Behavioural theories and the proof of behavioural properties

Michel Bidoit [a] and Rolf Hennicker [b]

[a] *LIENS, C.N.R.S. U.R.A. 1327 & Ecole Normale Supérieure, 45 Rue d'Ulm, F–75230 Paris Cedex 05, France*

[b] *Institut für Informatik, Ludwig-Maximilians-Universität München, Oettingenstr. 67, D–80538 München, Germany*

Abstract

Behavioural theories are a generalization of first-order theories where the equality predicate symbol is interpreted by a behavioural equality of objects (and not by their identity). In this paper we first consider arbitrary behavioural equalities determined by some (partial) congruence relation and we show how to reduce the behavioural theory of any class of algebras to (a subset of) the standard theory of some corresponding class of algebras. This reduction is the basis of a method for proving behavioural theorems whenever an axiomatization of the behavioural equality is provided. Then we focus on the important special case of (partial) observational equalities where two elements are observationally equal if they cannot be distinguished by observable computations over some set of input values. We provide general conditions under which an obvious infinite axiomatization of the observational equality can be replaced by a finitary one and we provide methodological guidelines for finding such finitary axiomatizations. As a consequence, any proof system for first-order logic can be used to prove the behavioural validity of first-order formulas with respect to a given (partial) observational equality.

1 Introduction

Behavioural abstraction plays a prominent rôle in formal software development, since it provides a suitable basis for defining adequate correctness concepts (cf. e.g. [12,21,19,23,2,20]). 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.

Behavioural correctness concepts can be formalized using a behavioural logic, where the usual satisfaction relation of first-order logic with equality is generalized to a behavioural satisfaction relation determined by a family of (partial) congruence relations (one for each algebra), called *behavioural equality*. The most important examples of behavioural equalities are observational equalities relating any two elements of an algebra which cannot be distinguished by observable computations. In the literature there are different definitions of observational equalities according to different choices of the input values that are allowed for observable computations. For instance [21] considers a "total" observational equality imposing no restriction on the input values while in [19] a "partial" observational equality is used where only observable inputs are allowed for observable computations. Since we consider arbitrary partial congruence relations both notions are captured by our general concept of behavioural equality.

For performing correctness proofs in a behavioural framework it is crucial to show that the axioms of a given specification are behaviourally satisfied by an implementation. In other words this means that the axioms of a given specification belong to the *behavioural theory* of the implementing specification where the behavioural theory of a class $\boldsymbol{C}$ of algebras is the set of all formulas that are behaviourally satisfied by all algebras of $\boldsymbol{C}$. Unfortunately it is usually difficult to prove that a formula belongs to the behavioural theory of a given class $\boldsymbol{C}$ of algebras, and the behavioural satisfaction relation does not even fulfil the satisfaction condition of institutions. Therefore we are interested in finding "nice" characterizations of behavioural theories which allow us to prove behavioural theorems using standard proof techniques as implemented, for instance, in any available theorem prover for (standard) first-order logic. We split this task into two major parts: First we show how to reduce for any arbitrary behavioural equality and for any class $\boldsymbol{C}$ of algebras the behavioural theory of $\boldsymbol{C}$ to (a subset of) the standard theory of some corresponding class of algebras and we show that this reduction is useful for proving behavioural theorems if an axiomatization of the behavioural equality is provided. Then we focus on (partial) observational equalities and we provide general conditions under which an obvious infinite axiomatization of the observational equality can be replaced by a finitary one.

More precisely we proceed as follows. In a first step (Section 4) we provide a general construction (called "lift operator") which introduces explicit predicate symbols for denoting the behavioural equality. We show that a formula ϕ is behaviourally valid with respect to a class $\boldsymbol{C}$ of algebras if and only if its lifted version $\mathcal{L}(\phi)$ is valid in the standard sense with respect to all lifted algebras of $\mathcal{L}(\boldsymbol{C})$ (Theorem 4.2). The usefulness of this characterization of behavioural theories still depends on the possibility to prove standard theorems over $\mathcal{L}(\boldsymbol{C})$. Therefore we introduce in Section 5 a general notion of (infinitary) axiomatization of a behavioural equality and we show that such an axiomatization allows us to characterize the class $\mathcal{L}(\boldsymbol{C})$ of lifted algebras (Theorem 5.5).

In particular, we see that given an axiomatizable class $\boldsymbol{C}$ (i.e. $\boldsymbol{C}$ is the model class of a flat standard specification) and given an axiomatization of the behavioural equality then $\mathcal{L}(\boldsymbol{C})$ is axiomatizable as well. However, we still have the problem that the axiomatization of the behavioural equality may be given by a set of *infinitary* formulas since in concrete examples (as in the case of observational equalities) only an infinitary axiomatization may be immediately deduced from the definition of the given behavioural equality. Hence, in the next step (Section 6) we consider finitary axiomatizations of the behavioural equality with the help of auxiliary hidden sorts and function symbols and we show that whenever such an axiomatization of the behavioural equality for a class $\boldsymbol{C}$ of algebras is provided, then one can prove behavioural theorems over $\boldsymbol{C}$ using standard proof techniques (Theorem 6.3).

These results lead to a general framework for the reduction of behavioural theories to standard theories. In Section 7 we study how a concrete finitary axiomatization (with hidden part) can be obtained in the case of (partial) observational equalities. For this purpose we set out from an obvious infinitary axiomatization of the partial observational equality which says that two elements a and b are observationally equal if they are denotable by terms which contain only input variables and if all applications of observable contexts to a and b yield the same result. Both properties are easily axiomatizable, but in general only by infinitary formulas, since there are usually infinitely many terms with input variables and also infinitely many observable contexts. How to find a finitary axiomatization of a (partial) observational equality for a given class $\boldsymbol{C}$ of algebras is based on two ideas: On one hand one has to specify the definedness (i.e. denotability by terms with input variables) with the help of new (hidden) predicate symbols. On the other hand we suggest to reduce the set of all observable contexts to a smaller set of contexts which is sufficient for describing the observational equality on all algebras of $\boldsymbol{C}$. We show that any smaller set of observable contexts which induces a congruence relation on the algebras of $\boldsymbol{C}$ is appropriate for this purpose (Proposition 7.4 and Theorem 7.7). If we can find a finite set of observable contexts with this property we can immediately derive a finitary axiomatization of the observational equality (Corollary 7.6). If we can only find an appropriate infinite subset of the observable contexts then the equality induced by this infinite set has to be axiomatized using contexts with hidden function symbols (Theorem 7.11). As a last result we show that it is always possible to construct a finitary axiomatization with hidden part of any partial observational equality (Proposition 7.13). However, this result is mainly of theoretical interest, since it relies on a technically complex encoding of the observational equality.

In Section 8 we summarize our method for proving the behavioural validity of formulas with respect to a given (partial) observational equality, and we apply our method to various concrete examples.

In Section 9 we introduce an alternative technique for obtaining a finitary axiomatization of partial observational equalities. The main idea of this section is to show that it is always possible to consider, instead of a partial observational equality, a corresponding total observational equality, provided we replace the class $\boldsymbol{C}$ of algebras we are interested in by another related class of algebras (Theorem 9.4). Using this alternative technique, one first get rid of the problems raised by partiality, and then the method described in Section 8 is applied, i.e. we have to obtain a suitable finitary axiomatization of the corresponding total observational equality. However this task is in general much simpler for total observational equalities than for partial ones.

In Section 10 we extend our study by considering the behavioural theories of "behavioural" and "abstractor" specifications (cf. [7]). We show that for such specifications we can first apply useful reductions before using our general proof method. Moreover, we show how to use a characterization of "fully abstract algebras" (w.r.t. the considered behavioural equality) to further optimize our proof techniques (Theorem 10.11).

In Section 11 we end by some concluding remarks and we discuss related approaches in the literature.

2 Basic notions

We assume that the reader is familiar with algebraic specifications (cf. e.g. [9], [28]). 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.*[1] To each function symbol $f \in F$ is associated an *arity* $s_1 \dots s_n \to s$ 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 $s_1 \dots s_n \to s$, 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). A Σ-algebra A' is a *subalgebra* of a Σ-algebra A if $A'_s \subseteq A_s$ for all $s \in S$ and if for all $f \in F$ the restriction of f^A to A' is the function $f^{A'}$. If $S' \subseteq S$ and $F' \subseteq F$ then $\Sigma' = (S', F')$ is called a *subsignature* of $\Sigma = (S, F)$. If Σ' is a subsignature of Σ, the *restriction* of a Σ-algebra A to the subsignature Σ', denoted by $A|_{\Sigma'}$, is defined by $(A|_{\Sigma'})_s = A_s$ for each $s \in S'$ and $f^{A|_{\Sigma'}} = f^A$ for each $f \in F'$. The restriction of a class $\boldsymbol{C}$ of Σ-algebras to a subsignature Σ', denoted by $\boldsymbol{C}|_{\Sigma'}$, is defined by $\boldsymbol{C}|_{\Sigma'} \stackrel{\text{def}}{=} \{A|_{\Sigma'} \mid A \in \boldsymbol{C}\}$. Throughout

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

this paper we always assume that the carrier sets A_s of a Σ-algebra A are not empty. Σ-morphisms are defined as usual. The category of all Σ-algebras is denoted by $\mathrm{Alg}(\Sigma)$.

Given an arbitrary S-sorted family $X = (X_s)_s \in S$ of sets X_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). In several occasions we will consider a subset $\mathrm{In} \subseteq S$ and we will choose $X_s = \emptyset$ for all $s \in S \setminus \mathrm{In}$ (and $X_s \neq \emptyset$ for all $s \in \mathrm{In}$). In that case, due to the non-empty carrier set requirement of above, we will always assume that the signature Σ is *sensible* w.r.t. In, i.e. that for all $s \in S \setminus \mathrm{In}$ (and hence for all $s \in S$), there exists a term t of sort s built with function symbols of Σ and variables of the non-empty sets $X_{s'}$ with $s' \in \mathrm{In}$. Given a Σ-algebra A, a *valuation* $\alpha : X \to A$ is a family of mappings $(\alpha_s : X_s \to A_s)_{s\in S}$. Any valuation $\alpha : X \to A$ uniquely extends to a Σ-morphism $I_\alpha : T_\Sigma(X) \to A$, called the *interpretation* associated to α.

A *partial Σ-congruence* on a Σ-algebra A is a family $\approx_A = (\approx_{A,s})_{s\in S}$ of partial equivalence relations (i.e. symmetric and transitive relations) $\approx_{A,s}$ on A_s compatible with the signature Σ, i.e. for all $f \in F$ of arity $s_1 \dots s_n \to s$, for all $a_i, b_i \in A_{s_i}$, if $a_i \approx_{A,s_i} b_i$ then $f^A(a_1, \dots, a_n) \approx_{A,s} f^A(b_1, \dots, b_n)$.[2] A *$\Sigma$-congruence* $\approx_A$ is *total* if for all a in A, $a \approx_A a$, i.e. all relations $\approx_{A,s}$ are reflexive. The "definition domain" of a partial congruence $\approx_A$, denoted by $\mathrm{Dom}(\approx_A)$, is defined by $\{a \in A \mid a \approx_A a\}$ and is a subalgebra of A (moreover the restriction of $\approx_A$ to $\mathrm{Dom}(\approx_A)$ is a total Σ-congruence on $\mathrm{Dom}(\approx_A)$). In the following $A/{\approx_A}$ denotes the quotient algebra of $\mathrm{Dom}(\approx_A)$ by $\approx_A$.

In the sequel of this paper we assume given an arbitrary but fixed family $X = (X_s)_{s\in S}$ of countably infinite sets X_s of variables of sort $s \in S$. *First-order Σ-formulas* are defined as usual, from equations $l = r$, the logical connectives $\neg, \wedge, \vee, \dots$ and the quantifiers $\forall, \exists$. We will also use *infinitary Σ-formulas* of the form $\bigwedge_{i\in I} \phi_i$ and $\bigvee_{i\in I} \phi_i$, 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 $t \in T_\Sigma(X)$, $\mathrm{Var}(t)$ denotes the set of variables occurring in t, and similarly $\mathrm{Var}(l, r)$ for a couple of terms l, r. For instance a universally quantified equation will be denoted by $\forall \mathrm{Var}(l, r).\, l = r$. Moreover, $\mathrm{FreeVar}(\phi)$ denotes the set of the free variables of the formula ϕ, which is defined as usual.

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 predicate symbol $=$ is interpreted in the carrier sets of the algebra by the set-theoretic equality. Due to the requirement of non-empty carrier sets, no pathological situations can occur with respect to the satisfaction relation

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

(cf. [16]).

A *basic (algebraic) specification* SP is a tuple $\langle \Sigma, \mathcal{A}x \rangle$ where $\Sigma = (S, F)$ is a signature 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, i.e. $\mathrm{Mod}(\mathrm{SP}) \stackrel{\text{def}}{=} \{A \in \mathrm{Alg}(\Sigma) \mid A \models \phi \text{ for all } \phi \in \mathcal{A}x\}$.

An algebraic specification language is called *ASL-like* if to any specification SP is associated a signature, denoted by Sig(SP), and a class of models, denoted by Mod(SP), such that $\mathrm{Mod}(\mathrm{SP}) \subseteq \mathrm{Alg}(\mathrm{Sig}(\mathrm{SP}))$, and if the language contains basic specifications as defined above and (at least) an operator $+$ for the combination of specifications SP and SP$'$ such that $\mathrm{Sig}(\mathrm{SP}+\mathrm{SP}') = \mathrm{Sig}(\mathrm{SP}) \cup \mathrm{Sig}(\mathrm{SP}')$ and $\mathrm{Mod}(\mathrm{SP}+\mathrm{SP}') = \{A \in \mathrm{Alg}(\mathrm{Sig}(\mathrm{SP}+\mathrm{SP}')) \mid A|_{\mathrm{Sig}(\mathrm{SP})} \in \mathrm{Mod}(\mathrm{SP})$ and $A|_{\mathrm{Sig}(\mathrm{SP}')} \in \mathrm{Mod}(\mathrm{SP}')\}$. In the following, by an ASL-like structured specification SP, we always refer to a specification written with such an ASL-like specification language (cf. e.g. [25]).

For any class $\boldsymbol{C}$ of Σ-algebras and any ASL-like structured specification SP$'$ with signature Σ', we denote by $\boldsymbol{C} + \mathrm{SP}'$ the class of $(\Sigma \cup \Sigma')$-algebras defined by $\boldsymbol{C} + \mathrm{SP}' \stackrel{\text{def}}{=} \{A \in \mathrm{Alg}(\Sigma \cup \Sigma') \mid A|_\Sigma \in \boldsymbol{C} \text{ and } A|_{\Sigma'} \in \mathrm{Mod}(\mathrm{SP}')\}$. This notation is consistent with the sum of specifications, since obviously we have $\mathrm{Mod}(\mathrm{SP}) + \mathrm{SP}' = \mathrm{Mod}(\mathrm{SP} + \mathrm{SP}')$.

The *(standard) theory* of a class $\boldsymbol{C}$ of Σ-algebras, denoted by $\mathrm{Th}(\boldsymbol{C})$, is defined by $\mathrm{Th}(\boldsymbol{C}) \stackrel{\text{def}}{=} \{\Sigma\text{-formula } \phi \mid A \models \phi \text{ for all } A \in \boldsymbol{C}\}$. In the following $\boldsymbol{C} \models \phi$ is an equivalent notation for $\phi \in \mathrm{Th}(\boldsymbol{C})$ and similarly $\mathrm{SP} \models \phi$ is an equivalent notation for $\phi \in \mathrm{Th}(\mathrm{Mod}(\mathrm{SP}))$. Note that we will always consider theories including infinitary Σ-formulas. However it is obvious that all our results remain valid if we restrict to first-order theories, i.e. theories consisting only of finitary (first-order) Σ-formulas.

In practice it is often useful to consider, instead of arbitrary Σ-algebras, algebras that are finitely generated by a distinguished subset of the function symbols, called constructors. In these algebras all elements can be denoted by a constructor term (which is built only with constructor symbols and variables of those sorts for which no constructor is defined). More precisely, such generation principles can be formalized using reachability constraints as follows:

(i) A *reachability constraint* over a signature $\Sigma = (S, F)$ is a pair $\mathcal{R} = (S_\mathcal{R}, F_\mathcal{R})$ such that $S_\mathcal{R} \subseteq S$, $F_\mathcal{R} \subseteq F$ and for any $f \in F_\mathcal{R}$ with arity $s_1 \ldots s_n \to s$ the sort s belongs to $S_\mathcal{R}$. A sort $s \in S_\mathcal{R}$ is called *constrained sort* and a function symbol $f \in F_\mathcal{R}$ is called *constructor symbol* (or briefly *constructor*). We assume also that for each constrained sort $s \in S_\mathcal{R}$ there exists at least one constructor in $F_\mathcal{R}$ with s as codomain. (This ensures that Σ is sensible w.r.t. $S \setminus S_\mathcal{R}$.)

(ii) A *constructor term* is a term $t \in T_{\Sigma'}(X')_s$ of sort $s \in S_{\mathcal{R}}$, where $\Sigma' = (S, F_{\mathcal{R}})$, $X' = (X'_s)_{s \in S}$ with $X'_s = X_s$ if $s \in S \setminus S_{\mathcal{R}}$ and $X'_s = \emptyset$ if $s \in S_{\mathcal{R}}$. The set of constructor terms is denoted by $T_{\mathcal{R}}$.

(iii) A Σ-algebra A *satisfies a reachability constraint* $\mathcal{R} = (S_{\mathcal{R}}, F_{\mathcal{R}})$, denoted by $A \models \mathcal{R}$, if for any $s \in S_{\mathcal{R}}$ and any $a \in A_s$, there exists a constructor term $t \in (T_{\mathcal{R}})_s$ and a valuation $\alpha : X' \to A$ such that $I_\alpha(t) = a$. (Note that this definition is independent of X because X_s is countably infinite for all $s \in S$.)

It is important to note that reachability constraints can be expressed by infinitary sentences:

Fact 2.1 *Let A be a Σ-algebra and $\mathcal{R} = (S_{\mathcal{R}}, F_{\mathcal{R}})$ be a reachability constraint over Σ. Then $A \models \mathcal{R}$ if and only if $A \models \bigwedge_{s \in S_{\mathcal{R}}} \mathrm{GEN}_s^{\mathcal{R}}$, where $\mathrm{GEN}_s^{\mathcal{R}}$ is the infinitary Σ-sentence defined by $\mathrm{GEN}_s^{\mathcal{R}} \stackrel{\text{def}}{=} \forall x{:}s. \bigvee_{t \in (T_{\mathcal{R}})_s} \exists \mathrm{Var}(t).\, x = t$.*

According to this fact, specifications with finitary axioms and reachability constraints can be defined as a particular kind of basic specifications with infinitary axioms as follows. Let Σ be a signature, $\mathcal{R} = (S_{\mathcal{R}}, F_{\mathcal{R}})$ be a reachability constraint over Σ and $\mathcal{A}x$ be a set of finitary Σ-sentences. Then the triple $\mathrm{SP} = \langle \Sigma, \mathcal{R}, \mathcal{A}x \rangle$ is, by definition, the basic specification $\langle \Sigma, \mathcal{A}x \cup \{\mathrm{GEN}_s^{\mathcal{R}} \mid s \in S_{\mathcal{R}}\} \rangle$. In the following a specification will be called *smooth* if it is built using the + specification-building primitive and specifications with finitary axioms and reachability constraints. To put emphasis on smooth specifications is justified by the fact that for such specifications it is easy to obtain sound proof rules by combining a sound (and complete) proof system for many-sorted first order logic with equality and structural induction w.r.t. the defined constructors.

Example 2.2 Let us consider the following `CONTAINER` specification.

spec: `CONTAINER`
  use: `ELEM, NAT, BOOL`
  sort: `Container`
  generated by:
    $\emptyset$: $\to$ `Container`
    `insert : Elem Container` $\to$ `Container`
  operations:
    $\_ \cup \_$: `Container Container` $\to$ `Container`
    `remove : Elem Container` $\to$ `Container`
    $\_ \in \_$: `Elem Container` $\to$ `Bool`
    `card : Container` $\to$ `Nat`
    `subset : Container Container` $\to$ `Bool`
  axioms:
    $\forall$ `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)
e ≠ e' ⇒ remove(e,insert(e',S)) = insert(e',remove(e,S))
e ∈ ∅ = false
[e ∈ insert(e',S) = true] ⇔ [(e = e') ∨ (e ∈ S = true)]
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 sort Container is constrained by the constructors ∅ 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 ∅ and insert) that: [3]

CONTAINER $\models$ S ∪ ∅ = S ,
CONTAINER $\models$ [e ∈ (S ∪ S') = true] ⇔
[(e ∈ S = true) ∨ (e ∈ S' = true)] ,

but it is important to note that:

CONTAINER $\not\models$ insert(e,insert(e,S)) = insert(e,S) ,
CONTAINER $\not\models$ insert(e,insert(e',S)) = insert(e',insert(e,S)) ,

i.e. the above equations are not valid in all models of CONTAINER. As a consequence, the CONTAINER specification cannot be considered as a correct implementation of the usual specification of sets if we require that an implementation satisfies (in the standard sense) all axioms of the given specification.

3 Behavioural theories

The main idea underlying the behavioural approach to the satisfaction of formulas is to assume that any Σ-algebra A is equipped with a partial congruence relation $\approx_A$, called the *behavioural equality* w.r.t. A, where for any two elements $a, b \in A$, $a \approx_A b$ holds whenever a and b are considered to be "behaviourally indistinguishable". More precisely:

[3] For the sake of simplicity the variables occurring in the formulas used in our examples are implicitly universally quantified.

Definition 3.1 (Behavioural equality and behaviour) *Given a signature Σ, a Σ-behavioural equality is a family $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ of partial Σ-congruence relations. For any Σ-algebra A, the quotient algebra $A/{\approx_A}$ is called the behaviour of A (w.r.t. $\approx$).*

First remember that $A/{\approx_A}$ denotes the quotient algebra of $\mathrm{Dom}(\approx_A)$ by $\approx_A$. Hence the behaviour $A/{\approx_A}$ of an algebra A forgets "irrelevant" elements (i.e. those not in $\mathrm{Dom}(\approx_A)$) and identifies all elements which are indistinguishable "from the outside", i.e. that are behaviourally equal. Hence the behaviour $A/{\approx_A}$ can be considered as the "black box view" of the algebra A. The most important examples of behavioural equalities are observational equalities relating any two elements of an algebra which cannot be distinguished by observable computations, cf. Section 3.2. In practice it is often the case that some additional properties are required for behavioural equalities (cf. e.g. [7,6] and Section 10), but since our results hold for arbitrary behavioural equalities there is no need for any extra assumption here.

As explained in the introduction, adequate correctness concepts must be defined with respect to some notion of behavioural abstraction. For instance, the above definition of the behaviour of an algebra leads to the notion of behavioural implementation studied in [6], where a specification SP-I is called a behavioural implementation of a specification SP (w.r.t. a given behavioural equality) if the behaviour of any model of SP-I (restricted to the signature of SP) is a model of SP. Thereby SP is considered as a specification of "intended behaviours" (cf. Example 3.12).

3.1 Behavioural satisfaction and behavioural theories

We start by generalizing the standard satisfaction relation to a behavioural satisfaction relation w.r.t. a given behavioural equality $\approx$. The idea is to interpret the variables occurring in a formula not by all values of an algebra A but only by values in $\mathrm{Dom}(\approx_A)$, and to interpret the equality predicate symbol not by the set-theoretic equality but by the given partial congruence relation $\approx_A$. A relationship between the standard satisfaction relation and the behavioural satisfaction relation is provided in Theorem 3.4.

Definition 3.2 (Behavioural satisfaction relation) *Let Σ be a signature and $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ be a Σ-behavioural equality. The behavioural satisfaction relation w.r.t. $\approx$, denoted by $\models_\approx$, is defined as follows.*
Let A be a Σ-algebra, $l, r \in T_\Sigma(X)_s$ be two terms of sort s, ϕ, ψ be two Σ-formulas, $\{\phi_i \mid i \in I\}$ be a countable family of Σ-formulas and $\alpha : X \to \mathrm{Dom}(\approx_A)$ be a valuation.

(i) $A, \alpha \models_\approx l = r$ holds if and only if $I_\alpha(l) \approx_A I_\alpha(r)$.

(ii) $A, \alpha \models_\approx \neg\phi$ *holds if and only if* $A, \alpha \models_\approx \phi$ *does not hold,*
$A, \alpha \models_\approx \phi \wedge \psi$ *holds if and only if both* $A, \alpha \models_\approx \phi$ *and* $A, \alpha \models_\approx \psi$ *hold.*

(iii) $A, \alpha \models_\approx \forall x{:}s.\phi$ *holds if and only if, for all valuations* $\beta : X \to \mathrm{Dom}(\approx_A)$ *with* $\beta(y) = \alpha(y)$ *for all* $y \neq x$, $A, \beta \models_\approx \phi$ *holds.*

(iv) $A, \alpha \models_\approx \bigwedge_{i \in I} \phi_i$ *holds if and only if, for all* $i \in I$, $A, \alpha \models_\approx \phi_i$ *holds.*

(v) $A \models_\approx \phi$ *if and only if* $A, \alpha \models_\approx \phi$, *for all valuations* $\alpha : X \to \mathrm{Dom}(\approx_A)$.

Hence Definition 3.2 is quite similar to the definition of the standard satisfaction relation, the main difference being for *(i)* where $I_\alpha(l) = I_\alpha(r)$ is replaced by $I_\alpha(l) \approx_A I_\alpha(r)$. Moreover, it is important to note that valuations have their range in $\mathrm{Dom}(\approx_A)$ and not in A, to take into account the fact that $\approx_A$ is a partial congruence.

According to the generalization of the standard satisfaction relation to the behavioural satisfaction relation with respect to a behavioural equality $\approx$, we consider the behavioural theory with respect to $\approx$ of a given class $\boldsymbol{C}$ of Σ-algebras.

Definition 3.3 (Behavioural theory) *Let* Σ *be a signature and* $\approx$ *be a* Σ-*behavioural equality. Let* $\boldsymbol{C} \subseteq \mathrm{Alg}(\Sigma)$ *be a class of* Σ-*algebras. The behavioural theory of* $\boldsymbol{C}$ *w.r.t.* $\approx$, *denoted by* $\mathrm{Th}_\approx(\boldsymbol{C})$, *is defined by:*
$\mathrm{Th}_\approx(\boldsymbol{C}) \stackrel{\mathrm{def}}{=} \{\Sigma\text{-}formula\ \phi \mid A \models_\approx \phi \text{ for all } A \in \boldsymbol{C}\}$.
In the following $\boldsymbol{C} \models_\approx \phi$ *is an equivalent notation for* $\phi \in Th_\approx(\boldsymbol{C})$ *and similarly* $\mathrm{SP} \models_\approx \phi$ *is an equivalent notation for* $\phi \in \mathrm{Th}_\approx(\mathrm{Mod}(\mathrm{SP}))$.

The following theorem, proved in [7], provides a relationship between the behavioural satisfaction relation and the standard one, hence between behavioural theories and standard theories.

Theorem 3.4 ([7, Theorem 3.11]) *Let* Σ *be a signature and* $\approx$ *be a* Σ-*behavioural equality. For any* Σ-*algebra* A, Σ-*formula* ϕ, *class* $\boldsymbol{C}$ *of* Σ-*algebras, the following holds:*

(i) $A \models_\approx \phi$ *if and only if* $A/\!\approx_A \models \phi$.

(ii) $\mathrm{Th}_\approx(\boldsymbol{C}) = \mathrm{Th}(\boldsymbol{C}/\!\approx)$,
where $\boldsymbol{C}/\!\approx$ *denotes the behavioural quotient of* $\boldsymbol{C}$, *defined by:*
$\boldsymbol{C}/\!\approx \stackrel{\mathrm{def}}{=} \{A/\!\approx_A \mid A \in \boldsymbol{C}\}$.

This theorem shows that an algebra behaviourally satisfies a formula if and only if its behaviour satisfies (in the standard sense) the same formula. Thus all the results developed in the following for the behavioural theory of a class $\boldsymbol{C}$ of algebras are equivalently valid for the standard theory of the behavioural quotient of $\boldsymbol{C}$. For instance, to show that a specification SP-I is a behavioural implementation of a basic specification $\mathrm{SP} = \langle \Sigma, \mathcal{A}x \rangle$, i.e. to show that the behaviour of any model of SP-I (restricted to the signature Σ) satisfies the axioms $\mathcal{A}x$, it is equivalent to show that $\mathcal{A}x$ is included in the behavioural theory of the model class of SP-I (restricted to the signature Σ), i.e. to prove that the sentences of $\mathcal{A}x$ are behavioural theorems w.r.t. this class of models (cf. [6]).

In practice it is usually difficult to prove behavioural theorems due to the generalized satisfaction relation. Although Theorem 3.4 *(ii)* shows that in principle behavioural theories can be reduced to standard theories, this result is of little practical interest because even if the class $\boldsymbol{C}$ is axiomatizable we have (in general) no straightforward proof system for the standard theory of the class $\boldsymbol{C}/\approx$ (since the formation of quotients does not preserve the validity of arbitrary formulas). Therefore we are interested in finding other characterizations of behavioural theories which allow us to prove behavioural theorems using standard proof techniques. For this purpose our general strategy will be first to reduce the behavioural theory $\mathrm{Th}_{\approx}(\boldsymbol{C})$ of some class $\boldsymbol{C}$ of algebras to (a subset of) the standard theory $\mathrm{Th}(\boldsymbol{D})$ of some other class $\boldsymbol{D}$ of algebras and then to look for an appropriate axiomatization (or a proof system) for $\boldsymbol{D}$ (provided that an axiomatization or a proof system for $\boldsymbol{C}$ is given).

This will be the topic of the next sections. Before let us study in more details the important special case of observational equalities.

3.2 The observational case

The most important examples of behavioural equalities are *observational equalities* of objects. The intuition behind observational equalities is the following one: Two objects of an algebra are considered to be observationally equal if they cannot be distinguished by "experiments" with "observable" results. In order to provide the necessary formalization, we will proceed as follows. First, we will define *contexts*, which are a special kind of terms representing the "experiments". Then we will show how a very general notion of *contextual equality* of objects can be associated to any choice of a set of contexts. Finally we define the observational equality of objects we are interested in as a special case of contextual equalities.

Definition 3.5 (Context) *Let $\Sigma = (S, F)$ be a signature and let $X = (X_s)_{s \in S}$ be the generally assumed family of countably infinite sets of variables of sort s. Let $Z = (\{z_s\})_{s \in S}$ be a disjoint S-sorted family of singleton sets.*

(i) A Σ-context is a Σ-term $C \in T_\Sigma(X \cup Z)$ which contains, besides variables in X, one or many occurrences of exactly one variable $z_s \in Z$, called the context variable *of C.*
(ii) By exception, $\mathrm{Var}(C)$ *will denote the set of variables occurring in C apart from the context variable of C.*

(iii) $C[t]$ *denotes the term obtained by substituting the term* $t \in T_\Sigma(X)_s$ *for the context variable* z_s *(of sort* s*) in* C*.*

(iv) Given an arbitrary set $\mathcal{C}$ *of* Σ*-contexts, we denote by* $\mathcal{C}(s)$ *the (possibly empty) subset of the contexts of* $\mathcal{C}$ *with context variable of sort* s*.*

$T_\Sigma(X \cup Z)$ represents the set of all possible experiments (with arbitrary input variables X). In general we may not want to allow experiments on any values in an algebra A, but only on those values that can be denoted by a term t, where some restrictions apply to the variables that may occur in t. More precisely, we will choose a subset of *input sorts* $\mathrm{In} \subseteq S$, and consider the smallest subalgebra $A[X_{\mathrm{In}}]$ of A generated by Σ and X_{In}, where X_{In} is the S-sorted family of variables defined by $(X_{\mathrm{In}})_s = \emptyset$ if $s \notin \mathrm{In}$ and $(X_{\mathrm{In}})_s = X_s$ if $s \in \mathrm{In}$ (where $X = (X_s)_{s \in S}$ is the generally assumed family of countably infinite sets of variables of sort s). This leads to the following definition of the (partial) contextual equality relation induced by some input sorts In and by a set $\mathcal{C}$ of Σ-contexts.

Definition 3.6 (Partial contextual equality) *Let* $\Sigma = (S, F)$ *be a signature,* $\mathrm{In} \subseteq S$ *be a set of input sorts such that* Σ *is sensible w.r.t.* In *(cf. Section 2),* $\mathcal{C}$ *be an arbitrary set of* Σ*-contexts and* A *be a* Σ*-algebra. The partial contextual equality on* A *induced by* $\mathcal{C}$ *and* In*, denoted by* $\approx_{\mathcal{C},\mathrm{In},A}$*, is defined as follows.*
Let $A[X_{\mathrm{In}}]$ *be the smallest subalgebra of* A *generated by* Σ *and* X_{In}*. The carrier sets of* $A[X_{\mathrm{In}}]$ *are defined by* $A[X_{\mathrm{In}}]_s = \{a \in A_s \mid \text{there exists a term } t \in T_\Sigma(X_{\mathrm{In}})_s \text{ and a valuation } \alpha : X_{\mathrm{In}} \to A \text{ such that } I_\alpha(t) = a\}$*.*
Two elements $a, b \in A_s$ *of sort* s *are contextually equal (w.r.t.* $\mathcal{C}$ *and* In*), denoted by* $a \approx_{\mathcal{C},\mathrm{In},A} b$*, if and only if both* a *and* b *belong to* $A[X_{\mathrm{In}}]_s$ *and, for all contexts* $C \in \mathcal{C}$ *with context variable* z_s *of sort* s*, for all valuations* $\alpha : X \to A[X_{\mathrm{In}}]$*, we have* $I_{\alpha_a}(C) = I_{\alpha_b}(C)$*, where* $\alpha_a, \alpha_b : X \cup \{z_s\} \to A[X_{\mathrm{In}}]$ *are the unique extensions of* α *defined by* $\alpha_a(z_s) = a$ *and* $\alpha_b(z_s) = b$*.*
Note that, if there is no context $C \in \mathcal{C}$ *with context variable of sort* s*, then we have* $a \approx_{\mathcal{C},\mathrm{In},A} b$*, for all* $a, b \in A[X_{\mathrm{In}}]_s$ *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 Σ-contexts if they belong to the chosen subalgebra $A[X_{\mathrm{In}}]$ and if they cannot be distinguished by at least one of the computations represented by the contexts of $\mathcal{C}$. Note that $\approx_{\mathcal{C},\mathrm{In},A}$ is a family of partial equivalence relations (one for each sort $s \in S$). However, $\approx_{\mathcal{C},\mathrm{In},A}$ is not necessarily a partial congruence relation, i.e. $\approx_{\mathcal{C},\mathrm{In},A}$ is not necessarily compatible with the signature Σ.

Even if $\approx_{\mathcal{C},\mathrm{In},A}$ is not a partial Σ-congruence, we can consider its "definition domain", denoted by $\mathrm{Dom}(\approx_{\mathcal{C},\mathrm{In},A})$, and defined by $\{a \in A \mid a \approx_{\mathcal{C},\mathrm{In},A} a\}$. Obviously we have $\mathrm{Dom}(\approx_{\mathcal{C},\mathrm{In},A}) = A[X_{\mathrm{In}}]$.

We are now ready to define (partial) observational equalities. For this, we assume given a signature $\Sigma = (S, F)$ and a distinguished set $\mathrm{Obs} \subseteq S$ of *observable* sorts (which denote the carrier sets of observable values). Moreover, we assume given a set $\mathrm{In} \subseteq S$ of *input* sorts such that Σ is sensible w.r.t. In. Then two objects of an algebra are considered to be observationally equal if they cannot be distinguished by "experiments" with observable results and with inputs chosen accordingly to In.

Definition 3.7 (Observable context and observational equality) *Let $\Sigma = (S, F)$ be a signature,* $\mathrm{Obs} \subseteq S$ *be a set of observable sorts and* $\mathrm{In} \subseteq S$ *be a set of input sorts, such that Σ is sensible w.r.t.* In.

(i) The set of all observable Σ-contexts, denoted by $\mathcal{C}_{\Sigma}^{\mathrm{Obs}}$, is defined as being the set of all Σ-contexts of observable sort $s \in$ Obs.
(ii) Let A be a Σ-algebra. The partial contextual equality on A induced by $\mathcal{C}_{\Sigma}^{\mathrm{Obs}}$ and In *(cf. Definition 3.6) is called the* (partial) observational equality *on A induced by* Obs *and* In *and is denoted by $\approx_{\mathrm{Obs},\mathrm{In},A}$.*

Remark 3.8 In [7], the partial observational equality induced by Obs and In is defined in a slightly different way. Instead of considering the observable contexts $\mathcal{C}_{\Sigma}^{\mathrm{Obs}}$ built from the signature Σ and arbitrary variables X, one restricts to observable contexts built from the signature Σ and variables of an input sort only (i.e. $T_{\Sigma}(X_{\mathrm{In}} \cup Z)$ is used instead of $T_{\Sigma}(X \cup Z)$ in Definition 3.5). Then two elements $a, b \in A_s$ of sort s are observationally equal if and only if both a and b belong to $A[X_{\mathrm{In}}]_s$ and, for all observable contexts $C \in T_{\Sigma}(X_{\mathrm{In}} \cup Z)$ with context variable z_s of sort s, for all valuations $\alpha : X_{\mathrm{In}} \to A$, we have $I_{\alpha_a}(C) = I_{\alpha_b}(C)$, where $\alpha_a, \alpha_b : X_{\mathrm{In}} \cup \{z_s\} \to A$ are the unique extensions of α defined by $\alpha_a(z_s) = a$ and $\alpha_b(z_s) = b$. It is fairly obvious that this leads to a definition equivalent to our Definition 3.7. However, Definition 3.7 will prove to be more convenient in the sequel.

Lemma 3.9 *The observational equality $\approx_{\mathrm{Obs},\mathrm{In},A}$ on A is a partial Σ-congruence (and* $\mathrm{Dom}(\approx_{\mathrm{Obs},\mathrm{In},A}) = A[X_{\mathrm{In}}]$*).*

The family $(\approx_{\mathrm{Obs},\mathrm{In},A})_{A \in \mathrm{Alg}(\Sigma)}$ of partial observational equalities (which in particular are partial Σ-congruences) will be denoted by $\approx_{\mathrm{Obs},\mathrm{In}}$. Most examples studied in the literature are captured by our definition:

1. If we choose $\mathrm{In} = S$, i.e. the elements of all carrier sets can be used as input for observable computations, then $A[X_{\mathrm{In}}] = A$ and $\approx_{\mathrm{Obs},S,A}$ is a total congruence on A which corresponds to the behavioural equality used e.g. in [21] and in [4].
2. If we choose $\mathrm{In} = \mathrm{Obs}$, i.e. only values generated from observable ones can be used as input for observable computations, then $\approx_{\mathrm{Obs},\mathrm{Obs},A}$ is a partial congruence on A and the resulting behavioural satisfaction relation corresponds to the one used in [19] for the behavioural satisfaction

of equations. The advantage here is that non-observable junk (i.e. values which are not reachable from the observable ones) will not be considered for the behavioural satisfaction of formulas and hence cannot cause problems, for instance, with respect to the correctness of implementations (cf. e.g. [20]).

Lemma 3.10 *Let us consider the total observational equality* $\approx_{\mathrm{Obs},S}$ *induced by a set* Obs *of observable sorts. Let* A *be a* Σ*-algebra and* $\forall \mathrm{Var}(l,r).\, l = r$ *be a universally quantified equation. Let* s *be the common sort of* l *and* r*.*

(i) If s *is an observable sort, then* $A \models_{\approx_{\mathrm{Obs},S}} \forall \mathrm{Var}(l,r).\, l = r$ *if and only if* $A \models \forall \mathrm{Var}(l,r).\, l = r$ *.*
(ii) If s *is a non-observable sort, then* $A \models_{\approx_{\mathrm{Obs},S}} \forall \mathrm{Var}(l,r).\, l = r$ *if and only if, for all observable contexts* $C \in \mathcal{C}^{\mathrm{Obs}}_{\Sigma}(s)$*,* $A \models \forall \mathrm{Var}(C) \cup \mathrm{Var}(l,r).\, C[l] = C[r]$ *.*

Remark 3.11 Lemma 3.10 is often used in the literature to define directly (i.e. without introducing explicitly the total observational equality) the behavioural satisfaction of equations. However the explicit definition we have chosen (cf. Definition 3.2) is necessary to define the behavioural satisfaction of arbitrary Σ-formulas w.r.t. an arbitrary behavioural equality (this idea is even extended to higher-order logic in [15]). On the other hand, this lemma suggests that, in the total observational framework, to prove the behavioural satisfaction of an equation $l = r$ (between non-observable terms of sort s), it is equivalent to prove the standard satisfaction of the infinite set of equations $C[l] = C[r]$, for all $C \in \mathcal{C}^{\mathrm{Obs}}_{\Sigma}(s)$. *Context Induction* (a specialized version of structural induction) was introduced in [13] as a means to prove such infinite sets of equations and has been implemented in the ISAR system (cf. [1]). Unfortunately proofs by context induction are quite complicated and difficult to handle in practice, especially in a first-order framework.

Example 3.12 Let us consider again our `CONTAINER` specification and assume that the observable sorts are `Elem`, `Nat` and `Bool` and that all sorts are input sorts. Let $\approx_{\mathrm{Cont}}$ denote the corresponding total observational equality. Two objects of sort `Container` will be considered as observationally 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 observationally 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 any observable context. For the same reasons, it is intuitively clear that the two characteristic equations of sets, `insert(e,insert(e,S)) = insert(e,S)` and `insert(e,insert(e',S)) = insert(e',insert(e,S))`, are behaviourally satisfied by all models of the `CONTAINER` specification. Indeed no observable experiment can distinguish the left and right-hand sides of these equations. Therefore, using Theorem 3.4, the behaviour of any model of `CONTAINER` is a model of the usual specification of sets, and `CONTAINER` can be considered as a correct behavioural implementation of sets. The aim of this paper is to provide a general proof technique to formally establish that the above intuition is right.

4 The lift operator

The main idea for reducing the behavioural theory of some class $\boldsymbol{C}$ of algebras to the standard theory of a related class $\boldsymbol{D}$ of algebras is to introduce explicit denotations for the given behavioural equality. For this purpose we use predicate symbols to denote the equivalence relations. [4]

Definition 4.1 (Lift operator) *Given a signature Σ, a Σ-algebra A, a class $\boldsymbol{C}$ of Σ-algebras and a Σ-formula ϕ, we define their lifted versions with respect to a given Σ-behavioural equality $\approx$ as follows:*

(i) $\mathcal{L}(\Sigma) \stackrel{\text{def}}{=} \Sigma \cup \{\sim_s\colon s\ s\}_{s\in S}$, *i.e.* $\mathcal{L}(\Sigma)$ *is the signature* Σ *enriched by binary predicate symbols* $\sim_s\colon s\ s$ *(to denote the behavioural equality), for each sort* s *in* S. *We will adopt an infix notation for the binary predicate symbols* $\sim_s$, *i.e. we write* $l \sim_s r$ *instead of* $\sim_s(l,r)$.
(ii) $\mathcal{L}(A)$ *is the unique* $\mathcal{L}(\Sigma)$*-algebra extension of* A *defined by:*
  (a) $\mathcal{L}(A)|_\Sigma \stackrel{\text{def}}{=} A$
  (b) For any s *in* S, $\sim_s^{\mathcal{L}(A)} \stackrel{\text{def}}{=} \approx_{A,s}$,
  i.e. for any a, b *in* $\mathcal{L}(A)_s$ $(= A_s)$, $a \sim_s^{\mathcal{L}(A)} b$ *if and only if* $a \approx_{A,s} b$.
(iii) $\mathcal{L}(\boldsymbol{C}) \stackrel{\text{def}}{=} \{\mathcal{L}(A) \mid A \in \boldsymbol{C}\}$.
(iv) $\mathcal{L}(\phi) \stackrel{\text{def}}{=} \left[\left(\bigwedge_{y:s\in\mathrm{FreeVar}(\phi)} D_s(y)\right) \Rightarrow \phi^*\right]$, *where* $D_s(y)$ *is an abbreviation for* $y \sim_s y$ *and* ϕ^* *is defined by induction on the structure of* ϕ *as follows:* [5]
  (a) If ϕ *is an equation* $l = r$ *between two terms of sort* s, *then* ϕ^* *is* $l \sim_s r$,
  (b) $(\neg\phi)^* = \neg(\phi^*)$, $(\phi_1 \wedge \phi_2)^* = (\phi_1^*) \wedge (\phi_2^*)$, $(\phi_1 \vee \phi_2)^* = (\phi_1^*) \vee (\phi_2^*)$, *and similarly for infinite conjunctions and disjunctions,*
  (c) $(\forall x{:}s.\phi)^* = \forall x{:}s.\,[D_s(x) \Rightarrow \phi^*]$.
  Note that if ϕ *is a closed* Σ*-formula then* $\mathcal{L}(\phi)$ *coincides with* ϕ^*.

The following theorem shows that for any class $\boldsymbol{C}$ of algebras the behavioural theory of $\boldsymbol{C}$ consists of all formulas ϕ whose lifted version $\mathcal{L}(\phi)$ belongs to the

[4] We assume the reader to be familiar with the usual notions of predicate symbols and their interpretations.
[5] Similar constructions, called relativizations, are used in [29,15].

standard theory of $\mathcal{L}(\boldsymbol{C})$.

Theorem 4.2 *Let Σ be a signature and $\approx$ be a Σ-behavioural equality. For any Σ-algebra A, class $\boldsymbol{C}$ of Σ-algebras, ASL-like structured specification* SP *with signature Σ (cf. Section 2), and Σ-formula ϕ, we have:*

(i) $A \models_{\approx} \phi$ *if and only if* $\mathcal{L}(A) \models \mathcal{L}(\phi)$.
(ii) $\boldsymbol{C} \models_{\approx} \phi$ *if and only if* $\mathcal{L}(\boldsymbol{C}) \models \mathcal{L}(\phi)$.
(iii) $\mathrm{SP} \models_{\approx} \phi$ *if and only if* $\mathcal{L}(\mathrm{Mod}(\mathrm{SP})) \models \mathcal{L}(\phi)$.

We will see in the next section that when SP is a basic specification then $\mathcal{L}(\mathrm{Mod}(\mathrm{SP}))$ can be axiomatized with the help of an axiomatization of the behavioural equality. More generally, if SP is an ASL-like structured specification, we will see that $\mathcal{L}(\mathrm{Mod}(\mathrm{SP}))$ can be expressed by a specification of the ASL-like language as soon as an axiomatization of the behavioural equality is provided (cf. Theorem 5.5).

The proof of Theorem 4.2 relies on the following two lemmas.

Lemma 4.3 *Let A be a Σ-algebra and ϕ be a Σ-formula. For all valuations $\alpha : X \to \mathrm{Dom}(\approx_A)$, the following conditions are equivalent:*

(i) $A, \alpha \models_{\approx} \phi$
(ii) $\mathcal{L}(A), \alpha \models \phi^*$
(iii) $\mathcal{L}(A), \alpha \models \mathcal{L}(\phi)$

Proof. Since A and $\mathcal{L}(A)$ have the same carrier sets, a valuation from X to $\mathrm{Dom}(\approx_A)$ is a special case of a valuation from X to $\mathcal{L}(A)$. Moreover, for any valuation $\alpha : X \to \mathrm{Dom}(\approx_A)$ and any variable y of sort s, we have $\mathcal{L}(A), \alpha \models D_s(y)$. It is therefore trivial to see that *(ii)* $\Leftrightarrow$ *(iii)*. We prove *(i)* $\Leftrightarrow$ *(ii)* by induction on the form of ϕ:

Case $l = r$: Let $\alpha : X \to \mathrm{Dom}(\approx_A)$ be an arbitrary valuation. $A, \alpha \models_{\approx} l = r$ iff $I_\alpha(l) \approx_A I_\alpha(r)$ iff $I_\alpha(l) \sim_s^{\mathcal{L}(A)} I_{\alpha(r)}$ (where s is the sort of l and r) iff $\mathcal{L}(A), \alpha \models l \sim_s r$ iff $\mathcal{L}(A), \alpha \models (l = r)^*$.

Case $\neg\phi$, $\phi \wedge \psi$, $\ldots$: For formulas of the form $\neg\phi$ and $\phi \wedge \psi$ the result follows directly from the induction hypothesis. This is also true for infinite conjunctions and disjunctions.

Case $\forall x{:}s.\phi$: Let $\alpha : X \to \mathrm{Dom}(\approx_A)$ be an arbitrary valuation. $A, \alpha \models_{\approx} \forall x{:}s.\phi$ iff, for all valuations $\beta : X \to \mathrm{Dom}(\approx_A)$ with $\beta(y) = \alpha(y)$ if $y \neq x$, $A, \beta \models_{\approx} \phi$ iff (by induction hypothesis) for all valuations $\beta : X \to \mathrm{Dom}(\approx_A)$ with $\beta(y) = \alpha(y)$ if $y \neq x$, $\mathcal{L}(A), \beta \models \phi^*$ iff (justification given below) for all valuations $\gamma : X \to \mathcal{L}(A)$ with $\gamma(y) = \alpha(y)$ if $y \neq x$, $\mathcal{L}(A), \gamma \models D_s(x) \Rightarrow \phi^*$ iff $\mathcal{L}(A), \alpha \models \forall x{:}s.[D_s(x) \Rightarrow \phi^*]$ iff $\mathcal{L}(A), \alpha \models (\forall x{:}s.\phi)^*$.
We still have to justify the central step. Hence we must prove that the two following conditions are equivalent:

(a) For all valuations $\beta : X \to \mathrm{Dom}(\approx_A)$ with $\beta(y) = \alpha(y)$ if $y \neq x$, $\mathcal{L}(A), \beta \models \phi^*$
(b) For all valuations $\gamma : X \to \mathcal{L}(A)$ with $\gamma(y) = \alpha(y)$ if $y \neq x$, $\mathcal{L}(A), \gamma \models D_s(x) \Rightarrow \phi^*$

The direction (b) $\Rightarrow$ (a) is obvious, since $\mathcal{L}(A), \beta \models D_s(x)$. To prove the other direction, assume (a) holds and let $\gamma : X \to \mathcal{L}(A)$ be an arbitrary valuation with $\gamma(y) = \alpha(y)$ if $y \neq x$.

Case $\gamma(x) \in \mathrm{Dom}(\approx_A)$: Then in (a) we can take $\beta = \gamma$ and we are done.

Case $\gamma(x) \notin \mathrm{Dom}(\approx_A)$: Then $\mathcal{L}(A), \gamma \not\models D_s(x)$, hence $\mathcal{L}(A), \gamma \models D_s(x) \Rightarrow \phi^*$. □

Lemma 4.4 *Let A be a Σ-algebra and ϕ be a Σ-formula. The following conditions are equivalent:*

(i) For all valuations $\alpha : X \to \mathrm{Dom}(\approx_A)$: $\mathcal{L}(A), \alpha \models \mathcal{L}(\phi)$
(ii) For all valuations $\beta : X \to \mathcal{L}(A)$: $\mathcal{L}(A), \beta \models \mathcal{L}(\phi)$

Proof. The direction *(ii)* $\Rightarrow$ *(i)* is obvious. To prove *(i)* $\Rightarrow$ *(ii)*, assume *(i)* holds and let $\beta : X \to \mathcal{L}(A)$ be an arbitrary valuation.

Case $\beta(y) \in \mathrm{Dom}(\approx_A)$ for all $y \in \mathrm{FreeVar}(\phi)$: Then there exists a valuation $\alpha : X \to \mathrm{Dom}(\approx_A)$ with $\alpha(y) = \beta(y)$ for all $y \in \mathrm{FreeVar}(\phi)$. By assumption $\mathcal{L}(A), \alpha \models \mathcal{L}(\phi)$. But then we conclude that $\mathcal{L}(A), \beta \models \mathcal{L}(\phi)$ since α and β coincide on all free variables of ϕ, hence of $\mathcal{L}(\phi)$.

Case $\beta(y) \notin \mathrm{Dom}(\approx_A)$ for some $y \in \mathrm{FreeVar}(\phi)$:
Then $\mathcal{L}(A), \beta \not\models \bigwedge_{y:s \in \mathrm{FreeVar}(\phi)} D_s(y)$ and therefore $\mathcal{L}(A), \beta \models \mathcal{L}(\phi)$ (by definition of $\mathcal{L}(\phi)$). □

Proof of Theorem 4.2. Let A be a Σ-algebra and ϕ be a Σ-formula. $A \models_\approx \phi$ iff (by definition) for all valuations $\alpha : X \to \mathrm{Dom}(\approx_A)$, $A, \alpha \models_\approx \phi$ iff (by Lemma 4.3) for all valuations $\alpha : X \to \mathrm{Dom}(\approx_A)$, $\mathcal{L}(A), \alpha \models \mathcal{L}(\phi)$ iff (by Lemma 4.4) for all valuations $\beta : X \to \mathcal{L}(A)$, $\mathcal{L}(A), \beta \models \mathcal{L}(\phi)$ iff (by definition) $\mathcal{L}(A) \models \mathcal{L}(\phi)$. Hence *(i)* is proved. *(ii)* and *(iii)* follow directly from *(i)*. □

Theorem 4.2 provides a means for reducing the set of behavioural theorems over an arbitrary class $\boldsymbol{C}$ of Σ-algebras to a set of standard theorems over a corresponding class $\mathcal{L}(\boldsymbol{C})$ of $\mathcal{L}(\Sigma)$-algebras. However, the usefulness of this reduction still depends on the possibility to perform proofs of standard theorems over $\mathcal{L}(\boldsymbol{C})$. Since $\mathcal{L}(\boldsymbol{C})$ is constructed on top of $\boldsymbol{C}$ by introducing a denotation for the given behavioural equality, we claim that for proving standard theorems over $\mathcal{L}(\boldsymbol{C})$ we need, on one hand, a proof system for proving standard theorems over $\boldsymbol{C}$ (in the best case $\boldsymbol{C}$ is axiomatizable) and, on the other hand, we need an axiomatization of the behavioural equality which will be considered in the next sections.

5 Axiomatization of the behavioural equality

In the previous section we have shown how to replace the behavioural theory of some given class $\boldsymbol{C}$ of algebras by (a subset of) the standard theory of another related class of algebras. The next step is to provide a characterization of this class of algebras in terms of an axiomatization of the behavioural equality.

Indeed, symmetrically to the semantic lifting induced by a Σ-behavioural equality, we can define the "axiomatic lifting" induced by a family of appropriate formulas. This "axiomatic lifting" can be defined for any Σ_1-algebra, using Σ_1-formulas, where $\Sigma \subseteq \Sigma_1$: [6]

Definition 5.1 (Axiomatic lifting) *Let $\Sigma = (S, F)$ and Σ_1 be two signatures such that $\Sigma \subseteq \Sigma_1$, and let $\Sigma_1^{\mathcal{L}} \stackrel{\text{def}}{=} \mathcal{L}(\Sigma) \cup \Sigma_1$. Let $\phi = (\phi_s)_{s \in S}$ be an S-sorted family of arbitrary Σ_1-formulas, and assume that each ϕ_s has exactly two free variables, say x_s and y_s, both of sort $s \in S$. Let $\mathrm{AXL}[\phi]$ be the $\Sigma_1^{\mathcal{L}}$-sentence defined by $\mathrm{AXL}[\phi] \stackrel{\text{def}}{=} \bigwedge_{s \in S} \forall x_s, y_s{:}s.\, [\phi_s \Leftrightarrow x_s \sim_s y_s]$. Then for any Σ_1-algebra A, $\mathcal{AL}[\phi](A)$ denotes the unique[7] $\Sigma_1^{\mathcal{L}}$-algebra such that:*

(i) $\mathcal{AL}[\phi](A)|_{\Sigma_1} = A$.
(ii) $\mathcal{AL}[\phi](A) \models \mathrm{AXL}[\phi]$.

$\mathcal{AL}[\phi](A)$ is called the axiomatic lifting of A induced by the family ϕ.

Note that when $\Sigma_1 = \Sigma$ and ϕ is a family of Σ-formulas, then $\mathrm{AXL}[\phi]$ is itself a $\mathcal{L}(\Sigma)$-formula (since then $\Sigma_1^{\mathcal{L}} = \mathcal{L}(\Sigma)$).

An axiomatization of a Σ-behavioural equality $\approx$ will be a family of Σ-formulas such that the induced axiomatic lifting coincides with the semantic lifting associated to $\approx$ defined in the previous section.

Definition 5.2 (Axiomatization of the behavioural equality) *Let $\Sigma = (S, F)$ be a signature. An axiomatization of a Σ-behavioural equality $\approx$ is an S-sorted family $\mathrm{Beh} = (\mathrm{Beh}_s(x_s, y_s))_{s \in S}$ of (possibly infinitary) Σ-formulas, where x_s and y_s are the only free variables (both of sort s) of $\mathrm{Beh}_s(x_s, y_s)$, such that, for any Σ-algebra A, $\mathcal{AL}[\mathrm{Beh}](A) = \mathcal{L}(A)$. Whenever such an axiomatization exists, we say that the behavioural equality $\approx$ is axiomatizable.*

The following proposition provides simple criteria to check whether a family of formulas is indeed an axiomatization of a given behavioural equality.

[6] The motivations for using a larger signature Σ_1 will become clear in the next section when we consider axiomatizations with hidden parts.
[7] Since the sentence $\mathrm{AXL}[\phi]$ defines unambiguously the interpretation of the predicate symbol $\sim_s$, for each $s \in S$.

Proposition 5.3 (Criteria for infinitary axiomatizations) *Let $\Sigma = (S, F)$ be a signature and let* $\mathrm{Beh} = (\mathrm{Beh}_s(x_s, y_s))_{s\in S}$ *be an S-sorted family of (possibly infinitary) Σ-formulas, where x_s and y_s are the only free variables (both of sort s) of* $\mathrm{Beh}_s(x_s, y_s)$. Beh *is an axiomatization of a Σ-behavioural equality $\approx$ if and only if, for any Σ-algebra A, any sort s in S and any valuation $\alpha : \{x_s, y_s\} \to A$, $A, \alpha \models \mathrm{Beh}_s(x_s, y_s)$ if and only if $\alpha(x_s) \approx_A \alpha(y_s)$.*

Proof. Note that for any Σ-algebra A, the algebras A, $\mathcal{L}(A)$ and $\mathcal{AL}[\mathrm{Beh}](A)$ have the same carrier sets. Moreover, $\mathcal{AL}[\mathrm{Beh}](A) = \mathcal{L}(A)$ iff for all sorts $s \in S$, $\sim_s^{\mathcal{L}(A)}$ and $\sim_s^{\mathcal{AL}[\mathrm{Beh}](A)}$ coincide. But we have, for any sort $s \in S$ and any valuation $\alpha : \{x_s, y_s\} \to A$ (hence for any two values $a, b \in A_s$):
– $A, \alpha \models \mathrm{Beh}_s(x_s, y_s)$ iff (since A and $\mathcal{AL}[\mathrm{Beh}](A)$ have the same carrier sets and since $\mathrm{Beh}_s(x_s, y_s)$ is a Σ-formula) $\mathcal{AL}[\mathrm{Beh}](A), \alpha \models \mathrm{Beh}_s(x_s, y_s)$ iff (by Definition 5.1) $\alpha(x_s) \sim_s^{\mathcal{AL}[\mathrm{Beh}](A)} \alpha(y_s)$ on one hand, and
– $\alpha(x_s) \approx_A \alpha(y_s)$ iff (by Definition 4.1) $\alpha(x_s) \sim_s^{\mathcal{L}(A)} \alpha(y_s)$ on the other.
This is enough to obtain the desired result. □

Example 5.4 Using the above criteria, it is straightforward to find infinitary axiomatizations of observational equalities:

1. The total observational equality $\approx_{\mathrm{Obs},S}$ induced by a set Obs of observable sorts (cf. Section 3.2) is axiomatized by the following infinitary formulas:
   $\mathrm{Beh}_s(x_s, y_s) \stackrel{\mathrm{def}}{=} \bigwedge_{C\in\mathcal{C}_\Sigma^{\mathrm{Obs}}(s)} \forall \mathrm{Var}(C).\, C[x_s] = C[y_s]$.
   Note that if s is an observable sort then $\mathrm{Beh}_s(x_s, y_s)$ is equivalent to $x_s = y_s$, since then the trivial context z_s belongs to $\mathcal{C}_\Sigma^{\mathrm{Obs}}(s)$.
2. The partial observational equality $\approx_{\mathrm{Obs},\mathrm{In}}$ induced by a set Obs of observable sorts and a set In of input sorts (cf. Section 3.2) is axiomatized by the following infinitary formulas:
   $\mathrm{Beh}_s(x_s, y_s) \stackrel{\mathrm{def}}{=} \mathrm{Def}_s(x_s) \wedge \mathrm{Def}_s(y_s) \wedge$
   $\bigwedge_{C\in\mathcal{C}_\Sigma^{\mathrm{Obs}}(s)} \forall \mathrm{Var}(C).\, \mathrm{DEF}(\mathrm{Var}(C)) \Rightarrow C[x_s] = C[y_s]$,
   where $\mathrm{Def}_s(x_s)$ is an abbreviation for $\bigvee_{t\in T_\Sigma(X_{\mathrm{In}})_s} \exists \mathrm{Var}(t).\, x_s = t$ and $\mathrm{DEF}(\mathrm{Var}(C))$ stands for $\bigwedge_{v_{s'}\in\mathrm{Var}(C)} \mathrm{Def}_{s'}(v_{s'})$.
   Note that if s is an input sort, then $\mathrm{Beh}_s(x_s, y_s)$ is equivalent to:
   $\bigwedge_{C\in\mathcal{C}_\Sigma^{\mathrm{Obs}}(s)} \forall \mathrm{Var}(C).\, \mathrm{DEF}(\mathrm{Var}(C)) \Rightarrow C[x_s] = C[y_s]$.
   If s is an observable sort then $\mathrm{Beh}_s(x_s, y_s)$ is equivalent to:
   $\mathrm{Def}_s(x_s) \wedge \mathrm{Def}_s(y_s) \wedge x_s = y_s$.
   Moreover, $\mathrm{Beh}_s(x_s, x_s)$ is always equivalent to $\mathrm{Def}_s(x_s)$.

Theorem 5.5 *Let Σ be a signature and* Beh *be an axiomatization of a Σ-behavioural equality $\approx$, and let* AXL[Beh] *be the corresponding $\mathcal{L}(\Sigma)$-sentence (cf. Definition 5.1). For any Σ-algebra A, class $\boldsymbol{C}$ of Σ-algebras, ASL-like structured specification* SP *with signature Σ, and Σ-formula ϕ, we have:*

(i) $A \models_\approx \phi$ *if and only if* $\mathcal{AL}[\mathrm{Beh}](A) \models \mathcal{L}(\phi)$.
(ii) $\boldsymbol{C} \models_\approx \phi$ *if and only if* $(\boldsymbol{C} + \langle \mathcal{L}(\Sigma), \mathrm{AXL}[\mathrm{Beh}]\rangle) \models \mathcal{L}(\phi)$.

(iii) SP $\models_\approx \phi$ *if and only if* $(\mathrm{SP} + \langle \mathcal{L}(\Sigma), \mathrm{AXL}[\mathrm{Beh}]\rangle) \models \mathcal{L}(\phi)$.

Proof. *(i)* follows directly from Theorem 4.2 and Definition 5.2; *(ii)* follows from *(i)* and the fact that $\mathcal{L}(\boldsymbol{C}) = \{\mathcal{AL}[\mathrm{Beh}](A) \mid A \in \boldsymbol{C}\} = \boldsymbol{C} + \langle \mathcal{L}(\Sigma), \mathrm{AXL}[\mathrm{Beh}]\rangle$ (cf. Section 2), *(iii)* is a special case of *(ii)*. □

Theorem 5.5 allows us to characterize behavioural theories in terms of standard theories using an axiomatization Beh of the given behavioural equality. However, in general these axiomatizations are infinitary ones. To perform proofs is therefore still difficult. One possibility is to replace these infinitary sentences by infinitary proof rules (such as *context induction* in the particular case of the total observational equality) and to show that the resulting proof system is sound and complete (cf. e.g. [14]). Another possibility, studied in the next sections, is to find special (but general enough) cases where the infinitary axiomatization can be replaced by a finitary one.

6 Finitary axiomatization of the behavioural equality

In practice it is in general simple to find an infinitary axiomatization of a behavioural equality (indeed in the observational framework this axiomatization is directly deduced from the definition of the observational equality, cf. Example 5.4). Unfortunately, for arbitrary behavioural equalities it is usually difficult to find a finitary axiomatization Beh which is valid for any algebra A (i.e. such that, for any algebra A, $\mathcal{AL}[\mathrm{Beh}](A) = \mathcal{L}(A)$). But since we are always interested in the behavioural theory of some given class $\boldsymbol{C}$ of algebras we do not really need a finitary axiomatization of the behavioural equality for any arbitrary algebra but rather just for the algebras in the class $\boldsymbol{C}$ we are interested in.

In the most simple cases, it turns out that, provided the class $\boldsymbol{C}$ we are interested in satisfies some simple property expressed by a finitary sentence, we can find a finitary axiomatization of the behavioural equality for the algebras belonging to $\boldsymbol{C}$ (cf. Section 7, Corollary 7.6). In general, however, it may be necessary to introduce additional hidden sorts and function symbols that will prove useful in getting rid of the infinitary axiomatization. This idea leads to the notion of finitary axiomatization with hidden part as formalized below.

Definition 6.1 (Finitary axiomatization with hidden part) *Let Σ be a signature, $\approx$ be a Σ-behavioural equality and $\boldsymbol{C}$ be a class of Σ-algebras.*

(a) *Let* HID *be a specification with* **finitary axioms** *plus* **reachability constraints** *of the form $\langle \Sigma_H, \mathcal{R}_H, \mathcal{A}x_H\rangle$, with $\Sigma \subseteq \Sigma_H$, which defines hidden sorts (possibly constrained) and hidden function symbols.*

(b) *Let* $\mathrm{HBeh} = (\mathrm{HBeh}_s(x_s, y_s))_{s\in S}$ *be an* S*-sorted family of* **finitary** Σ_H**-formulas**, *where* x_s *and* y_s *are the only free variables (both of sort* s*) of* $\mathrm{HBeh}_s(x_s, y_s)$.

$(\mathrm{HID}, \mathrm{HBeh})$ *is called a finitary axiomatization with hidden part of the behavioural equality* $\approx$ *with respect to the class* $\boldsymbol{C}$ *of* Σ*-algebras, if the following two conditions are satisfied:*

(i) HID *is a conservative extension of* $\boldsymbol{C}$, *i.e. all* Σ*-algebras* $A \in \boldsymbol{C}$ *can be extended to (at least) an algebra* $A_H \in \mathrm{Mod}(\mathrm{HID})$, *with* $A_H|_\Sigma = A$ *(hence* $(\boldsymbol{C} + \mathrm{HID})|_\Sigma = \boldsymbol{C}$*).*

(ii) *For any* Σ_H*-algebra* $A_H \in (\boldsymbol{C}+\mathrm{HID})$, $(\mathcal{AL}[\mathrm{HBeh}](A_H))|_{\mathcal{L}(\Sigma)} = \mathcal{L}(A_H|_\Sigma)$, *where* $\mathcal{AL}[\mathrm{HBeh}](A_H)$ *denotes the axiomatic lifting of* A_H *induced by* HBeh *(cf. Definition 5.1).*

HID *is called the hidden part of the finitary axiomatization.*

Some comments may help to provide a better understanding of the definition above:

1. The main goal is to replace the family Beh of infinitary formulas by a family HBeh of finitary formulas. Then obviously the induced sentence AXL[HBeh] (cf. Definition 5.1) will be a finitary sentence as well (since we assume that S is finite).
2. It is a standard idea to use auxiliary hidden sorts and function symbols to replace formulas expressed in some logic by "equivalent" formulas expressed in a less powerful one (cf. e.g. [27]). It is therefore natural to introduce the hidden part HID in our axiomatization.
3. The finitary axiomatization is "adequate" for the class $\boldsymbol{C}$ if HID is a conservative extension of $\boldsymbol{C}$ (cf. condition *(i)*) and if, moreover, for all algebras in $\boldsymbol{C}$ extended accordingly to HID, the axiomatic lifting induced by the finitary axiomatization HBeh "coincides" with the semantic lifting associated to the behavioural equality (cf. condition *(ii)*).
4. In the most simple cases no hidden part is necessary, i.e. we can choose HID equal to $\langle \Sigma, \emptyset, \emptyset\rangle$. Then $(\boldsymbol{C} + \mathrm{HID}) = \boldsymbol{C}$ and HBeh is simply a family of finitary Σ-formulas. Then condition *(i)* is trivially satisfied and $(\langle \Sigma, \emptyset, \emptyset\rangle, \mathrm{HBeh})$ is a finitary axiomatization of the given behavioural equality with respect to a class $\boldsymbol{C}$ of Σ-algebras if and only if, for any Σ-algebra $A \in \boldsymbol{C}$, $\mathcal{AL}[\mathrm{HBeh}](A) = \mathcal{L}(A)$.

Remark 6.2 Let $\Sigma = (S, F)$ be a signature and let $(\mathrm{HID}, \mathrm{HBeh})$ be a finitary axiomatization with hidden part of a Σ-behavioural equality $\approx$ with respect to a class $\boldsymbol{C}$ of Σ-algebras. Let $A \in \boldsymbol{C}$ be a Σ-algebra. Then, according to the condition *(i)*, there exists at least one Σ_H-algebra $A_H \in (\boldsymbol{C} + \mathrm{HID})$ such that $A_H|_\Sigma = A$ (in particular, for any sort $s \in S$, $A_{H,s} = A_s$). Then the condition *(ii)* implies that $\mathcal{AL}[\mathrm{HBeh}](A_H)|_{\mathcal{L}(\Sigma)} = \mathcal{L}(A)$. Hence $\{\mathcal{L}(A)\} =$

$(\{A\}+\mathrm{HID}+\langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}]\rangle)|_{\mathcal{L}(\Sigma)}$. Using a similar reasoning as in the proof of Proposition 5.3, we conclude that, for any sort $s \in S$ and any valuation $\alpha : \{x_s, y_s\} \to A_H$, $A_H, \alpha \models \mathrm{HBeh}_s(x_s, y_s)$ if and only if $\alpha(x_s) \approx_A \alpha(y_s)$.

It is obvious that the results obtained in the previous section (cf. Theorem 5.5) carry over to finitary axiomatizations with hidden part:

Theorem 6.3 *Let Σ be a signature and* $(\mathrm{HID}, \mathrm{HBeh})$ *be a finitary axiomatization with hidden part of a Σ-behavioural equality $\approx$ with respect to a class $\boldsymbol{C}$ of Σ-algebras. Let Σ_H be the signature of* HID *and let* AXL[HBeh] *be the corresponding $\Sigma_H^{\mathcal{L}}$-sentence (cf. Definition 5.1).*

(i) $\boldsymbol{C} \models_{\approx} \phi$ *if and only if* $(\boldsymbol{C} + \mathrm{HID} + \langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}]\rangle) \models \mathcal{L}(\phi)$.

(ii) In particular, if $\boldsymbol{C}$ is the model class of some ASL-like structured specification SP*, we have:*
SP $\models_{\approx} \phi$ *if and only if* $(\mathrm{SP} + \mathrm{HID} + \langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}]\rangle) \models \mathcal{L}(\phi)$.
It is important to note that, if SP *is a smooth specification, then the specification* $\mathrm{SP} + \mathrm{HID} + \langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}]\rangle$ *is smooth as well (cf. Section 2).*

Proof. Similar to the proof of Theorem 5.5, using the fact that, for any Σ-algebra $A \in \boldsymbol{C}$, $\{\mathcal{L}(A)\} = (\{A\} + \mathrm{HID} + \langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}]\rangle)|_{\mathcal{L}(\Sigma)}$ (cf. Remark 6.2), hence $\mathcal{L}(\boldsymbol{C}) = (\boldsymbol{C} + \mathrm{HID} + \langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}]\rangle)|_{\mathcal{L}(\Sigma)}$. □

Remark 6.4 If the hidden part is empty, i.e. $\mathrm{HID} = \langle \Sigma, \emptyset, \emptyset\rangle$, and if $\boldsymbol{C}$ is the model class of some ASL-like structured specification SP, we have:
SP $\models_{\approx} \phi$ if and only if $(\mathrm{SP} + \langle \mathcal{L}(\Sigma), \mathrm{AXL}[\mathrm{HBeh}]\rangle) \models \mathcal{L}(\phi)$.

As shown by the above theorem, once a finitary axiomatization (with hidden part) of the behavioural equality is provided, we can use "standard" proof techniques to prove the behavioural validity of some formula with respect to a given smooth specification. The aim of the next section is to explain how one can find adequate finitary axiomatizations with hidden part in the particular case of observational equalities.

7 Axiomatization of the observational equality

We will now focus on observational equalities, and throughout this section we assume given a signature $\Sigma = (S, F)$, a set $\mathrm{Obs} \subseteq S$ of observable sorts and a set $\mathrm{In} \subseteq S$ of input sorts (and we assume that Σ is sensible w.r.t. In). X_{In} is the S-sorted family of variables defined by $(X_{\mathrm{In}})_s = \emptyset$ if $s \notin \mathrm{In}$ and $(X_{\mathrm{In}})_s = X_s$ if $s \in \mathrm{In}$ (where $X = (X_s)_{s \in S}$ is the generally assumed family of countably infinite sets of variables of sort s). The problem to be solved is to find a finitary axiomatization of the observational equality $\approx_{\mathrm{Obs,In}}$ (w.r.t. a given class $\boldsymbol{C}$ of Σ-algebras). Remember that for any Σ-algebra A, the definition domain $\mathrm{Dom}(\approx_{\mathrm{Obs,In},A})$ of $\approx_{\mathrm{Obs,In},A}$ is equal to $A[X_{\mathrm{In}}]$, the smallest subalgebra of A generated by Σ and X_{In}.

Let us consider again the infinitary axiomatization of $\approx_{\mathrm{Obs,In}}$ given in Example 5.4:

$$\mathrm{Beh}_s(x_s, y_s) \stackrel{\mathrm{def}}{=} \mathrm{Def}_s(x_s) \wedge \mathrm{Def}_s(y_s) \wedge \bigwedge_{C \in \mathcal{C}_\Sigma^{\mathrm{Obs}}(s)} \forall \mathrm{Var}(C).\, \mathrm{DEF}(\mathrm{Var}(C)) \Rightarrow C[x_s] = C[y_s]\,,$$

where $\mathrm{Def}_s(x_s)$ is an abbreviation for $\bigvee_{t \in T_\Sigma(X_{\mathrm{In}})_s} \exists \mathrm{Var}(t).\, x_s = t\,,$
and $\mathrm{DEF}(\mathrm{Var}(C))$ stands for $\bigwedge_{v_{s'} \in \mathrm{Var}(C)} \mathrm{Def}_{s'}(v_{s'})\,.$

It is clear that we have two distinct reasons for obtaining infinitary formulas: on one hand we must consider an infinite set of observable contexts $\mathcal{C}_\Sigma^{\mathrm{Obs}}$ which represents the infinitely many experiments with observable results, and this leads to the infinitary conjunction $\bigwedge_{C \in \mathcal{C}_\Sigma^{\mathrm{Obs}}(s)} \cdots$; on the other hand we must only consider values that are denotable by a term $t \in T_\Sigma(X_{\mathrm{In}})$, and this leads to the infinitary disjunctions $\bigvee_{t \in T_\Sigma(X_{\mathrm{In}})_s} \cdots$. A natural idea to get rid of the first cause of infinitary axiomatization 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. Indeed, if successful, this idea would be enough to solve our problem in the special case of the total observational equality.

In general, however, it is clear that this idea will not be powerful enough. First, if we consider a partial observational equality, we must still face the infinitary $\mathrm{Def}_s(.)$ parts. Moreover, it is not always possible to consider only a *finite* subset of observable contexts. In these cases we must find an appropriate hidden part to obtain a finitary axiomatization of the observational equality $\approx_{\mathrm{Obs,In}}$.

7.1 Using a smaller set of observable contexts

Before we consider more concretely how to construct a hidden part in order to obtain a finitary axiomatization of the given observational equality, we will first provide a general characterization of finitary axiomatizations of the observational equality. This characterization will point out that under some conditions it is enough to axiomatize, instead of the given observational equality, a contextual equality (cf. Definition 3.6) with respect to some (smaller) subset $\mathcal{C}$ of the set $\mathcal{C}_\Sigma^{\mathrm{Obs}}$ of all observable contexts.

First let us point out that all definitions and results of Sections 4, 5 and 6 can be generalized in a straightforward way to contextual equalities.[8] This is in particular the case for Definition 5.2 and Example 5.4. Hence, the partial contextual equality $\approx_{\mathcal{C},\mathrm{In}}$ induced by an arbitrary set $\mathcal{C}$ of Σ-contexts and by a set In of input sorts is axiomatized by the S-sorted family of formulas $\mathrm{Beh}^{\mathcal{C}} = (\mathrm{Beh}^{\mathcal{C}}_s(x_s, y_s))_{s\in S}$, with:

$$\mathrm{Beh}^{\mathcal{C}}_s(x_s, y_s) \stackrel{\mathrm{def}}{=} \mathrm{Def}_s(x_s) \wedge \mathrm{Def}_s(y_s) \wedge \bigwedge_{C\in\mathcal{C}(s)} \forall \mathrm{Var}(C).\, \mathrm{DEF}(\mathrm{Var}(C)) \Rightarrow C[x_s] = C[y_s]\,,$$

where $\mathrm{Def}_s(x_s)$ and $\mathrm{DEF}(\mathrm{Var}(C))$ are the same abbreviations as above.

In a first step we will study some sufficient conditions under which the contextual equality induced by a set $\mathcal{C}$ of Σ-contexts and a set In of input sorts coincides with the observational equality $\approx_{\mathrm{Obs},\mathrm{In}}$.

Lemma 7.1 *Let A be a Σ-algebra and $\approx_A$ be an arbitrary partial congruence on A. If $\mathrm{Dom}(\approx_A) = \mathrm{Dom}(\approx_{\mathrm{Obs},\mathrm{In},A})(= A[X_{\mathrm{In}}])$ and if $\approx_A$ "coincides" with the set-theoretic equality on $\mathrm{Dom}(\approx_A)_s$ for all observable sorts s (i.e. if for all $a, b \in \mathrm{Dom}(\approx_A)_s$, with $s \in \mathrm{Obs}$, we have $a \approx_A b$ if and only if $a = b$) then $\approx_A \subseteq \approx_{\mathrm{Obs},\mathrm{In},A}$.*

Proof. Let a, b be two elements of A_s, for some sort $s \in S$, and assume that $a \approx_A b$. Then both a and b belong to $A[X_{\mathrm{In}}]$. Since $\approx_A$ is a partial congruence (hence is compatible with the signature Σ), we have, for any observable context $C \in \mathcal{C}^{\mathrm{Obs}}_{\Sigma}(s)$ and for any valuation $\alpha : X \to A[X_{\mathrm{In}}]$, $I_{\alpha_a}(C) \approx_A I_{\alpha_b}(C)$, where $\alpha_a, \alpha_b : X \cup \{z_s\} \to A[X_{\mathrm{In}}]$ are the unique extensions of α defined by $\alpha_a(z_s) = a$ and $\alpha_b(z_s) = b$, where z_s is the context variable of C. Since C is an observable context, both $I_{\alpha_a}(C)$ and $I_{\alpha_b}(C)$ belong to $A[X_{\mathrm{In}}]_{s'}$, for some observable sort s'. Then, by assumption, $I_{\alpha_a}(C) \approx_A I_{\alpha_b}(C)$ implies $I_{\alpha_a}(C) = I_{\alpha_b}(C)$, hence we have $a \approx_{\mathrm{Obs},\mathrm{In},A} b$. Therefore $\approx_A \subseteq \approx_{\mathrm{Obs},\mathrm{In},A}$. □

Remark 7.2 Indeed it is easy to prove that the set of all partial congruences on A which have $A[X_{\mathrm{In}}]$ as definition domain and which "coincide" with the set-theoretic equality for each observable sort is a complete lattice, the smallest element of which is the restriction of the set-theoretic equality to $A[X_{\mathrm{In}}]$, the greatest element being the observational equality $\approx_{\mathrm{Obs},\mathrm{In},A}$.

Lemma 7.3 *Let $\mathcal{C} \subseteq \mathcal{C}^{\mathrm{Obs}}_{\Sigma}$ be an arbitrary subset of observable Σ-contexts such that, for any observable sort $s \in \mathrm{Obs}$, $z_s \in \mathcal{C}$ and let A be a Σ-algebra. Then $\approx_{\mathcal{C},\mathrm{In},A} = \approx_{\mathrm{Obs},\mathrm{In},A}$ if and only if $\approx_{\mathcal{C},\mathrm{In},A}$ is a partial Σ-congruence.*

Proof. First note that, according to Definition 3.6, $\mathrm{Dom}(\approx_{\mathcal{C},\mathrm{In},A}) = A[X_{\mathrm{In}}] = \mathrm{Dom}(\approx_{\mathrm{Obs},\mathrm{In},A})$. Assume that $\approx_{\mathcal{C},\mathrm{In},A}$ is a partial Σ-congruence. Since $\mathcal{C} \subseteq \mathcal{C}^{\mathrm{Obs}}_{\Sigma}$, obviously we have $\approx_{\mathrm{Obs},\mathrm{In},A} \subseteq \approx_{\mathcal{C},\mathrm{In},A}$. To prove that $\approx_{\mathcal{C},\mathrm{In},A} \subseteq \approx_{\mathrm{Obs},\mathrm{In},A}$, by

[8] Indeed, the only result for which we must assume that the behavioural equality considered is a family of partial congruence relations, and not only a family of arbitrary partial equivalence relations, is Theorem 3.4.

Lemma 7.1, it is enough to prove that $\approx_{\mathcal{C},\mathrm{In},A}$ "coincides" with the set-theoretic equality on $A[X_{\mathrm{In}}]_s$ for each observable sort s. 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. $\square$

Using the last lemma we can infer necessary and sufficient conditions under which the family of formulas $\mathrm{Beh}^{\mathcal{C}}$ is "equivalent" to the axiomatization of the observational equality $\approx_{\mathrm{Obs},\mathrm{In}}$.

Proposition 7.4 (Criteria for using a smaller set of contexts) *Let $\mathcal{C} \subseteq \mathcal{C}_{\Sigma}^{\mathrm{Obs}}$ be an arbitrary subset of observable Σ-contexts such that, for any observable sort $s \in \mathrm{Obs}$, $z_s \in \mathcal{C}$. Let* Beh *and* $\mathrm{Beh}^{\mathcal{C}}$ *be the respective axiomatizations of $\approx_{\mathrm{Obs},\mathrm{In}}$ and $\approx_{\mathcal{C},\mathrm{In}}$. For any Σ_H-algebra A_H, with $\Sigma \subseteq \Sigma_H$, the following two conditions are equivalent:*

(i) $\mathcal{AL}[\mathrm{Beh}^{\mathcal{C}}](A_H) = \mathcal{AL}[\mathrm{Beh}](A_H)$
(ii) $\mathcal{AL}[\mathrm{Beh}^{\mathcal{C}}](A_H) \models \mathrm{CONG}_{\Sigma}^{\sim}$

where $\mathrm{CONG}_{\Sigma}^{\sim}$ is the finitary $\mathcal{L}(\Sigma)$-sentence defined by:
$\mathrm{CONG}_{\Sigma}^{\sim} \stackrel{\text{def}}{=} \bigwedge_{f \in F} \mathrm{CONG}_{f}^{\sim}$,
where for each function symbol $f \in F$ of arity $s_1 \ldots s_n \to s$, the sentence $\mathrm{CONG}_{f}^{\sim}$ is defined by:

$$\mathrm{CONG}_{f}^{\sim} \stackrel{\text{def}}{=} \forall x_1, y_1{:}s_1, \ldots , x_n, y_n{:}s_n.$$
$$\Big[\Big(\textstyle\bigwedge_{1 \le i \le n} x_i \sim_{s_i} y_i\Big) \Rightarrow f(x_1, \ldots , x_n) \sim_s f(y_1, \ldots , y_n)\Big].$$

Proof.
$\Longrightarrow$: Is obvious since Beh is the axiomatization of the partial Σ-congruence $\approx_{\mathrm{Obs},\mathrm{In}}$, hence $\mathcal{AL}[\mathrm{Beh}](A_H) \models \mathrm{CONG}_{\Sigma}^{\sim}$.
$\Longleftarrow$: Assume that $\mathcal{AL}[\mathrm{Beh}^{\mathcal{C}}](A_H) \models \mathrm{CONG}_{\Sigma}^{\sim}$. Then, since $\mathrm{Beh}^{\mathcal{C}}$ is an axiomatization of the contextual equality $\approx_{\mathcal{C},\mathrm{In}}$, we know that $\approx_{\mathcal{C},\mathrm{In},A_H|_{\Sigma}}$ is a partial Σ-congruence, and using Lemma 7.3 we conclude that $\approx_{\mathcal{C},\mathrm{In},A_H|_{\Sigma}} = \approx_{\mathrm{Obs},\mathrm{In},A_H|_{\Sigma}}$. Hence $\mathcal{AL}[\mathrm{Beh}^{\mathcal{C}}](A_H|_{\Sigma}) = \mathcal{AL}[\mathrm{Beh}](A_H|_{\Sigma})$, and therefore $\mathcal{AL}[\mathrm{Beh}^{\mathcal{C}}](A_H) = \mathcal{AL}[\mathrm{Beh}](A_H)$. $\square$

Remark 7.5 Let us stress that in the following cases some important simplifications apply to the general $\mathrm{CONG}_{\Sigma}^{\sim}$ formula given in Proposition 7.4:

1. If $\mathcal{C}$ is generated by a subsignature of Σ, i.e. if $\mathcal{C} = \mathcal{C}_{\Sigma'}^{\mathrm{Obs}}$ for some subsignature Σ' of Σ, with $\Sigma' = (S, F')$, then we know that $\mathrm{Beh}^{\mathcal{C}}$ is the axiomatization of a partial Σ'-congruence, and we can therefore omit in the conjunction all $\mathrm{CONG}_{f}^{\sim}$, for $f \in F'$.
2. If all observable sorts are input sorts (i.e. $\mathrm{Obs} \subseteq \mathrm{In}$), and in particular if we consider a total observational equality (since then $\mathrm{Obs} \subseteq \mathrm{In} = S$), we know that $\approx_{\mathcal{C},\mathrm{In}}$ coincides with the set-theoretic equality on the carrier

sets of all observable sorts (since $\mathcal{C}$ contains the "trivial" contexts z_s when s is an observable sort). Hence in such cases $\sim_s$ is interpreted by the set-theoretic equality for all observable sorts s. We can then apply the following simplifications:
(a) If a function symbol f has only observable sorts as arguments, then one can omit $\mathrm{CONG}_f^{\sim}$ in the conjunction.
(b) In the premisses, one can omit $x_i \sim_{s_i} y_i$ for all observable sorts s_i, and then replace in the conclusion y_i by x_i.
(c) In the conclusion, one can replace $\sim_s$ by $=$ if s is an observable sort.

From Proposition 7.4 we can immediately deduce the following corollary which shows that any finite set $\mathcal{C} \subset \mathcal{C}_\Sigma^{\mathrm{Obs}}$ of observable Σ-contexts (which contains, for any observable sort $s \in \mathrm{Obs}$, the trivial context z_s) induces a finitary axiomatization of the total observational equality $\approx_{\mathrm{Obs},S}$ w.r.t. any class $\boldsymbol{C}$ for which the contextual equality $\approx_{\mathcal{C},S}$ is a Σ-congruence.

Corollary 7.6 *Let $\boldsymbol{C}$ be a class of Σ-algebras and let $\mathcal{C} \subset \mathcal{C}_\Sigma^{\mathrm{Obs}}$ be a* **finite** *set of observable Σ-contexts such that, for any observable sort $s \in \mathrm{Obs}$, $z_s \in \mathcal{C}$. Let* TBeh *be the S-sorted family of finitary Σ-formulas defined by:*
$\mathrm{TBeh}_s(x_s, y_s) \stackrel{\mathrm{def}}{=} \bigwedge_{C\in\mathcal{C}(s)} \forall \mathrm{Var}(C).\, C[x_s] = C[y_s]\,.$
TBeh *is a finitary axiomatization (with an empty hidden part) of the total observational equality $\approx_{\mathrm{Obs},S}$ with respect to the class $\boldsymbol{C}$ of Σ-algebras if and only if $(\boldsymbol{C} + \langle \mathcal{L}(\Sigma), \mathrm{AXL}[\mathrm{TBeh}]\rangle) \models \mathrm{CONG}_\Sigma^{\sim}$.*

Proof. The family TBeh is obviously a finitary axiomatization of the total contextual equality $\approx_{\mathcal{C},S}$. Moreover, $\{\mathcal{AL}[\mathrm{TBeh}](A) \mid A \in \boldsymbol{C}\} = \boldsymbol{C} + \langle \mathcal{L}(\Sigma), \mathrm{AXL}[\mathrm{TBeh}]\rangle$. TBeh is a finitary axiomatization of the total observational equality $\approx_{\mathrm{Obs},S}$ with respect to the class $\boldsymbol{C}$ of Σ-algebras if and only if, for any algebra $A \in \boldsymbol{C}$, $\mathcal{AL}[\mathrm{TBeh}](A) = \mathcal{L}(A) = \mathcal{AL}[\mathrm{Beh}](A)$, where Beh is the infinitary axiomatization of $\approx_{\mathrm{Obs},S}$. Hence the corollary is a direct consequence of Proposition 7.4. □

The above corollary solves our problem in the simplest cases, i.e. when we consider a total observational equality and when, moreover, we can find a finite subset $\mathcal{C}$ of observable Σ-contexts such that $(\boldsymbol{C}+\langle \mathcal{L}(\Sigma), \mathrm{AXL}[\mathrm{TBeh}]\rangle) \models \mathrm{CONG}_\Sigma^{\sim}$: in this case we can immediately apply Theorem 6.3, with an empty hidden part and with AXL[TBeh].

More generally, from Proposition 7.4 we can infer the following characterization of finitary axiomatizations of the observational equality $\approx_{\mathrm{Obs,In}}$. Intuitively this characterization says that for axiomatizing the observational equality $\approx_{\mathrm{Obs,In}}$, it is enough to axiomatize some contextual equality $\approx_{\mathcal{C},\mathrm{In}}$, provided that $\approx_{\mathcal{C},\mathrm{In},A}$ is a partial Σ-congruence for all algebras A in the class $\boldsymbol{C}$ of interest.

Theorem 7.7 (Characterization of finitary axiomatizations of $\approx_{\text{Obs,In}}$) *Let* $\boldsymbol{C}$ *be a class of* Σ*-algebras, and let* $(\text{HID}, \text{HBeh})$ *be as in Definition 6.1.* $(\text{HID}, \text{HBeh})$ *is a finitary axiomatization with hidden part of the observational equality* $\approx_{\text{Obs,In}}$ *with respect to the class* $\boldsymbol{C}$ *of* Σ*-algebras if and only if the following conditions are satisfied:*

(i) $(\boldsymbol{C} + \text{HID})|_{\Sigma} = \boldsymbol{C}$.
(ii) *For any* Σ_H*-algebra* $A_H \in (\boldsymbol{C} + \text{HID})$, $\mathcal{AL}[\text{HBeh}](A_H) \models \text{CONG}_{\Sigma}^{\approx}$, *i.e.* $(\boldsymbol{C} + \text{HID} + \langle \Sigma_H^{\mathcal{L}}, \text{AXL}[\text{HBeh}] \rangle) \models \text{CONG}_{\Sigma}^{\approx}$.
(iii) *There exists a (possibly infinite) set* $\mathcal{C} \subseteq \mathcal{C}_{\Sigma}^{\text{Obs}}$ *of observable* Σ*-contexts which contains, for any observable sort* $s \in \text{Obs}$, *the trivial context* z_s, *and such that, for any* Σ_H*-algebra* $A_H \in (\boldsymbol{C} + \text{HID})$, $\mathcal{AL}[\text{HBeh}](A_H) = \mathcal{AL}[\text{Beh}^{\mathcal{C}}](A_H)$, *where* $\text{Beh}^{\mathcal{C}}$ *is the axiomatization of the contextual equality* $\approx_{\mathcal{C},\text{In}}$.

Proof. It is obviously enough to show that the condition *(ii)* of Definition 6.1 is equivalent to the conditions *(ii)* and *(iii)* above. Let Beh be the infinitary axiomatization of the observational equality $\approx_{\text{Obs,In}}$ and let $A_H \in (\boldsymbol{C} + \text{HID})$.

$\Longrightarrow$**:** Assume that $\mathcal{AL}[\text{HBeh}](A_H)|_{\mathcal{L}(\Sigma)} = \mathcal{L}(A_H|_{\Sigma})$. But then $\mathcal{AL}[\text{HBeh}](A_H) = \mathcal{AL}[\text{Beh}](A_H)$. Hence the condition *(ii)* is obviously satisfied. On the other hand the condition *(iii)* is trivially satisfied as well by choosing $\mathcal{C} = \mathcal{C}_{\Sigma}^{\text{Obs}}$.
$\Longleftarrow$**:** Assume that the conditions *(ii)* and *(iii)* hold. Then $\mathcal{AL}[\text{HBeh}](A_H) \models \text{CONG}_{\Sigma}^{\approx}$ and $\mathcal{AL}[\text{HBeh}](A_H) = \mathcal{AL}[\text{Beh}^{\mathcal{C}}](A_H)$, for some $\mathcal{C} \subseteq \mathcal{C}_{\Sigma}^{\text{Obs}}$ with $z_s \in \mathcal{C}$ for all $s \in \text{Obs}$. Hence $\mathcal{AL}[\text{Beh}^{\mathcal{C}}](A_H) \models \text{CONG}_{\Sigma}^{\approx}$ and therefore, by Proposition 7.4, $\mathcal{AL}[\text{Beh}^{\mathcal{C}}](A_H) = \mathcal{AL}[\text{Beh}](A_H)$. Thus $\mathcal{AL}[\text{HBeh}](A_H) = \mathcal{AL}[\text{Beh}](A_H)$, and therefore $\mathcal{AL}[\text{HBeh}](A_H)\,|_{\mathcal{L}(\Sigma)} = \mathcal{AL}[\text{Beh}](A_H)\,|_{\mathcal{L}(\Sigma)} = \mathcal{AL}[\text{Beh}](A_H|_{\Sigma}) = \mathcal{L}(A_H|_{\Sigma})$, i.e. the condition *(ii)* of Definition 6.1 holds. □

Hence, unless we can apply Corollary 7.6, we still have to find an adequate hidden part to obtain a suitable finitary axiomatization of the observational equality $\approx_{\text{Obs,In}}$. However, Theorem 7.7 suggests that our task may be simplified by considering some appropriate subset $\mathcal{C}$ of observable contexts.

7.2 Hidden part for the observational equality

The aim of this subsection is to explain how to find a finitary axiomatization with hidden part $(\text{HID}, \text{HBeh})$ of the partial observational equality $\approx_{\text{Obs,In}}$ w.r.t. some class $\boldsymbol{C}$ of Σ-algebras. In a first step we must get rid of the infinitary $\text{Def}_s(.)$ parts. For this we suggest to use auxiliary hidden predicate symbols $D_{H,s}$ axiomatized in such a way that $D_{H,s}(x_s)$ holds if and only if $\text{Def}_s(x_s)$ holds. In a second step we must get rid of the infinitary conjunctions over observable contexts. The previous subsection suggests to replace the set of all observable contexts by some smaller subset $\mathcal{C}$. However, if we cannot find a convenient finite subset $\mathcal{C}$, then we still have infinitary conjunctions over the contexts of $\mathcal{C}$. In that case we suggest to use auxiliary hidden function symbols that will intuitively provide the same observations as $\mathcal{C}$ does, and to use a finite set $\mathcal{C}_H$ of observable contexts built with the help of these hidden function symbols.

Let us first explain how a family of finitary formulas is induced by the predicate symbols $D_{H,s}$ and by the choice of a finite set of contexts $\mathcal{C}_H$.

Definition 7.8 *Let Σ_H be a signature such that $\Sigma \subseteq \Sigma_H$, and such that for each sort $s \in S$, there exists a unary predicate symbol $D_{H,s}$ of domain s in Σ_H. Let $\mathcal{C}_H$ be an arbitrary finite set of Σ_H-contexts (with variables in X_H, where $X_H = (X_{H,s})_{s \in S_H}$ is the generally assumed family of countably infinite sets of variables of sort $s \in S_H$). The S-sorted family of finitary Σ_H-formulas induced by the predicate symbols $D_{H,s}$ and the contexts $\mathcal{C}_H$, denoted by* $\mathrm{HBeh}[D_H, \mathcal{C}_H]$, *is defined by:*

$\mathrm{HBeh}[D_H, \mathcal{C}_H]_s(x_s, y_s) \stackrel{\mathrm{def}}{=} D_{H,s}(x_s) \wedge D_{H,s}(y_s) \wedge$
$\qquad \bigwedge_{C_H \in \mathcal{C}_H(s)} \forall \mathrm{Var}(C_H).\, \mathrm{DEF}_H(\mathrm{Var}(C_H)) \Rightarrow C_H[x_s] = C_H[y_s]\,,$

where $\mathrm{DEF}_H(\mathrm{Var}(C_H))$ *stands for* $\bigwedge_{v_{s'} \in \mathrm{Var}(C_H), s' \in S} D_{H,s'}(v_{s'})$.

Remark 7.9 Note that $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ is not necessarily the axiomatization of some contextual equality.

Theorem 7.7 provides an abstract characterization of finitary axiomatizations of the observational equality $\approx_{\mathrm{Obs,In}}$, which is valid for any couple (HID, HBeh). However, if we know that we are going to use the family $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ induced by the predicate symbols $D_{H,s}$ and by a finite set of contexts $\mathcal{C}_H$, we can derive more precise conditions. The idea is to find criteria for $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ to be indeed the axiomatization of some contextual equality (cf. Proposition 7.10). Then one can use these criteria to check the condition *(iii)* of Theorem 7.7.

Proposition 7.10 (Criteria for the hidden part) *Let $\mathcal{C}$ be an arbitrary set of Σ-contexts and let $\mathrm{Beh}^{\mathcal{C}}$ be the axiomatization of the contextual equality $\approx_{\mathcal{C},\mathrm{In}}$. Let Σ_H be a signature such that $\Sigma \subseteq \Sigma_H$, and such that for each sort $s \in S$, there exists a unary predicate symbol $D_{H,s}$ of domain s in Σ_H. Let $\mathcal{C}_H$ be an arbitrary finite set of Σ_H-contexts (with variables in X_H). Let* $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ *be the S-sorted family of finitary Σ_H-formulas induced by the predicate symbols $D_{H,s}$ and by the contexts $\mathcal{C}_H$.*
For any Σ_H-algebra A_H, $\mathcal{AL}[\mathrm{HBeh}[D_H, \mathcal{C}_H]](A_H) = \mathcal{AL}[\mathrm{Beh}^{\mathcal{C}}](A_H)$ if the following three conditions are satisfied:

(i) For each sort $s \in S$ and any value $a \in (A_H)_s$, $D^{A_H}_{H,s}(a)$ if and only if there exists a term $t \in T_\Sigma(X_{\mathrm{In}})$ and a valuation $\alpha : X_{\mathrm{In}} \to A_H$ such that $I_\alpha(t) = a$.

(ii) *For each sort* $s \in S$, *for each* Σ_H*-context* $C_H \in \mathcal{C}_H(s)$, *for each valuation* $\alpha : X_H \to A_H$ *such that* $D^{A_H}_{H,s'}(\alpha(v_{s'}))$ *for all* $s' \in S$ *and all* $v_{s'} \in \mathrm{Var}(C_H)$, *there exists a* Σ*-context* $C \in \mathcal{C}(s)$ *and a valuation* $\beta : X \to A_H$ *with* $D^{A_H}_{H,s'}(\beta(v_{s'}))$ *for all* $s' \in S$ *and all* $v_{s'} \in \mathrm{Var}(C)$, *such that, for any value* $a \in (A_H)_s$ *with* $D^{A_H}_{H,s}(a)$, $I_{\beta_a}(C) = I_{\alpha_a}(C_H)$, *where* I_{β_a} *and* I_{α_a} *are the unique extensions of* I_β *and* I_α *respectively defined by* $I_{\beta_a}(z_s) = a$ *and* $I_{\alpha_a}(z_s) = a$.

(iii) *For each sort* $s \in S$, *for each* Σ*-context* $C \in \mathcal{C}(s)$, *for each valuation* $\alpha : X \to A_H$ *such that* $D^{A_H}_{H,s'}(\alpha(v_{s'}))$ *for all* $s' \in S$ *and all* $v_{s'} \in \mathrm{Var}(C)$, *there exists a* Σ_H*-context* $C_H \in \mathcal{C}_H(s)$ *and a valuation* $\beta : X_H \to A_H$ *with* $D^{A_H}_{H,s'}(\beta(v_{s'}))$ *for all* $s' \in S$ *and all* $v_{s'} \in \mathrm{Var}(C_H)$, *such that, for any value* $a \in (A_H)_s$ *with* $D^{A_H}_{H,s}(a)$, $I_{\beta_a}(C_H) = I_{\alpha_a}(C)$, *where* I_{β_a} *and* I_{α_a} *are the unique extensions of* I_β *and* I_α *respectively defined by* $I_{\beta_a}(z_s) = a$ *and* $I_{\alpha_a}(z_s) = a$.

Proof. Let A_H be an arbitrary Σ_H-algebra. From *(i)* we can conclude that for any sort $s \in S$ and any valuation $\alpha : \{x_s\} \to A_H$, $A_H, \alpha \models D_{H,s}(x_s)$ if and only if $A_H, \alpha \models \mathrm{Def}_s(x_s)$. Using this fact together with the conditions *(ii)* and *(iii)* we can infer that, for any sort $s \in S$ and any valuation $\alpha : \{x_s, y_s\} \to A_H$, $A_H, \alpha \models \mathrm{HBeh}[D_H, \mathcal{C}_H]_s(x_s, y_s)$ if and only if $A_H, \alpha \models \mathrm{Beh}^{\mathcal{C}}_s(x_s, y_s)$. Hence $\mathcal{AL}[\mathrm{HBeh}[D_H, \mathcal{C}_H]](A_H) = \mathcal{AL}[\mathrm{Beh}^{\mathcal{C}}](A_H)$. □

The combination of Theorem 7.7 and Proposition 7.10 leads to the following criteria for finite axiomatizations of the observational equality $\approx_{\mathrm{Obs,In}}$.

Theorem 7.11 (Criteria for finitary axiomatizations of $\approx_{\mathrm{Obs,In}}$) *Let* $\boldsymbol{C}$ *be a class of* Σ*-algebras. Let* HID *be a specification with finitary axioms plus reachability constraints of the form* $\langle \Sigma_H, \mathcal{R}_H, \mathcal{A}x_H \rangle$, *with* $\Sigma \subseteq \Sigma_H$, *such that for each sort* $s \in S$, *there exists a unary predicate symbol* $D_{H,s}$ *of domain* s *in* Σ_H. *Let* $\mathcal{C}_H$ *be an arbitrary finite set of* Σ_H*-contexts (with variables in* X_H*). Let* $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ *be the* S*-sorted family of finitary* Σ_H*-formulas induced by the predicate symbols* $D_{H,s}$ *and the contexts* $\mathcal{C}_H$.
$(\mathrm{HID}, \mathrm{HBeh}[D_H, \mathcal{C}_H])$ *is a finitary axiomatization with hidden part of the observational equality* $\approx_{\mathrm{Obs,In}}$ *with respect to the class* $\boldsymbol{C}$ *of* Σ*-algebras if the following conditions are satisfied:*

(i) $(\boldsymbol{C} + \mathrm{HID})|_\Sigma = \boldsymbol{C}$.
(ii) $(\boldsymbol{C} + \mathrm{HID} + \langle \Sigma^{\mathcal{L}}_H, \mathrm{AXL}[\mathrm{HBeh}[D_H, \mathcal{C}_H]] \rangle) \models \mathrm{CONG}^{\approx}_\Sigma$.
(iii) *There exists a (possibly infinite) set* $\mathcal{C} \subseteq \mathcal{C}^{\mathrm{Obs}}_\Sigma$ *of observable* Σ*-contexts which contains, for any observable sort* $s \in \mathrm{Obs}$, *the trivial context* z_s *and such that any* Σ_H*-algebra* $A_H \in (\boldsymbol{C} + \mathrm{HID})$ *satisfies the three conditions (i), (ii) and (iii) of Proposition 7.10 (w.r.t.* $\mathcal{C}$*).*

Proof. Follows directly from Theorem 7.7 and Proposition 7.10. □

Theorem 7.11 provides the key for constructing a finitary axiomatization of the observational equality $\approx_{\mathrm{Obs,In}}$ with respect to a given class $\boldsymbol{C}$ of Σ-algebras. Indeed the conditions *(i)*, *(ii)* and *(iii)* can be considered as proof obligations. Condition *(i)* requires that all algebras in $\boldsymbol{C}$ can be extended to a model of the specification HID, i.e. HID does not introduce any "confusion" on the algebras of $\boldsymbol{C}$. Condition *(ii)* requires that the relation axiomatized by the formulas $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ is a Σ-congruence and condition *(iii)* provides a criterion for this relation to coincide with the contextual equality induced by some set $\mathcal{C}$ of Σ-contexts and by the input sorts In. In practice, to apply Theorem 7.11, we will proceed as follows (cf. the method described in Section 8). First we select some set $\mathcal{C}$ of observable contexts (guidelines for this selection are provided in Section 8.1). Then it is important to distinguish the following cases:

1. If we consider the total observational equality (i.e. $\mathrm{In} = S$), and if the selected set $\mathcal{C}$ of observable contexts is finite, then we can directly apply Corollary 7.6: there is no need for an hidden part to obtain a finitary axiomatization of $\approx_{\mathrm{Obs},S}$, provided the congruence property is satisfied by the algebras in the class $\boldsymbol{C}$ we are interested in. An example of such a situation is given in Section 8.2.
2. If we consider the total observational equality but the selected set $\mathcal{C}$ of contexts is infinite, then we must introduce an hidden part and use auxiliary function symbols that will intuitively provide the same observations as $\mathcal{C}$ does. However, since we are in the total case, condition *(iii)* of Theorem 7.11 reduces to conditions *(ii)* and *(iii)* of Proposition 7.10. These conditions look rather technical, however in concrete examples it turns out that both conditions can be checked quite easily because one has only to relate contexts of $\mathcal{C}_H$ to contexts of $\mathcal{C}$ and vice versa (cf. e.g. Section 8.3).
3. If we consider a partial observational equality and if the selected set $\mathcal{C}$ of observable contexts is finite, then we must introduce a hidden part with the $D_{H,s}$ predicates, but the condition *(iii)* of Theorem 7.11 reduces to the condition *(i)* of Proposition 7.10. An example of such a case is given in Section 8.4.
4. In the most general case (partial observational equality and infinite set $\mathcal{C}$ of observable contexts), no simplification applies and one has to introduce an adequate hidden part and to check in particular all three conditions of Proposition 7.10.

7.3 A general finitary axiomatization of the observational equality

In this subsection we will show that, for any partial observational equality $\approx_{\mathrm{Obs,In}}$ and any class $\boldsymbol{C}$ of Σ-algebras, it is always possible to construct a finitary axiomatization with hidden part of $\approx_{\mathrm{Obs,In}}$ w.r.t. $\boldsymbol{C}$. The idea is to encode the observable Σ-contexts and the Σ-terms with input variables in the hidden part, and to specify the application of contexts and the interpretation of terms by corresponding hidden function symbols. However, it should be clear that (at least) the encoding of the contexts is so complex that this result is of purely theoretical interest. In particular the encoding of the contexts uses reachability constraints, and it is unfortunately clear that structural induction w.r.t. these reachability constraints will have the same complexity as context induction. In practice it is both more efficient and simpler to define an ad hoc hidden part and to discharge the proof obligations provided by Theorem 7.11.

Definition 7.12 (Encoding of definition domains and contexts) *Let $\Sigma = (S, F)$ be a signature, $\mathrm{Obs} \subseteq S$ be a set of observable sorts and $\mathrm{In} \subseteq S$ be a set of input sorts such that Σ is sensible w.r.t. In. The specification $\mathrm{HID}[\Sigma, \mathrm{Obs}, \mathrm{In}] = \langle \Sigma_H, \mathcal{R}_H, \mathcal{A}x_H \rangle$, with $\Sigma_H = (S_H, F_H)$, is defined by:*

(i) Let $Ar[\Sigma] \stackrel{\mathrm{def}}{=} \{s' \to s \mid s, s' \in S$ and there exists (at least) a Σ-context of sort s with context variable of sort $s'\}$.

(ii) S_H is equal to S plus:

- *(a) for each sort $s \in S \setminus \mathrm{In}$, a new sort $T[s]$,*
- *(b) for each $s' \to s \in Ar[\Sigma]$, a new sort $Ct[s' \to s]$.*

(iii) F_H is equal to F plus:

Function symbols for the encoding of definition domains:

- *(a) for each function symbol $f \in F$ of arity $s_1 \dots s_n \to s$, with $s \notin \mathrm{In}$, a new function symbol $T[f]$ of arity $\zeta_1 s_1 \dots \zeta_n s_n \to T[s]$, with $\zeta_i s_i = s_i$ if $s_i \in \mathrm{In}$ and $\zeta_i s_i = T[s_i]$ otherwise,*
- *(b) for each sort $s \in S \setminus \mathrm{In}$, a new function symbol $I[s]$ of arity $T[s] \to s$,*
- *(c) for each sort $s \in S$, a new unary predicate symbol $D_{H,s}$ of domain s,*

Function symbols for the encoding of Σ-contexts:

- *(d) for each sort $s \in S$, a new constant $Ct[z_s]$ of sort $Ct[s \to s]$,*
- *(e) for each non constant function symbol $f \in F$ of arity $s_1 \dots s_n \to s$, for each $i \in \{1, \dots, n\}$ and for each sort $s' \in S$ such that $s' \to s_i \in Ar[\Sigma]$, a new function symbol $Ct[f, i, s']$ of arity $\zeta_1 s_1 \dots Ct[s' \to s_i] \dots \zeta_n s_n \to Ct[s' \to s]$, with $\zeta_j s_j = s_j$ if $s_j \in \mathrm{In}$ and $\zeta_j s_j = T[s_j]$ otherwise,*
- *(f) for each $s' \to s \in Ar[\Sigma]$, a new function symbol $apply_{s' \to s}$ of arity $Ct[s' \to s]\ s' \to s$.*

(iv) $\mathcal{R}_H = (S_{\mathcal{R}}, F_{\mathcal{R}})$ is defined by:

- *(a) $S_{\mathcal{R}} \stackrel{\mathrm{def}}{=} S_H \setminus S$.*
- *(b) $F_{\mathcal{R}} \stackrel{\mathrm{def}}{=} \{T[f] \mid f \in F\} \cup \{Ct[z_s] \mid s \in S\} \cup$ $\{Ct[f, i, s'] \mid f \in F$ of arity $s_1 \dots s_n \to s,\ n > 0,\ 1 \le i \le n,$ $s' \to s_i \in Ar[\Sigma]\}$.*

(v) $\mathcal{A}x_H$ is the union of:

Axioms for the encoding of definition domains:

- *(a) for each sort $s \in S \setminus \mathrm{In}$ and for each $T[f]$, the (implicitly universally quantified) equation $I[s](T[f](x_1, \dots, x_n)) = f(\zeta_1 x_1, \dots, \zeta_n x_n)$, with*

$\zeta_i x_i = x_i$ *if* $s_i \in \mathrm{In}$ *and* $\zeta_i x_i = I[s_i](x_i)$ *otherwise,*

(b) *for each sort* $s \in S \setminus \mathrm{In}$*, the sentence* $\forall x_s{:}s.\, D_{H,s}(x_s) \Leftrightarrow \exists y_s{:}T[s].\, I[s](y_s) = x_s$,

(c) *for each sort* $s \in \mathrm{In}$*, the sentence* $\forall x_s{:}s.\, D_{H,s}(x_s)$,

Axioms for the encoding of Σ-contexts:

(d) *for each sort* $s \in S$*, the sentence* $\forall x_s{:}s.\ apply_{s \to s}(Ct[z_s], x_s) = x_s$,

(e) *for each* $s' \to s \in Ar[\Sigma]$ *and for each* $Ct[f, i, s']$ *the (implicitly universally quantified) equation:*
$$apply_{s' \to s}(Ct[f, i, s'](x_1, \ldots, c_i, \ldots, x_n), x_{s'}) = f(\zeta_1 x_1, \ldots, apply_{s' \to s_i}(c_i, x_{s'}), \ldots, \zeta_n x_n)\,,$$
where $\zeta_j x_j = x_j$ *if* $s_j \in \mathrm{In}$ *and* $\zeta_j x_j = I[s_j](x_j)$ *otherwise.*

Proposition 7.13 *Let* $\mathrm{HID}[\Sigma, \mathrm{Obs}, \mathrm{In}]$ *be as in Definition 7.12. Let* $\mathcal{C}_H$ *be the set of* Σ_H*-contexts defined by* $\mathcal{C}_H \stackrel{\text{def}}{=} \{apply_{s' \to s}(ctx, z_{s'}) \mid s' \to s \in Ar[\Sigma]$ *such that* $s \in \mathrm{Obs}$, $s' \in S$*, ctx is an arbitrary but fixed variable of sort* $Ct[s' \to s]$ *and* $z_{s'}$ *is the context variable of sort* $s'\}$. *Let* $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ *be the* S*-sorted family of finitary* Σ_H*-formulas induced by the predicate symbols* $D_{H,s}$ *and the contexts* $\mathcal{C}_H$*.*
Then $(\mathrm{HID}[\Sigma, \mathrm{Obs}, \mathrm{In}], \mathrm{HBeh}[D_H, \mathcal{C}_H])$ *is a finitary axiomatization of the observational equality* $\approx_{\mathrm{Obs},\mathrm{In}}$ *with respect to any class* $\boldsymbol{C}$ *of* Σ*-algebras.*

Proof. We must show that the three conditions of Theorem 7.11 are satisfied. It is easy to show that any Σ-algebra A can be extended to a model of $\mathrm{HID}[\Sigma, \mathrm{Obs}, \mathrm{In}]$ by a straightforward construction, i.e. condition *(i)* is satisfied. Condition *(ii)* is trivially satisfied if we show that condition *(iii)* holds with respect to the set $\mathcal{C}_\Sigma^{\mathrm{Obs}}$ of all observable Σ-contexts. To prove this, we must check that the conditions *(i)*, *(ii)* and *(iii)* of Proposition 7.10 are satisfied w.r.t. the set of all observable Σ-contexts $\mathcal{C}_\Sigma^{\mathrm{Obs}}$ and for any Σ_H-algebra $A_H \in \mathrm{Mod}(\mathrm{HID}[\Sigma, \mathrm{Obs}, \mathrm{In}])$. Let $A_H \in \mathrm{Mod}(\mathrm{HID}[\Sigma, \mathrm{Obs}, \mathrm{In}])$.

Proof of *(i)*: W.l.o.g. let $s \in S \setminus \mathrm{In}$. We have the following lemmas:

Lemma 7.14 *For all values* $T[a] \in (A_H)_{T[s]}$ *there exists a term* $t \in T_\Sigma(X_{\mathrm{In}})_s$ *and a valuation* $\alpha : X_{\mathrm{In}} \to A_H$ *such that* $I[s]^{A_H}(T[a]) = I_\alpha(t)$ *.*
(Proof by induction on the constructors of the sort $T[s]$.)

Lemma 7.15 *For any term* $t \in T_\Sigma(X_{\mathrm{In}})_s$ *and any valuation* $\alpha : X_{\mathrm{In}} \to A_H$*, there exists a value* $T[a] \in (A_H)_{T[s]}$ *such that* $I[s]^{A_H}(T[a]) = I_\alpha(t)$ *.*
(Proof by structural induction on $T_\Sigma(X_{\mathrm{In}})$.)

From Lemmas 7.14 and 7.15 we conclude that for any value $a \in (A_H)_s$ the following holds: $D_{H,s}^{A_H}(a)$ iff (using the axioms defining $D_{H,s}$) there exists $T[a] \in (A_H)_{T[s]}$ such that $I[s]^{A_H}(T[a]) = a$ iff (by Lemmas 7.14 and 7.15) there exists a term $t \in T_\Sigma(X_{\mathrm{In}})$ and a valuation $\alpha : X_{\mathrm{In}} \to A_H$ such that $I_\alpha(t) = a$.

Proof of *(ii)*: Let $s' \in S$ be arbitrary. We have to consider Σ_H-contexts of the form $apply_{s' \to s}(ctx, z_{s'})$ with $s \in \mathrm{Obs}$. Condition *(ii)* can now be easily derived from the following lemma:

Lemma 7.16 *For all sorts $Ct[s' \to s] \in S_H$ and for all values $a_H \in (A_H)_{Ct[s' \to s]}$ there exists a Σ-context C of sort s with context variable $z_{s'}$ and a valuation $\beta : X \to A_H$ with $D^{A_H}_{H,s''}(\beta(v_{s''}))$ for all $s'' \in S$ and all $v_{s''} \in \mathrm{Var}(C)$ such that for any value $a \in (A_H)_{s'}$ we have $I_{\beta_a}(C) = I_{\alpha_a}(apply_{s' \to s}(ctx, z_{s'}))$ where $\alpha_a(ctx) = a_H$, $\alpha_a(z_{s'}) = a$ and $\beta_a(z_{s'}) = a$.*
(Proof by induction on the constructors of the sort $Ct[s' \to s]$. Note that Lemma 7.16 is formulated for arbitrary sorts s, not only for observable ones, which allows us to use the necessary induction hypotheses.)

Proof of *(iii)*: Condition *(iii)* follows from another lemma which is symmetric to Lemma 7.16 and can be proved by induction on the structure of arbitrary (observable or not) Σ-contexts. □

8 How to prove behavioural properties: Examples

In this section we discuss general guidelines for proving behavioural properties with respect to an observational equality $\approx_{\mathrm{Obs,In}}$ and we show how to apply the results obtained in the previous sections to various examples.

In the following we assume given a signature $\Sigma = (S, F)$, a set $\mathrm{Obs} \subseteq S$ of observable sorts, a set $\mathrm{In} \subseteq S$ of input sorts (and we assume that Σ is sensible w.r.t. In), and an arbitrary specification SP with signature Σ. We want to prove that $\mathrm{SP} \models_{\approx_{\mathrm{Obs,In}}} \phi$, for some Σ-formula ϕ.

The results obtained in the previous sections lead to the following method:

1. Select an appropriate subset $\mathcal{C} \subseteq \mathcal{C}^{\mathrm{Obs}}_{\Sigma}$ of observable contexts (which contains the trivial contexts z_s for $s \in \mathrm{Obs}$).
2. If $\mathrm{In} \neq S$ then define an appropriate hidden part with predicate symbols $D_{H,s}$.
3. If $\mathcal{C}$ is infinite then complete the hidden part as needed and select some finite set of contexts $\mathcal{C}_H$ (built with hidden function symbols). Otherwise let $\mathcal{C}_H = \mathcal{C}$.
4. Consider the S-sorted family of finitary formulas $\mathrm{HBeh}[D_H, \mathcal{C}_H]$ induced by the predicate symbols $D_{H,s}$ (if any) and by the contexts $\mathcal{C}_H$, and construct the sentence $\mathrm{AXL}[\mathrm{HBeh}[D_H, \mathcal{C}_H]]$.
5. Check that:
   (a) The resulting hidden part HID is a conservative extension of SP.
   (b) $(\mathrm{SP} + \mathrm{HID} + \langle \Sigma^{\mathcal{L}}_H, \mathrm{AXL}[\mathrm{HBeh}[D_H, \mathcal{C}_H]] \rangle) \models \mathrm{CONG}^{\sim}_{\Sigma}$.
   (c) The criteria for the hidden part given in Proposition 7.10 are satisfied, for any Σ_H-algebra $A_H \in \mathrm{Mod}(\mathrm{SP} + \mathrm{HID})$, w.r.t. the selected set of contexts $\mathcal{C}$.

Once these steps are done (once for all), Theorem 6.3 implies that:

SP $\models_{\approx_{\mathrm{Obs,In}}} \phi$ if and only if $(\mathrm{SP} + \mathrm{HID} + \langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}[D_H, \mathcal{C}_H]] \rangle) \models \mathcal{L}(\phi)$, for any Σ-formula ϕ. [9]

Remark 8.1 It is important to note that since we consider a partial observational equality (and not an arbitrary behavioural equality), we can apply some obvious simplifications to the lift operator:

1. If all observable sorts are input sorts (i.e. $\mathrm{Obs} \subseteq \mathrm{In}$), and in particular if we consider a total observational equality (since then $\mathrm{Obs} \subseteq \mathrm{In} = S$), then on observable sorts, the observational equality coincides with the set-theoretic equality, hence the predicate symbols $\sim_s$ are not necessary for the observable sorts s.
2. If we consider a total observational equality $\approx_{\mathrm{Obs},S}$, then for any Σ-formula ϕ, the lifted formula $\mathcal{L}(\phi)$ is equivalent to the formula ϕ where all equations $l = r$ between terms of a non-observable sort s are replaced by $l \sim_s r$.
3. If we consider a partial observational equality $\approx_{\mathrm{Obs,In}}$, then we need predicate symbols $D_{H,s}$ for non-input sorts s only. Moreover, for each non-input sort s, we know that $x_s \sim_s x_s$ is equivalent to $D_{H,s}(x_s)$, and this fact can be used to simplify the lifting of formulas.

It is clear that the selection of an adequate subset $\mathcal{C}$ of $\mathcal{C}_\Sigma^{\mathrm{Obs}}$ is a crucial step in the method. In particular, if we succeed in finding a convenient finite set $\mathcal{C}$, then the definition of the hidden part is considerably simplified (it is enough to introduce the predicate symbols $D_{H,s}$) and the proof obligations are much easier to discharge. But even if the selected set $\mathcal{C}$ is infinite, it is important to understand that the subsequent steps will be much easier if we consider a strict subset of $\mathcal{C}_\Sigma^{\mathrm{Obs}}$ instead of the set of all observable contexts. In the next subsection we explain how in general such a set $\mathcal{C}$ can be easily deduced from the specification we are interested in. Then we provide various examples which show how our method can be applied.

8.1 Crucial contexts

The problem now is to find an adequate subset $\mathcal{C}$ of $\mathcal{C}_\Sigma^{\mathrm{Obs}}$, if possible, a finite subset $\mathcal{C}$. Remember that observable contexts represent experiments with observable results. A typical (simple) 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 (more complex experiments will be arbitrary combinations of these simple ones). The cru-

[9] Note that if SP is a smooth specification, then the specification $\mathrm{SP} + \mathrm{HID} + \langle \Sigma_H^{\mathcal{L}}, \mathrm{AXL}[\mathrm{HBeh}[D_H, \mathcal{C}_H]] \rangle$ is smooth as well.

cial 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 could be enough to consider the "crucial contexts" of the form $f(x_1, \ldots, x_{k-1}, z_{s_k}, x_{k+1}, \ldots, x_n)$, with $f \in F$ of arity $s_1 \ldots s_{k-1}\, s_k\, s_{k+1} \ldots s_n \to s$, $s \in \mathrm{Obs}$ and $s_k \in S \setminus \mathrm{Obs}$. If the contextual equality induced by these crucial contexts is a Σ-congruence (for the algebras in the class $\boldsymbol{C}$ we are interested in), then using Proposition 7.4 we know that it is fine to replace the infinite set of all observable contexts $\mathcal{C}_\Sigma^{\mathrm{Obs}}$ by the finite set of the "crucial contexts".

It should be clear that the selection of the "crucial contexts" is mainly a starting point. As we will see in the examples described in the next subsections, in some cases the set of the "crucial contexts" is not optimal (i.e. an even smaller set of contexts is adequate), while in other cases the set of the "crucial contexts" is not adequate (i.e. we have to select a larger, infinite set of contexts). However, from our experience, the set of the "crucial contexts" is always a very useful starting point and when not adequate it provides nevertheless the "right intuition" about which set of contexts $\mathcal{C}$ must be selected.

8.2 The `CONTAINER` *example*

Let us consider again the `CONTAINER` specification introduced in Examples 2.2 and 3.12. The observable sorts are {`Elem, Nat, Bool`} and all sorts are input sorts (i.e. we consider the total observational equality).

We are interested in proving behavioural properties of the `CONTAINER` specification. First, as explained in Remark 8.1, note that we can apply some obvious simplifications, since the observational equality we consider is a total one. Hence we can considerably simplify the lifting of formulas, the axiomatization of the observational equality and the induced sentence AXL[.] by using just one predicate symbol $\sim_{\mathrm{Cont}}$ (for the sake of clarity we abbreviate the sort `Container` by `Cont`).

Here the set of the "crucial contexts" is:
$\{\mathtt{e} \in z_{\mathrm{Cont}},\ \mathtt{card}(z_{\mathrm{Cont}}),\ \mathtt{subset}(\mathtt{S},z_{\mathrm{Cont}}),\ \mathtt{subset}(z_{\mathrm{Cont}},\mathtt{S}),\ z_{\mathrm{Bool}},\ z_{\mathrm{Nat}},\ z_{\mathrm{Elem}}\}$. However, intuitively here the "crucial context" $\mathtt{e} \in z_{\mathrm{Cont}}$ is enough to observe containers, and the other observing operations (`card` and `subset`) do not observe "more" than the operation $\in$ does. Hence we will select the set $\mathcal{C} = \{\mathtt{e} \in z_{\mathrm{Cont}},\ z_{\mathrm{Bool}},\ z_{\mathrm{Nat}},\ z_{\mathrm{Elem}}\}$ of observable contexts.

Since we consider the total observational equality, this leads directly to the following formula $\mathrm{TBeh}_{\mathrm{Cont}}(\mathtt{S},\mathtt{S'})$ (cf. Corollary 7.6):
$\mathrm{TBeh}_{\mathrm{Cont}}(\mathtt{S},\mathtt{S'}) \stackrel{\mathrm{def}}{=} [\forall \mathtt{e}:\mathtt{Elem}.\ \mathtt{e} \in \mathtt{S} = \mathtt{e} \in \mathtt{S'}]$.

The corresponding axiomatization is the sentence AXL[$\text{TBeh}_{\text{Cont}}$]:
$\forall$ `S,S':Container. ([`$\forall$`e:Elem. e` $\in$ `S = e` $\in$ `S'] ` $\Leftrightarrow$ ` S ` $\sim_{\text{Cont}}$ ` S').`

The formula $\text{CONG}_\Sigma^{\tilde{}}$ is (after some obvious simplifications, cf. Remark 7.5):

$\forall$ `e:Elem, S,S',S1,S1':Container.`
    `[(S` $\sim_{\text{Cont}}$ `S')` $\wedge$ `(S1` $\sim_{\text{Cont}}$ `S1')` $\Rightarrow$
        `insert(e,S)` $\sim_{\text{Cont}}$ `insert(e,S')` $\wedge$
        `(S` $\cup$ `S1)` $\sim_{\text{Cont}}$ `(S'` $\cup$ `S1')` $\wedge$
        `remove(e,S)` $\sim_{\text{Cont}}$ `remove(e,S')` $\wedge$
        `card(S) = card(S')` $\wedge$
        `subset(S,S1) = subset(S',S1') ]`

Let $\texttt{CONTAINER}^{\sim} \stackrel{\text{def}}{=}$
$$(\texttt{CONTAINER} + \langle \text{Sig}(\texttt{CONTAINER}) \cup \{\sim_{\text{Cont}}\}, \text{AXL}[\text{TBeh}_{\text{Cont}}]\rangle).$$

In a first step we must check that $\texttt{CONTAINER}^{\sim} \models \text{CONG}_\Sigma^{\tilde{}}$, which is not difficult.

It is then very easy to prove that:
$\texttt{CONTAINER}^{\sim} \models$ `insert(e,insert(e,S))` $\sim_{\text{Cont}}$ `insert(e,S)`
and that:
$\texttt{CONTAINER}^{\sim} \models$ `insert(e,insert(e',S))` $\sim_{\text{Cont}}$ `insert(e',insert(e,S))`
which means that the two corresponding equations are behaviourally valid in the model class of the `CONTAINER` specification. This means as well that this specification can be considered as a correct behavioural implementation of sets.

8.3 The `STACK` *example*

Let us consider the following `STACK` specification.

spec: STACK
  use: ELEM
  sort: Stack
  generated by:
    empty : $\rightarrow$ Stack
    push : Elem Stack $\rightarrow$ Stack
  operations:
    pop : Stack $\rightarrow$ Stack
    top : Stack $\rightarrow$ Elem
  axioms:
    $\forall$ S:Stack, e:Elem.
    pop(empty) = empty
    pop(push(e,S)) = S

```
    top(push(e,S)) = e
end STACK.
```

We assume that all sorts are input sorts (i.e. we consider the total observational equality) and that the sort `Elem` is observable. We are interested in proving behavioural properties of stacks. We can again apply obvious simplifications and use just one predicate symbol $\sim_{\mathrm{Stack}}$ (cf. Remark 8.1).

Here the set of the "crucial contexts" is reduced to $\{\mathtt{top(}z_{\mathrm{Stack}}\mathtt{)},\ z_{\mathrm{Elem}}\}$. It is clear that the context $\mathtt{top(}z_{\mathrm{Stack}}\mathtt{)}$ is not enough to observe stacks: two stacks may have the same top element without being observationally equal. Intuitively two stacks must be considered as observationally equal if the elements stored in the stacks are the same (and if they are stored in the same order). This can be checked by first looking to the top element, then popping the stacks and looking again to the top element, etc. until both stacks are (simultaneously) empty. Hence we will select the set $\mathcal{C} = \{\mathtt{top(pop}^n\mathtt{(}z_{\mathrm{Stack}}\mathtt{))} \mid n \in \mathbb{N}\} \cup \{z_{\mathrm{Elem}}\}$.

Since we select an infinite set of observable contexts, we must find an adequate hidden part HID and a set of contexts $\mathcal{C}_H$ to be used instead of $\mathcal{C}$ itself. But the selected set $\mathcal{C}$ of observable contexts is simple enough to suggest the following solution. Let HID be the specification:

$$\begin{aligned}\mathrm{HID} \stackrel{\mathrm{def}}{=} \mathtt{NATP} + \langle &\mathrm{Sig}(\mathtt{STACK}) \cup \mathrm{Sig}(\mathtt{NATP}) \cup \{\mathtt{topn} : \mathtt{Nat\ Stack} \to \mathtt{Elem}\},\\ &\{\forall\, \mathtt{S:Stack,\ x:Nat.}\\ &\quad \mathtt{topn(0,S) = top(S)}\\ &\quad \mathtt{topn(s(x),S) = topn(x,pop(S))}\}\rangle\end{aligned}$$

whereby it is assumed that `NATP` is a monomorphic specification of Peano's natural numbers having $\mathbb{N}$ as model. Let $\mathcal{C}_H \stackrel{\mathrm{def}}{=} \{\mathtt{topn(x,}z_{\mathrm{Stack}}\mathtt{)},\ z_{\mathrm{Elem}}\}$.

Since we consider the total observational equality, we do not need any hidden predicate symbol, and we obtain as resulting formula:
$\mathrm{HBeh}[\emptyset,\mathtt{topn}]_{\mathrm{Stack}}\mathtt{(S,S')} \stackrel{\mathrm{def}}{=} [\forall \mathtt{x:Nat.\ topn(x,S) = topn(x,S')}]$.

The corresponding axiomatization is the sentence $\mathrm{AXL}[\mathrm{HBeh}[\emptyset,\mathtt{topn}]_{\mathrm{Stack}}]$:
$\forall\ \mathtt{S,S':Stack.}\ ([\forall \mathtt{x:Nat.\ topn(x,S) = topn(x,S')}] \Leftrightarrow \mathtt{S} \sim_{\mathrm{Stack}} \mathtt{S'})$.

The formula $\mathrm{CONG}_{\Sigma}^{\sim}$ is (after some obvious simplifications, cf. Remark 7.5): [10]
$\forall\ \mathtt{e:Elem,\ S,S':Stack.}\ (\mathtt{S} \sim_{\mathrm{Stack}} \mathtt{S'}) \Rightarrow \mathtt{push(e,S)} \sim_{\mathrm{Stack}} \mathtt{push(e,S')}$

Now let `STACK-IMPL` be a specification (with signature Sig(`STACK`)) which is supposed to describe a concrete implementation of stacks (for instance by

[10] In particular, note that the set $\mathcal{C}$ is generated by the subsignature where `pop` and `top` are the only function symbols. Hence it is not necessary to check that
$(\mathtt{S} \sim_{\mathrm{Stack}} \mathtt{S'}) \Rightarrow \mathtt{pop(S)} \sim_{\mathrm{Stack}} \mathtt{pop(S')}$ or that
$(\mathtt{S} \sim_{\mathrm{Stack}} \mathtt{S'}) \Rightarrow \mathtt{top(S) = top(S')}$.

means of arrays and pointers). Let $\texttt{STACK-IMPL-H}^{\sim} \stackrel{\text{def}}{=} (\texttt{STACK-IMPL} + \text{HID} + \langle \text{Sig(HID)} \cup \{\sim_{\text{Stack}}\}, \text{AXL}[\text{HBeh}[\emptyset, \texttt{topn}]_{\text{Stack}}]\rangle)$.

Obviously the condition *(iii)* of Theorem 7.11 holds (independently of the choice of the specification STACK-IMPL), since to any context $\texttt{topn(x,}z_{\text{Stack}}\texttt{)}$ and any valuation $\alpha : \{\texttt{x}\} \to \mathbb{N}$ corresponds the context $\texttt{top(pop}^{\alpha(\texttt{x})}\texttt{(}z_{\text{Stack}}\texttt{))}$ (and reciprocally).

Hence, if HID is a conservative extension of the STACK-IMPL specification and if moreover $\texttt{STACK-IMPL-H}^{\sim} \models \text{CONG}^{\sim}_{\Sigma}$, we can apply Theorem 7.11, and as a consequence we know that, for any formula ϕ:
$\texttt{STACK-IMPL} \models_{\approx_{\text{Obs},S}} \phi$ if and only if $\texttt{STACK-IMPL-H}^{\sim} \models \mathcal{L}(\phi)$.

This means that for checking that the specification STACK-IMPL is indeed a behavioural implementation of STACK, it is then enough to show that, for all STACK axioms ϕ, $\texttt{STACK-IMPL-H}^{\sim} \models \mathcal{L}(\phi)$.

8.4 The LIST *example*

As a last example we will consider a classical implementation of sets by non-redundant lists. Let us first consider the following LIST specification:

spec: LIST
  use: ELEM, BOOL
  sort: List
  generated by:
    empty : → List
    cons : Elem List → List
  operations:
    head : List → Elem
    tail : List → List
    _ ∈ _ : Elem List → Bool
    insert : Elem List → List
    remove : Elem List → List
  axioms:
    ∀ L,L':List, e,e':Elem.
    head(cons(e,L)) = e
    tail(empty) = empty
    tail(cons(e,L)) = L
    e ∈ empty = false
    [e ∈ cons(e',L) = true] ⇔ [(e = e') ∨ (e ∈ L = true)]
    [e ∈ L = true] ⇒ insert(e,L) = L
    [e ∈ L = false] ⇒ insert(e,L) = cons(e,L)

```
    remove(e,empty) = empty
    remove(e,cons(e,L)) = L
    e ≠ e' ⇒ remove(e,cons(e',L)) = cons(e',remove(e,L))
end LIST.
```

First note that `remove` just removes the first occurrence of an element. Hence it is clear that $\texttt{LIST} \not\models \texttt{e} \in \texttt{remove(e,L) = false}$.

We are now interested in studying the behaviour of lists if we forget the operations `cons`, `head` and `tail`. (Indeed, to use lists as an implementation of sets, we must first forget these operations and in addition perform an appropriate renaming, but we will not take this renaming into account here for the sake of simplicity.) Formally, we consider the signature Σ-`Set` defined by $\Sigma\texttt{-Set} \stackrel{\text{def}}{=} \mathrm{Sig}(\texttt{LIST}) \setminus \{\texttt{cons, head, tail}\}$, and we consider the specification $\texttt{LIST}|_{\Sigma-\mathrm{Set}}$, which has exactly Σ-`Set` as signature, and the model class of which is defined (as usual) by $\mathrm{Mod}(\texttt{LIST}|_{\Sigma-\mathrm{Set}}) \stackrel{\text{def}}{=} \mathrm{Mod}(\texttt{LIST})|_{\Sigma-\mathrm{Set}}$.

The behavioural theory of $\texttt{LIST}|_{\Sigma-\mathrm{Set}}$ will now be studied w.r.t. the partial observational equality $\approx_{\mathrm{Obs,Obs}}$ induced on Σ-`Set`-algebras by $\mathrm{Obs} = \{\texttt{Elem}, \texttt{Bool}\}$. This observational equality is a family of partial Σ-`Set`-congruences. Indeed, since our aim is to prove that $\texttt{LIST}|_{\Sigma-\mathrm{Set}}$ is a correct behavioural implementation of sets, it is important not to choose `List` as an input sort, since we must only consider lists built with `empty`, `insert` and `remove` for the behavioural satisfaction of the set axioms (only these lists actually represent sets).

The set of the "crucial contexts" is $\mathcal{C} = \{\texttt{e} \in z_{\mathrm{List}}, z_{\mathrm{Bool}}, z_{\mathrm{Elem}}\}$, and looks fine enough to be selected.

Since we consider a partial observational equality $\approx_{\mathrm{Obs,Obs}}$, we must find an adequate hidden part HID with a predicate symbol $D_{H,\mathrm{List}}$. (As for the previous examples, using Remark 8.1, we apply obvious simplifications and use just one predicate symbol $\sim_{\mathrm{List}}$ in the lifting; similarly we need predicate symbols $D_{H,s}$ for non-input sorts s only, hence here only for the sort `List`.) Since the selected set of contexts is finite there is no need for contexts built with auxiliary hidden function symbols.

The rôle of the predicate symbol $D_{H,\mathrm{List}}$ is to denote the lists built with the Σ-`Set` operations (and observable values), i.e. the lists built with `empty`, `insert` and `remove` (and not with `cons` or `tail`). Intuitively it is clear that these lists are exactly the lists with no duplicates. Hence it is quite obvious that the following hidden part HID should be adequate:

$\mathrm{HID} \stackrel{\text{def}}{=} \langle \Sigma\texttt{-Set} \cup \{\texttt{nodup : List}\},$
$\quad \{\forall \texttt{L:List. nodup(L)} \Leftrightarrow (\forall \texttt{e:Elem. e} \in \texttt{remove(e,L) = false})\}\rangle.$

We obtain as resulting formula $\mathrm{HBeh}[\mathtt{nodup}, \in]_{\mathrm{List}}$`(L,L')`:
`[nodup(L) ∧ nodup(L') ∧ (∀e:Elem. e ∈ L = e ∈ L')]`.

The corresponding axiomatization is the sentence $\mathrm{AXL}[\mathrm{HBeh}[\mathtt{nodup}, \in]_{\mathrm{List}}]$:
`∀ L,L':List.`
`([nodup(L) ∧ nodup(L') ∧ (∀e:Elem. e ∈ L = e ∈ L')] ⇔`
$\mathtt{L} \sim_{\mathrm{List}} \mathtt{L'}$).

In particular we have: $\forall$`L:List. nodup(L)` $\Leftrightarrow$ $\mathtt{L} \sim_{\mathrm{List}} \mathtt{L}$.

The formula $\mathrm{CONG}^{\sim}_{\Sigma-\mathrm{Set}}$ is (after some obvious simplifications, cf. Remark 7.5):
`∀ e:Elem, L,L':List.`
$[(\mathtt{L} \sim_{\mathrm{List}} \mathtt{L'}) \Rightarrow$
$\mathtt{insert(e,L)} \sim_{\mathrm{List}} \mathtt{insert(e,L')} \wedge$
$\mathtt{remove(e,L)} \sim_{\mathrm{List}} \mathtt{remove(e,L')}\ ]$

Let $\mathtt{LIST}|_{\Sigma-\mathrm{Set}}\text{-}\mathtt{H}^{\sim} \stackrel{\mathrm{def}}{=}$
$(\mathtt{LIST}|_{\Sigma-\mathrm{Set}} + \mathrm{HID} + \langle \mathrm{Sig}(\mathrm{HID}) \cup \{\sim_{\mathrm{List}}\}, \mathrm{AXL}[\mathrm{HBeh}[\mathtt{nodup}, \in]_{\mathrm{List}}]\rangle)$
and let
$\mathtt{LIST\text{-}H}^{\sim} \stackrel{\mathrm{def}}{=} (\mathtt{LIST} + \mathrm{HID} + \langle \mathrm{Sig}(\mathrm{HID}) \cup \{\sim_{\mathrm{List}}\}, \mathrm{AXL}[\mathrm{HBeh}[\mathtt{nodup}, \in]_{\mathrm{List}}]\rangle)$.

It is obvious that HID is a conservative extension of the $\mathtt{LIST}|_{\Sigma-\mathrm{Set}}$ specification (indeed it is also a conservative extension of the `LIST` specification). To check that $\mathtt{LIST}|_{\Sigma-\mathrm{Set}}\text{-}\mathtt{H}^{\sim} \models \mathrm{CONG}^{\sim}_{\Sigma-\mathrm{Set}}$, it is equivalent to check that $\mathtt{LIST\text{-}H}^{\sim} \models \mathrm{CONG}^{\sim}_{\Sigma-\mathrm{Set}}$, which is not difficult. Then we must check that the condition *(iii)* of Theorem 7.11 holds, i.e. that the condition *(i)* of Proposition 7.10 is satisfied. This follows from an easy proof by induction w.r.t. $T_{\Sigma-\mathrm{Set}}(X_{\{\mathrm{Elem},\mathrm{Bool}\}})_{\mathrm{List}}$.

Hence we can apply Theorem 7.11, and as a consequence we know that, for any Σ-`Set`-formula ϕ, $\mathtt{LIST}|_{\Sigma-\mathrm{Set}} \models_{\approx_{\mathrm{Obs},\mathrm{Obs}}} \phi$ if and only if $\mathtt{LIST}|_{\Sigma-\mathrm{Set}}\text{-}\mathtt{H}^{\sim} \models \mathcal{L}(\phi)$ if and only if $\mathtt{LIST\text{-}H}^{\sim} \models \mathcal{L}(\phi)$.

For instance, to prove that:
$\mathtt{LIST}|_{\Sigma-\mathrm{Set}} \models_{\approx_{\mathrm{Obs},\mathrm{Obs}}}$ `insert(e,insert(e',L)) = insert(e',insert(e,L))`
and that:
$\mathtt{LIST}|_{\Sigma-\mathrm{Set}} \models_{\approx_{\mathrm{Obs},\mathrm{Obs}}}$ `e ∈ remove(e,L) = false`,
it is equivalent to prove that (since `nodup(L)` is equivalent to $\mathtt{L} \sim_{\mathrm{List}} \mathtt{L}$):
$\mathtt{LIST\text{-}H}^{\sim} \models$
$\mathtt{nodup(L)} \Rightarrow \mathtt{insert(e,insert(e',L))} \sim_{\mathrm{List}} \mathtt{insert(e',insert(e,L))}$
and that:
$\mathtt{LIST\text{-}H}^{\sim} \models \mathtt{nodup(L)} \Rightarrow \mathtt{e} \in \mathtt{remove(e,L)} = \mathtt{false}$,
which is not difficult. This indeed is the crucial step required to conclude that our `LIST` specification can be considered as a correct behavioural implementation of sets.

9 Reducing partial observational equalities to total ones

In Section 7 we have explained how to find a finitary axiomatization of a partial observational equality. An important part of this task was the introduction of adequate hidden predicate symbols $D_{H,s}$ axiomatized in a suitable way. The main idea of this section is to show that it is always possible to consider, instead of a partial observational equality, a corresponding total observational equality, provided we replace the class $\boldsymbol{C}$ of algebras we are interested in by another related class of algebras. Using this alternative technique, one first gets rid of the problems raised by partiality, and then the method described in Section 8 is applied, i.e. we have to obtain a suitable finitary axiomatization of the corresponding total observational equality. However this task is in general much simpler for total observational equalities than for partial ones, since in particular there is no need to guess and axiomatize auxiliary hidden predicate symbols $D_{H,s}$ (i.e. the second step of the method described in Section 8 is not necessary).

Throughout this section we still assume given a signature $\Sigma = (S, F)$, a set $\mathrm{Obs} \subseteq S$ of observable sorts and a set $\mathrm{In} \subseteq S$ of input sorts (and we assume that Σ is sensible w.r.t. In).

Intuitively, to check the behavioural validity of a formula ϕ w.r.t. a Σ-algebra A and the partial observational equality $\approx_{\mathrm{Obs,In}}$, it is equivalent to check the behavioural validity of the formula ϕ w.r.t. the subalgebra $A[X_{\mathrm{In}}]$ and the total observational equality $\approx_{\mathrm{Obs},S}$. More formally:

Proposition 9.1 *Let A be a Σ-algebra, $\boldsymbol{C}$ be a class of Σ-algebras and ϕ be a Σ-formula.*

(i) $A \models_{\approx_{\mathrm{Obs,In}}} \phi$ *if and only if* $A[X_{\mathrm{In}}] \models_{\approx_{\mathrm{Obs},S}} \phi$ *.*
(ii) $\boldsymbol{C} \models_{\approx_{\mathrm{Obs,In}}} \phi$ *if and only if* $\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}(\boldsymbol{C}) \models_{\approx_{\mathrm{Obs},S}} \phi$ *,*
where $\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}(\boldsymbol{C}) \stackrel{\mathrm{def}}{=} \{A[X_{\mathrm{In}}] \mid A \in \boldsymbol{C}\}$ *.*

Proof. *(ii)* follows directly from *(i)*. *(i)* is implied by the fact that for any Σ-algebra A, any valuation $\alpha : X \to A[X_{\mathrm{In}}]$ and any Σ-formula ϕ, $A, \alpha \models_{\approx_{\mathrm{Obs,In}}} \phi$ iff $A[X_{\mathrm{In}}], \alpha \models_{\approx_{\mathrm{Obs},S}} \phi$. This fact follows from an easy induction on the structure of the formula ϕ, using the fact that a valuation $\alpha : X \to A[X_{\mathrm{In}}]$ induces an interpretation I_α from $T_\Sigma(X)$ to $A[X_{\mathrm{In}}]$, and that for any elements $a, b \in A[X_{\mathrm{In}}]$, $a \approx_{\mathrm{Obs,In},A} b$ iff $a \approx_{\mathrm{Obs},S,A[X_{\mathrm{In}}]} b$. □

Note that our $\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}$ operator corresponds to the "restriction functor" considered e.g. in [10] and in [24]. Our aim is now to show that the $\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}$ operator can be considered as a derived operator of the ASL-like specification language. For this we will assume that, in addition to basic specifications and the sum operator $+$, the ASL-like specification language considered contains also the following two specification-building primitives:

Renaming: For any specification SP with signature Σ and any *bijective* signature morphism $\sigma : \Sigma \to \Sigma'$, the expression **rename** SP **by** σ is a specification, with signature $\Sigma' = \sigma(\mathrm{Sig}(\mathrm{SP}))$, and with model class $\sigma(\mathrm{Mod}(\mathrm{SP}))$, where for any class $\boldsymbol{C}$ of Σ-algebras, $\sigma(\boldsymbol{C}) \stackrel{\text{def}}{=} \{A' \in \mathrm{Alg}(\Sigma') \mid A'|_\sigma \in \boldsymbol{C}\}$ (thereby "$|_\sigma$" denotes the forgetful functor induced by the bijective signature morphism σ, cf. e.g. [9]). In particular, when SP is a basic specification $\langle \Sigma, \mathcal{A}x \rangle$, then the specification **rename** SP **by** σ is equivalent to the basic specification $\langle \Sigma', \sigma(\mathcal{A}x) \rangle$, where σ is extended in a straightforward way to Σ-terms and Σ-formulas.

Export: For any specification SP with signature Σ and any subsignature Σ' of Σ, the expression $\mathrm{SP}|_{\Sigma'}$ is a specification, with signature Σ' and with model class $\mathrm{Mod}(\mathrm{SP})|_{\Sigma'}$. [11]

To show that the $\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}$ operator can be defined as a derived operator of any ASL-like language containing the rename and export specification-building primitives, we start by introducing an auxiliary specification associated to the signature Σ and to the input sorts In:

Definition 9.2 (Encoding of the $\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}$ operator) *Let $\Sigma = (S, F)$ be a signature and $\mathrm{In} \subseteq S$ be a set of input sorts such that Σ is sensible w.r.t.* In.

(i) Let $S_{\neg\mathrm{In}} \stackrel{\text{def}}{=} S \setminus \mathrm{In}$ (hence $S = \mathrm{In} \oplus S_{\neg\mathrm{In}}$).
(ii) Let $F_{\mathrm{In}} \stackrel{\text{def}}{=} \{f \in F \mid$ all sorts in the arity of f are input sorts$\}$, and let $F_{\neg\mathrm{In}}$ be its complement (hence $F = F_{\mathrm{In}} \oplus F_{\neg\mathrm{In}}$, and for each $f \in F_{\neg\mathrm{In}}$, the arity of f involves at least one non-input sort).
(iii) For each sort $s \in S_{\neg\mathrm{In}}$, let $c[s]$ be a "new" sort (i.e. we assume $c[s] \notin S$).
(iv) For each function symbol $f \in F_{\neg\mathrm{In}}$ of arity $s_1 \ldots s_n \to s$, let $c[f]$ be a "new" function symbol of arity $\zeta_1 s_1 \ldots \zeta_n s_n \to \zeta s$, with $\zeta_i s_i = s_i$ if $s_i \in \mathrm{In}$ and $\zeta_i s_i = c[s_i]$ otherwise, and similarly for ζs (again we assume $c[f] \notin F$).
(v) Let $\Sigma_c = (S_c, F_c)$ be the signature defined by:
$S_c \stackrel{\text{def}}{=} \mathrm{In} \oplus \{c[s] \mid s \in S_{\neg\mathrm{In}}\}$ and $F_c \stackrel{\text{def}}{=} F_{\mathrm{In}} \oplus \{c[f] \mid f \in F_{\neg\mathrm{In}}\}$.

Obviously we have a bijective signature morphism $\sigma_c : \Sigma \to \Sigma_c$, defined by $\sigma_c(s) = s$ if $s \in \mathrm{In}$ and $\sigma_c(s) = c[s]$ if $s \notin \mathrm{In}$, $\sigma_c(f) = f$ if $f \in F_{\mathrm{In}}$ and $\sigma_c(f) = c[f]$ if $f \in F_{\neg\mathrm{In}}$.

The specification $\mathrm{RESTRICT}[\Sigma, \mathrm{In}] = \langle \Sigma_{\mathrm{RESTR}}, \mathcal{R}_{\mathrm{RESTR}}, \mathcal{A}x_{\mathrm{RESTR}} \rangle$ *is defined*

[11] Note that we have already implicitly used this export operator in our `LIST` example described in Section 8.4.

by:

(i) Σ_{RESTR} *is equal to* $\Sigma \cup \Sigma_c$ *plus, for each sort* $s \in S_{\neg\mathrm{In}}$, *a new function symbol* $I[s]$ *of arity* $s \to c[s]$.

(ii) $\mathcal{R}_{\mathrm{RESTR}} = (S_{\neg\mathrm{In}}, \{f \in F_{\neg\mathrm{In}}$ *of arity* $s_1 \ldots s_n \to s$, *with* $s \notin \mathrm{In}\})$.

(iii) $\mathcal{A}x_{\mathrm{RESTR}}$ *is the union of:*

(a) *For each function symbol* $f \in F_{\neg\mathrm{In}}$ *of arity* $s_1 \ldots s_n \to s$, *with* $s \notin \mathrm{In}$, *the (implicitly universally quantified) equation:* $I[s](f(x_1, \ldots, x_n)) = c[f](\zeta_1 x_1, \ldots, \zeta_n x_n)$, *with* $\zeta_i x_i = x_i$ *if* $s_i \in \mathrm{In}$ *and* $\zeta_i x_i = I[s_i](x_i)$ *otherwise.*

(b) *For each function symbol* $f \in F_{\neg\mathrm{In}}$ *of arity* $s_1 \ldots s_n \to s$, *with* $s \in \mathrm{In}$, *the (implicitly universally quantified) equation:* $f(x_1, \ldots, x_n) = c[f](\zeta_1 x_1, \ldots, \zeta_n x_n)$, *with* $\zeta_i x_i = x_i$ *if* $s_i \in \mathrm{In}$ *and* $\zeta_i x_i = I[s_i](x_i)$ *otherwise.*

(c) *For each sort* $s \in S_{\neg\mathrm{In}}$, *the sentence:* $\forall x_s, y_s{:}s.\ I[s](x_s) = I[s](y_s) \Rightarrow x_s = y_s$.

Proposition 9.3 ($\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}$ **is a derived operator)** *Let* Σ, In, Σ_c, σ_c *and* $\mathrm{RESTRICT}[\Sigma, \mathrm{In}]$ *be as in Definition 9.2. For any class* $\boldsymbol{C}$ *of* Σ*-algebras closed under isomorphism, we have:*
$\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}(\boldsymbol{C}) = (\sigma_c(\boldsymbol{C}) + \mathrm{RESTRICT}[\Sigma, \mathrm{In}])|_\Sigma$.
In particular, if $\boldsymbol{C}$ *is the model class of some ASL-like structured specification* SP *with signature* Σ, *we have:*
$\mathrm{RESTR}_{[\Sigma,\mathrm{In}]}(\mathrm{Mod}(\mathrm{SP})) = ($**rename** SP **by** $\sigma_c + \mathrm{RESTRICT}[\Sigma, \mathrm{In}])|_\Sigma$.

Proof. Let $\boldsymbol{C}$ be a class of Σ-algebras closed under isomorphism. Note that $\Sigma \cap \Sigma_c = (S_{\mathrm{In}}, F_{\mathrm{In}})$.

$\subseteq$**:** Let $A \in \boldsymbol{C}$ be a Σ-algebra. Consider the Σ_{RESTR}-algebra B defined by:

(i) $B|_{\Sigma_c} = \sigma_c(A)$, i.e. $B|_{\Sigma_c}$ is the Σ_c-algebra A_c defined by $(A_c)|_{\sigma_c} = A$. Note that for any sort $s \in S_{\neg\mathrm{In}}$, we have $B_{c[s]} = A_s$, and for any sort $s \in \mathrm{In}$, we have $B_s = A_s$.

(ii) $B|_\Sigma = A[X_{\mathrm{In}}]$.

(iii) For each sort $s \in S_{\neg\mathrm{In}}$, $I[s]^B : B_s (= A[X_{\mathrm{In}}]_s) \to B_{c[s]} (= A_s)$ is the canonical inclusion.

B is well-defined since, for any sort $s \in \mathrm{In}$, $B_s = A_s = A[X_{\mathrm{In}}]_s$, and for any function symbol $f \in F_{\mathrm{In}}$, $f^B = f^{\sigma_c(A)} = f^A = f^{A[X_{\mathrm{In}}]}$. Moreover, obviously $B \in \mathrm{Mod}(\mathrm{RESTRICT}[\Sigma, \mathrm{In}])$ and $B|_{\Sigma_c} \in \sigma_c(\boldsymbol{C})$, i.e. $B \in (\sigma_c(\boldsymbol{C}) + \mathrm{RESTRICT}[\Sigma, \mathrm{In}])$. But by construction $A[X_{\mathrm{In}}] = B|_\Sigma$, hence $A[X_{\mathrm{In}}] \in (\sigma_c(\boldsymbol{C}) + \mathrm{RESTRICT}[\Sigma, \mathrm{In}])|_\Sigma$.

$\supseteq$**:** Let $B \in (\sigma_c(\boldsymbol{C}) + \mathrm{RESTRICT}[\Sigma, \mathrm{In}])$ be a Σ_{RESTR}-algebra, and let A be the Σ-algebra defined by $A \stackrel{\mathrm{def}}{=} (B|_{\Sigma_c})|_{\sigma_c}$. Obviously $A \in \boldsymbol{C}$. Now, note that the applications $I[s]^B$, for $s \in S_{\neg\mathrm{In}}$, together with the identities on B_s, for $s \in \mathrm{In}$, induce an injective Σ-morphism h from $B|_\Sigma$ to A, since $B \models \mathcal{A}x_{\mathrm{RESTR}}$. Hence $B|_\Sigma$ is isomorphic to the image of h, and due to the reachability constraint $\mathcal{R}_{\text{RESTR}}$, we conclude that $B|_\Sigma$ is isomorphic to $A[X_{\text{In}}]$. Hence $B|_\Sigma \in \text{RESTR}_{[\Sigma,\text{In}]}(\boldsymbol{C})$, since we have assumed that $\boldsymbol{C}$ is closed under isomorphism (hence $\text{RESTR}_{[\Sigma,\text{In}]}(\boldsymbol{C})$ is also closed under isomorphism). □

As shown by the above proof, the basic idea underlying the construction is to make a "copy" (by σ_c) of a given Σ-algebra A, to add a Σ-algebra generated by Σ and X_{In}, and to make sure that this added algebra is in fact isomorphic to $A[X_{\text{In}}]$ (using the $I[s]$ function symbols axiomatized in a suitable way). The whole construction and proof would be simpler if we would have made a "plain copy" of Σ, but in practice it is obviously "more efficient" to restrict the copy to the non-input sorts. We can now derive the following theorem:

Theorem 9.4 (Reducing partial observational equalities to total ones) *Let $\Sigma = (S, F)$ be a signature,* Obs $\subseteq S$ *be a set of observable sorts and* In $\subseteq S$ *be a set of input sorts such that Σ is sensible w.r.t.* In. *Let $\boldsymbol{C}$ be a class of Σ-algebras closed under isomorphism and let ϕ be a Σ-formula.*

(i) $\boldsymbol{C} \models_{\approx_{\text{Obs},\text{In}}} \phi$ *if and only if* $(\sigma_c(\boldsymbol{C}) + \text{RESTRICT}[\Sigma,\text{In}])|_\Sigma \models_{\approx_{\text{Obs},S}} \phi$.
(ii) In particular, if $\boldsymbol{C}$ is the model class of some ASL-like structured specification SP*, we have:*
SP $\models_{\approx_{\text{Obs},\text{In}}} \phi$ *if and only if*
(**rename** SP **by** $\sigma_c + \text{RESTRICT}[\Sigma,\text{In}])|_\Sigma \models_{\approx_{\text{Obs},S}} \phi$.

Proof. Follows directly from Propositions 9.1 and 9.3. □

Hence, when we consider a partial observational equality $\approx_{\text{Obs},\text{In}}$, we can first use the reduction provided by Theorem 9.4, and then we have to find a finitary axiomatization of the total observational equality $\approx_{\text{Obs},S}$, but now with respect to the model class of (**rename** SP **by** σ_c + RESTRICT$[\Sigma,\text{In}])|_\Sigma$ instead of the model class of SP. In particular, this means that applying the method described in Section 8 will be simpler, since there will be no need for an hidden part to encode the definition domain of the partial observational equality by predicates $D_{H,s}$ (hence it could be the case that no hidden part at all will be necessary, provided we can select an appropriate finite set of observable contexts). The expression (**rename** SP **by** σ_c + RESTRICT$[\Sigma,\text{In}])|_\Sigma$ does not denote a smooth specification as defined in Section 2, but this is not a problem at all since it is well-known how to handle standard proofs with respect to such structured specifications.

Example 9.5 Let us consider again the `LIST` example studied in Section 8.4. Remember that we are interested in the behavioural theory of the specification `LIST`$|_{\Sigma-\text{Set}}$, w.r.t. the partial observational equality $\approx_{\text{Obs},\text{Obs}}$ induced on Σ-`Set`-algebras by Obs = {`Elem, Bool`}. The only non-input sort is `List`.

In a first step we apply the transformation suggested by Theorem 9.4, and

we must therefore consider the total observational equality $\approx_{\mathrm{Obs},S}$ w.r.t. the specification LIST-RESTR defined by:
LIST-RESTR $\stackrel{\mathrm{def}}{=}$ (**rename** LIST$|_{\Sigma-\mathrm{Set}}$ **by** σ_c+RESTRICT[Σ-Set, List])$|_{\Sigma-\mathrm{Set}}$,
where the renaming σ_c and the specification RESTRICT[Σ-Set, List] are defined as in Definition 9.2 (in particular, we introduce a new sort c[List], and new function symbols c[empty], c[$\in$], c[insert] and c[remove], and also a function symbol I[List]: List $\rightarrow$ c[List] with a suitable axiomatization).

As in Section 8.4, we select the set of "crucial contexts":
$\mathcal{C} = \{\mathtt{e} \in z_{\mathrm{List}}, z_{\mathrm{Bool}}, z_{\mathrm{Elem}}\}$.

The corresponding formula TBeh$_{\mathrm{List}}$(L,L') is (cf. Corollary 7.6):
[$\forall$e:Elem. e $\in$ L = e $\in$ L'] .

The corresponding axiomatization is the sentence AXL[TBeh$_{\mathrm{List}}$]:
$\forall$ L,L':List. ([$\forall$e:Elem. e $\in$ L = e $\in$ L'] $\Leftrightarrow$ L $\sim_{\mathrm{List}}$ L') .

The formula CONG$^{\sim}_{\Sigma-\mathrm{Set}}$ is:
$\forall$ e:Elem, L,L':List.
    [(L $\sim_{\mathrm{List}}$ L') $\Rightarrow$
        insert(e,L) $\sim_{\mathrm{List}}$ insert(e,L') $\wedge$
        remove(e,L) $\sim_{\mathrm{List}}$ remove(e,L')]

Let LIST-RESTR$^{\sim}$ $\stackrel{\mathrm{def}}{=}$ (LIST-RESTR + $\langle \Sigma\text{-Set} \cup \{\sim_{\mathrm{List}}\}, \mathrm{AXL}[\mathrm{TBeh}_{\mathrm{List}}]\rangle$) .

In a first step we must check that LIST-RESTR$^{\sim}$ $\models$ CONG$^{\sim}_{\Sigma-\mathrm{Set}}$.
This is not difficult, due to the axiomatization of I[List]. (Note that, for instance, we have among the axioms on I[List]: e $\in$ L = e c[$\in$] I[List](L); moreover, due to the reachability constraint $\mathcal{R}_{\mathrm{RESTR}}$, we can reason by structural induction w.r.t. the constructors of the List sort, i.e. w.r.t. empty, insert and remove, even if the constructors of the c[List] sort are empty and cons.)

Then, using Theorem 9.4, Corollary 7.6 and Theorem 6.3, we know that, for any Σ-Set-formula ϕ, LIST$|_{\Sigma-\mathrm{Set}}$ $\models_{\approx_{\mathrm{Obs},\mathrm{Obs}}}$ ϕ if and only if LIST-RESTR$^{\sim}$ $\models$ $\mathcal{L}(\phi)$.[12]

For instance, to prove that:
LIST$|_{\Sigma-\mathrm{Set}}$ $\models_{\approx_{\mathrm{Obs},\mathrm{Obs}}}$ insert(e,insert(e',L)) = insert(e',insert(e,L))
and that:

[12] Note that here, in contrast with Section 8.4, we have first reduced the partial observational equality to a total one. This means in particular that we can apply to the lifting of formulas the simplifications explained in Remark 8.1, i.e. for any Σ-Set-formula ϕ, the lifted formula $\mathcal{L}(\phi)$ is equivalent to the formula ϕ where all equations $l = r$ between terms of sort List are replaced by $l \sim_{\mathrm{List}} r$.

$\texttt{LIST}|_{\Sigma-\mathrm{Set}} \models_{\approx_{\mathrm{Obs,Obs}}} \texttt{e} \in \texttt{remove(e,L) = false}$,
it is equivalent to prove that:
$\texttt{LIST-RESTR}^{\sim} \models \texttt{insert(e,insert(e',L))} \sim_{\mathrm{List}} \texttt{insert(e',insert(e,L))}$
and that:
$\texttt{LIST-RESTR}^{\sim} \models \texttt{e} \in \texttt{remove(e,L) = false}$,
which is again not difficult.

10 Behavioural theories of behavioural and abstractor specifications

All results developed so far were obtained for arbitrary structured specifications, and we have not yet considered any interaction between behavioural equalities and the ASL-like specification language used to express specifications. In practice, however, behavioural equalities are not only used to define a behavioural satisfaction relation, but they can also be used to enrich the given specification language by "behavioural specifications". Behavioural specifications are a generalization of standard specifications which allow us to describe the behaviour of data structures and programs with respect to a given behavioural equality. The essential difference to the standard case is that instead of considering the usual models of a specification, one considers all algebras the behaviour of which is a model of the given underlying specification. More precisely, we define behavioural specifications as follows.

Definition 10.1 (Behavioural specification) *Let* SP *be an arbitrary ASL-like structured specification with signature Σ, and let $\approx$ be a Σ-behavioural equality.*

(i) The expression **behaviour** SP **w.r.t.** $\approx$ *is a behavioural specification, with signature Σ.*
(ii) The model class of a behavioural specification is defined by:
$\mathrm{Mod}(\textbf{behaviour}\ \mathrm{SP}\ \textbf{w.r.t.}\ \approx) \stackrel{\mathrm{def}}{=} \mathrm{Beh}_{\approx}(\mathrm{Mod}(\mathrm{SP}))$,
where, for any class $\boldsymbol{C}$ of Σ-algebras, the behaviour operator $\mathrm{Beh}_{\approx}$ *is defined by* $\mathrm{Beh}_{\approx}(\boldsymbol{C}) \stackrel{\mathrm{def}}{=} \{A \in \mathrm{Alg}(\Sigma) \mid A/{\approx_A} \in \boldsymbol{C}\}$.
(iii) A class $\boldsymbol{C}$ of Σ-algebras is called behaviourally closed (w.r.t. $\approx$) if $\boldsymbol{C} \subseteq$ $\mathrm{Beh}_{\approx}(\boldsymbol{C})$ *(or, equivalently, if $\boldsymbol{C}/{\approx} \subseteq \boldsymbol{C}$). In particular, the specification* SP *is behaviourally closed (w.r.t. $\approx$) if* $\mathrm{Mod}(\mathrm{SP}) \subseteq \mathrm{Beh}_{\approx}(\mathrm{Mod}(\mathrm{SP}))$ *(or, equivalently, if* $\mathrm{Mod}(\mathrm{SP})/{\approx} \subseteq \mathrm{Mod}(\mathrm{SP})$*).*

Remark 10.2 An algebra A is a model of a behavioural specification **behaviour** SP **w.r.t.** $\approx$ if and only if its behaviour $A/{\approx_A}$ (cf. Definition 3.1) is a model (in the standard sense) of the underlying specification SP. In particular, from Theorem 3.4 we can conclude that if SP is a basic specification $\langle \Sigma, \mathcal{A}x \rangle$, then a Σ-algebra A is a model of **behaviour** SP **w.r.t.** $\approx$ if and only if the axioms of SP are behaviourally satisfied by A, i.e. $A \models_{\approx} \mathcal{A}x$. An algebraic specification SP is behaviourally closed if the behaviours of all models of SP are also models of SP. If this is not the case, this means that there is some "inconsistency" between the properties required by the specification SP (e.g. its axioms) and the chosen behavioural equality (cf. e.g. [7]). [13]

Another approach which allows us to relax the standard semantics of algebraic specifications are "abstractor specifications" (cf. [25,24]). In this case an equivalence relation between algebras is used for abstracting from the (standard) model class of a specification. Abstractor specifications are defined as follows.

Definition 10.3 (Abstractor specification) *Let* SP *be an arbitrary ASL-like structured specification with signature Σ, and let $\equiv$ be an arbitrary equivalence relation on* $\mathrm{Alg}(\Sigma)$.

(i) *The expression* **abstract** SP **w.r.t.** $\equiv$ *is an abstractor specification, with signature Σ.*
(ii) *The model class of an abstractor specification is defined by:*
$\mathrm{Mod}(\textbf{abstract}\ \mathrm{SP}\ \textbf{w.r.t.} \equiv) \stackrel{\mathrm{def}}{=} \mathrm{Abs}_{\equiv}(\mathrm{Mod}(\mathrm{SP}))$,
where, for any class $\boldsymbol{C}$ of Σ-algebras, $\mathrm{Abs}_{\equiv}(\boldsymbol{C})$ *denotes the closure of $\boldsymbol{C}$ under $\equiv$, i.e.* $\mathrm{Abs}_{\equiv}(\boldsymbol{C}) \stackrel{\mathrm{def}}{=} \{B \in \mathrm{Alg}(\Sigma) \mid B \equiv A \textit{ for some } A \in \boldsymbol{C}\}$.

Behavioural specifications and abstractor specifications are based on the same intention, namely to provide a more general view of the semantics of specifications. This proves to be especially useful to define implementation relations where implementations may relax (some of) the properties of the given requirement specification (cf. e.g. abstractor implementations in [24] or behavioural implementations in [13,6] and similarly in [17]; for a survey on implementation concepts see [20]). [7] and [8] provide an in-depth study of the relationships between behavioural and abstractor specifications.

The underlying idea of the behavioural approach is to consider the behaviour $A/{\approx_A}$ of any Σ-algebra A w.r.t. a given Σ-behavioural equality $\approx$, while the abstractor approach is based on an equivalence relation $\equiv$ between Σ-algebras, where intuitively two algebras A and B are equivalent if they have the same behaviour. Hence to relate both approaches, an obvious preliminary requirement is that in both cases "behaviour" has the same meaning:

Definition 10.4 (Factorizability) *Given a signature Σ, a Σ-behavioural equality $\approx$ and an equivalence relation $\equiv$ between Σ-algebras, we say that the equivalence $\equiv$ is factorizable by $\approx$ if for all Σ-algebras A and B, $A \equiv B$ if and only if $A/{\approx_A}$ and $B/{\approx_B}$ are isomorphic.*

[13] "Behaviourally closed" was called "behaviourally consistent" in [7], but the terminology proposed here seems more adequate.

Example 10.5 (Observational equivalences) In Section 3.2 we have seen how to define observational equalities. Given a signature $\Sigma = (S, F)$, we can also define *observational equivalences* $\equiv_{\mathrm{Obs,In}}$ associated to a choice of input sorts $\mathrm{In} \subseteq S$ and of observable sorts $\mathrm{Obs} \subseteq S$ as follows:
Two Σ-algebras A and B are called observationally equivalent w.r.t. Obs and In, denoted by $A \equiv_{\mathrm{Obs,In}} B$, if there exists an S-sorted family of variables Y_{In} with $(Y_{\mathrm{In}})_s = \emptyset$ for all $s \in S \setminus \mathrm{In}$ and two valuations $\alpha_A : Y_{\mathrm{In}} \to A$ and $\alpha_B : Y_{\mathrm{In}} \to B$ with surjective mappings $\alpha_{A,s} : (Y_{\mathrm{In}})_s \to A_s$ and $\alpha_{B,s} : (Y_{\mathrm{In}})_s \to B_s$ for all $s \in \mathrm{In}$ such that for all terms $l, r \in T_\Sigma(Y_{\mathrm{In}})_s$ of observable sort $s \in \mathrm{Obs}$ the following holds: $I_{\alpha_A}(l) = I_{\alpha_A}(r)$ if and only if $I_{\alpha_B}(l) = I_{\alpha_B}(r)$.
We refer the reader to [7] where a more detailed discussion of observational equivalences is provided as well as a proof of the fact that the observational equivalence $\equiv_{\mathrm{Obs,In}}$ is factorizable by the observational equality $\approx_{\mathrm{Obs,In}}$.

The factorizability of the equivalence $\equiv$ by the behavioural equality $\approx$ is the technical condition under which meaningful relationships between behavioural and abstractor specifications can be established (cf. [7]). Fully abstract algebras play also an important rôle for the study of these relationships. Following Milner's notion (cf. [18]), we define full abstractness with respect to a given behavioural equality as follows:

Definition 10.6 (Fully abstract algebra and regularity) *Let Σ be a signature and $\approx$ be a Σ-behavioural equality.*

(i) *A Σ-algebra A is called fully abstract with respect to $\approx$ (or briefly fully abstract) if $\approx_A$ coincides with the set-theoretic equality on the carrier sets of A. (In particular $\approx_A$ is total.)*
(ii) *For any class $\boldsymbol{C}$ of Σ-algebras, $\mathrm{FA}_\approx(\boldsymbol{C})$ denotes the subclass of the fully abstract algebras of $\boldsymbol{C}$, i.e. $\mathrm{FA}_\approx(\boldsymbol{C}) \overset{\text{def}}{=} \{A \in \boldsymbol{C} \mid A \text{ is fully abstract}\}$.*
(iii) *The behavioural equality $\approx$ is called regular if, for any Σ-algebra A, its behaviour $A/\approx_A$ is fully abstract.*

Example 10.7 Let us consider again the `CONTAINER` specification described in Examples 2.2 and 3.12. We still assume that the sorts `Elem`, `Nat` and `Bool` are observable, and that all sorts are input sorts, and we consider the associated observational equality. In that case, the algebra of finite sequences of elements, which is a model of `CONTAINER`, is not a fully abstract algebra, since two distinct sequences may be observationally equal. The algebra of finite sets of elements, which is also a model of `CONTAINER`, is fully abstract.

We will show at the end of this section that observational equalities are always regular (see also [7] for a direct proof and [8] for more general properties of observational equalities).

We will now focus on behavioural theories of behavioural and abstractor specifications. Let us first recall a result established in [7].

Theorem 10.8 (Behavioural theories of behavioural and abstractor specifications [7, Section 7]) *Let Σ be a signature, $\approx$ be a regular Σ-behavioural equality, and $\equiv$ be an equivalence relation between Σ-algebras factorizable by $\approx$. For any class $\boldsymbol{C}$ of Σ-algebras, we have:*

(i) $\mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(\boldsymbol{C})) = \mathrm{Th}_{\approx}(\boldsymbol{C})$.
(ii) $\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(\boldsymbol{C})) = \mathrm{Th}(\mathrm{FA}_{\approx}(\boldsymbol{C}))$.
(iii) If $\boldsymbol{C}$ is closed under isomorphism and behaviourally closed w.r.t. $\approx$, then $\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(\boldsymbol{C})) = \mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(\boldsymbol{C})) = \mathrm{Th}_{\approx}(\boldsymbol{C}) = \mathrm{Th}(\mathrm{FA}_{\approx}(\boldsymbol{C}))$.
(iv) In particular, if SP *is an ASL-like structured specification with signature Σ, we have:*
  (a) $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{abstract}\ \mathrm{SP}\ \textbf{w.r.t.}\ \equiv)) = \mathrm{Th}_{\approx}(\mathrm{Mod}(\mathrm{SP}))$.
  (b) $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{behaviour}\ \mathrm{SP}\ \textbf{w.r.t.}\ \approx)) = \mathrm{Th}(\mathrm{FA}_{\approx}(\mathrm{Mod}(\mathrm{SP})))$.
  (c) If Mod(SP) *is closed under isomorphism and if* SP *is behaviourally closed w.r.t. $\approx$, then:*
  $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{behaviour}\ \mathrm{SP}\ \textbf{w.r.t.}\ \approx)) =$
  $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{abstract}\ \mathrm{SP}\ \textbf{w.r.t.}\ \equiv)) =$
  $\mathrm{Th}_{\approx}(\mathrm{Mod}(\mathrm{SP})) = \mathrm{Th}(\mathrm{FA}_{\approx}(\mathrm{Mod}(\mathrm{SP})))$.

Theorem 10.8 has various important consequences. Let us assume in the following discussion that the ASL-like specification language has been enriched by behavioural and abstractor specifications as shown in Definitions 10.1 and 10.3, and assume as well that the behavioural equality is regular and that the equivalence between algebras is factorizable by the behavioural equality. Moreover, let us assume that all ASL-like structured specifications have model classes closed under isomorphism, as it is usually the case. [14] Then from Theorem 10.8 we derive the following proof methods:

1. If we want to prove that some formula ϕ is behaviourally satisfied by an abstractor specification **abstract** SP **w.r.t.** $\equiv$, it is equivalent to prove that the same formula ϕ is behaviourally satisfied by SP. In particular, we do not directly apply Theorems 5.5 or 6.3 to abstractor specifications, but we first use the reduction provided by Theorem 10.8*(iv)(a)*. From a practical point of view, this means that we must find a finitary axiomatization of the behavioural equality **not** w.r.t. the model class of the

[14] Note that for behavioural and abstractor specifications, this is guaranteed if the behavioural equality is "isomorphism compatible" (i.e. if two isomorphic algebras have isomorphic behaviours) and if the equivalence between algebras is "isomorphism protecting" (i.e. if two isomorphic algebras are equivalent). Under the assumption of factorizability, these two properties are equivalent. Observational equalities and observational equivalences are respectively isomorphism compatible and isomorphism protecting, cf. [7].

abstractor specification, **but only** w.r.t. the (in general much smaller) model class of the underlying specification SP.

2. If we want to prove that some formula ϕ is behaviourally satisfied by a behavioural specification **behaviour** SP **w.r.t.** $\approx$, it is equivalent to prove that the same formula ϕ is satisfied (in the standard sense) by the fully abstract models of the underlying specification SP. In particular, it would be sufficient to prove that this formula is satisfied in the standard sense by SP.
3. If the specification SP is behaviourally closed,[15] then to prove that a formula ϕ is behaviourally satisfied by either SP or **abstract** SP **w.r.t.** $\equiv$ or **behaviour** SP **w.r.t.** $\approx$, it is equivalent to prove that the same formula ϕ is satisfied (in the standard sense) by the fully abstract models of SP. Again, in particular it would be sufficient to prove that this formula is satisfied in the standard sense by SP.

The above discussion points out the importance of fully abstract algebras, and more precisely of the standard theory of the fully abstract models of a given specification SP (or of a given class $\boldsymbol{C}$ of algebras). Indeed, in Section 4 we have shown how to reduce the behavioural theory of a class $\boldsymbol{C}$ of algebras to (a subset of) the standard theory of the lifted class $\mathcal{L}(\boldsymbol{C})$ (cf. Theorem 4.2), and then in the following sections we have studied how to use this reduction once an axiomatization Beh of the behavioural equality is provided: the basic idea was "to characterize lifted algebras" by the sentence AXL[Beh]. Similarly, our aim is now to show how to derive "a characterization of full abstractness" (by some sentence FA[Beh]) from an axiomatization Beh of the behavioural equality. For this purpose we define the following sentence:

Definition 10.9 *Let $\Sigma = (S, F)$ and Σ_1 be two signatures such that $\Sigma \subseteq \Sigma_1$, and let $\phi = (\phi_s)_{s \in S}$ be an S-sorted family of Σ_1-formulas, where each formula ϕ_s has exactly two free variables x_s and y_s (both of sort $s \in S$). Let* FA[ϕ] *be the Σ_1-sentence defined by:*
$\mathrm{FA}[\phi] \stackrel{\mathrm{def}}{=} \bigwedge_{s \in S} \; \forall x_s, y_s{:}s.\, [\phi_s \Leftrightarrow x_s = y_s]\,.$

Proposition 10.10 (Characterization of fully abstract algebras) *Let $\Sigma = (S, F)$ be a signature, $\approx$ be a Σ-behavioural equality and let $\boldsymbol{C}$ be a class of Σ-algebras. Let* Beh *be a (possibly infinitary) axiomatization of the Σ-behavioural equality $\approx$, and let* (HID, HBeh) *be a finitary axiomatization with hidden part of the Σ-behavioural equality $\approx$ with respect to the given class $\boldsymbol{C}$ of Σ-algebras. Let Σ_H be the signature of* HID*, and let* FA[Beh] *and* FA[HBeh] *be the sentences induced by* Beh *and* HBeh *respectively (cf. Definition 10.9).*

[15] See [7] and [6] for sufficient conditions. In particular, basic specifications where axioms are restricted to conditional equations with observable premisses are behaviourally closed w.r.t. any total observational equality, cf. [22].

(i) *A Σ-algebra A is fully abstract (w.r.t. $\approx$) if and only if* $A \models \mathrm{FA[Beh]}$,
hence $\mathrm{FA}_{\approx}(\boldsymbol{C}) = (\boldsymbol{C} + \langle \Sigma, \mathrm{FA[Beh]} \rangle)$.

(ii) *A Σ-algebra $A \in \boldsymbol{C}$ is fully abstract (w.r.t. $\approx$) if and only if*
$(\{A\} + \mathrm{HID}) \models \mathrm{FA[HBeh]}$,
hence $\mathrm{FA}_{\approx}(\boldsymbol{C}) = (\boldsymbol{C} + \mathrm{HID} + \langle \Sigma_H, \mathrm{FA[HBeh]} \rangle)|_{\Sigma}$.

Proof. *(i)* follows directly from the definition of fully abstract algebras and Proposition 5.3; *(ii)* is derived in a similar way using Remark 6.2. □

This characterization of fully abstract algebras leads directly to the following theorem.

Theorem 10.11 *Let Σ be a signature, $\approx$ be a regular Σ-behavioural equality, and $\equiv$ be an equivalence relation between Σ-algebras factorizable by $\approx$. Let $\boldsymbol{C}$ be a class of Σ-algebras. Let* Beh *be a (possibly infinitary) axiomatization of the Σ-behavioural equality $\approx$, and let* (HID, HBeh) *be a finitary axiomatization with hidden part of the Σ-behavioural equality $\approx$ with respect to the given class $\boldsymbol{C}$ of Σ-algebras. Let Σ_H be the signature of* HID*, and let* FA[Beh] *and* FA[HBeh] *be the sentences induced by* Beh *and* HBeh *respectively (cf. Definition 10.9).*

(i) $\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(\boldsymbol{C})) = \mathrm{Th}(\boldsymbol{C} + \langle \Sigma, \mathrm{FA[Beh]} \rangle) =$
$\mathrm{Th}((\boldsymbol{C} + \mathrm{HID} + \langle \Sigma_H, \mathrm{FA[HBeh]} \rangle)|_{\Sigma})$.

(ii) *If $\boldsymbol{C}$ is closed under isomorphism and behaviourally closed w.r.t. $\approx$, then* $\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(\boldsymbol{C})) = \mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(\boldsymbol{C})) = \mathrm{Th}_{\approx}(\boldsymbol{C}) = \mathrm{Th}(\boldsymbol{C} + \langle \Sigma, \mathrm{FA[Beh]} \rangle) =$
$\mathrm{Th}((\boldsymbol{C} + \mathrm{HID} + \langle \Sigma_H, \mathrm{FA[HBeh]} \rangle)|_{\Sigma})$.

(iii) *In particular, if $\boldsymbol{C}$ is the model class of some ASL-like structured specification* SP*, we have:*

  (a) $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{behaviour } \mathrm{SP} \textbf{ w.r.t. } \approx)) = \mathrm{Th}(\mathrm{SP} + \langle \Sigma, \mathrm{FA[Beh]} \rangle) =$
  $\mathrm{Th}((\mathrm{SP} + \mathrm{HID} + \langle \Sigma_H, \mathrm{FA[HBeh]} \rangle)|_{\Sigma})$.

  (b) *If moreover* Mod(SP) *is closed under isomorphism and if* SP *is behaviourally closed w.r.t. $\approx$, then:*
  $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{behaviour } \mathrm{SP} \textbf{ w.r.t. } \approx)) =$
  $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{abstract } \mathrm{SP} \textbf{ w.r.t. } \equiv)) =$
  $\mathrm{Th}_{\approx}(\mathrm{Mod}(\mathrm{SP})) =$
  $\mathrm{Th}(\mathrm{SP} + \langle \Sigma, \mathrm{FA[Beh]} \rangle) = \mathrm{Th}((\mathrm{SP} + \mathrm{HID} + \langle \Sigma_H, \mathrm{FA[HBeh]} \rangle)|_{\Sigma})$.

Proof. Follows directly from Theorem 10.8 and Proposition 10.10. □

Hence, as soon as we have found a finitary axiomatization (HID, HBeh) of a behavioural equality $\approx$ with respect to the model class of a specification SP (e.g. following the method described in Section 8 for observational equalities), we obtain the corresponding finitary sentences AXL[HBeh] and FA[HBeh], and then we can use "standard" proof techniques to prove the behavioural validity of formulas with respect to either SP, or the behavioural or abstractor specifications built on top of SP. In the case of a behavioural specification, or when the specification SP is known to be behaviourally closed, then we can use the sentence FA[HBeh] instead of the sentence AXL[HBeh].

Example 10.12 Let us consider again the `CONTAINER` specification studied in Section 8.2. It is not difficult to show that this specification is behaviourally closed with respect to the total observational equality induced by the observable sorts `Elem`, `Nat` and `Bool`. Hence we proceed as in Section 8.2, but instead of using the sentence AXL[$\mathrm{TBeh}_{\mathrm{Cont}}$], we use the sentence FA[$\mathrm{TBeh}_{\mathrm{Cont}}$] defined by $\forall$ `S,S':Container. ([`$\forall$`e:Elem. e` $\in$ `S = e` $\in$ `S'] ` $\Leftrightarrow$ ` S = S').` Now let `CONTAINER-FA` $\stackrel{\text{def}}{=}$ (`CONTAINER` $+ \langle \mathrm{Sig}($`CONTAINER`$), \mathrm{FA}[\mathrm{TBeh}_{\mathrm{Cont}}]\rangle)$. To prove that the two characteristic equations of sets are behaviourally valid in the model class of the `CONTAINER` specification (or in the model class of the corresponding behavioural or abstractor specifications), it is equivalent to prove that `CONTAINER-FA` $\models$ `insert(e,insert(e,S)) = insert(e,S)` and that `CONTAINER-FA` $\models$ `insert(e,insert(e',S)) = insert(e',insert(e,S))`.

In the rest of this section we study a last application of axiomatizations of behavioural equalities, namely we point out that, whenever an axiomatization of a behavioural equality $\approx$ is provided, one can characterize the regularity of $\approx$ by a certain property (called invariance) of this axiomatization.

Definition 10.13 (Invariant formula) *Let Σ be a signature and $\approx$ be a Σ-behavioural equality. A Σ-formula ϕ is called invariant w.r.t. $\approx$ if, for any Σ-algebra A and valuation $\alpha : X \to \mathrm{Dom}(\approx_A)$, $A, \alpha \models_{\approx} \phi$ if and only if $A, \alpha \models \phi$.*

Remark 10.14 If ϕ is closed then, due to Theorem 4.2, ϕ is invariant if and only if ϕ and $\mathcal{L}(\phi)$ are equivalent formulas w.r.t. the class of all lifted algebras $\mathcal{L}(A)$, for $A \in \mathrm{Alg}(\Sigma)$. In particular, in the special case of partial observational equalities induced by a set Obs of observable sorts and by a set In of input sorts, sentences built with the logical connectives, quantifiers and from equations between observable terms (i.e. terms of an observable sort) that contain only variables of input sorts are invariant formulas.

Proposition 10.15 *Let Σ be a signature and* Beh *be an axiomatization of a Σ-behavioural equality $\approx$. The behavioural equality $\approx$ is regular if and only if its axiomatization* Beh *is invariant.*

Proof. From Proposition 5.3 we know that, since Beh is an axiomatization of $\approx$, for any Σ-algebra A, any sort $s \in S$ and any valuation $\alpha : X \to A$, $A, \alpha \models \mathrm{Beh}_s(x_s, y_s)$ iff $\alpha(x_s) \approx_A \alpha(y_s)$. In particular, for any valuation $\alpha : X \to \mathrm{Dom}(\approx_A)$, $A, \alpha \models \mathrm{Beh}_s(x_s, y_s)$ iff $A, \alpha \models_{\approx} x_s = y_s$. Hence the axiomatization Beh of $\approx$ is invariant if and only if, for any Σ-algebra A and any valuation $\alpha : X \to \mathrm{Dom}(\approx_A)$, $A, \alpha \models_{\approx} \mathrm{Beh}_s(x_s, y_s)$ iff $A, \alpha \models \mathrm{Beh}_s(x_s, y_s)$ iff $A, \alpha \models_{\approx} x_s = y_s$. From this we conclude that the axiomatization Beh of

$\approx$ is invariant if and only if, for any Σ-algebra A, $A \models_{\approx}$ FA[Beh]. Using Theorem 3.4*(i)*, we conclude that the axiomatization Beh is invariant if and only if, for any Σ-algebra A, $A/{\approx_A} \models$ FA[Beh], i.e., using Proposition 10.10, if and only if, for any Σ-algebra A, $A/{\approx_A}$ is fully abstract, which is the definition of regularity. $\square$

Using Proposition 10.15, it is easy to show that the observational equality $\approx_{\mathrm{Obs,In}}$ induced by a set Obs of observable sorts and a set In of input sorts is always regular (cf. Example 5.4).

11 Conclusion

In the literature several approaches formalize behavioural correctness concepts by introducing some kind of behavioural semantics (cf. e.g. [12,21,19,23,2,20]). 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 [13] or the correspondence relation in [26]). In this paper we have developed a proof theoretical framework for checking the behavioural validity of arbitrary first-order formulas. We have shown that in the case of partial observational equalities this framework leads to a method that allows us to prove observational theorems using any arbitrary theorem prover for standard first-order logic. For concrete examples we have successfully proved observational theorems with the Larch Prover LP (cf. [11]).

Recently, it has been shown in [15] that our results can be extended to a higher-order logical framework. Note that in a higher-order logical framework it is always easy to find a finitary (higher-order) axiomatization of an observational equality, but then the problem is how to handle the complexity of the higher-order axiomatization when performing proofs. Note also that our approach is completely different from the one used in [14], where sound and complete proof systems with *infinitary proof rules* are given for structured specifications with observability operators.

One of the main ideas for obtaining the results described in this paper was to use the fact that an observational equality is the greatest congruence relation which "coincides" with the set-theoretic equality for observable sorts. In [17], a similar idea is used, in the framework of *hidden order sorted algebras*, to derive efficient techniques for proving the correctness of refinements. However, in [17], the idea merely leads to an optimization of the context induction principle of [13], using a distinction between "generators" and "defined functions" (moreover in the hidden order sorted algebra framework, axioms are restricted to equations, and only total observational equalities are considered). Hence an essential difference between the technique proposed in [17] and the method described in this paper is the following one. With our method, context induction is entirely "subsumed" by the proof, once for all, of the "congruence property" (cf. Step 5(b) of the method described in Section 8), and then the proof of the behavioural validity of a formula ϕ amounts to the proof of the standard validity of the lifted formula $\mathcal{L}(\phi)$ (w.r.t. an enriched specification). With the method described in [17], one still has to prove the behavioural validity of each formula by context induction, but optimized in the sense that not all observable contexts have to be considered, but only the observable contexts built with the so-called "defined functions". It is however interesting to note that the sets of observable contexts considered in the examples given in [17] are (not surprisingly) essentially the same as the sets of observable contexts we select in the first step of our method for the same examples.

The most important application of our proof technique is the verification of the correctness of behavioural implementations. Using the results developed in this paper, in [6] we have investigated proof rules that allow us to establish, in a modular way, the correctness of behavioural implementations of structured specifications. According to the relationships between behavioural specifications and abstractor specifications (for "factorizable" equivalences, cf. Section 10 and [7]), these proof rules are also correct for abstractor implementations in the sense of [24]. As a consequence of our results it is an objective of future work to built an environment for proving behavioural theorems and the correctness of behavioural implementations on top of some existing theorem prover such as e.g. the Larch Prover.

Acknowledgements: This work is partially supported by the French-German cooperation programme PROCOPE and by the ESPRIT Basic Research Working Group 6112 COMPASS-II. We would like to thank for stimulating discussions and helpful comments Hubert Baumeister, Fernando Orejas, Don Sannella, Andrzej Tarlecki and Martin Wirsing. Preliminary results of parts of this work were developed in [3], [4] and [5].

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] M. Bidoit and R. Hennicker. How to prove observational theorems with LP.

In *Proc. of the First International Workshop on Larch*, pages 18–35. Springer-Verlag Workshops in Computing Series, 1993.

[4] M. Bidoit and R. Hennicker. Proving behavioural theorems with standard first-order logic. In *Proc. of ALP'94*, pages 41–58. Springer-Verlag L.N.C.S. 850, 1994.

[5] M. Bidoit and R. Hennicker. Behavioural theories. In *Recent Trends in Data Type Specification*, pages 153–169. Springer-Verlag L.N.C.S. 906, 1995.

[6] M. Bidoit and R. Hennicker. Modular correctness proofs of behavioural implementations. Available by WWW: http://www.pst.informatik.uni-muenchen.de/~hennicke/, 1995. A short version appeared as: Proving the correctness of behavioural implementations, in *Proc. of AMAST'95*, Springer-Verlag L.N.C.S. 936, pages 152–168, 1995.

[7] M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. *Science of Computer Programming*, 25(2–3), 1995. A short version appeared as: Characterizing behavioural semantics and abstractor semantics, in *Proc. of ESOP'94*, Springer-Verlag L.N.C.S. 788, pages 105–119, 1994.

[8] M. Bidoit and A. Tarlecki. Behavioural satisfaction and equivalence in concrete model categories. In *Proc. of CAAP'96*. Springer-Verlag L.N.C.S., 1996. To appear.

[9] 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.

[10] H. Ehrig and B. Mahr. *Fundamentals of algebraic specification 2*, volume 21 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1990.

[11] 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.

[12] 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.

[13] R. Hennicker. Context induction: a proof principle for behavioural abstractions and algebraic implementations. *Formal Aspects of Computing*, 3(4):326–345, 1991.

[14] R. Hennicker, M. Wirsing, and M. Bidoit. Proof systems for structured specifications with observability operators. *To appear in Theoretical Computer Science (special issue on the 10th WADT)*, 1996.

[15] M. Hofmann and D. Sannella. On behavioural abstraction and behavioural satisfaction in higher-order logic. In *Proc. of TAPSOFT'95*, pages 247–261. Springer-Verlag L.N.C.S. 915, 1995.

[16] G. Kreisel and J.L. Krivine. *Eléments de Logique Mathématique*. Dunod (Paris), 1967.

[17] G. Malcolm and J. Goguen. Proving correctness of refinement and implementation. Technical Monograph PRG–114, Oxford University Computing Laboratory, 1994.

[18] R. Milner. Fully abstract models of typed λ-calculi. *Theoretical Computer Science*, 4:1–22, 1977.

[19] 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.

[20] 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.

[21] H. Reichel. Initial restrictions of behaviour. In *Proc. of IFIP Working Conference, The Role of Abstract Models in Information Processing*, 1985.

[22] H. Reichel. Initial computability, algebraic specifications, and partial algebras. International Series of Monographs in Computer Science 2, Oxford: Clarendon Press, 1987.

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

[24] D. Sannella and A. Tarlecki. Toward formal development of programs from algebraic specification: implementation revisited. *Acta Informatica*, 25:233–281, 1988.

[25] D. Sannella and M. Wirsing. A kernel language for algebraic specifications and implementation. In *Proc. of FCT'83*, pages 413–427. Springer-Verlag L.N.C.S. 158, 1983.

[26] O. Schoett. Behavioural correctness of data representation. *Science of Computer Programming*, 14:43–57, 1990.

[27] J.W. Thatcher, E.G. Wagner, and J.B. Wright. Data type specification: parameterization and the power of specification techniques. *ACM Transactions on Programming Languages and Systems*, 4:711–732, 1982.

[28] M. Wirsing. *Algebraic specification*. Handbook of Theoretical Computer Science. Elsevier Science Publishers B. V., 1990.

[29] M. Wirsing. *Structured specifications: syntax, semantics and proof calculus*. Logic and Algebra of Specification, International Summer School Marktoberdorf 1991. Springer-Verlag, 1993.