Proving Behavioural Theorems with Standard First-Order Logic

M. BIDOIT
R. HENNICKER*

Laboratoire d'Informatique, URA 1327 du CNRS
Département de Mathématiques et d'Informatique
Ecole Normale Supérieure
*Institut für Informatik
Ludwig-Maximilians-Universität München

LIENS - 94 - 11

September 1994

Proving Behavioural Theorems with Standard First-Order Logic

Michel Bidoit[1] and Rolf Hennicker[2]

[1] LIENS, C.N.R.S. U.R.A. 1327 & Ecole Normale Supérieure
45, Rue d'Ulm, F-75230 Paris Cedex 05, France
[2] Institut für Informatik, Ludwig-Maximilians-Universität München
Leopoldstr. 11B, D-80802 München, Germany

Abstract. Behavioural logic is a generalization of first-order logic where the equality predicate is interpreted by a behavioural equality of objects (and not by their identity). We establish simple and general sufficient conditions under which the behavioural validity of some first-order formula with respect to a given first-order specification is equivalent to the standard validity of the same formula in a suitably enriched specification. As a consequence any proof system for first-order logic can be used to prove the behavioural validity of first-order formulas.

1 Introduction

Observability plays a prominent role in formal software development, since it provides a suitable basis for defining adequate correctness concepts. For instance, for proving the correctness of a program with respect to a given specification, many examples show that it is essential to abstract from internal implementation details and to rely only on the observable behaviour of the program. A similar situation is the notion of equivalence between concurrent processes and the abstraction from single step transitions to input-output operational semantics.

Behavioural correctness concepts can be formalized by using a behavioural logic, where the usual satisfaction relation of first-order logic with equality is generalized to a *behavioural satisfaction relation* (cf. e.g. [14, 12]). The key idea is to interpret the equality predicate symbol by a *behavioural equality*, where two objects are behaviourally equal if they cannot be distinguished by experiments with observable results. Hence to prove the behavioural validity of a formula we have to consider in general infinitely many observable experiments which are formally represented by an infinite set of "observable contexts".

The problem considered in this paper is how to prove the behavioural validity of some first-order formula ϕ with respect to a given first-order specification SP. We prove that, when the (first-order) specification SP satisfies some general and simple property (called the "Observability Kernel assumption"), the behavioural validity of the first-order formula ϕ (w.r.t. SP) is equivalent to the standard validity of the same formula ϕ with respect to the specification SP enriched by an adequate **finitary** first-order formula that represents the infinite set of all observable experiments. To this end, we use a general characterization of behavioural theories that we established in [5], and we show that the behavioural first-order theory of *SP* is equal to the standard first-order theory of the class of the fully abstract (standard) models of *SP*. Then we provide an infinitary axiomatization of full abstractness, and finally we show that, under the "Observability Kernel assumption", this infinitary axiomatization is equivalent to a finitary one.

The main significance of our result is that any available theorem prover for standard first-order logic with equality can be used, first to discharge the "Observability Kernel assumption", and then to prove the behavioural validity of first-order formulas. The soundness and completeness of behavioural proofs only rely on the soundness and completeness of the actually used standard proof system. Our result is fairly general since we do not need any restriction neither on the first-order specification *SP* nor on the first-order formula ϕ to be proved.

In the literature several approaches formalize behavioural correctness concepts by introducing some kind of behavioural semantics (cf. e.g. [8], [14], [12], [15], [2], [13]). The main drawback of these approaches is that they either do not provide a proof-theoretical framework or suggest technically complicated proof techniques which are only of limited interest for practical applications (cf. the context induction principle in [10] or the correspondance relation in [16]). [4] can be considered as a preliminary result, but restricted to the behavioural proof of equations w.r.t. equational specifications, with only one non observable sort.

This paper is organized as follows. In Section 2 we briefly summarize the basic notions of algebraic specifications that will be used later on. In Section 3 we define the behavioural equality and the associated behavioural satisfaction relation, we explain how all usual notions can be generalized in a behavioural framework and we point out the crucial role of fully abstract algebras. In Section 4 we study sufficient conditions (the "Observability Kernel" assumption) under which it is enough to consider a finite number of observable experiments. In Section 5 we show how the "Observability Kernel" assumption leads to a general method to prove behavioural theorems using any theorem prover for standard first-order logic.

2 Basic Notions

We assume that the reader is familiar with algebraic specifications [9, 6]. The basic concepts and notations that will be used hereafter are briefly summarized in this section.

A (many sorted) *signature* Σ is a pair (S, F) where S is a set of *sorts* and F is a set of *function symbols*.[3] To each function symbol $f \in F$ is associated an *arity*

[3] In this paper we assume that both S and F are finite.

$s_1 \dots s_n \rightarrow s$ with $s, s_1, \dots, s_n \in S$. If $n = 0$ then f is called *constant* of sort s. A *total Σ-algebra* $A = (\ (A_s)_{s \in S}, (f^A)_{f \in F}\ )$ over a signature $\Sigma = (S, F)$ consists of a family of carrier sets $(A_s)_{s \in S}$ and a family of functions $(f^A)_{f \in F}$ such that, if f has arity $s_1 \dots s_n \rightarrow s$, then f^A is a (total) function from $A_{s_1} \times \dots \times A_{s_n}$ to A_s (if $n = 0$ then f^A denotes a constant object of A_s). Σ-morphisms are defined as usual. The category of all Σ-algebras is denoted by $Alg(\Sigma)$.

Throughout this paper, given a signature $\Sigma = (S, F)$, we assume given an arbitrary but fixed family $X = (X_s)_{s \in S}$ of countably infinite sets X_s of variables of sort $s \in S$. $T_\Sigma(X)$ denotes the *Σ-term algebra freely generated by X*, the carrier sets of which are the sets $T_\Sigma(X)_s$ of *terms* of sort s (and with variables in X). Given a Σ-algebra A, a *valuation* $\alpha : X \rightarrow A$ is a family of mappings $(\alpha_s : X_s \rightarrow A_s)_{s \in S}$. Any valuation $\alpha : X \rightarrow A$ uniquely extends to a Σ-morphism $I_\alpha : T_\Sigma(X) \rightarrow A$, called the *interpretation* associated to α.

Given a Σ-algebra A, a *Σ-congruence* on A is a family $\approx_A = (\approx_{A,s})_{s \in S}$ of equivalence relations $\approx_{A,s}$ on A_s compatible with the signature Σ, i.e. for all $f \in F$ of arity $s_1 \dots s_n \rightarrow s$, for all $a_i, b_i \in A_{s_i}$, if $a_i \approx_{A,s_i} b_i$ then $f^A(a_1, \dots, a_n) \approx_{A,s} f^A(b_1, \dots, b_n)$.[4]

In practice it is often useful to consider instead of arbitrary Σ-algebras those algebras that are finitely generated by a distinguished subset Ω of the function symbols, called *constructors*. In these algebras all elements can be denoted by a constructor term (which is built only by constructor symbols and by variables of those sorts for which no constructor is defined). More precisely, let $\Sigma = (S, F)$ be a signature and $\Omega \subseteq F$ be a distinguished subset of constructors. A term t is called a *constructor term* if $t \in T_{\Sigma'}(X')$, where $\Sigma' = (S, \Omega)$, $X' = (X'_s)_{s \in S}$ with $X'_s = X_s$ if s is not the result sort of some constructor $f \in \Omega$ and $X'_s = \emptyset$ otherwise. The set of constructor terms is denoted by T_Ω. A Σ-algebra A is called *finitely generated by Ω* if for any $a \in A$ there exists a constructor term $t \in T_\Omega$ and a valuation $\alpha : X' \rightarrow A$ such that $I_\alpha(t) = a$. In particular, if $\Omega = \emptyset$, then any algebra $A \in Alg(\Sigma)$ is finitely generated by the empty set of constructors. (Note also that the definition of the generation principle is independent of X because X_s is countably infinite for all $s \in S$.)

Given a signature Σ and a set of variables X, the set $\mathrm{WFF}(\Sigma, X)$ of (well-formed) *finitary first-order Σ-formulas* is defined as usual, from equations $l = r$, the logical connectives $\neg, \wedge, \dots$ and the quantifiers $\forall, \exists$. Here the only predicate symbol is equality. In some occasions we will also use *infinitary Σ-formulas* of the form $\bigwedge_{i \in I} \phi_i$, where $(\phi_i)_{i \in I}$ is a countable family of Σ-formulas. A *Σ-sentence* is a Σ-formula which contains no free variable. In the sequel we will use the following abbreviations: For any term $t \in T_\Sigma(X)$, $var(t)$ denotes the set of variables occurring in t, and similarly $var(l, r)$ for a couple of terms l, r.

[4] In the sequel, for sake of clarity, we will omit the subscript s and write $a \approx_A b$ instead of $a \approx_{A,s} b$.

Hence a universally quantified equation will be denoted by $\forall var(l,r)\ .\ l = r$.

The *(standard) satisfaction* of a Σ-formula ϕ (finitary or not) by a Σ-algebra A, denoted by $A \models \phi$, is defined as usual in the first-order predicate calculus: the only predicate symbol $=$ is interpreted by the set-theoretic equality over the carrier sets of the algebra.

A *standard algebraic specification* SP is a tuple $(\Sigma, \Omega, \mathcal{A}x)$ where $\Sigma = (S, F)$ is a signature, $\Omega \subseteq F$ is a distinguished subset of constructors and $\mathcal{A}x$ is a set of Σ-sentences, called *axioms* of SP. The *model class* of SP, denoted by $Mod(SP)$, is the class of all Σ-algebras which satisfy the axioms of SP and which are finitely generated by Ω, i.e. $Mod(SP) \overset{\text{def}}{=} \{A \in Alg(\Sigma) \mid A \models \phi \text{ for all } \phi \in \mathcal{A}x \text{ and } A \text{ is finitely generated by } \Omega\}$.

Remember that if $\Omega = \emptyset$ then any algebra $A \in Alg(\Sigma)$ is finitely generated by the empty set of constructors. Hence in that case $Mod(SP)$ is simply the class of all Σ-algebras satisfying the axioms of SP. Therefore our assumption that any specification includes a declaration of a set of constructors is not a restriction but, on the contrary, it allows to apply our results to specifications with or without reachability constraints.

The *(standard) theory* of a class $\mathbf{C} \subseteq Alg(\Sigma)$ of Σ-algebras, denoted by $Th(\mathbf{C})$, is defined by $Th(\mathbf{C}) \overset{\text{def}}{=} \{\phi \in \mathrm{WFF}(\Sigma, X) \mid A \models \phi \text{ for all } A \in \mathbf{C}\}$.
In the following $SP \models \phi$ is an equivalent notation for $\phi \in Th(Mod(SP))$.

Example. Let us consider the following `CONTAINER` specification.

```
spec : CONTAINER
 use : ELEM, NAT, BOOL
 sort : Container
 generated by :
  ∅ : → Container
  insert : Elem Container → Container
 operations :
  _ ∪ _ : Container Container → Container
  remove : Elem Container → Container
  _ ∈ _ : Elem Container → Bool
  card : Container → Nat
  subset : Container Container → Bool
 axioms :
    ∀ S,S' : Container, e,e' : Elem .
  ∅ ∪ S = S
  insert(e,S) ∪ S' = insert(e,S ∪ S')
  remove(e,∅) = ∅
  remove(e,insert(e,S)) = remove(e,S)
  not(e = e') ⇒ remove(e,insert(e',S)) = insert(e',remove(e,S))
  e ∈ ∅ = false
  e ∈ insert(e',S) = ((e eq e') | (e ∈ S))
  card(∅) = 0
  (e ∈ S = true) ⇒ card(insert(e,S)) = card(S)
  (e ∈ S = false) ⇒ card(insert(e,S)) = succ(card(S))
  (subset(S,S') = true) ⇔
                  (∀ e : Elem . (e ∈ S = true) ⇒ (e ∈ S' = true))
end CONTAINER.
```

We do not detail the subspecifications `ELEM`, `NAT` and `BOOL` which are the usual ones. Note that the models of `CONTAINER` are finitely generated by the operations $\emptyset$ and `insert`. Since the `CONTAINER` specification is rather loose, its model class contains, among other algebras, the algebra of finite sets of elements, the algebra of finite multisets of elements, as well as the algebra of finite sequences of elements. It is quite easy to show (by structural induction w.r.t. the constructors $\emptyset$ and `insert`) that[5] `CONTAINER` $\models$ `S ∪ ∅ = S`, for instance, or that `CONTAINER` $\models$ `e ∈ S ∪ S' = (e ∈ S) | (e ∈ S')`, but it is important to note that `CONTAINER` $\not\models$ `insert(x,insert(x,S)) = insert(x,S)` and that `CONTAINER` $\not\models$ `insert(x,insert(y,S)) = insert(y,insert(x,S))`. As a consequence, the (standard) `CONTAINER` specification cannot be considered as a correct abstract implementation of a (standard) specification of sets. $\diamondsuit$

3 Behavioural Specifications and Behavioural Theories

As explained in the Introduction, we want to reflect the following idea: Some data structures are observable with respect to some *observable sorts*. (For instance, in the example given in the previous section, `Container`s are observable with respect to `Bool`eans and `Nat`ural numbers by means of the $\in$, `subset` and `card` operations.) The underlying intuition of our approach is that two objects are *behaviourally equal* if they cannot be distinguished by experiments with observable results. In the definitions below, experiments are formalized through *contexts* and experiments with observable results through *observable contexts*. Then we will generalize first-order logic to *behavioural first-order logic*: Instead of the set-theoretic equality, we use the *behavioural equality* for the interpretation of the $=$ predicate symbol. This behavioural equality is defined with respect to the observable contexts (hence the observable sorts) and is used to define a *behavioural satisfaction relation*. Similar approaches can be found in [14, 12, 10, 2, 3]. We provide now the necessary technical definitions:

Definition 1 (Context). Let $\Sigma = (S, F)$ be a signature and $Y = (Y_s)_{s \in S}$ be an S-sorted family of countably infinite sets Y_s of variables of sort s.[6] Let

[5] For sake of clarity the variables occurring in the equations used in our examples are implicitly universally quantified.

[6] For sake of clarity we assume that the sets of variables Y_s used for contexts are disjoint from the sets of variables X_s used for formulas.

$Z = \{z_s \mid s \in S\}$ be an S-sorted set of variables such that $z_s \notin Y_s$ for all $s \in S$.[7]

1. A *context* is a term $C \in T_\Sigma(Y \cup Z)$ that contains, besides variables in Y, one or many occurrences of exactly one variable $z_s \in Z$, called the *context variable* of C.
2. By exception, $var(C)$ will denote the set of variables occurring in C *but* the context variable of C.
3. The arity of a context C is $s \rightarrow s'$, where s is the sort of the context variable of C and s' is the sort of C.
4. $C[t]$ denotes the term obtained by substituting the term t (of sort s) for the context variable z_s (of sort s) of C.
5. Given a distinguished subset S_D of S and a sort s in S, we denote by $\mathcal{C}_s^{S_D}$ the set of all contexts C of arity $s \rightarrow s_d$, with $s_d \in S_D$.

Definition 2 (Contextual equality). Let $\Sigma = (S, F)$ be a signature, $\mathcal{C}$ be an arbitrary set of contexts and A be a Σ-algebra. The *contextual equality* on A induced by $\mathcal{C}$, denoted by $\approx_{\mathcal{C},A}$, is defined as follows:
Two elements $a, b \in A_s$ of sort s are contextually equal (w.r.t. $\mathcal{C}$), i.e. $a \approx_{\mathcal{C},A} b$, if and only if, for all contexts $C \in \mathcal{C}$ with context variable z_s of sort s, for all valuations $\alpha : Y \rightarrow A$, we have $I_{\alpha_1}(C) = I_{\alpha_2}(C)$, where $\alpha_1, \alpha_2 : Y \cup \{z_s\} \rightarrow A$ are the unique extensions of α defined by $\alpha_1(z_s) = a$ and $\alpha_2(z_s) = b$.
Note that, if there is no context $C \in \mathcal{C}$ with context variable of sort s, then we have $a \approx_{\mathcal{C},A} b$, for all $a, b \in A_s$ of sort s.

The intuition behind this definition is that two elements a and b are contextually equal w.r.t. a given set $\mathcal{C}$ of contexts if they cannot be distinguished by at least one of the computations represented by the contexts of $\mathcal{C}$. Note that $\approx_{\mathcal{C},A}$ is a family of equivalence relations (one for each sort $s \in S$), in particular $\approx_{\mathcal{C},A}$ always contains the set-theoretic equality. However, $\approx_{\mathcal{C},A}$ is not necessarily a congruence relation, i.e. $\approx_{\mathcal{C},A}$ is not necessarily compatible with the signature Σ. From our definition of the contextual equality induced by a given set of contexts $\mathcal{C}$ we immediately deduce the following characterization:

Lemma 3. *Let $\Sigma = (S, F)$ be a signature and $\mathcal{C}$ be an arbitrary set of contexts. For any Σ-algebra A, any sort $s \in S$ and any $a, b \in A_s$, let $x_L, x_R \in X_s$ and $\beta : \{x_L, x_R\} \rightarrow A$ be the valuation defined by $\beta(x_L) = a$ and $\beta(x_R) = b$. Then $a \approx_{\mathcal{C},A} b$ if and only if $A, \beta \models \bigwedge_{C \in \mathcal{C}(s)} \forall var(C) \,.\, C[x_L] = C[x_R]$*
where $\mathcal{C}(s)$ denotes the subset of all the contexts in $\mathcal{C}$ with context variable of sort s. As usual, when $\mathcal{C}(s)$ is empty, then the empty conjunction $\bigwedge_{C \in \mathcal{C}(s)} \dots$ is equivalent to true.
Note that the formula axiomatizing the contextual equality is an infinitary one if $\mathcal{C}(s)$ is an infinite set of contexts.

In the following we assume given a signature $\Sigma = (S, F)$ and a subset of *observable sorts* $S_{Obs} \subseteq S$. $S_{\neg Obs}$ denotes the complementary subset of non observable sorts, i.e. $S_{\neg Obs} = S \backslash S_{Obs}$.

[7] We assume as well that $z_s \notin X_s$.

Definition 4 (Observable context and behavioural equality). The set $\mathcal{C}_{Obs} \overset{\text{def}}{=} \bigcup_{s \in S} \mathcal{C}_s^{S_{Obs}}$ denotes the set of all *observable contexts.*
Let A be a Σ-algebra. The contextual equality on A induced by $\mathcal{C}_{Obs}$ (cf. Definition 2) is called the *behavioural equality* on A and is denoted by $\approx_{Obs,A}$.

Lemma 5. *The behavioural equality $\approx_{Obs,A}$ on A is a Σ-congruence.*

Note that, on observable sorts, the behavioural equality coincides with the set-theoretic equality, since $\mathcal{C}_s^{S_{Obs}}$ always contains the "trivial" context z_s when s is an observable sort. For the non observable sorts, the behavioural equality contains the set-theoretic equality, but there may be also distinct values which are behaviourally equal.

Now we can define the behavioural satisfaction relation with respect to S_{Obs}:

Definition 6 (Behavioural satisfaction relation). The *behavioural satisfaction relation* w.r.t. S_{Obs}, denoted by $\models_{Obs}$, is defined as follows:
Let A be a Σ-algebra.

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

Hence Definition 6 is quite similar to the definition of the standard satisfaction relation $\models$, the only difference concerns (1) where $I_\alpha(l) \approx_{Obs,A} I_\alpha(r)$ replaces $I_\alpha(l) = I_\alpha(r)$.

Lemma 7. *Let A be a Σ-algebra and $\forall var(l, r) \,.\, l = r$ be a universally quantified equation. Let s be the common sort of l and r.*
If s is an observable sort, then $A \models_{Obs} \forall var(l, r) \,.\, l = r$ if and only if $A \models \forall var(l, r) \,.\, l = r$.
If s is a non observable sort, then $A \models_{Obs} \forall var(l, r) \,.\, l = r$ if and only if, for all observable contexts $C \in \mathcal{C}_s^{S_{Obs}}$, $A \models \forall var(C) \cup var(l, r) \,.\, C[l] = C[r]$.

Remark. Lemma 7 is often used in the literature to define directly (i.e. without introducing explicitly the behavioural equality) the behavioural satisfaction of equations. However the explicit definition we have chosen is necessary to define the behavioural satisfaction of arbitrary first-order formulas. On the other hand, this lemma suggests that to prove the behavioural satisfaction of an equation $l = r$ (between non observable terms), it is equivalent to prove the standard satisfaction of the infinite set of equations $C[l] = C[r]$, for all $C \in \mathcal{C}_s^{S_{Obs}}$. *Context Induction* (a specialized version of structural induction) was introduced by R. Hennicker in [10] as a means to prove such infinite sets of equations and has been implemented in the ISAR system (cf. [1]). In [4] it is explained how an explicit use of context induction can be avoided under some assumptions. Unfortunately, none of these ideas directly extends to the proof of the behavioural satisfaction of arbitrary first-order formulas.

In a similar way to what was done for the satisfaction relation we also generalize the generation principle of algebras to take into account the behavioural equality:

Definition 8 (Behaviourally finitely generated algebra). Let $\Omega \subseteq F$ be a distinguished subset of constructors and A be a Σ-algebra. A is called *behaviourally finitely generated by* Ω (w.r.t. S_{Obs}) if for any $a \in A$ there exists a constructor term $t \in T_\Omega$ and a valuation $\alpha : var(t) \rightarrow A$ such that $I_\alpha(t) \approx_{Obs,A} a$. In particular, if $\Omega = \emptyset$, then any algebra $A \in Alg(\Sigma)$ is behaviourally finitely generated by the empty set of constructors.

Behavioural specifications can be built on top of standard specifications as follows:

Definition 9 (Behavioural specification).

1. A *behavioural specification* is a tuple $SP\text{–}Obs = (SP, S_{Obs})$ such that $SP = (\Sigma, \Omega, \mathcal{A}x)$ is a standard specification (with signature $\Sigma = (S, F)$), and $S_{Obs} \subseteq S$ is a distinguished subset of observable sorts.
2. The *model class of* $SP\text{–}Obs$, denoted by $Mod(SP\text{–}Obs)$, is the class of all Σ-algebras which behaviourally satisfy the axioms of SP and which are behaviourally finitely generated by Ω, i.e.:
   $Mod(SP\text{–}Obs) \overset{\text{def}}{=} \{A \in Alg(\Sigma) \mid A \models_{Obs} \phi$ for all $\phi \in \mathcal{A}x$ and
   A is behaviourally finitely generated by $\Omega\}$.

Now we can consider the behavioural theory with respect to S_{Obs} of a given class $\mathbf{C}$ of Σ-algebras:

Definition 10 (Behavioural theory). Let $\mathbf{C} \subseteq Alg(\Sigma)$ be a class of Σ-algebras. The *behavioural theory* of $\mathbf{C}$ w.r.t. S_{Obs}, denoted by $Th_{Obs}(\mathbf{C})$, is defined by $Th_{Obs}(\mathbf{C}) \overset{\text{def}}{=} \{\phi \in \mathrm{WFF}(\Sigma, X) \mid A \models_{Obs} \phi$ for all $A \in \mathbf{C}\}$.
In the following, $SP\text{–}Obs \models_{Obs} \phi$ means $\phi \in Th_{Obs}(Mod(SP\text{–}Obs))$. In this case ϕ is called a *behavioural theorem* (w.r.t. $SP\text{–}Obs$).

Example. Let us consider again our `CONTAINER` specification and assume that the observable sorts are `Elem`, `Nat` and `Bool`. Then we obtain a behavioural specification (`CONTAINER`, {`Elem, Nat, Bool`}). Two objects of sort `Container` will be considered as behaviourally equal if they cannot be distinguished by observable contexts. Here, all observable contexts (with context variable of sort `Container`) must contain either $\in$, `subset` or `card`. If we consider the algebra of finite sequences of elements, it is intuitively clear that two distinct sequences will be behaviourally equal if they contain the same elements (not necessarily with the same number of occurrences or in the same order), because these sequences cannot be distinguished by the operations $\in$, `subset` or `card`. For the same reasons, it is intuitively clear that the two characteristic equations of sets, `insert(x,insert(x,S)) = insert(x,S)` and `insert(x,insert(y,S)) = insert(y,insert(x,S))`, are behaviourally satisfied by all models of the behavioural specification (`CONTAINER`, {`Elem, Nat, Bool`}). Indeed no observable experiment (done with the $\in$, `card` and `subset` operations) can distinguish the left and right-hand sides of these equations. The aim of this paper is to provide a proof technique to formally establish that this intuition is right. $\diamondsuit$

Fully abstract algebras play an important role for the characterization of behavioural theories. Following Milner's notion (cf. [11]), we define full abstractness with respect to the observable sorts S_{Obs} as follows:

Definition 11 (Fully abstract algebra).

1. A Σ-algebra A is called *fully abstract* with respect to S_{Obs} if $\approx_{Obs,A}$ coincides with the set-theoretic equality over the carrier sets of A.
2. For any class $\mathbf{C} \subseteq Alg(\Sigma)$ of Σ-algebras, $FA_{Obs}(\mathbf{C})$ denotes the subclass of the fully abstract algebras of $\mathbf{C}$, i.e.
   $FA_{Obs}(\mathbf{C}) \overset{\text{def}}{=} \{A \in \mathbf{C} \mid A \text{ is fully abstract w.r.t. } S_{Obs}\}$.

Example. Consider again our behavioural specification (`CONTAINER`, {`Elem, Nat, Bool`}). The algebra of finite sequences of elements is not fully abstract (we have pointed out above that two distinct sequences may be behaviourally equal), while the algebra of finite sets of elements is fully abstract. $\diamondsuit$

Since an algebra A is fully abstract if and only if all behaviourally equal objects are identical, we have:

Proposition 12 (Infinitary characterization of fully abstract algebras). *A Σ-algebra A is fully abstract w.r.t. S_{Obs} if and only if A satisfies (in the standard sense) the following infinitary formula FA^{∞}_{Obs} : $\bigwedge_{s\in S_{\neg Obs}} FA^{\infty}_{Obs}(s)$, where, for each non observable sort $s \in S_{\neg Obs}$, $FA^{\infty}_{Obs}(s)$ is:*
$\forall x_L, x_R : s \,.\, \left[\left(\bigwedge_{C\in\mathcal{C}_s^{S_{Obs}}} \forall var(C) \,.\, C[x_L] = C[x_R]\right) \Longrightarrow x_L = x_R\right]$

Proof. Straightforward from Definition 11 and Lemma 3. □

The crucial role of fully abstract algebras is outlined by the following result:

Theorem 13 (Characterization of behavioural theories). *Let $SP\text{–}Obs = (SP, S_{Obs})$ be a behavioural specification.*

1. *$Th_{Obs}(Mod(SP\text{–}Obs)) = Th(FA_{Obs}(Mod(SP)))$.*
2. *Let $SP\text{–}FA^{\infty}_{Obs}$ be the (standard) specification SP augmented by the infinitary axiom FA^{∞}_{Obs} defined in Proposition 12. Then:*
   $Th_{Obs}(Mod(SP\text{–}Obs)) = Th(Mod(SP\text{–}FA^{\infty}_{Obs}))$ i.e.
   $SP\text{–}Obs \models_{Obs} \phi$ if and only if $SP\text{–}FA^{\infty}_{Obs} \models \phi$, for all $\phi \in WFF(\Sigma, X)$.

Proof. (1) is a special case of a more general theorem given in [5]. (2) is a direct consequence of (1) and of Proposition 12. □

According to Theorem 13, the behavioural satisfaction of a given first-order formula ϕ by the model class of the behavioural specification SP–Obs is equivalent to the standard satisfaction of the same formula ϕ by the standard specification SP–FA^{∞}_{Obs}. Unfortunately this result is up to now of limited practical interest, since Proposition 12 only provides an infinitary axiomatization of fully abstract algebras, hence the specification SP–FA^{∞}_{Obs} contains an infinitary axiom. In the sequel we study sufficient conditions for getting rid of this infinitary axiomatization.

4 The Observability Kernel

Since the behavioural equality is defined with respect to an infinite set of observable contexts which represent the infinitely many experiments with observable results, it is not surprising that our characterization of fully abstract algebras involves an infinitary formula (cf. Proposition 12). A natural idea to get rid of this infinitary formula is to check whether, under some conditions, it would be enough to consider some adequate finite set of observable contexts instead of the infinite set of all observable contexts.

In a first step we will study some sufficient conditions under which the contextual equality induced by an arbitrary set of contexts $\mathcal{C}$ coincides with the behavioural equality.

Lemma 14. *Let A be a Σ-algebra and $\approx_A$ be an arbitrary congruence on A. If $\approx_A$ coincides with the set-theoretic equality on the carrier sets of all observable sorts $s \in S_{Obs}$, then $\approx_A \subseteq \approx_{Obs,A}$.*

Proof. Let a, b be two elements of A_s, for some sort $s \in S$ and assume that $a \approx_A b$. Since $\approx_A$ is a congruence (hence is compatible with the signature Σ), we have, for any observable context $C \in \mathcal{C}_s^{S_{Obs}}$ and for any valuation $\alpha : Y \to A$, $I_{\alpha_1}(C) \approx_A I_{\alpha_2}(C)$, where $\alpha_1, \alpha_2 : Y \cup \{z_s\} \to A$ are the unique extensions of α defined by $\alpha_1(z_s) = a$ and $\alpha_2(z_s) = b$, where z_s is the context variable of C. Since C is an observable context, both $I_{\alpha_1}(C)$ and $I_{\alpha_2}(C)$ belong to the carrier set of some observable sort. But since we have assumed that $\approx_A$ coincides with the set-theoretic equality on the carrier sets of the observable sorts, $I_{\alpha_1}(C) \approx_A I_{\alpha_2}(C)$ implies $I_{\alpha_1}(C) = I_{\alpha_2}(C)$, hence we have $a \approx_{Obs,A} b$. Therefore $\approx_A \subseteq \approx_{Obs,A}$. □

Remark. Indeed it is easy to prove that the set of all congruences on A that coincides with the set-theoretic equality on each observable sort is a complete lattice, the smallest element of which is the set-theoretic equality, the greatest element being the behavioural equality.

Lemma 15. *Let $\mathcal{C} \subseteq \mathcal{C}_{Obs}$ be an arbitrary subset of observable contexts such that, for any observable sort $s \in S_{Obs}$, $z_s \in \mathcal{C}$ and let A be a Σ-algebra. Then $\approx_{\mathcal{C},A} = \approx_{Obs,A}$ if and only if $\approx_{\mathcal{C},A}$ is a Σ-congruence.*

Proof. Assume that $\approx_{\mathcal{C},A}$ is a Σ-congruence. Since $\mathcal{C} \subseteq \mathcal{C}_{Obs}$, obviously we have $\approx_{Obs,A} \subseteq \approx_{\mathcal{C},A}$. To prove that $\approx_{\mathcal{C},A} \subseteq \approx_{Obs,A}$, by Lemma 14, it is enough to prove that $\approx_{\mathcal{C},A}$ coincides with the set-theoretic equality for each carrier set of an observable sort. But this holds since $\mathcal{C}$ contains by assumption all the "trivial" contexts z_s when s is an observable sort. The converse direction is trivial. □

Notation. In the following, C_k^f denotes a context of the form $f(y_1, \ldots, y_{k-1}, z_{s_k}, y_{k+1}, \ldots, y_n)$ built from a function symbol $f \in F$ of arity $s_1 \ldots s_{k-1}\, s_k\, s_{k+1} \ldots s_n \rightarrow s$, an integer k with $1 \leq k \leq n$, a context variable z_{s_k} of sort s_k, and pairwise distinct variables $y_i \in Y$. Provided they are pairwise distinct, the actual names of the variables y_i are irrelevant and these variables can be left implicit in the notation C_k^f. Moreover, when the context C_k^f is substituted for the context variable of another context C to form the context $C[C_k^f]$, we assume w.l.o.g. that the variables y_i of C_k^f are distinct from the variables y_j occurring in C (i.e. we assume that $var(C) \cap var(C_k^f) = \emptyset$).

Proposition 16. *Let $\mathcal{C}$ be an arbitrary set of contexts and A be a Σ-algebra. The contextual equality $\approx_{\mathcal{C},A}$ on A induced by $\mathcal{C}$ is a Σ-congruence if and only if A satisfies the following (possibly infinitary) formulas, for each $s \in S$, for each function symbol $f \in F$ of arity $s_1 \ldots s_n \rightarrow s'$, for each integer k with $1 \leq k \leq n$ and $s_k = s$:*

$$\forall x_L, x_R : s \,.\, \Big[\Big(\bigwedge_{C\in\mathcal{C}(s)} \forall var(C) \,.\, C[x_L] = C[x_R]\Big) \Longrightarrow \bigwedge_{C'\in\mathcal{C}(s')} \forall var(C') \cup var(C_k^f) \,.\, C'[C_k^f[x_L]] = C'[C_k^f[x_R]]\Big]$$

where $\mathcal{C}(s)$ ($\mathcal{C}(s')$ resp.) denotes the subset of all the contexts in $\mathcal{C}$ with context variable of sort s (s' resp.).[8]

Proof. For the proof we use the following lemma:

Lemma 17. *Given a Σ-algebra A, a family $\approx_A$ of equivalence relations is a Σ-congruence on A if and only if, for all $s \in S$, for all $f \in F$ of arity $s_1 \ldots s_n \rightarrow s'$, for all k with $1 \leq k \leq n$ and $s_k = s$, for all $a, b \in A_s$, if $a \approx_A b$ then for all $c_i \in A_{s_i}$, $f^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n) \approx_A f^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$.*

Now let A be a Σ-algebra, and $\mathcal{C}$ be an arbitrary set of contexts. Let $s \in S$ and let $f \in F$ be an arbitrary function symbol of arity $s_1 \ldots s_n \rightarrow s'$, let k be an arbitrary integer with $1 \leq k \leq n$ and $s_k = s$. According to the lemma above, $\approx_{\mathcal{C},A}$ is a Σ-congruence if and only if for all $a, b \in A_s$, $a \approx_{\mathcal{C},A} b$ implies that $f^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n) \approx_{\mathcal{C},A} f^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$ holds for all $c_i \in A_{s_i}$. We will now show that this implication is equivalent to the fact that A satisfies the formula given in the proposition. According to Lemma 3, $a \approx_{\mathcal{C},A} b$ iff $A, \beta \models \bigwedge_{C\in\mathcal{C}(s)} \forall var(C) \,.\, C[x_L] = C[x_R]$, where $\beta : \{x_L, x_R\} \rightarrow A$ is the valuation defined by $\beta(x_L) = a$ and $\beta(x_R) = b$. Similarly, $f^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n) \approx_{\mathcal{C},A} f^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$ iff $A, \beta' \models \bigwedge_{C'\in\mathcal{C}(s')} \forall var(C') \,.\, C'[x'_L] = C'[x'_R]$, where $\beta' : \{x'_L, x'_R\} \rightarrow A$ is

[8] Remember that an empty conjunction is as usual equivalent to true.

the valuation defined by $\beta'(x'_L) = f^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n)$ and $\beta'(x'_R) = f^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$. Now let $\alpha : var(C^f_k) \rightarrow A$ be the valuation defined by $\alpha(y_i) = c_i$. Then $\beta'(x'_L) = I_{\beta\cup\alpha}(C^f_k[x_L])$ and $\beta'(x'_R) = I_{\beta\cup\alpha}(C^f_k[x_R])$. Hence $f^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n) \approx_{\mathcal{C},A} f^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$ holds for all $c_i \in A_{s_i}$ iff $A, \beta\cup\alpha \models \bigwedge_{C'\in\mathcal{C}(s')} \forall var(C') \,.\, C'[C^f_k[x_L]] = C'[C^f_k[x_R]]$ holds for all $\alpha : var(C^f_k) \rightarrow A$, i.e. iff $A, \beta \models \bigwedge_{C'\in\mathcal{C}(s')} \forall var(C')\cup var(C^f_k) \,.\, C'[C^f_k[x_L]] = C'[C^f_k[x_R]]$, which shows that the required implication is exactly the one provided by the proposition. □

Lemma 15 points out that we can replace the infinite set $\mathcal{C}_{Obs}$ of all observable contexts by any subset $\mathcal{C} \subseteq \mathcal{C}_{Obs}$ (in particular by any finite subset $\mathcal{C}$), provided that $\mathcal{C}$ contains the trivial observable contexts and that the contextual equality induced by $\mathcal{C}$ is a congruence. Moreover, Proposition 16 provides a necessary and sufficient condition for this contextual equality to be a Σ-congruence. The problem now is to find an adequate finite subset of $\mathcal{C}_{Obs}$. Remember that observable contexts represent experiments with observable results. A typical experiment will start by some computations involving mainly non observable values and providing non observable results, then there will be a computation providing an observable result, possibly followed by more computations over observable values. The crucial idea is that intuitively only the step going from non observable values to observable ones is critical. Hence our intuition suggests that, in addition to the trivial observable contexts, it would be enough to consider the contexts of the form C^f_k, with $f \in \Sigma$ of arity $s_1 \ldots s_{k-1}\, s_k\, s_{k+1} \ldots s_n \rightarrow s$, $s \in S_{Obs}$ and $s_k \in S_{\neg Obs}$, provided that the contextual equality induced by these contexts is a Σ-congruence. To make this intuition more precise we start by introducing some useful notations.

Notation. According to the partition of the sorts S into S_{Obs} and $S_{\neg Obs}$, the set of function symbols F can be split into:

1. The subset F_O of function symbols of arity $f_O : s_1 \ldots s_n \rightarrow s$, with $s \in S_{Obs}$, and at least one $s_i \in S_{\neg Obs}$;
2. The subset F_I of function symbols of arity $f_I : s_1 \ldots s_m \rightarrow s$, with $s \in S_{\neg Obs}$, and at least one $s_j \in S_{\neg Obs}$;
3. The subset of all other function symbols.

Definition 18 (Crucial observable contexts). The set of the *crucial observable contexts*, denoted by $\mathcal{CC}_{Obs}$, is defined by:
$\mathcal{CC}_{Obs} = \{C^{f_O}_i \mid f_O : s_1 \ldots s_{i-1}\, s_i\, s_{i+1} \ldots s_n \rightarrow s \in F_O$ and
i is such that $s_i \in S_{\neg Obs}\} \cup \{z_s \mid s \in S_{Obs}\}$.
Note that, for each adequate choice of $f_O \in F_O$ and i, we make an arbitrary choice for the (pairwise distinct) variables y_k left implicit in the notation $C^{f_O}_i$. Hence the set of the crucial observable contexts is finite.

Example. Consider again our behavioural specification (`CONTAINER`, {`Elem`, `Nat`, `Bool`}). We have F_O = {∈, `card`, `subset`} and F_I = {`insert`, ∪, `remove`}.

$\mathcal{CC}_{Obs} = \{x \in z_{Cont},\ \mathtt{card}(z_{Cont}),\ \mathtt{subset}(S, z_{Cont}),\ \mathtt{subset}(z_{Cont}, S),\ z_{Bool},\ z_{Nat},\ z_{Elem}\}$. ♢

Before stating the main result of this section we observe the following property:

Lemma 19. *The contextual equality induced by the set of the crucial observable contexts $\mathcal{CC}_{Obs}$ is always compatible with the function symbols $f_O \in F_O$.*

Proof. Let A be a Σ-algebra and $\approx_{\mathcal{CC}_{Obs},A}$ be the contextual equality induced over A by the crucial contexts $\mathcal{CC}_{Obs}$. We know that $\approx_{\mathcal{CC}_{Obs},A}$ coincides with the set-theoretic equality on the carrier sets of the observable sorts. Hence it is enough to check the compatibility w.r.t. non observable sorts. Let $s \in S_{\neg Obs}$ and $a, b \in A_s$ such that $a \approx_{\mathcal{CC}_{Obs},A} b$. Let $f_O \in F_O$ of arity $f_O : s_1 \ldots s_n \to s$, and let k such that $s_k = s$. Let $c_i \in A_{s_i}$ be arbitrary values. Then $f_O^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n)$ ($f_O^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$ resp.) is equal to $I_{\alpha_1}(C_k^{f_O})$ ($I_{\alpha_2}(C_k^{f_O})$ resp.), where $\alpha : var(C_k^{f_O}) \to A$ is the valuation defined by $\alpha(y_i) = c_i$ and $\alpha_1, \alpha_2 : Y \cup \{z_s\} \to A$ are the unique extensions of α defined by $\alpha_1(z_s) = a$ and $\alpha_2(z_s) = b$. By definition, $a \approx_{\mathcal{CC}_{Obs},A} b$ implies $I_{\alpha_1}(C_k^{f_O}) = I_{\alpha_2}(C_k^{f_O})$, i.e. $f_O^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n) = f_O^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$. Hence $f_O^A(c_1, \ldots, c_{k-1}, a, c_{k+1}, \ldots, c_n) \approx_{\mathcal{CC}_{Obs},A} f_O^A(c_1, \ldots, c_{k-1}, b, c_{k+1}, \ldots, c_n)$ and Lemma 17 shows that $\approx_{\mathcal{CC}_{Obs},A}$ is a (S, F_O)-congruence. □

We have now the necessary ingredients to state our main result:

Theorem 20 (Observability Kernel). *Let A be a Σ-algebra. The contextual equality $\approx_{\mathcal{CC}_{Obs},A}$ on A induced by the set of the crucial observable contexts $\mathcal{CC}_{Obs}$ coincides with the behavioural equality $\approx_{Obs,A}$ if and only if the algebra A satisfies (in the standard sense) the following finitary first-order formula $OK_{\Sigma,S_{Obs}}$, called the Observability Kernel associated to Σ and S_{Obs}, and defined by $OK_{\Sigma,S_{Obs}} \overset{\text{def}}{=} \bigwedge_{s \in S_{\neg Obs}} OK_\Sigma(s)$, where for each non observable sort $s \in S_{\neg Obs}$, $OK_\Sigma(s)$ is the following implication:*

$$\forall x_L, x_R : s\ .\ \left[\left(\bigwedge_{f_O \in F_O}^{\wedge i}\ \forall var(C_i^{f_O})\ .\ C_i^{f_O}[x_L] = C_i^{f_O}[x_R]\right) \Longrightarrow \right.$$
$$\left. \bigwedge_{f_I \in F_I}^{\wedge j} \bigwedge_{f_O \in F_O}^{\wedge i}\ \forall var(C_i^{f_O}) \cup var(C_j^{f_I})\ .\ C_i^{f_O}[C_j^{f_I}[x_L]] = C_i^{f_O}[C_j^{f_I}[x_R]]\right]$$

and where:

1. *$\bigwedge_{f_O \in F_O}^{\wedge i}$ is an abbreviation for the conjunction over all contexts $C_i^{f_O}$, for all $f_O \in F_O$ and all choices of i such that the sort of the context variable of the context $C_i^{f_O}$ is s.*
2. *Similarly $\bigwedge_{f_I \in F_I}^{\wedge j} \bigwedge_{f_O \in F_O}^{\wedge i}$ is an abbreviation for the conjunction over all contexts $C_j^{f_I}$ and $C_i^{f_O}$, for all $f_I \in F_I$, $f_O \in F_O$ and all choices of j and i such that the sort of the context variable of the context $C_j^{f_I}$ is s and the sort of the context variable of the context $C_i^{f_O}$ is the sort of the context $C_j^{f_I}$.*

Proof. According to Lemma 15, the contextual equality induced by $\mathcal{CC}_{Obs}$ coincides with the behavioural equality if and only if it is a Σ-congruence, i.e. if it is compatible with all $f \in F$. Since the contextual equality induced by $\mathcal{CC}_{Obs}$ coincides with the set-theoretic equality on the carrier sets of the observable sorts, and since it is compatible with F_O (cf. Lemma 19), it is enough to check the compatibility w.r.t. non observable sorts and function symbols in F_I. But then the Observability Kernel is exactly the conjunction, for all non observable sorts and for all $f_I \in F_I$, of the formulas given in Proposition 16 (with $\mathcal{CC}_{Obs}$ as a special case for $\mathcal{C}$). □

It is important to note that, since the signature Σ is finite, $OK_{\Sigma,S_{Obs}}$ is a finitary first-order formula.

Example. Consider again our behavioural specification (`CONTAINER`, {`Elem, Nat, Bool`}). The Observability Kernel is the following formula (here there is just one non observable sort, `Container`):[9]

```
∀ CL, CR : Container .
  [ (∀ x : Elem, S : Container .
     x ∈ CL = x ∈ CR ∧
     card(CL) = card(CR) ∧
     subset(S,CL) = subset(S,CR) ∧
     subset(CL,S) = subset(CR,S) )
   ⟹ (∀ y, z : Elem, S1, S2 : Container .
        z ∈ insert(y,CL) = z ∈ insert(y,CR) ∧
        z ∈ (S1 ∪ CL) = z ∈ (S1 ∪ CR) ∧
        z ∈ (CL ∪ S1) = z ∈ (CR ∪ S1) ∧
        z ∈ remove(y,CL) = z ∈ remove(y,CR) ∧
        card(insert(y,CL)) = card(insert(y,CR)) ∧
        card(S1 ∪ CL) = card(S1 ∪ CR) ∧
        card(CL ∪ S1) = card(CR ∪ S1) ∧
        card(remove(y,CL)) = card(remove(y,CR)) ∧
        subset(S1, insert(y,CL)) = subset(S1, insert(y,CR)) ∧
        subset(S1, S2 ∪ CL) = subset(S1, S2 ∪ CR) ∧
        subset(S1, CL ∪ S2) = subset(S1, CR ∪ S2) ∧
        subset(S1, remove(y,CL)) = subset(S1, remove(y,CR)) ∧
        subset(insert(y,CL), S1) = subset(insert(y,CR), S1) ∧
        subset(S2 ∪ CL, S1) = subset(S2 ∪ CR, S1) ∧
        subset(CL ∪ S2, S1) = subset(CR ∪ S2, S1) ∧
        subset(remove(y,CL), S1) = subset(remove(y,CR), S1) ) ]
```

Note that in the next section we will study further simplifications that will considerably improve the Observability Kernel, and as a consequence will lead to much more simpler formulas to be proved. ♢

[9] For sake of clarity we have chosen more adequate names for the variables and we have moved the universal quantifiers in front of the conjunctions.

Corollary 21. *Let A be a Σ-algebra.* **If** $A \models OK_{\Sigma,S_{Obs}}$ **then** *for any non observable sort $s \in S_{\neg Obs}$, for any $a, b \in A_s$ and valuation $\beta : \{x_L, x_R\} \to A$ with $\beta(x_L) = a$ and $\beta(x_R) = b$, the following conditions are equivalent:*

1. $a \approx_{Obs,A} b$
2. $a \approx_{\mathcal{CC}_{Obs},A} b$
3. $A, \beta \models \bigwedge_{C \in \mathcal{C}_s^{S_{Obs}}} \forall var(C) \;.\; C[x_L] = C[x_R]$
4. $A, \beta \models \bigwedge^{\wedge i}_{f_O \in F_O} \; \forall var(C_i^{f_O}) \;.\; C_i^{f_O}[x_L] = C_i^{f_O}[x_R]$

Proof. Follows immediately from Theorem 20 and Lemma 3. In particular, according to Lemma 3 we know that $a \approx_{\mathcal{CC}_{Obs},A} b$ is equivalent to $A, \beta \models \bigwedge_{C \in \mathcal{CC}_{Obs(s)}} \forall var(C) \;.\; C[x_L] = C[x_R]$. Since s is a non observable sort, the latter formula is the same as the one given in 4. □

5 How to Prove Behavioural Theorems

We assume given a behavioural specification *SP–Obs* $= (SP, S_{Obs})$ with signature $\Sigma = (S, F)$ and with observable sorts $S_{Obs} \subseteq S$. We keep the notations F_O and F_I introduced in the previous section. $\mathcal{CC}_{Obs}$ denotes the set of the crucial observable contexts.

We can now combine the results obtained in Section 3 (especially Proposition 12 and Theorem 13) with the simplifications induced by the Observability Kernel assumption (cf. Theorem 20 and Corollary 21).

Proposition 22 (Finitary characterization of fully abstract algebras). *Let A be a Σ-algebra.* **If** $A \models OK_{\Sigma,S_{Obs}}$ **then** *A is fully abstract w.r.t. S_{Obs} if and only if A satisfies (in the standard sense) the following finitary formula $FA_{Obs} : \bigwedge_{s \in S_{\neg Obs}} FA_{Obs}(s)$, where, for each non observable sort $s \in S_{\neg Obs}$, $FA_{Obs}(s)$ is:*
$\forall x_L, x_R : s \;.\; \left[\left(\bigwedge^{\wedge i}_{f_O \in F_O} \; \forall var(C_i^{f_O}) \;.\; C_i^{f_O}[x_L] = C_i^{f_O}[x_R]\right) \Longrightarrow x_L = x_R\right]$

Proof. Follows from Proposition 12 and Corollary 21 ($3 \Leftrightarrow 4$). □

Now we obtain our final result:

Theorem 23. *Let SP* **partitioned by** *F_O be the (standard) specification SP augmented by the finitary axiom FA_{Obs} defined in Proposition 22.*
If *SP* $\models OK_{\Sigma,S_{Obs}}$ **then**
$Th_{Obs}(Mod(SP–Obs)) = Th(Mod(SP$ **partitioned by** *F_O)), i.e. for all $\phi \in WFF(\Sigma, X)$, SP–Obs $\models_{Obs} \phi$ if and only if SP* **partitioned by** *$F_O \models \phi$.*

Proof. By Theorem 13.1 we have $Th_{Obs}(Mod(SP–Obs)) = Th(FA_{Obs}(Mod(SP)))$. The assumption $SP \models OK_{\Sigma,S_{Obs}}$ and Proposition 22 imply that $FA_{Obs}(Mod(SP)) = Mod(SP$ **partitioned by** $F_O)$. □

Theorem 23 provides a very general and powerful method to prove the behavioural satisfaction of arbitrary first-order formulas by *SP*-Obs: First we compute the Observability Kernel $OK_{\Sigma,S_{Obs}}$ and we prove, once for all, that:

$$(A)\ SP \models OK_{\Sigma,S_{Obs}}$$

Then, for any first-order formula ϕ, to prove *SP–Obs* $\models_{Obs} \phi$ we prove:

$$(B)\ SP\ \textbf{partitioned by}\ F_O \models \phi$$

The result is general since we have made no assumption neither on the axioms of *SP* nor on the number of non observable sorts. The result is powerful since for both (A) and (B) we can use any available theorem prover for first-order logic. Our method has been successfully applied to various examples using the Larch Prover V3.0 [7].[10]

A last improvement can be obtained using the following remark. In most cases it is possible to split F_O into two sets F_{O1} and F_{O2}, with the following property (for all $s \in S_{\neg Obs}$):

$$(R)\ SP \models \forall x_L, x_R : s\ .\ \Big[\Big(\bigwedge\nolimits^{\wedge i}_{fo\in F_{O1}}\ \forall var(C_i^{fo})\ .\ C_i^{fo}[x_L] = C_i^{fo}[x_R]\Big) \Longrightarrow \bigwedge\nolimits^{\wedge i}_{fo\in F_{O2}}\ \forall var(C_i^{fo})\ .\ C_i^{fo}[x_L] = C_i^{fo}[x_R]\Big]$$

Then the proof of (A) is split into the proof of (R) (for all $s \in S_{\neg Obs}$) and the proof of the Reduced Observability Kernel, which is similar to the Observability Kernel, but where the conjunctions $\bigwedge^{\wedge i}_{fo\in F_O}$ are restricted to $\bigwedge^{\wedge i}_{fo\in F_{O1}}$.

Example. Consider again our behavioural specification (**CONTAINER**, {`Elem, Nat, Bool`}). Using our last improvement, we can split F_O into $F_{O1} = \{\in\}$ and $F_{O2} = \{$`card, subset`$\}$. Then the proof of the Observability Kernel is split into the proof of the *Simplifiability of the Observability Kernel*:

```
∀ CL, CR : Container .
  [ (∀ x : Elem . x ∈ CL = x ∈ CR )
   ⟹ (∀ S : Container .
       card(CL) = card(CR) ∧
       subset(S,CL) = subset(S,CR) ∧
       subset(CL,S) = subset(CR,S) ) ]
```

and the proof of the *Reduced Observability Kernel*:

```
∀ CL, CR : Container .
  [ (∀ x : Elem . x ∈ CL = x ∈ CR )
   ⟹ (∀ y, z : Elem, S : Container .
       z ∈ insert(y,CL) = z ∈ insert(y,CR) ∧
       z ∈ (S ∪ CL) = z ∈ (S ∪ CR) ∧
       z ∈ (CL ∪ S) = z ∈ (CR ∪ S) ∧
       z ∈ remove(y,CL) = z ∈ remove(y,CR) ) ]
```

[10] Indeed the **partitioned by** construct was inspired by the Larch Prover where it is available.

It is not difficult to show that `CONTAINER` $\models$ "Simplifiability of the Observability Kernel" and that `CONTAINER` $\models$ "Reduced Observability Kernel". Hence we now consider the enriched specification `CONTAINER partitioned by` $\in$, i.e. the specification `CONTAINER` enriched by the axiom:

$\forall$ `CL, CR : Container . [` $(\forall$ `x : Elem . x` $\in$ `CL = x` $\in$ `CR )` $\Longrightarrow$ `CL = CR ]`

and it is then very easy to prove that:

`CONTAINER partitioned by` $\in \models$ `insert(x,insert(x,S)) = insert(x,S)`

and that:

`CONTAINER partitioned by` $\in \models$

`insert(x,insert(y,S)) = insert(y,insert(x,S))`

which means that these two equations are behaviourally valid in the model class of the behavioural specification (`CONTAINER`, {`Elem, Nat, Bool`}). This means as well that this behavioural specification can be considered as a correct abstract implementation of sets. ◊

6 Conclusion

We have provided a technique that allows us to reduce the infinitary characterization of behavioural equality to a finitary one. Hence to prove behavioural theorems we can use arbitrary theorem provers for standard first-order logic. Our technique relies on the so-called "Observability Kernel Assumption". Unfortunately there are interesting examples where this condition is not satisfied (like the usual specification of stacks.) However, in all such cases that we have considered so far we can define a conservative extension of the given specification by introducing appropriate auxiliary function symbols such that the extended specification satisfies the Observability Kernel Assumption. It is an interesting objective of further research to study under which conditions appropriate conservative extensions exist and to develop a general method for the construction of such extensions.

Acknowledgements: This work is partially supported by the French-German cooperation program Procope, a joint CNRS-NSF grant, the E.C. ESPRIT Working Group COMPASS, the E.C. HCM project MeDiCis, and the German BMFT project SPECTRUM. Special thanks are due to Fernando Orejas, Andrzej Tarlecki and Martin Wirsing for fruitful discussions, to Steve Garland for suggesting improvements in our LP scripts and to Franck Vedrine for running the examples with LP. It is a pleasure to acknowledge also Hubert Baumeister who pointed out that the proofs of our results could be simplified by avoiding any reference to the context induction principle.

References

1. B. Bauer and R. Hennicker. Proving the correctness of algebraic implementations by the ISAR system. In *Proc. of DISCO'93*, pages 2–16. Springer-Verlag L.N.C.S. 722, 1993.

2. G. Bernot and M. Bidoit. Proving the correctness of algebraically specified software: modularity and observability issues. In *Proc. of AMAST'91*, pages 216–242. Springer-Verlag Workshops in Computing Series, 1992.
3. G. Bernot, M. Bidoit, and T. Knapik. Towards an adequate notion of observation. In *Proc. of ESOP'92*, pages 39–55. Springer-Verlag L.N.C.S. 582, 1992.
4. M. Bidoit and R. Hennicker. How to prove observational theorems with LP. In *Proc. of the First International Workshop on Larch*. Springer-Verlag Workshops in Computing Series, 1993.
5. M. Bidoit, R. Hennicker, and M. Wirsing. Characterizing behavioural semantics and abstractor semantics. In *Proc. of ESOP'94*, pages 105–119. Springer-Verlag L.N.C.S. 788, 1994.
6. H. Ehrig and B. Mahr. *Fundamentals of algebraic specification 1. Equations and initial semantics*, volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.
7. S. Garland and J. Guttag. An overview of LP, the Larch Prover. In *Proc. of the Third International Conference on Rewriting Techniques and Applications*, pages 137–151. Springer-Verlag L.N.C.S. 355, 1989.
8. J. Goguen and J. Meseguer. Universal realization, persistent interconnection and implementation of abstract modules. In *Proc. of 9th ICALP*, pages 265–281. Springer-Verlag L.N.C.S. 140, 1982.
9. J.A. Goguen, J.W. Thatcher, and E.G. Wagner. *An initial approach to the specification, correctness, and implementation of abstract data types*, volume 4 of *Current Trends in Programming Methodology*. Prentice Hall, 1978.
10. R. Hennicker. Context induction: a proof principle for behavioural abstractions and algebraic implementations. *Formal Aspects of Computing*, 3(4):326–345, 1991.
11. R. Milner. Fully abstract models of typed λ-calculi. *Theoretical Computer Science*, 4:1–22, 1977.
12. P. Nivela and F. Orejas. Initial behaviour semantics for algebraic specification. In *Recent Trends in Data Type Specification*, pages 184–207. Springer-Verlag L.N.C.S. 332, 1988.
13. F. Orejas, M. Navarro, and A. Sànches. Implementation and behavioural equivalence: A survey. In *Recent Trends in Data Type Specification*, pages 93–125. Springer-Verlag L.N.C.S. 655, 1993.
14. H. Reichel. Initial restrictions of behaviour. In *Proc. of IFIP Working Conference, The Role of Abstract Models in Information Processing*, 1985.
15. D. Sannella and A. Tarlecki. On observational equivalence and algebraic specification. In *Proc. of TAPSOFT'85*, pages 308–322. Springer-Verlag L.N.C.S. 185, 1985.
16. O. Schoett. Behavioural correctness of data representation. *Science of Computer Programming*, 14:43–57, 1990.

This article was processed using the LaTeX macro package with LLNCS style