notion of an observer (used in observational signatures) allows us to define a powerful notion of observational signature morphism which guarantees, nevertheless, that the (observational) satisfaction condition of the institution is valid. Then, in Section 6, we define a sound and complete proof system for observational logic.

The results obtained so far allow us, first, to apply a generic construction of structured specifications over an arbitrary institution (cf. [24]), thus obtaining a basic language of structured observational specifications. Secondly, we can also apply a generic construction of a sound and complete proof system for structured specifications (cf. [6]) which leads to a corresponding proof system for structured observational specifications.

2 Algebraic Preliminaries

We assume the reader to be familiar with the basic notions of algebraic specifications (cf. e.g. [18]), like the notions of (many-sorted) *signature* $\Sigma = (S, OP)$ (with a set S of *sorts* and a set OP of *operation symbols* op: $s_1,\ldots,s_n \to s$), *signature morphism* $\sigma: \Sigma \to \Sigma'$, *total* Σ-*algebra* $A = ((A_s)_{s\in S}, (f^A)_{f\in F})$, Σ-*congruence*, Σ-*term algebra* $T(\Sigma, X)$, *valuation* $\alpha : X \to A$ and *interpretation* I_α: $T(\Sigma, X) \to A$. Throughout this paper we assume that the carrier sets A_s of a Σ-algebra are not empty and that $X = (X_s)_{s\in S}$ is a family of countably infinite sets X_s of variables of sort $s \in S$. The class of all Σ-algebras is denoted by $Alg(\Sigma)$. Together with Σ-homomorphisms this class forms a category, for simplicity also denoted by $Alg(\Sigma)$.

---

[1] Indeed our notion of an "observer" is a generalization of an "action" in the sense of [21] and of a "behavioural operation" in the sense of [8].

For any signature morphism $\sigma: \Sigma \to \Sigma'$ the *reduct functor* $_|_\sigma : Alg(\Sigma') \to Alg(\Sigma)$ is defined as usual. The *reduct* of a relation $\varphi' \subseteq A' \times B'$ w.r.t. $\sigma: \Sigma \to \Sigma'$ is denoted by $\varphi'|_\sigma$ where $\varphi'|_\sigma \subseteq A'|_\sigma \times B'|_\sigma$ is defined by $(\varphi'|_\sigma)_s =_{def} \varphi'_{\sigma(s)}$ for all $s \in S$.
The set of (many-sorted) *first-order Σ-formulas* is defined as usual whereby we will also admit infinitary Σ-formulas built by countably infinite conjunctions (or disjunctions). A *finitary* Σ-formula is a Σ-formula which contains no infinitary conjunction (disjunction resp.) and a Σ-*sentence* is a Σ-*formula* which contains no free variable. The *(standard) satisfaction relation*, denoted by $A \models \phi$, is defined as usual in the first-order predicate calculus (with a straightforward extension to infinitary formulas, cf. e.g. [17]). The notation $A \models \phi$ is extended in the usual way to classes of algebras and sets of formulas. A Σ-sentence ϕ is a *semantic consequence* of a set Φ of Σ-sentences, also denoted by $\Phi \models \phi$, if for any Σ-algebra A with $A \models \Phi$ we have $A \models \phi$.

3 The Category of Observational Algebras

An observational signature is a generalization of a standard algebraic signature with a distinguished set of observable sorts (determining the carrier sets of the observable values) and with a distinguished set of observer operations (determining the experiments that can be used to distinguish non-observable elements, often called "states"). An n-ary operation op: $s_1,\ldots, s_n \to s$ with several non-observable argument sorts may also be used as an observer (which is not the case in [21] and in [8]). In this case op is equipped with a "position number" $1 \le i \le n$ which indicates the argument sort of the states to be observed by op. For instance, if op: $s_1, s_2 \to s$ is a binary operation then we can declare either (op, 1) or (op, 2) or both, (op, 1) and (op, 2), as observer thus obtaining as much flexibility as needed in practical examples.

Definition 3.1 *(Observational signature)* Let $\Sigma = (S, OP)$ be a signature and $S_{Obs} \subseteq S$ be a set of *observable sorts*. An *observer* is a pair (op, i) where $(op: s_1,\ldots, s_n \to s) \in OP$ is an operation symbol such that $1 \le i \le n$ and $s_i \notin S_{Obs}$. (op, i) is a *direct observer* of s_i if $s \in S_{Obs}$; otherwise it is an *indirect observer*. If $op: s_1 \to s$ is a unary observer we will simply write op instead of (op, 1). An *observational signature* $\Sigma_{Obs} = (\Sigma, S_{Obs}, OP_{Obs})$ consists of a signature $\Sigma = (S, OP)$, a set $S_{Obs} \subseteq S$ of observable sorts and a set OP_{Obs} of observers (op, i) with $op \in OP$. ♦

Convention We implicitly assume in the following (if not stated otherwise) that whenever we consider an observational signature Σ_{Obs}, then $\Sigma_{Obs} = (\Sigma, S_{Obs}, OP_{Obs})$ with $\Sigma = (S, OP)$ and similarly for Σ'_{Obs} etc.

Example 3.2 The following is a simple observational signature for bank accounts with observer "bal" determining the balance of an account and an operation "update" subsuming the usual credit and debit operations. Here and in the following examples we use postfix notation for unary operations and infix notation for binary operations.

sorts {account, int}
observable sorts {int}
observers { _.bal: account → int}
operations {new: → account, _.update_ : account, int → account}

A more advanced signature for bank accounts may be obtained by introducing also an indirect observer "_ .undo: account → account" intended to reconstruct the previous state of an account after having performed an action. ♦

Any observational signature determines a set of observable contexts which represent those experiments which allow us to distinguish elements by the given observers.

Definition 3.3 *(Σ_{Obs}-context)* Let Σ_{Obs} be an observational signature, let $X = (X_s)_{s \in S}$ be the generally assumed family of variable sets and let $Z = (\{z_s\})_{s \in S}$ be a disjoint S-sorted family of singleton sets. For all s, s' ∈ S the set $C(\Sigma_{Obs})_{s \to s'}$ of Σ_{Obs}-*contexts* with "application sort" s and "result sort" s' is inductively defined as follows:

(1) For each $s \in S$, $z_s \in C(\Sigma_{Obs})_{s \to s}$.

(2) For each $(op, i) \in OP_{Obs}$ with $op: s_1, \ldots, s_n \to s'$, for each $c \in C(\Sigma_{Obs})_{s \to s_i}$ and pairwise disjoint variables $x_1, \ldots, x_n$ (not occurring in c) of sort $s_1, \ldots, s_n$,

$$op(x_1, \ldots, x_{i-1}, c, x_{i+1}, \ldots, x_n) \in C(\Sigma_{Obs})_{s \to s'}.$$

Each context $c \in C(\Sigma_{Obs})_{s \to s'}$ contains, besides variables in X, exactly one occurrence of the "context variable" z_s. The application of a context $c \in C(\Sigma_{Obs})_{s \to s'}$ to a term t of sort s, denoted by c[t], is the term obtained by substituting the term t for z_s.
An *observable* Σ_{Obs}-*context* is a Σ_{Obs}-context with observable result sort $s' \in S_{Obs}$. We denote by $C(\Sigma_{Obs})_{s \to S_{Obs}}$ the set of observable contexts with application sort s. ♦

In Example 3.2 the only observable context is "$z_{account}$.bal". If we additionally use the indirect observer "undo" then there are infinitely many observable contexts of the form "$z_{account}$.undo.undo … .bal".
Elements which cannot be distinguished by the experiments of an observational signature are considered to be observationally equal, formally defined as follows.

Definition 3.4 *(Σ_{Obs}-equality)* Let Σ_{Obs} be an observational signature. For any Σ-algebra $A \in Alg(\Sigma)$ the *observational* Σ_{Obs}-*equality* on A is denoted by $\approx_{\Sigma_{Obs},A}$ and defined by:
For all $s \in S$, two elements $a, b \in A_s$ are observationally equal w.r.t. Σ_{Obs} i.e. $a \approx_{\Sigma_{Obs},A} b$, if and only if for all observable contexts $c \in C(\Sigma_{Obs})_{s \to S_{Obs}}$ and for all valuations $\alpha, \beta: X \cup \{z_s\} \to A$ with $\alpha(x) = \beta(x)$ if $x \in X$, $\alpha(z_s) = a$, $\beta(z_s) = b$, we have $I_\alpha(c) = I_\beta(c)$. Obviously, if s is an observable sort, then for all $a, b \in A$, $a \approx_{\Sigma_{Obs},A} b$ is equivalent to $a = b$. ♦

For any Σ-algebra A, $\approx_{\Sigma_{Obs},A}$ is an equivalence relation on A. But it is important to note that for an arbitrary Σ-algebra A there may exist (non-observer) operations which are not compatible with the observational equality $\approx_{\Sigma_{Obs},A}$, i.e. $\approx_{\Sigma_{Obs},A}$ is in general not a Σ-congruence on A.
In this paper we follow the loose semantics approach to algebraic specifications where a specification can be considered as a description of all admissible implementations (represented by the models of the specification). The basic assumption of the present approach is that, having declared a set of observers, an implementation can only be admissible if all its operations respect the observational equality determined by the

given observers. Formally, this is expressed by the following notion of an observational algebra.

Definition 3.5 *(Observational algebra)* Let Σ_{Obs} be an observational signature. An *observational* Σ_{Obs}*-algebra* is a Σ-algebra A such that $\approx_{\Sigma_{Obs},A}$ is a Σ-congruence on A.[2] The class of all observational Σ_{Obs}-algebras is denoted by $Alg_{Obs}(\Sigma_{Obs})$. ♦

Note that in the special case where all operations op ∈ OP are declared as observers (for each non-observable argument sort) any Σ-algebra is an observational Σ_{Obs}-algebra and $\approx_{\Sigma_{Obs},A}$ is just the (total) observational equality of elements defined in [4][3] and similarly in other approaches in the literature like, for instance, in [22, 11].
Let us now point out the relationship to the coalgebraic framework (cf. e.g. [16, 23]). For this purpose assume that $\Sigma_{Obs} = (\Sigma, S_{Obs}, OP_{Obs})$ is an observational signature such that for any observer (op, i) ∈ OP_{Obs}, s_i is the only non-observable argument sort of op. Moreover, assume that any observable sort s ∈ S_{Obs} is interpreted in any observational Σ_{Obs}-algebra by the same fixed set of observable values (e.g. integers, booleans etc.). Then a (polynomial) functor T: Set → Set can be associated to OP_{Obs} which captures the functionality of the observer symbols.[4] Any observational Σ_{Obs}-algebra A is an extension of a T-coalgebra C which has the same carrier sets as C and defines the non-observer operations op ∈ OP\$OP_{Obs}$ on top of C. The fact that the extension A is an observational algebra is equivalent to the fact that each operation of OP preserves bisimilarity of elements (cf. e.g. [16]).

In order to obtain a category of observational algebras we still need an appropriate morphism notion. Of course, since any observational algebra is a Σ-algebra, one could simply use standard homomorphisms between Σ-algebras. But this does not reflect the relationships between the *observable behaviour* of algebras. Therefore, we have chosen another definition where an observational homomorphism is defined as an appropriate relation which is compatible with observational equalities.

Definition 3.6 *(Observational homomorphism)* Let A, B ∈ $Alg_{Obs}(\Sigma_{Obs})$. An *observational* Σ_{Obs}*-homomorphism* φ: A → B is an S-sorted family $(\varphi_s)_{s\in S}$ of relations $\varphi_s \subseteq A_s \times B_s$ with the following properties for all s ∈ S:
(1) For all a ∈ A_s there exists b ∈ B_s such that a φ_s b.
(2) For all a ∈ A_s, b, b' ∈ B_s, if a φ_s b then (a φ_s b' if and only if b $\approx_{\Sigma_{Obs},B}$ b').
(3) For all a, a' ∈ A_s, b ∈ B_s, if a φ_s b and a $\approx_{\Sigma_{Obs},A}$ a' then a' φ_s b.
(4) For all (op: $s_1,\ldots, s_n \to s$) ∈ OP and $a_i \in A_{s_i}$, $b_i \in B_{s_i}$,
if a_i φ_{s_i} b_i for i = 1,…, n then $op^A(a_1,\ldots, a_n)$ φ_s $op^B(b_1,\ldots, b_n)$. ♦

---

[2] Obviously, for this it is sufficient that all non-observer operations are compatible with $\approx_{\Sigma_{Obs},A}$.

[3] In [4] also partial observational equalities are considered. It should be straightforward to extend our approach to this case.

[4] If there is more than one non-observable sort in S\S_{Obs} then a set of functors has to be associated to Σ_{Obs}.

Theorem 3.7 *(The category of observational algebras)*
For each observational signature Σ_{Obs}, the class $Alg_{Obs}(\Sigma_{Obs})$ together with observational Σ_{Obs}-homomorphisms is a category which, by abuse of notation, will also be denoted by $Alg_{Obs}(\Sigma_{Obs})$. Thereby the composition of observational homomorphisms is the usual composition of relations and, for each observational Σ_{Obs}-algebra A, the identity id_A: A $\rightarrow$ A in the category $Alg_{Obs}(\Sigma_{Obs})$ is the observational equality $\approx_{\Sigma_{Obs},A}$. (The proof is straightforward; see [14].) ♦

Since for any observational Σ_{Obs}-algebra A the observational equality $\approx_{\Sigma_{Obs},A}$ is a Σ-congruence, we can construct the quotient algebra $A/\approx_{\Sigma_{Obs},A}$ which identifies all elements of A which are indistinguishable "from the outside". $A/\approx_{\Sigma_{Obs},A}$ can be considered as the "black box view" of A thus representing the "observable behaviour" of A w.r.t. Σ_{Obs}. Using this behaviour construction we obtain (for any observational signature Σ_{Obs}) a functor from the category $Alg_{Obs}(\Sigma_{Obs})$ of observational algebras into the category $Alg(\Sigma)$ of (standard) Σ-algebras which establishes a one to one correspondence between observational homomorphisms φ: A $\rightarrow$ B and standard homomorphisms h: $A/\approx_{\Sigma_{Obs},A} \rightarrow B/\approx_{\Sigma_{Obs},B}$, i.e. the functor is full and faithful.

Theorem 3.8 *(Behaviour functor)* For any observational signature Σ_{Obs}, $\mathcal{F}_{\Sigma_{Obs}}$: $Alg_{Obs}(\Sigma_{Obs}) \rightarrow Alg(\Sigma)$ is a full and faithful functor where $\mathcal{F}_{\Sigma_{Obs}}$ is defined by:
For each A $\in$ $Alg_{Obs}(\Sigma_{Obs})$, $\mathcal{F}_{\Sigma_{Obs}}(A) =_{def} A/\approx_{\Sigma_{Obs},A}$ and
for each observational Σ_{Obs}-homomorphism φ: A $\rightarrow$ B,

$$\mathcal{F}_{\Sigma_{Obs}}(\varphi): A/\approx_{\Sigma_{Obs},A} \rightarrow B/\approx_{\Sigma_{Obs},B} \text{ is defined by } \mathcal{F}_{\Sigma_{Obs}}(\varphi)([a]) =_{def} [b] \text{ if a } \varphi \text{ b.}$$

(For A $\in$ $Alg_{Obs}(\Sigma_{Obs})$, $\mathcal{F}_{\Sigma_{Obs}}(A)$ is called the *observational behaviour* of A. The proof of the theorem is straightforward; see [14].) ♦

Remark 3.9 Since $\mathcal{F}_{\Sigma_{Obs}}$ is full and faithful, it is obvious that two observational algebras are observationally isomorphic if and only if they have isomorphic behaviours. Hence, as a consequence of a result in [4], observational isomorphism coincides with usual notions of observational equivalence between algebras (cf. e.g. [22]).[5] This also points out the adequacy of our morphism notion. ♦

4 Observational Satisfaction

The underlying idea of the observational satisfaction relation is to interpret the equality symbol "=" occurring in a first-order formula ϕ not by the set-theoretic equality but by the observational equality of elements.

Definition 4.1 The *observational satisfaction relation* between observational Σ_{Obs}-algebras and Σ-formulas is denoted by $\models_{\Sigma_{Obs}}$ and defined as follows:
(1) For any two terms t, r $\in$ $T(\Sigma, X)_s$ of the same sort s and for any valuation α: X $\rightarrow$ A, A, $\alpha \models_{\Sigma_{Obs}} t = r$ holds if $I_\alpha(t) \approx_{\Sigma_{Obs},A} I_\alpha(r)$.

[5] To our knowledge [20] is the first paper where observational equivalence of algebras is characterized by isomorphism of some category.

(2) For any arbitrary Σ-formula ϕ and for any valuation α: $X \to A$, $A, \alpha \models_{\Sigma_{Obs}} \phi$ is defined by induction over the structure of the formula ϕ in the usual way.
(3) For any arbitrary Σ-formula ϕ, $A \models_{\Sigma_{Obs}} \phi$ holds if for all valuations α: $X \to A$, $A, \alpha \models_{\Sigma_{Obs}} \phi$ holds.

The notation $A \models_{\Sigma_{Obs}} \phi$ is extended in the usual way to classes of observational algebras and sets of formulas. ♦

Technically the observational satisfaction relation could be defined in the same way for arbitrary Σ-algebras which do not necessarily belong to $Alg_{Obs}(\Sigma_{Obs})$. This is the approach of extended hidden algebra (cf. [8]) where a special predicate symbol "~" is introduced for representing the observational equality of (non-observable) elements. But then the congruence rule of the equational calculus is only sound w.r.t. "~" if one can prove that all operations of a specification are "behaviourally coherent", i.e. are compatible with the given observational equality.[6]

Definition 4.2 *(Observational consequence)* A Σ-sentence ϕ is an *observational consequence* of a set Φ of Σ-sentences, also denoted by $\Phi \models_{\Sigma_{Obs}} \phi$, if for any observational Σ_{Obs}-algebra A, $A \models_{\Sigma_{Obs}} \Phi$ implies $A \models_{\Sigma_{Obs}} \phi$. ♦

The next proposition shows that the behaviour functor defined in Theorem 3.8 is compatible with the observational and the standard satisfaction relations. (For the proof see [14].)

Proposition 4.3 For any $A \in Alg_{Obs}(\Sigma_{Obs})$ and any Σ-formula ϕ,
$A \models_{\Sigma_{Obs}} \phi$ if and only if $\mathcal{F}_{\Sigma_{Obs}}(A) \models \phi$.[7] ♦

As a consequence of Remark 3.9 and Proposition 4.3 we can generalize Scott's theorem (cf. e.g. [17]) to observational algebras and observational satisfaction (taking into account that Σ-formulas may be infinitary; cf. Section 2).

Corollary 4.4 *(Observational version of Scott's theorem)*[8]
Let $A, B \in Alg_{Obs}(\Sigma_{Obs})$ be two observational Σ_{Obs}-algebras such that $\mathcal{F}_{\Sigma_{Obs}}(A)$ and $\mathcal{F}_{\Sigma_{Obs}}(B)$ are countable. The following conditions are equivalent:
(1) A and B are observationally isomorphic.
(2) For all (possibly infinitary) Σ-formulas ϕ, $A \models_{\Sigma_{Obs}} \phi$ if and only if $B \models_{\Sigma_{Obs}} \phi$. ♦

We are now able to define the syntax and semantics of flat observational specifications. Structured specifications will be considered in Section 7.

---

[6] The idea of introducing a denotation "~" for observational equalities is suggested in [2, 3] as a proof-theoretic means for proving behavioural theorems and implementation correctness.

[7] In an abstract category-theoretic setting this fact can be used as a definition of behavioural satisfaction as in [7] and [5].

[8] A related result, but formulated in terms of observational equivalence of algebras instead of observational isomorphism, is given in [4].

Definition 4.5 A *flat observational specification* $SP = \langle \Sigma_{Obs}, Ax \rangle$ consists of an observational signature $\Sigma_{Obs} = (\Sigma, S_{Obs}, OP_{Obs})$ and a set Ax of Σ-sentences, called the *axioms* of SP. The semantics of SP is given by its signature $Sig_{Obs}(SP)$ and by its class of models $Mod_{Obs}(SP)$ which are defined by

$$Sig_{Obs}(SP) =_{def} \Sigma_{Obs},\ Mod_{Obs}(SP) =_{def} \{A \in Alg_{Obs}(\Sigma_{Obs}) \mid A \models_{\Sigma_{Obs}} Ax\}.\ \blacklozenge$$

For any observational specification SP, the class $Mod_{Obs}(SP)$ is closed under observational isomorphisms.

Example 4.6 The following specification of bank accounts has additionally to the account operations of Example 3.2 an operation "paycharge" which reduces the balance of an account by a constant monthly fee.

spec ACCOUNT =
  sorts {account, int}
  observable sorts {int}
  observers { _.bal: account → int, _.undo: account → account}
  operations "operations for the integers" ∪
    {new: → account, _.update_ : account, int → account,
    _.paycharge: account → account}
  axioms "axioms for the integers" ∪
    {∀x: int, s: account.
    new.bal = 0, new.undo = new,
    s.update(x).bal = s.bal+x, s.update(x).undo = s,
    s.paycharge.bal = s.bal-10, s.paycharge.undo = s}

A possible model of the specification ACCOUNT which satisfies the axioms even literally can be defined in terms of lists of integers. Another model which satisfies the axioms observationally (but not literally) can be constructed by using the well-known array with pointer realization of lists.
In the above specification the behaviour of the operations is uniquely specified w.r.t. the given observers. A proper loose specification can be obtained, for instance, by removing the equations for the "paycharge" operation. Then the semantics of the specification is still restricted to those models where the interpretation of "paycharge" is compatible with the given observational equality (since only observational algebras are admissible models). ♦

5 The Institution of Observational Logic

The category of observational algebras is the basis for defining an institution (cf. [10]) of observational logic which captures the model-theoretic view of the observable behaviour of systems. An essential ingredient to build an institution is an appropriate morphism notion for observational signatures which is defined as follows.

Definition 5.1 *(Observational signature morphism)* Let $\Sigma_{Obs} = (\Sigma, S_{Obs}, OP_{Obs})$ and $\Sigma'_{Obs} = (\Sigma', S'_{Obs}, OP'_{Obs})$ be two observational signatures with $\Sigma = (S, OP)$ and $\Sigma' = (S', OP')$. An *observational signature morphism* $\sigma: \Sigma_{Obs} \to \Sigma'_{Obs}$ is a signature morphism $\sigma: \Sigma \to \Sigma'$ such that the following conditions are satisfied:

(1) For all $s \in S$, $s \in S_{Obs}$ if and only if $\sigma(s) \in \sigma(S_{Obs})$.
(2) If $(op, i) \in OP_{Obs}$ then $(\sigma(op), i) \in OP'_{Obs}$.
(3) If $(op', i) \in OP'_{Obs}$ such that $op': s_1', \ldots, s_n' \rightarrow s'$ and $s_i' = \sigma(s_i)$ for some $s \in S$ then there exists $(op, i) \in OP_{Obs}$, $op: s_1, \ldots, s_n \rightarrow s$ such that $op' = \sigma(op)$. ♦

Condition (1) is standard. It requires that observable and non-observable sorts are preserved by σ. Condition (2) requires that also observers are preserved by σ. Condition (3) is essential for the satisfaction condition presented below. It says that whenever the image $\sigma(s)$ of some "old" sort s of Σ_{Obs} is observed by an observer op' of Σ'_{Obs} then there must be a corresponding observer op of Σ_{Obs} which observes s and which is mapped to op'. Thus no "new" observations can be introduced for "old" sorts. However, it is important to note that there is still sufficient flexibility for the following:
1. We can introduce new operation symbols in $OP' \backslash OP'_{Obs}$ which are *not* in the image of σ but nevertheless may have argument sorts which *are in the image* of σ. A standard example may be given by the signature of a specification of a bank which is based on (i.e. imports) a specification of accounts. Then the bank specification may introduce a new operation "_.add_: bank, account → bank" for adding an account to a bank. This is not a problem as long as "add" is not used as an observer for accounts (i.e. (add, 2) is not an observer which indeed would be strange). Examples like this, where some argument sorts of an operation are imported from another signature frequently occur in practice, in particular, in object-oriented programming. This situation cannot be dealt with by hidden signature morphisms; cf. e.g. [19]. Extended hidden algebra, however, solves this problem; cf. [8].
2. We can introduce new observers (op', i) in OP'_{Obs} as long as the observed sort s_i' is not in the image of σ. For instance, in the bank example one has definitely to introduce some observer(s) for the new sort "bank". This can neither be done in hidden algebra nor in extended hidden algebra.

Definition 5.2 *(Observational reduct functor)* For any observational signature morphism $\sigma: \Sigma_{Obs} \rightarrow \Sigma'_{Obs}$, $Alg_{Obs}(\sigma): Alg_{Obs}(\Sigma'_{Obs}) \rightarrow Alg_{Obs}(\Sigma_{Obs})$ is defined by:
For each $A' \in Alg_{Obs}(\Sigma'_{Obs})$, $Alg_{Obs}(\sigma)(A') =_{def} A'|_\sigma$
For each observational Σ'_{Obs}-homomorphism $\varphi': A' \rightarrow B'$,

$$Alg_{Obs}(\sigma)(\varphi'): A'|_\sigma \rightarrow B'|_\sigma \text{ is defined by } Alg_{Obs}(\sigma)(\varphi') =_{def} \varphi'|_\sigma$$

(See Section 2 for the definition of the reducts $A'|_\sigma$ and $\varphi'|_\sigma$) ♦

The following lemma is essential for proving that $Alg_{Obs}(\sigma)$ is indeed a well-defined functor (cf. Theorem 5.4) and also for checking the (observational) satisfaction condition (cf. Theorem 5.5). It says that the observational equality is preserved by observational reducts. (The proof of the lemma and of the subsequent theorems is given in [14].)

Lemma 5.3 For any observational signature morphism $\sigma: \Sigma_{Obs} \rightarrow \Sigma'_{Obs}$ and observational Σ'_{Obs}-algebra $A' \in Alg_{Obs}(\Sigma'_{Obs})$, $(\approx_{\Sigma'_{Obs}, A'})|_\sigma = \approx_{\Sigma_{Obs}, (A'|_\sigma)}$. ♦

Theorem 5.4 For any observational signature morphism $\sigma: \Sigma_{Obs} \rightarrow \Sigma'_{Obs}$, $Alg_{Obs}(\sigma): Alg_{Obs}(\Sigma'_{Obs}) \rightarrow Alg_{Obs}(\Sigma_{Obs})$ is a well-defined functor. ♦

We are now able to state the satisfaction condition for observational logic which generalizes the satisfaction condition for hidden algebra. It guarantees encapsulation in the sense that observational properties are respected when composing specifications. In Section 7 we will consider structured specifications and we will see how a straightforward sound and complete proof system for structured specifications can be constructed which needs the validity of the satisfaction condition.

Theorem 5.5 *(Observational satisfaction condition)* Let $\sigma: \Sigma_{Obs} \rightarrow \Sigma'_{Obs}$ be an observational signature morphism. For any $A' \in Alg_{Obs}(\Sigma'_{Obs})$ and Σ-sentence ϕ,

$$A' |=_{\Sigma'_{Obs}} \sigma(\phi) \text{ if and only if } Alg_{Obs}(\sigma)(A') |=_{\Sigma_{Obs}} \phi$$

where $\sigma(\phi)$ is the usual extension of a signature morphism to Σ-sentences. ♦

Corollary 5.6 *(The institution of observational logic)*
The quadruple $INS_{Obs} = (Sig_{Obs}, Sen_{IFOLEQ}, Alg_{Obs}, |=_{Obs})$ is an institution whereby:

- Sig_{Obs} is the category of observational signatures and observational signature morphisms.
- The functor $Sen_{IFOLEQ}: Sig_{Obs} \rightarrow$ Set maps
  - each observational signature $\Sigma_{Obs} = (\Sigma, S_{Obs}, OP_{Obs})$ to the set of (possibly infinitary) many-sorted first-order Σ-sentences (cf. Section 2) and
  - each observational signature morphism $\sigma: \Sigma_{Obs} \rightarrow \Sigma'_{Obs}$ to the obvious translation function which transforms Σ-sentences into Σ'-sentences.
- The functor $Alg_{Obs}: (Sig_{Obs})^{op} \rightarrow$ Cat maps
  - each observational signature Σ_{Obs} to the category $Alg_{Obs}(\Sigma_{Obs})$ of observational Σ_{Obs}-algebras and observational Σ_{Obs}-homomorphisms and
  - each observational signature morphism $\sigma: \Sigma_{Obs} \rightarrow \Sigma'_{Obs}$ to the observational reduct functor $Alg_{Obs}(\sigma): Alg_{Obs}(\Sigma'_{Obs}) \rightarrow Alg_{Obs}(\Sigma_{Obs})$.
- $|=_{Obs} = (|=_{\Sigma_{Obs}})_{\Sigma_{Obs} \in |Sig_{Obs}|}$ where, for each observational signature Σ_{Obs}, $|=_{\Sigma_{Obs}}$ is the observational satisfaction relation of a Σ-sentence by an observational Σ_{Obs}-algebra. ♦

According to Proposition 4.3 the family $(\mathcal{F}_{\Sigma_{Obs}})_{\Sigma_{Obs} \in |Sig_{Obs}|}$ of (full and faithful) functors $\mathcal{F}_{\Sigma_{Obs}}: Alg_{Obs}(\Sigma_{Obs}) \rightarrow Alg(\Sigma)$ can be extended to an institution morphism (cf. [10]) which maps the institution of observational logic to the institution of (standard) infinitary first order logic.

6 A Proof System for Observational Logic

In this section we study the proof theory for observational logic. For defining an appropriate proof system we first associate to any observational signature Σ_{Obs} the following set $FA_{\Sigma_{Obs}}$ of Σ-sentences.

Definition 6.1 Let $\Sigma_{Obs} = (\Sigma, S_{Obs}, OP_{Obs})$ be an observational signature with $\Sigma = (S, F)$. $FA_{\Sigma_{Obs}} =_{def} \{FA_{\Sigma_{Obs}}(s) \mid s \in S \backslash S_{Obs}\}$ where for any $s \in S \backslash S_{Obs}$,

$$FA_{\Sigma_{Obs}}(s) =_{def} \forall\, x_L, x_R{:}s.\ \Big(\bigwedge_{c \in C(\Sigma_{Obs})_{s \to S_{Obs}}} \forall\, Var(c).\ c[x_L] = c[x_R]\Big) \Rightarrow x_L = x_R.$$

Thereby Var(c) denotes the set of all variables occurring in c besides the context variable z_s. ♦

The underying idea for considering $FA_{\Sigma_{Obs}}$ stems from a result in [4] where it is shown that the behavioural theory of a class C of Σ-algebras coincides with the standard theory of the fully abstract algebras of C. The following theorem shows that indeed the sentences $FA_{\Sigma_{Obs}}$ allow us to characterize the observational consequence relation in terms of the standard consequence relation. (For the proof see [14].)

Theorem 6.2 Let Σ_{Obs} be an observational signature, Φ be a set of Σ-sentences and ϕ be a Σ-sentence. $\Phi \models_{\Sigma_{Obs}} \phi$ if and only if $\Phi \cup FA_{\Sigma_{Obs}} \models \phi$. ♦

In the sequel we assume given, for each signature Σ, a sound and complete proof system $\Pi(\Sigma)$ for (many-sorted) infinitary first-order logic (see the discussion below). The proof system $\Pi(\Sigma_{Obs})$ for observational logic is then constructed by adding to the axioms and rules of $\Pi(\Sigma)$ the sentences $FA_{\Sigma_{Obs}}$ as further axioms.

Definition 6.3 *(Proof system for observational logic)*
For any observational signature Σ_{Obs}, $\Pi(\Sigma_{Obs}) =_{def} \Pi(\Sigma) \cup FA_{\Sigma_{Obs}}$.
We write $\Phi \vdash_{\Sigma_{Obs}} \phi$ ($\Phi \vdash_{\Sigma} \phi$ resp.) if ϕ is a Σ-sentence that can be deduced from a set Φ of Σ-sentences by the axioms and rules of $\Pi(\Sigma_{Obs})$ ($\Pi(\Sigma)$ resp.). ♦

Corollary 6.4 *(Soundness and completeness)* For any observational signature Σ_{Obs}, set Φ of Σ-sentences and Σ-sentence ϕ, $\Phi \vdash_{\Sigma_{Obs}} \phi$ if and only if $\Phi \models_{\Sigma_{Obs}} \phi$.

Proof: $\Phi \vdash_{\Sigma_{Obs}} \phi$ iff, by definition of $\Pi(\Sigma_{Obs})$, $\Phi \cup FA_{\Sigma_{Obs}} \vdash_{\Sigma} \phi$ iff, by soundness and completeness of $\Pi(\Sigma)$, $\Phi \cup FA_{\Sigma_{Obs}} \models \phi$ iff, by Theorem 6.2, $\Phi \models_{\Sigma_{Obs}} \phi$. ♦

The axioms $FA_{\Sigma_{Obs}}$ can be considered as a coinductive proof principle (cf. e.g. [16]) which, together with $\Pi(\Sigma)$, allows us to prove the observational validity not only of equations but of arbitrary first-order formulas. If Σ_{Obs} contains only direct observers there exist (up to α-conversion) only finitely many observable contexts and hence $FA_{\Sigma_{Obs}}$ is finitary. In this case $\Pi(\Sigma)$ can be chosen as a formal (i.e. finitary) proof system and any available theorem prover for first-order logic can be used to prove that ϕ is an observational consequence of Φ.[9]
If Σ_{Obs} contains indirect observers there may be infinitely many observable contexts and then $FA_{\Sigma_{Obs}}$ contains infinitary conjunctions. In this case we can choose for $\Pi(\Sigma)$ a proof system for infinitary first-order logic (for instance, the many-sorted variant of

[9] For instance, using the Larch Prover one can directly implement the axioms $FA_{\Sigma_{Obs}}$ by the "partitioned by" construct of LP; cf. [12].

the proof system in [17]). Then the above completeness result is mainly of theoretical interest. However, it is important to note that the infinitary formulas $FA_{\Sigma_{Obs}}$ can still be very useful because in practical examples the validity of (an instantiation of) the infinitary conjunction of $FA_{\Sigma_{Obs}}$ can often be verified by an induction proof (cf. Example 6.5 below). Using a result of [3] it is even possible to encode the infinitary formulas $FA_{\Sigma_{Obs}}$ by finitary ones if one introduces auxiliary symbols and reachability constraints. Hence the problem of the non-completeness of finitary proof systems for observational logic corresponds exactly to the non-completeness of finitary proof systems for inductively defined data types (in particular of arithmetic).

Example 6.5 Consider the signature of the ACCOUNT specification of Example 4.6. It induces the infinitary sentence FA(account) $=_{def}$

$$\forall\, s_L, s_R\text{: account. } (\bigwedge_{i \in \mathbf{N}} s_L.undo^i.bal = s_R.undo^i.bal) \Rightarrow s_L = s_R.$$

Now consider the implicitly universally quantified equation

$$s.paycharge = s.update(-10).$$

It is easy to prove by induction that for all $i \in \mathbf{N}$,

$$s.paycharge.undo^i.bal = s.update(-10).undo^i.bal$$

can be derived from the axioms of ACCOUNT. Then, using FA(account), we deduce

$$s.paycharge = s.update(-10)$$

and therefore, by Corollary 6.4, this equation is an observational consequence of the ACCOUNT specification. ♦

7 Structured Observational Specifications

In [6] (and similarly in [24]) a basic set of specification-building operations is defined which allows one to build structured specifications over an arbitrary institution. We will now apply these operators to the particular institution of observational logic thus obtaining the following set of operations for constructing structured observational specifications. The semantics of such a specification SP is determined by its (observational) signature, denoted by $Sig_{Obs}(SP)$, and by its class of models, denoted by $Mod_{Obs}(SP)$. In the following definition we assume that σ: $\Sigma_{Obs} \rightarrow \Sigma'_{Obs}$ is an injective observational signature morphism.[10]

basic: Any presentation $\langle \Sigma_{Obs}, Ax \rangle$ is an observational specification. Its semantics is defined in Definition 4.5.

union: For any two observational specifications SP1 and SP2 with $Sig_{Obs}(SP1) = Sig_{Obs}(SP2)$, the expression SP1 $\cup$ SP2 is an observational specification with semantics

$$Sig_{Obs}(SP1 \cup SP2) =_{def} Sig_{Obs}(SP1),$$
$$Mod_{Obs}(SP1 \cup SP2) =_{def} Mod_{Obs}(SP1) \cap Mod_{Obs}(SP2).$$

[10] The injectivity requirement ensures that the interpolation property for institutions (cf. [6]) needed for the completeness proof holds. Whether the interpolation property holds without this assumption seems to be an open question.

translate: For any observational specification SP with $Sig_{Obs}(SP) = \Sigma_{Obs}$, the expression **translate** SP **by** σ is an observational specification with semantics

Sig_{Obs}(**translate** SP **by** σ) $=_{def} \Sigma'_{Obs}$,
Mod_{Obs}(**translate** SP **by** σ) $=_{def} \{A' \in Alg_{Obs}(\Sigma'_{Obs}) \mid Alg_{Obs}(\sigma)(A') \in Mod_{Obs}(SP)\}$.

derive: For any observational specification SP' with $Sig_{Obs}(SP') = \Sigma'_{Obs}$, the expression **derivefrom** SP' **by** σ is an observational specification with semantics

Sig_{Obs}(**derivefrom** SP' **by** σ) $=_{def} \Sigma_{Obs}$,
Mod_{Obs}(**derivefrom** SP' **by** σ) $=_{def} \{Alg_{Obs}(\sigma)(A') \mid A' \in Mod_{Obs}(SP')\}$.

Definition 7.1 Let SP be an observational specification with signature Σ_{Obs}. A Σ-sentence ϕ is called an *observational theorem* of SP, written $SP \models_{\Sigma_{Obs}} \phi$, if $Mod_{Obs}(SP) \models_{\Sigma_{Obs}} \phi$. ♦

In the following we are interested in a proof system which allows us to prove observational theorems of structured (observational) specifications. For this purpose we instantiate the institution-independent proof system of [6] and obtain the following rules which generate, for each observational signature Σ_{Obs}, a relation $\vdash_{\Sigma_{Obs}}$ between observational specifications SP and Σ-sentences ϕ.

(pi-obs) $\dfrac{SP \vdash_{\Sigma_{Obs}} \phi_i \text{ for } i \in I, \{\phi_i \mid i \in I\} \vdash_{\Sigma_{Obs}} \phi}{SP \vdash_{\Sigma_{Obs}} \phi}$

(basic) $\dfrac{\phi \in Ax}{\langle \Sigma_{Obs}, Ax \rangle \vdash_{\Sigma_{Obs}} \phi}$

(union-1) $\dfrac{SP1 \vdash_{\Sigma_{Obs}} \phi}{SP1 \cup SP2 \vdash_{\Sigma_{Obs}} \phi}$

(union-2) $\dfrac{SP2 \vdash_{\Sigma_{Obs}} \phi}{SP1 \cup SP2 \vdash_{\Sigma_{Obs}} \phi}$

(translate) $\dfrac{SP \vdash_{\Sigma_{Obs}} \phi}{\textbf{translate } SP \textbf{ by } \sigma \vdash_{\Sigma'_{Obs}} \sigma(\phi)}$

(derive) $\dfrac{SP' \vdash_{\Sigma'_{Obs}} \sigma(\phi)}{\textbf{derivefrom } SP' \textbf{ by } \sigma \vdash_{\Sigma_{Obs}} \phi}$

According to the rule (pi-obs) the proof system for structured specifications is based on the proof system $\Pi(\Sigma_{Obs})$ for observational logic (cf. Section 6). The other rules correspond to the specification-building operations and hence proofs of observational theorems can be performed according to the structure of a given specification. An institution-independent proof of the soundness of the above rules is presented in [24]. The completeness can be checked by applying the results of [6] to the insti-tution INS_{Obs} of observational logic. For this purpose one has to show that INS_{Obs} satisfies the amalgamation and interpolation properties which is detailed in [14].

Theorem 7.2 *(Soundness and completeness)* Let SP be an observational specification with signature Σ_{Obs} and let ϕ be a Σ-sentence.

$SP \models_{\Sigma_{Obs}} \phi$ if and only if $SP \vdash\!\!\!\vdash_{\Sigma_{Obs}} \phi$. ♦

8 Conclusion

Observational logic provides a formal foundation for an observational specification methodology for state-based systems which works quite analogously to functional specifications of reachable data structures. In the latter case one usually starts by declaring a set of data type constructors. Similarly, in the observational case one starts by declaring a set of observers which do not tell us how elements are constructed but how elements can be observed. While data type constructors induce a generation principle which restricts the admissible models of a specification to reachable algebras, observer operations induce an observational equality which restricts the admissible models to observational algebras. Moreover, the operations on reachable data structures can be specified by inductive definitions while the operations on non-observable elements (i.e. states) can be defined (coinductively) by describing their effect w.r.t. the given observers. Analogously to abstract data type specifications, a loose observational specification describes a class of observational algebras which is closed under observational isomorphisms. Such a class can be considered as an "abstract behaviour type". If it contains only one observational isomorphism class its specification can be regarded as an "observationally monomorphic" specification of an object-oriented program.
The main topic of our next research steps is the consideration of refinement relations between (structured) observational specifications with an emphasis on refinement proofs. We hope that we can reuse several results of [2] and [13] but we are aware that these approaches do not deal with a built-in (internalised) observational semantics of structured specifications as considered in this paper. Another important direction of future research is concerned with an extension of observational logic to take into account concurrent systems specifications.

Acknowledgement We would like to thank Andrzej Tarlecki and the referees of this paper for several valuable remarks.

References

1. G. Bernot, M. Bidoit: Proving the correctness of algebraically specified software: modularity and observability issues. Proc. AMAST '91, 216-242, Springer-Verlag Works. in Comp. Series, 1992.
2. M. Bidoit, R. Hennicker: Proving the correctness of behavioural implementations. In Proc. AMAST '95, LNCS 936, 152-168, 1995. An extended version entitled "Modular correctness proofs of behavioural implementations" will appear in Acta Informatica.
3. M. Bidoit, R. Hennicker: Behavioural theories and the proof of behavioural properties. Theoretical Computer Science 175, 3-55, 1996.
4. M. Bidoit, R. Hennicker, M. Wirsing: Behavioural and abstractor specifications. Science of Computer Programming 25, 149-186, 1995.
5. M. Bidoit, A. Tarlecki: Behavioural satisfaction and equivalence in concrete model categories. Proc. CAAP '96, Trees in Algebra and Progr., LNCS 1059, 241-256, 1996.

6. T. Borzyszkowski: Completeness of a logical system for structured specifications. In: F. Parisi Presicce (ed.): Recent Trends in Algebraic Development Techniques, LNCS 1376, 107-121, 1998.
7. R. Burstall, R. Diaconescu: Hiding and behaviour: an institutional approach. In: A. W. Roscoe (ed.): A Classical Mind: Essays in Honour of C.A.R. Hoare, 75-92, Prentice-Hall, 1994.
8. R. Diaconescu: Behavioural coherence in object-oriented algebraic specification. Japan Advanced Institute for Science and Technology, IS-RR-98-0017F, 1998.
9. R. Diaconescu, K. Futatsugi: CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification, AMAST Series in Computing, Vol. 6, World Scientific, 1998.
10. J. Goguen, R. Burstall: Institutions: abstract model theory for specification and programming. Journal of the Association for Computing Machinery 39 (1), 95-146, 1992.
11. J. Goguen, G. Malcolm: A hidden agenda. Report CS97-538, Univ. of Calif. at San Diego, 1997.
12. J. Guttag, J. Horning: Larch: Languages and Tools for Formal Specification. Texts and Monographs in Computer Science, Springer, 1993.
13. R. Hennicker: Structured specifications with behavioural operators: semantics, proof methods and applications. Habilitation thesis, Institut für Informatik, Ludwig-Maximilians-Universität München, 1997.
14. R. Hennicker, M. Bidoit: Observational logic (long version). www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/, 1998.
15. R. Hennicker, F. Nickl: A behavioural algebraic framework for modular system design with reuse. In: H. Ehrig, F. Orejas (eds.): Recent Trends in Data Type Specification, LNCS 785, 220-234, 1994.
16. B. Jacobs, J. Rutten: A Tutorial on (Co)Algebras and (Co)Induction. EATCS Bulletin 62, 222-259, 1997.
17. H. J. Keisler: Model theory for infinitary logic. North-Holland, 1971.
18. J. Loeckx, H.-D. Ehrich, M. Wolf: Specification of Abstract Data Types. Wiley and Teubner, 1996.
19. G. Malcolm, J. A. Goguen: Proving correctness of refinement and implementation. Technical Monograph PRG-114, Oxford University Computing Laboratory, 1994.
20. P. Nivela, F. Orejas: Initial behaviour semantics for algebraic specifications. In: D. T. Sannella, A. Tarlecki (eds.): Recent Trends in Data Type Specification, Springer Lecture Notes in Computer Science 332, 184-207, 1988.
21. P. Padawitz: Swinging data types: syntax, semantics, and theory. In: M. Haveraaen, O. Owe, O.-J. Dahl (eds.): Recent Trends in Data Type Specification, LNCS 1130, 409-435, 1996.
22. H. Reichel: Initial computability, algebraic specifications, and partial algebras. International Series of Monographs in Computer Science No. 2, Oxford: Clarendon Press, 1987.
23. H. Reichel: An approach to object semantics based on terminal co-algebras. Math. Struct. Comp. Sci., 5, 129-152, 1995.
24. D. T. Sannella, A. Tarlecki: Specifications in an arbitrary institution. Information and Computation 76, 165-210, 1988.