# Behavioural Theories

Michel Bidoit[1] and Rolf Hennicker[2]

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

**Abstract.** Behavioural theories are a generalization of standard theories where the equality predicate is interpreted by an arbitrary (possibly partial) congruence relation, called *behavioural equality*. In this paper we first show how to reduce the behavioural theory of any class $C$ of $\Sigma$-algebras to (a subset of) the standard theory of some corresponding class of algebras. Then we consider infinitary axiomatizations of the behavioural equality and we define a general condition under which the infinitary axiomatization can be replaced by a finitary one. As a consequence, behavioural theorems over different kinds of specifications can be proved using standard proof techniques. All results are stated in an abstract way, i.e. independent of a particular behavioural equality. As an important example we can apply our results to observational theories where the behavioural equality is given by an observational equality between elements.

## 1 Introduction

Formal algebraic specifications provide an appropriate means for the development of correct software. Thereby an important role is played by correctness concepts which allow behavioural abstractions. The basic idea of these concepts is that a software product is regarded as a correct implementation of some given specification if it satisfies the intended behaviour independently of internal implementation details which may not satisfy all properties of the specification.

This idea has been formalized in several approaches in the literature (cf. e.g. [4, 9, 10, 8, 11]). For instance, in [10] a concrete specification $SP'$ is called *abstractor implementation* of a given specification $SP$ *via an abstractor* $\alpha$ (determined by an equivalence relation on the class of all $\Sigma$-algebras) if all models of $SP'$ are models of the *abstractor specification* $\alpha(SP)$, i.e. are equivalent to a model of $SP$. A different approach, but guided by the same motivation, are *behavioural specifications*. Thereby the key idea is to relax the interpretation of the equality predicate $=$ and to use a *behavioural satisfaction relation* where the equality predicate is interpreted by a *behavioural equality* determined by an arbitrary (possibly partial) congruence relation on any $\Sigma$-algebra. (An important example are *observational equalities* which relate any two elements of an algebra which cannot be distinguished by observable computations, cf. e.g. [9, 1, 8].)

For performing correctness proofs in the case of behavioural specifications 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 behavioural specification belong to the *behavioural theory* of the implementing specification where the behavioural theory of a class $C$ of $\Sigma$-algebras is the set of all formulas that are behaviourally satisfied by all algebras of $C$. Since we know from a result in [2] that behavioural semantics is in many cases the same but (at most) more restrictive than abstractor semantics (for "factorizable" abstractors) this proof strategy is also correct for abstractor implementations.

Unfortunately it is usually difficult to prove that a formula $\phi$ belongs to the behavioural theory of a given class $C$ of $\Sigma$-algebras. The behavioural satisfaction relation does not even fulfill 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. In [1] we have shown how this can be achieved in the particular case where $C$ is the model class of a behavioural specification and where the behavioural equality is a total observational equality. It is the aim of this work to obtain a similar but more general result for arbitrary classes $C$ of $\Sigma$-algebras (such that $C$ can be instantiated by the model class of different kinds of specifications as e.g. standard specifications, abstractor specifications, behavioural specifications, "ASL-like" specifications) and for arbitrary (possibly partial) behavioural equalities (thus avoiding technicalities like "observable contexts" which appear in the observational framework).

For this purpose we proceed as follows:

In a first step (Section 4) we provide a general construction (the so-called "lift operator") which allows us to characterize the behavioural theory of a class $C$ of $\Sigma$-algebras by (a subset of) the standard theory of a corresponding class $\mathcal{L}(C)$ of lifted algebras. More precisely, a formula $\phi$ is behaviourally valid in all algebras of $C$ if and only if its lifted version $\mathcal{L}(\phi)$ is valid in the standard sense in all lifted algebras of $\mathcal{L}(C)$. The usefulness of this characterization of behavioural theories still depends on the possibility to prove standard theorems over $\mathcal{L}(C)$.

Therefore we introduce in a next step (Section 5) a general notion of (infinitary) axiomatization of behavioural equalities and we show that such an axiomatization is useful for characterizing the class $\mathcal{L}(C)$ of lifted algebras. In particular, we see that if $C$ is axiomatizable (i.e. is the model class of a flat standard specification) and if an axiomatization of the behavioural equality is provided then $\mathcal{L}(C)$ is axiomatizable as well. However, we have still 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 last step (Section 6), it remains to get rid of such infinitary axiomatizations. For this purpose we consider conditional axiomatizations of the behavioural equality. The underlying idea is that if a class $C$ of $\Sigma$-algebras satisfies a certain condition (represented by a set of finitary formulas) then the infinitary axiomatization of the behavioural equality can be replaced by a finitary one. Combined with the previous reduction of behavioural theories over $C$ to standard theories over $\mathcal{L}(C)$ we can then prove behavioural theorems using standard proof techniques. As an example we consider in Section 7 the total observational equality and we show how one can reduce the observational theory of a multiset specification (which exports only a part of its operations) to a standard theory of another related specification.

## 2 Basic Notions

We assume that the reader is familiar with basic notions of algebraic specifications (cf. e.g. [3]) like the notions of (many sorted) *signature* $\Sigma = (S, F)$, *total* $\Sigma$*-algebra* $A = (\ (A_s)_{s\in S}, (f^A)_{f\in F}\ )$ where $A_s$ are the carrier sets of $A$ and $f^A$ are the (total) operations, $\Sigma$*-term algebra* $T_\Sigma(X)$, *valuation* $\alpha : X \to A$ and *interpretation* $I_\alpha : T_\Sigma(X) \to A$. Throughout this paper we assume that the carrier sets $A_s$ of a $\Sigma$-algebra $A$ are not empty and $Alg(\Sigma)$ denotes the class of all such $\Sigma$-algebras. We assume also that $X = (X_s)_{s\in S}$ is a family of countably infinite sets $X_s$ of variables of sort $s \in S$.

A *partial* $\Sigma$*-congruence* on a $\Sigma$-algebra $A$ is a family $\approx_A = (\approx_{A,s})_{s\in S}$ of partial equivalence (i.e. symmetric and transitive) relations $\approx_{A,s}$ on $A_s$ compatible with the signature $\Sigma$, 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 $f^A(a_1, \dots, a_n) \approx_{A,s} f^A(b_1, \dots, b_n)$.[3] A $\Sigma$*-congruence* $\approx_A$ is *total* if for all $a$ in $A$, $a \approx_A a$. 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 $\approx_A$ is a total $\Sigma$-congruence on $\mathrm{Dom}(\approx_A)$). In the sequel $A/\approx_A$ denotes the quotient algebra of $\mathrm{Dom}(\approx_A)$ by $\approx_A$.

*First-order* $\Sigma$*-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* $\Sigma$*-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 $\Sigma$-formulas. A $\Sigma$*-sentence* is a $\Sigma$-formula which contains no free variable. In the sequel $\mathrm{FreeVar}(\phi)$ denotes the set of the free variables of the formula $\phi$ and $\mathrm{Var}(t)$ denotes the variables of the term $t$.

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

A *standard (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 $\Sigma$-sentences, called *axioms* of $SP$. The *model class* of $SP$, denoted by $Mod(SP)$, is the class of all $\Sigma$-algebras which satisfy the axioms of $SP$, i.e. $Mod(SP) \stackrel{\text{def}}{=} \{A \in Alg(\Sigma) \mid A \models \phi \text{ for all } \phi \in \mathcal{A}x\}$.

A 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 $Mod(SP) \subseteq Alg(Sig(SP))$, and if the language contains standard specifications and (at least) an operator $+$ for the combination

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

of specifications $SP$ and $SP'$ such that $Sig(SP + SP') = Sig(SP) \cup Sig(SP')$ and $Mod(SP + SP') = \{A \in Alg(Sig(SP + SP')) \mid A|_{Sig(SP)} \in Mod(SP)$ and $A|_{Sig(SP')} \in Mod(SP')\}$.

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

Note that we will always consider theories including infinitary $\Sigma$-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) $\Sigma$-formulas.

## 3 Behavioural and Abstractor Specifications

In this section we summarize the basic definitions and results of [2] where the relationships between behavioural and abstractor specifications are studied.

Behavioural specifications are a generalization of standard specifications which allow to describe the behaviour of data structures and programs with respect to a given (partial) congruence relation. The essential difference to the standard case is that instead of the set-theoretic equality, the given congruence relation is used for the interpretation of the = predicate symbol.

Another approach which also allows to relax the standard semantics of algebraic specifications are abstractor specifications (cf. [10]). In this case an equivalence relation between algebras is used for abstracting from the (standard) model class of a specification.

In the sequel of this paper we always assume given a family $\approx\, = (\approx_A)_{A \in Alg(\Sigma)}$ of (possibly partial) $\Sigma$-congruences on the algebras of $Alg(\Sigma)$ and an equivalence relation $\equiv$ on $Alg(\Sigma)$ such that $\equiv$ is *factorizable* by $\approx$, i.e.:
For all $A, B \in Alg(\Sigma), A \equiv B$ if and only if $A/{\approx_A}$ and $B/{\approx_B}$ are isomorphic.

### 3.1 Behavioural and Abstractor Semantics

**Definition 1 (Behavioural satisfaction relation).** Let $A$ be a $\Sigma$-algebra. Moreover, let $l, r \in T_\Sigma(X)_s$ be two terms of sort $s$, $\phi$ be an arbitrary $\Sigma$-formula, and $\alpha : X \to \mathrm{Dom}(\approx_A)$ be a valuation. The behavioural satisfaction relation w.r.t. $\approx$, denoted by $\models_\approx$, is defined as follows:

1. $A, \alpha \models_\approx l = r$ if and only if $I_\alpha(l) \approx_A I_\alpha(r)$.
2. $A, \alpha \models_\approx \phi$ is defined by induction on the structure of the formula $\phi$.
3. $A \models_\approx \phi$ if and only if $A, \alpha \models_\approx \phi$, for all valuations $\alpha : X \to \mathrm{Dom}(\approx_A)$.

Hence Definition 1 is quite similar to the definition of the standard satisfaction relation, the only difference being for *(1)* 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 partial nature of $\approx_A$.

**Definition 2 (Behavioural specification).** Let $SP = \langle \Sigma, \mathcal{A}x \rangle$ be a standard specification. Then:

1. The expression **behaviour** $SP$ **w.r.t.** $\approx$ is a behavioural specification.
2. The model class of a behavioural specification is defined by
   $Mod(\textbf{behaviour } SP \textbf{ w.r.t. } \approx) \stackrel{\text{def}}{=} \{A \in Alg(\Sigma) \mid A \models_{\approx} \phi \text{ for all } \phi \in \mathcal{A}x\}$.

**Definition 3 (Abstractor operator).** For any class $C \subseteq Alg(\Sigma)$, $Abs_{\equiv}(C)$ denotes the closure of $C$ under $\equiv$, i.e.:
$Abs_{\equiv}(C) \stackrel{\text{def}}{=} \{B \in Alg(\Sigma) \mid B \equiv A \text{ for some } A \in C\}$.

**Definition 4 (Abstractor specification).** Let $SP = \langle \Sigma, \mathcal{A}x \rangle$ be a standard specification. Then:

1. The expression **abstract** $SP$ **w.r.t.** $\equiv$ is an abstractor specification.
2. The model class of an abstractor specification is defined by
   $Mod(\textbf{abstract } SP \textbf{ w.r.t. } \equiv) \stackrel{\text{def}}{=} Abs_{\equiv}(Mod(SP))$.

Fully abstract algebras play an important role for the characterization of behavioural semantics. Following Milner's notion (cf. [7]), we define full abstractness with respect to a given family $\approx$ of $\Sigma$-congruences as follows:

**Definition 5 (Fully abstract algebra).**

1. A $\Sigma$-algebra $A$ is called fully abstract with respect to $\approx$ (or briefly fully abstract) if $\approx_A$ coincides with the set-theoretic equality over the carrier sets of $A$. (In particular $\approx_A$ is total.)
2. For any class $C \subseteq Alg(\Sigma)$ of $\Sigma$-algebras, $FA_{\approx}(C)$ denotes the subclass of the fully abstract algebras of $C$, i.e. $FA_{\approx}(C) \stackrel{\text{def}}{=} \{A \in C \mid A \text{ is fully abstract}\}$.

**Definition 6 (Regularity).** A family $\approx = (\approx_A)_{A \in Alg(\Sigma)}$ of $\Sigma$-congruences is called regular if, for any $\Sigma$-algebra $A$, the quotient algebra $A/{\approx_A}$ is fully abstract.

**Definition 7 (Behavioural quotient operator).** For any class $C \subseteq Alg(\Sigma)$, $C/{\approx}$ denotes the behavioural quotient of $C$, i.e.: $C/{\approx} \stackrel{\text{def}}{=} \{A/{\approx_A} \mid A \in C\}$.

**Definition 8 (Behaviour operator).** For any class $C \subseteq Alg(\Sigma)$,
$Beh_{\approx}(C) \stackrel{\text{def}}{=} Abs_{\equiv}(FA_{\approx}(C))$.

A central result of [2] is the following characterization of behavioural semantics:

**Theorem 9.** *If the family $\approx$ is regular, then*
$Mod(\textbf{behaviour } SP \textbf{ w.r.t. } \approx) = Beh_{\approx}(Mod(SP))$.

*Example 1.* An important example in which all our results apply is the *partial observational equality* of elements as defined e.g. in [8].

Formally, 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. Let $X_{In}$ be the $S$-sorted family of variables defined by $(X_{In})_s = \emptyset$ if $s \notin In$ and $(X_{In})_s = X_s$ if $s \in In$ (where $X = (X_s)_{s \in S}$ is the generally assumed family of countably infinite sets of variables of sort $s$), and let $Z = (\{z_s\})_{s \in S}$ be a disjoint $S$-sorted family of singleton sets. Then:

1. An *observable $\Sigma$-context* is a $\Sigma$-term $C \in T_\Sigma(X_{In} \cup Z)$ of observable sort which contains besides input variables in $X_{In}$ exactly one distinguished variable $z_s$ (called *context variable*) of some sort $s \in S$.[4] The application of a context $C$ with context variable $z_s$ to a term $t$ of sort $s$ is denoted by $C[t]$. The set of observable contexts with context variable of sort $s$ and input variables in $In$ is denoted by $\mathcal{C}_s^{Obs,In}$.
2. If $A$ is a $\Sigma$-algebra then two elements $a, b \in A_s$ are observationally equal, denoted by $a \approx_{Obs,In,A} b$, if:
   (a) There exists a term $t \in T_\Sigma(X_{In})_s$ and a valuation $\alpha : X_{In} \to A$ such that $I_\alpha(t) = a$, and similarly for $b$.
   (b) For all contexts $C \in \mathcal{C}_s^{Obs,In}$ with context variable $z_s$, for all valuations $\alpha : X_{In} \to A$, we have $I_{\alpha_1}(C) = I_{\alpha_2}(C)$, where $\alpha_1, \alpha_2 : X_{In} \cup \{z_s\} \to A$ are the unique extensions of $\alpha$ defined by $\alpha_1(z_s) = a$ and $\alpha_2(z_s) = b$.
3. The family $(\approx_{Obs,In,A})_{A \in Alg(\Sigma)}$ of partial observational equalities (which in particular are partial $\Sigma$-congruences) is denoted by $\approx_{Obs,In}$. $\mathrm{Dom}(\approx_{Obs,In,A})$ is the smallest subalgebra of $A$ which is generated by the values of input sorts.
4. A special important case is obtained by chosing $In = S$. In that case the associated observational equality is a total $\Sigma$-congruence and correspond to the (total) observational equality of elements as defined in e.g. [9, 1].

### 3.2 Behavioural Theories

According to the generalization of the standard satisfaction relation to the satisfaction relation with respect to a family $\approx$ of $\Sigma$-congruences we consider the theory with respect to $\approx$ of a given class $C$ of $\Sigma$-algebras.

**Definition 10 (Behavioural theory).** Let $C \subseteq Alg(\Sigma)$ be a class of $\Sigma$-algebras. The behavioural theory of $C$, denoted by $Th_\approx(C)$, is defined by:
$Th_\approx(C) \stackrel{\text{def}}{=} \{\Sigma\text{-formula } \phi \mid A \models_\approx \phi \text{ for all } A \in C\}$.
In the following $C \models_\approx \phi$ is an equivalent notation for $\phi \in Th_\approx(C)$.

Another central result of [2] is the following theorem which leads to a characterization of behavioural theories:

**Theorem 11.** *For all $\Sigma$-algebras $A, B$ and $\Sigma$-formulas $\phi$ the following holds:*
1. *$A \models_\approx \phi$ if and only if $A/{\approx_A} \models \phi$.*
2. *If $A \equiv B$ then $A \models_\approx \phi$ if and only if $B \models_\approx \phi$.*
3. *If $A$ is fully abstract then $A \models_\approx \phi$ if and only if $A \models \phi$.*

**Corollary 12.** *For any class $C$ of $\Sigma$-algebras, we have:*
1. *$Th_\approx(C) = Th(C/\approx)$.*
2. *$Th_\approx(Abs_\equiv(C)) = Th_\approx(C)$.*
3. *$Th_\approx(FA_\approx(C)) = Th(FA_\approx(C))$.*
4. *$Th_\approx(Beh_\approx(C)) = Th_\approx(Abs_\equiv(FA_\approx(C))) = Th(FA_\approx(C))$.*

[4] By exception, in the sequel $\mathrm{Var}(C)$ will denote the set of variables occurring in $C$ *but* the context variable of $C$.

In practice it is usually difficult to prove behavioural theorems due to the generalized satisfaction relation. Although Corollary 12(1) shows that in principle behavioural theories can be reduced to standard theories this result is of little practical interest because even if the class $C$ is axiomatizable we have (in general) no straightforward proof system for the standard theory of the class $C/\approx$ (since the formation of quotients does not preserve the validity of arbitrary $\Sigma$-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 is first to reduce the behavioural theory $Th_{\approx}(C)$ of some class $C$ of $\Sigma$-algebras to (a subset of) the standard theory $Th(D)$ of some other class $D$ of algebras and then to look for an appropriate axiomatization (or a proof system) for $D$ (provided that an axiomatization or a proof system for $C$ is given). According to the previous results we know that this strategy works well in the following case: If $C$ is a class of **fully abstract** algebras then we know by Corollary 12(3) that $Th_{\approx}(C) = Th(C)$ i.e. in this case the behavioural theory of $C$ can be reduced to the standard theory of $C$. Moreover we have shown in [1] that if $C = FA_{\approx_{Obs,S}}(Mod(SP))$ is the class of the fully abstract models of a standard specification $SP$ where $\approx_{Obs,S}$ is a total observational equality defined by a set $Obs$ of observable sorts (cf. Example 1) then $C$ can be axiomatized by a set of infinitary $\Sigma$-formulas which under some condition (the so-called *observability kernel* assumption) can be replaced by a finitary axiomatization.

It is the aim of this work to obtain a similar result for arbitrary classes $C$ of $\Sigma$-algebras and arbitrary families $\approx$ of partial $\Sigma$-congruences.

## 4 The Lift Operator

In a first step we introduce a "lift" operator which provides an explicit denotation for the given family $\approx = (\approx_A)_{A \in Alg(\Sigma)}$ of partial $\Sigma$-congruences. For this purpose we use predicate symbols to denote the equivalence relations and their domains.[5]

**Definition 13 (Lift operator).** Given a signature $\Sigma$, a $\Sigma$-algebra $A$, a class $C$ of $\Sigma$-algebras and a $\Sigma$-formula $\phi$ we define their lifted versions as follows:

1. $\mathcal{L}(\Sigma) \stackrel{\text{def}}{=} \Sigma \cup \{\sim_s : s\ s\}_{s \in S} \cup \{D_s : s\}_{s \in S}$, i.e. $\mathcal{L}(\Sigma)$ is the signature $\Sigma$ enriched by, for each sort $s$ in $S$, a binary predicate $\sim_s : s\ s$ (to denote the behavioural equality) and a unary predicate $D_s$ (to denote the definedness of $\sim_s$). We will adopt an infix notation for the binary predicates $\sim_s$, i.e. we write $l \sim_s r$ instead of $\sim_s(l, r)$.
2. $\mathcal{L}(A)$ is the unique $\mathcal{L}(\Sigma)$-algebra extension of $A$ defined by:
   (a) $\mathcal{L}(A)|_{\Sigma} \stackrel{\text{def}}{=} A$

[5] We assume the reader to be familiar with the usual notions of predicates and their interpretations.

(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 $a \approx_{A,s} b$.
(c) For any $s$ in $S$, for any $a$ in $\mathcal{L}(A)_s$ $(= A_s)$, $D_s^{\mathcal{L}(A)}(a)$ if and only if $a \approx_{A,s} a$ (hence if and only if $a \sim_s^{\mathcal{L}(A)} a$).

3. $\mathcal{L}(C) \stackrel{\text{def}}{=} \{\mathcal{L}(A) \mid A \in C\}$.
4. $\mathcal{L}(\phi) \stackrel{\text{def}}{=} \left[\left(\bigwedge_{y:s \in \mathrm{FreeVar}(\phi)} D_s(y)\right) \Rightarrow \phi^*\right]$, where $\phi^*$ is defined by induction on the structure of $\phi$ as follows:[6]
   (a) If $\phi$ is an equation $l = r$ between two terms of sort $s$, then $\phi^*$ 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 $\phi$ is a closed $\Sigma$-formula then $\mathcal{L}(\phi)$ coincides with $\phi^*$.

**Definition 14.** Given a class $D$ of $\mathcal{L}(\Sigma)$-algebras, we define:
$Th^{\mathcal{L}}(D) \stackrel{\text{def}}{=} \{\Sigma\text{-formula } \phi \mid \mathcal{L}(\phi) \in Th(D)\}$.

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

**Theorem 15.** *For any $\Sigma$-algebra $A$ and $\Sigma$-formula $\phi$,*
$A \models_\approx \phi$ *if and only if* $\mathcal{L}(A) \models \mathcal{L}(\phi)$.
*Hence, for any class $C$ of $\Sigma$-algebras,*
$C \models_\approx \phi$ *if and only if* $\mathcal{L}(C) \models \mathcal{L}(\phi)$, *i.e.* $Th_\approx(C) = Th^{\mathcal{L}}(\mathcal{L}(C))$.

*Proof.* Relies on the following lemmas:

**Lemma 16.** *Let $A$ be a $\Sigma$-algebra and $\phi$ be a $\Sigma$-formula. For all valuations $\alpha : X \to \mathrm{Dom}(\approx_A)$, the following conditions are equivalent:*

1. $A, \alpha \models_\approx \phi$
2. $\mathcal{L}(A), \alpha \models \phi^*$
3. $\mathcal{L}(A), \alpha \models \mathcal{L}(\phi)$

**Lemma 17.** *Let $A$ be a $\Sigma$-algebra and $\phi$ be a $\Sigma$-formula. The following conditions are equivalent:*

1. *For all valuations* $\alpha : X \to \mathrm{Dom}(\approx_A)$: $\mathcal{L}(A), \alpha \models \mathcal{L}(\phi)$
2. *For all valuations* $\beta : X \to \mathcal{L}(A)$: $\mathcal{L}(A), \beta \models \mathcal{L}(\phi)$ □

*Remark.* Let $C$ be a class of fully abstract $\Sigma$-algebras. By Theorem 15 we have $Th_\approx(C) = Th^{\mathcal{L}}(\mathcal{L}(C))$. Moreover, $Th^{\mathcal{L}}(\mathcal{L}(C)) = Th(C)$ because for any fully abstract algebra $A$, $A \models \phi$ if and only if $\mathcal{L}(A) \models \mathcal{L}(\phi)$. Hence we obtain in this particular case again our previous result $Th_\approx(C) = Th(C)$.

[6] Similar constructions, called relativizations, were used in [12] and recently in [6].

Theorem 15 provides a means for reducing the set of behavioural theorems over an arbitrary class $C$ of $\Sigma$-algebras to a set of standard theorems over a corresponding class $\mathcal{L}(C)$ of $\mathcal{L}(\Sigma)$-algebras.[7] However, the usefulness of this reduction still depends on the possibility to perform proofs of standard theorems over $\mathcal{L}(C)$. Since $\mathcal{L}(C)$ is constructed on top of $C$ by introducing a denotation for the behavioural equality we claim that for proving standard theorems over $\mathcal{L}(C)$ we need, on one hand, a proof system for proving standard theorems over $C$ (in the best case $C$ is axiomatizable) and, on the other hand, we need an axiomatization of the behavioural equality which will be considered in the next section.

*Example 2.*

1. Let $C = Mod(SP)$ be the model class of a standard specification $SP = \langle \Sigma, \mathcal{A}x \rangle$. Then, by Theorem 15, $Th_{\approx}(Mod(SP)) = Th^{\mathcal{L}}(\mathcal{L}(Mod(SP)))$ and we will see in the next section that $\mathcal{L}(Mod(SP))$ can be axiomatized with the help of an axiomatization of the behavioural equality.
   More generally, if $SP$ is a structured ASL-like specification (cf. Section 2) we will see that $\mathcal{L}(Mod(SP))$ can be expressed by a specification of the ASL-like language as soon as an axiomatization of the behavioural equality is provided (cf. Example 7).
2. Let $C = Abs_{\equiv}(C')$ for some class $C'$ of $\Sigma$-algebras. Then an immediate application of Theorem 15 would lead to $Th_{\approx}(Abs_{\equiv}(C')) = Th^{\mathcal{L}}(\mathcal{L}(Abs_{\equiv}(C')))$. But even if $C'$ is axiomatizable there is no simple proof system for $Abs_{\equiv}(C')$ and hence also not for $\mathcal{L}(Abs_{\equiv}(C'))$. However, in this case we know, by Corollary 12(2), that $Th_{\approx}(Abs_{\equiv}(C')) = Th_{\approx}(C')$ and hence we can apply Theorem 15 to $C'$ instead of $Abs_{\equiv}(C')$. Thus we obtain $Th_{\approx}(Abs_{\equiv}(C')) = Th_{\approx}(C') = Th^{\mathcal{L}}(\mathcal{L}(C'))$. Hence for abstractor specifications we have:
   $Th_{\approx}(Mod(\textbf{abstract } SP \textbf{ w.r.t. } \equiv)) = Th_{\approx}(Mod(SP)) = Th^{\mathcal{L}}(\mathcal{L}(Mod(SP)))$, i.e. this case can be reduced to the case considered in Part 1 of the example.
3. Let $C = Beh_{\approx}(C')$ for some class $C'$ of $\Sigma$-algebras. Then an application of Theorem 15 would lead to $Th_{\approx}(Beh_{\approx}(C')) = Th^{\mathcal{L}}(\mathcal{L}(Beh_{\approx}(C')))$ which again is not a useful reduction. However, in this case we know, by Corollary 12(4), that $Th_{\approx}(Beh_{\approx}(C')) = Th(FA_{\approx}(C'))$. In this case we are interested in an axiomatization of full abstractness. In particular, **if the family $\approx$ is regular** then $Th_{\approx}(Mod(\textbf{behaviour } SP \textbf{ w.r.t. } \approx)) = Th(FA_{\approx}(Mod(SP)))$. (Note that in this case we do not actually use Theorem 15 but rather the more specific Theorem 9 together with Corollary 12(4).)
4. Let $C$ be a class of $\Sigma$-algebras such that $Abs_{\equiv}(C) = Beh_{\approx}(C)$ (see [2] for simple necessary and sufficient conditions). Then we have:
   $Th_{\approx}(C) = Th_{\approx}(Abs_{\equiv}(C)) = Th_{\approx}(Beh_{\approx}(C)) = Th(FA_{\approx}(C))$.

## 5 Axiomatization of the Behavioural Equality

In the previous section we have shown how to replace the behavioural theory of some given class $C$ of $\Sigma$-algebras by (a subset of) the standard theory of another

[7] In [6] it is shown how a similar result for "expressible" behavioural equalities can be obtained in the framework of higher-order logic.

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.

**Definition 18 (Axiomatization of the behavioural equality).** An axiomatization of the behavioural equality $\approx$ is a family of (possibly infinitary) $\Sigma$-formulas $(Beh_s^\infty(x_s, y_s))_{s\in S}$ (where $x_s$ and $y_s$ are the only free variables, of sort $s$, of $Beh_s^\infty(x_s, y_s)$) such that, for any $\Sigma$-algebra $A$, any sort $s$ in $S$ and any valuation $\alpha : X \to A$, $\quad A, \alpha \models Beh_s^\infty(x_s, y_s)$ if and only if $\alpha(x_s) \approx_A \alpha(y_s)$. Whenever such an axiomatization exists, we say that the behavioural equality $\approx$ is axiomatizable.

In the following, we assume given an axiomatization $(Beh_s^\infty(x_s, y_s))_{s\in S}$ of the behavioural equality $\approx$.

*Example 3 (Observational equality).*

- The (total) observational equality $\approx_{Obs,S}$ induced by a set $Obs$ of observable sorts (cf. Example 1) is axiomatized by the following infinitary formulas: $Beh_s^\infty(x_s, y_s) \stackrel{\text{def}}{=} \bigwedge_{C\in\mathcal{C}_s^{Obs,S}} \forall \mathrm{Var}(C)\ .\ C[x_s] = C[y_s]$. Note that if $s$ is an observable sort then $Beh_s^\infty(x_s, y_s)$ is equivalent to $x_s = y_s$, i.e. for observable elements the observational equality coincides with the set-theoretic equality.
- The (partial) observational equality $\approx_{Obs,In}$ induced by a set $Obs$ of observable sorts and a set $In$ of input sorts (cf. Example 1) is axiomatized by the following infinitary formulas:
  $Beh_s^\infty(x_s, y_s) \stackrel{\text{def}}{=} Def_s(x_s) \wedge Def_s(y_s) \wedge \bigwedge_{C\in\mathcal{C}_s^{Obs,In}} \forall \mathrm{Var}(C)\ .\ C[x_s] = C[y_s]$,
  where $Def_s(x_s)$ is an abbreviation for $\bigvee_{t\in T_\Sigma(X_{In})_s} \exists \mathrm{Var}(t)\ .\ x_s = t$.
  Note that if $s$ is an input sort, then $Beh_s^\infty(x_s, y_s)$ is equivalent to
  $\bigwedge_{C\in\mathcal{C}_s^{Obs,In}} \forall \mathrm{Var}(C)\ .\ C[x_s] = C[y_s]$; if $s$ is an observable sort then
  $Beh_s^\infty(x_s, y_s)$ is equivalent to $Def_s(x_s) \wedge Def_s(y_s) \wedge x_s = y_s$. Moreover, $Beh_s^\infty(x_s, x_s)$ is always equivalent to $Def_s(x_s)$.

Before we come back to the general case of lifted algebras in Section 5.3 we will consider the usefulness of an axiomatization of the behavioural equality in the case of fully abstract algebras.

### 5.1 Case of Fully Abstract Algebras

The following proposition provides an obvious characterization of full abstractness in terms of an axiomatization of the behavioural equality.

**Proposition 19 (Characterization of fully abstract algebras).**
*A $\Sigma$-algebra $A$ is fully abstract if and only if, for all $s$ in $S$,*
$A \models \forall x_s, y_s{:}s.\, [Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s = y_s]$. $\square$

From Proposition 19 we directly deduce the following result which says that for any class $C$ of $\Sigma$-algebras the subclass of the fully abstract algebras of $C$ can be characterized using an axiomatization of the behavioural equality. This fact will be used when considering the particular case of behavioural theories of behavioural specifications (cf. Example 4(3)).

**Theorem 20.** *Let $FA^\infty$ be the following formula:*
$\bigwedge_{s\in S}\ \forall x_s, y_s{:}s.\,[Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s = y_s]$.
*Then for any class $C$ of $\Sigma$-algebras, $FA_\approx(C) = C \cap Mod(\langle \Sigma, FA^\infty\rangle)$.*
*In particular, if $C = Mod(\langle \Sigma, \mathcal{A}x\rangle)$ then $FA_\approx(C) = Mod(\langle \Sigma, \mathcal{A}x \cup FA^\infty\rangle)$.* □

### 5.2 Invariance and Regularity

The results for fully abstract algebras are the basis for the characterization of behavioural theories of behavioural specifications since in this case we have $Th_\approx(Mod(\textbf{behaviour}\ SP\ \textbf{w.r.t.}\ \approx)) = Th(FA_\approx(Mod(SP)))$ whereby it is crucial that the given family $\approx$ is regular (cf. Example 2(3)). The aim of this section is to point out that whenever an axiomatization of the behavioural equality is provided one can characterize the regularity of $\approx$ by a certain property (called invariance) of this axiomatization.

**Definition 21 (Invariant formula).** A formula $\phi$ is called invariant w.r.t. $\approx$ if, for any $\Sigma$-algebra $A$ and valuation $\alpha : X \to \mathrm{Dom}(\approx_A)$, $A, \alpha \models_\approx \phi$ if and only if $A, \alpha \models \phi$.

The following lemmas will be used for proving the characterization of regularity given in Proposition 24.

**Lemma 22.** *The axiomatization of the behavioural equality $\approx$ is invariant if and only if, for any $\Sigma$-algebra $A$ and any sort $s$ in $S$:*
$A \models_\approx\ \forall x_s, y_s{:}s.\,[Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s = y_s]$.

*Proof.* Let $A$ be a $\Sigma$-algebra and $\alpha : X \to \mathrm{Dom}(\approx_A)$ be an arbitrary valuation.
$\Longrightarrow$: $A, \alpha \models_\approx Beh_s^\infty(x_s, y_s)$ iff (by invariance) $A, \alpha \models Beh_s^\infty(x_s, y_s)$ iff
$\alpha(x_s) \approx_A \alpha(y_s)$ iff $A, \alpha \models_\approx x_s = y_s$.
$\Longleftarrow$: $A, \alpha \models_\approx Beh_s^\infty(x_s, y_s)$ iff (by assumption) $A, \alpha \models_\approx x_s = y_s$ iff
$A, \alpha \models Beh_s^\infty(x_s, y_s)$. Hence the axiomatization is invariant. □

**Lemma 23.** *The axiomatization of the behavioural equality $\approx$ is invariant if and only if, for any $\Sigma$-algebra $A$ and any sort $s$ in $S$:*
$A/{\approx_A} \models\ \forall x_s, y_s{:}s.\,[Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s = y_s]$.

*Proof.* Follows from Theorem 11.1 and Lemma 22. □

**Proposition 24.** *The behavioural equality $\approx$ is regular if and only if its axiomatization is invariant.*

*Proof.* The behavioural equality $\approx$ is regular if and only if, for any $\Sigma$-algebra $A$, $A/{\approx_A}$ is fully abstract. According to Proposition 19 and Lemma 23, this is equivalent to the invariance of the axiomatization. □

It is therefore easy to show that the observational equality $\approx_{Obs,In}$ is always regular (cf. Example 3). In the following we always assume that the family $\approx$ is regular or equivalently that the axiomatization of the behavioural equality is invariant.

### 5.3 General Case

We will now generalize the results obtained for fully abstract algebras to the lifting operator. In a first step we obtain an appropriate characterization of lifted algebras.

**Proposition 25 (Characterization of lifted algebras).** *Let $B$ be an arbitrary $\mathcal{L}(\Sigma)$-algebra. Then $B \in \mathcal{L}(Alg(\Sigma))$ (i.e. $B$ is a lifted algebra) if and only if, for all $s \in S$:*
$B \models \forall x_s, y_s{:}s.\, [Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s \sim_s y_s]$ *and* $B \models \forall x_s{:}s.\, [x_s \sim_s x_s \Leftrightarrow D_s(x)]$.

*Proof.* Let $A$ be a $\Sigma$-algebra. Since $A$ and $\mathcal{L}(A)$ have the same carrier sets, it is not necessary to distinguish between valuations from $X$ to $A$ and valuations from $X$ to $\mathcal{L}(A)$. In the following we will denote such valuations by $\alpha : X \longrightarrow A, \mathcal{L}(A)$.

$\Longrightarrow$: Assume $B \in \mathcal{L}(Alg(\Sigma))$ and let $A \in Alg(\Sigma)$ such that $B = \mathcal{L}(A)$. Let $\alpha : X \longrightarrow A, \mathcal{L}(A)$ be an arbitrary valuation. $\mathcal{L}(A), \alpha \models Beh_s^\infty(x_s, y_s)$ iff (since $Beh_s^\infty(x_s, y_s)$ is a $\Sigma$-formula) $A, \alpha \models Beh_s^\infty(x_s, y_s)$ iff (by Definition 18) $\alpha(x_s) \approx_A \alpha(y_s)$ iff (by Definition 13) $\alpha(x_s) \sim_s^{\mathcal{L}(A)} \alpha(y_s)$ iff $\mathcal{L}(A), \alpha \models x_s \sim_s y_s$. Hence $B \models \forall x_s, y_s{:}s.\, [Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s \sim_s y_s]$. The fact that $B \models \forall x_s{:}s.\, [x_s \sim_s x_s \Leftrightarrow D_s(x)]$ follows directly from Definition 13(2.c).

$\Longleftarrow$: Assume $B \models \forall x_s, y_s{:}s.\, [Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s \sim_s y_s]$ and $B \models \forall x_s{:}s.\, [x_s \sim_s x_s \Leftrightarrow D_s(x)]$. Let $A = B\,|_\Sigma$. We prove that $B = \mathcal{L}(A)$. For this it is enough to show that for all $s \in S$ and for all $a, b \in B_s$ $(= A_s = \mathcal{L}(A)_s)$ $a \sim_s^B b$ iff $a \sim_s^{\mathcal{L}(A)} b$ and $D_s^B(a)$ iff $D_s^{\mathcal{L}(A)}(a)$.
Let $\alpha : X \to B, A, \mathcal{L}(A)$ be a valuation such that $\alpha(x_s) = a$ and $\alpha(y_s) = b$. Then $a \sim_s^B b$ iff $B, \alpha \models x_s \sim_s y_s$ iff $B, \alpha \models Beh_s^\infty(x_s, y_s)$ iff $A, \alpha \models Beh_s^\infty(x_s, y_s)$ iff $a \approx_A b$ iff $a \sim_s^{\mathcal{L}(A)} b$. Moreover, $D_s^B(a)$ iff $B, \alpha \models D_s(x)$ iff $B, \alpha \models x_s \sim_s x_s$ iff $a \sim_s^B a$ iff $a \sim_s^{\mathcal{L}(A)} a$ iff $D_s^{\mathcal{L}(A)}(a)$. Hence $B = \mathcal{L}(A)$. $\square$

From Proposition 25 we directly obtain that for any class $C$ of $\Sigma$-algebras the class $\mathcal{L}(C)$ of lifted algebras can be characterized using an axiomatization of the behavioural equality.

**Theorem 26.** *Let $BEH^\infty$ be the following formula:*
$\bigwedge_{s\in S} \forall x_s, y_s{:}s.\, [Beh_s^\infty(x_s, y_s) \Leftrightarrow x_s \sim_s y_s] \;\wedge\; \bigwedge_{s\in S} \forall x_s{:}s.\, [x_s \sim_s x_s \Leftrightarrow D_s(x_s)]$.
*For any class $C$ of $\Sigma$-algebras, let $Ext(C) \stackrel{\text{def}}{=} \{B \in Alg(\mathcal{L}(\Sigma)) \mid B\,|_\Sigma \in C\}$. Then $\mathcal{L}(C) = Ext(C) \cap Mod(\langle \mathcal{L}(\Sigma), BEH^\infty\rangle)$.*
*In particular, if $C = Mod(\langle \Sigma, \mathcal{A}x\rangle)$ then $Ext(C) = Mod(\langle \mathcal{L}(\Sigma), \mathcal{A}x\rangle)$, hence $\mathcal{L}(C) = Mod(\langle \mathcal{L}(\Sigma), \mathcal{A}x \cup BEH^\infty\rangle)$.* $\square$

### 5.4 Application to Various Classes of Algebras

We can apply the previous theorems to various classes of algebras of special interest.

*Example 4.* Let $SP = \langle \Sigma, \mathcal{A}x\rangle$ be a standard specification.

1. If $C = Mod(SP)$ then, by Theorem 26,
   $\mathcal{L}(Mod(SP)) = Mod(\langle \mathcal{L}(\Sigma), \mathcal{A}x \cup BEH^{\infty} \rangle)$ and hence, by Theorem 15,
   $Th_{\approx}(Mod(SP)) = Th^{\mathcal{L}}(Mod(\langle \mathcal{L}(\Sigma), \mathcal{A}x \cup BEH^{\infty} \rangle))$.
2. If $C = Mod($**abstract** $SP$ **w.r.t.** $\equiv)$ then
   $Th_{\approx}(Mod($**abstract** $SP$ **w.r.t.** $\equiv)) =$ (cf. Example 2(2)) $Th_{\approx}(Mod(SP)) =$ (cf. Case 1) $Th^{\mathcal{L}}(Mod(\langle \mathcal{L}(\Sigma), \mathcal{A}x \cup BEH^{\infty} \rangle))$.
3. If $C = Mod($**behaviour** $SP$ **w.r.t.** $\approx)$ then
   $Th_{\approx}(Mod($**behaviour** $SP$ **w.r.t.** $\approx)) =$ (cf. Example 2(3))
   $Th(FA_{\approx}(Mod(SP))) =$ (cf. Theorem 20) $Th(Mod(\langle \Sigma, \mathcal{A}x \cup FA^{\infty} \rangle))$.
4. Assume that $Mod($**abstract** $SP$ **w.r.t.** $\equiv) = Mod($**behaviour** $SP$ **w.r.t.** $\approx)$ (see [2] for simple necessary and sufficient conditions). Then
   $Th_{\approx}(Mod(SP)) = Th(Mod(\langle \Sigma, \mathcal{A}x \cup FA^{\infty} \rangle))$ (cf. Example 2(4) and Case 3).

## 6 The Behavioural Kernel

As we have seen in Example 4, the combination of the results in Section 4 and Section 5 allows us to characterize behavioural theories in terms of standard theories using an axiomatization of the behavioural equality. In particular, we have seen how we can axiomatize fully abstract algebras (through $FA^{\infty}$) and lifted algebras (through $BEH^{\infty}$). However, in general these axiomatizations are infinitary ones. To perform proofs is therefore still difficult. One possibility is to replace these infinitary formulas by infinitary proof rules (such as *context induction*) and to show that the resulting proof system is sound and complete (cf. [5]). Another possibility, studied in this section, is to find special (but general enough) cases where the infinitary axiomatization can be replaced by a finitary one.

In practice it may be simple to find an infinitary axiomatization of the behavioural equality (indeed in the observational framework this axiomatization is directly deduced from the definition of the observational equality). Unfortunately it is usually impossible to find a finitary axiomatization of $\approx$ which for any $\Sigma$-algebra $A$ is equivalent to the infinitary one. But since we are always interested in the behavioural theory of some given class $C$ of $\Sigma$-algebras we do not really need a finite axiomatization of the behavioural equality for any arbitrary $\Sigma$-algebra but rather for the algebras in the class $C$ we are interested in. In several examples it turns out that if the class $C$ satisfies some simple properties, then we can find a finitary axiomatization of the behavioural equality for the algebras belonging to $C$. This is formalized as follows:

**Definition 27 (Conditional axiomatization of the behavioural equality).** A conditional axiomatization of the behavioural equality $\approx$ is a pair $(BK, (Beh^F_s(x_s, y_s))_{s \in S})$, where $BK$ is a finitary first-order $\Sigma$-formula and $(Beh^F_s(x_s, y_s))_{s \in S}$ is a family of finitary first-order $\Sigma$-formulas (where $x_s$ and $y_s$ are the only free variables, of sort $s$, of $Beh^F_s(x_s, y_s)$) such that:
For any $\Sigma$-algebra $A$, **if** $A \models BK$ **then** for any $s$ in $S$ and valuation $\alpha : X \to A$, $A, \alpha \models Beh^F_s(x_s, y_s)$ if and only if $\alpha(x_s) \approx_A \alpha(y_s)$.
$BK$ is called the behavioural kernel of the conditional axiomatization.

From the definition above one may think that, given a class $C$ of algebras, one has to look for an adequate conditional axiomatization $(BK, (Beh_s^F(x_s, y_s))_{s\in S})$ such that each $A$ in $C$ satisfies $BK$. In practice we work the other way round, i.e. we start from a given infinitary axiomatization $(Beh_s^\infty(x_s, y_s))_{s\in S}$ and then we will look for some syntactical transformations leading to a (finitary) conditional axiomatization $(BK, (Beh_s^F(x_s, y_s))_{s\in S})$ such that whenever a $\Sigma$-algebra $A$ satisfies the condition BK the finitary axiomatization is equivalent to the infinitary one. More precisely:

**Lemma 28.** *$(BK, (Beh_s^F(x_s, y_s))_{s\in S})$ is a conditional axiomatization of the behavioural equality if and only if, for any $\Sigma$-algebra $A$:*
**If** $A \models BK$ **then** *for any $s$ in $S$ and valuation $\alpha : X \to A$,*
*$A, \alpha \models Beh_s^F(x_s, y_s)$ if and only if $A, \alpha \models Beh_s^\infty(x_s, y_s)$.* □

*Example 5.* In the case of the total observational equality $\approx_{Obs,S}$ a conditional axiomatization is given in [1] using as a condition the so-called *observability kernel*. For a concrete instantiation of this axiomatization see Section 7. To provide a conditional axiomatization for a partial observational equality $\approx_{Obs,In}$ is still a matter of further research.

It is obvious that all results obtained in the previous section carry over to conditional axiomatizations.

**Theorem 29.** *Let $(BK, (Beh_s^F(x_s, y_s))_{s\in S})$ be a conditional axiomatization of the behavioural equality. Let $A$ be a $\Sigma$-algebra, $B$ be a $\mathcal{L}(\Sigma)$-algebra and $C$ be a class of $\Sigma$-algebras such that $A \models BK$, $B \models BK$ and $C \models BK$.*

1. *$A$ is fully abstract if and only if, for all $s$ in $S$,*
 $A \models \forall x_s, y_s{:}s.\, \big[Beh_s^F(x_s, y_s) \Leftrightarrow x_s = y_s\big]$.
2. *Let $FA$ be the following formula:*
 $\bigwedge_{s\in S} \forall x_s, y_s{:}s.\, \big[Beh_s^F(x_s, y_s) \Leftrightarrow x_s = y_s\big]$.
 *Then $FA_\approx(C) = C \cap Mod(\langle \Sigma, FA\rangle)$.*
3. *In particular, if $C = Mod(\langle \Sigma, \mathcal{A}x\rangle)$ then $FA_\approx(C) = Mod(\langle \Sigma, \mathcal{A}x \cup FA\rangle)$.*
4. *$B \in \mathcal{L}(Alg(\Sigma))$ (i.e. $B$ is a lifted algebra) if and only if, for all $s \in S$,*
 $B \models \forall x_s, y_s{:}s.\, \big[Beh_s^F(x_s, y_s) \Leftrightarrow x_s \sim_s y_s\big]$ *and*
 $B \models \forall x_s{:}s.\, [x_s \sim_s x_s \Leftrightarrow D_s(x)]$.
5. *Let $BEH$ be the following formula:*
 $\bigwedge_{s\in S} \forall x_s, y_s{:}s.\, \big[Beh_s^F(x_s, y_s) \Leftrightarrow x_s \sim_s y_s\big] \wedge \bigwedge_{s\in S} \forall x_s{:}s.\, [x_s \sim_s x_s \Leftrightarrow D_s(x_s)]$.
 *Then $\mathcal{L}(C) = Ext(C) \cap Mod(\langle \mathcal{L}(\Sigma), BEH\rangle)$.*
6. *In particular, if $C = Mod(\langle \Sigma, \mathcal{A}x\rangle)$ then $\mathcal{L}(C) = Mod(\langle \mathcal{L}(\Sigma), \mathcal{A}x \cup BEH\rangle)$.*

In the following examples we assume given a conditional axiomatization of the behavioural equality with behavioural kernel $BK$.

*Example 6.* Let $SP = \langle \Sigma, \mathcal{A}x\rangle$ be a standard specification such that $SP \models BK$. Then we obtain the same results as in Example 4 but with $BEH$ ($FA$ resp.) instead of $BEH^\infty$ ($FA^\infty$ resp.).

*Example 7.* Let *SP* be an ASL-like specification such that $SP \models BK$. It is easy to show that $Ext(Mod(SP)) \cap Mod(\langle \mathcal{L}(\Sigma), BEH \rangle) = Mod(SP + \langle \mathcal{L}(\Sigma), BEH \rangle)$. Then Theorem 29(5) shows that $\mathcal{L}(Mod(SP)) = Mod(SP + \langle \mathcal{L}(\Sigma), BEH \rangle)$ and hence, by Theorem 15, $Th_{\approx}(Mod(SP)) = Th^{\mathcal{L}}(Mod(SP + \langle \mathcal{L}(\Sigma), BEH \rangle))$. Hence $SP \models_{\approx} \phi$ if and only if $(SP + \langle \mathcal{L}(\Sigma), BEH \rangle) \models \mathcal{L}(\phi)$, where *BEH* is a first-order formula. As an important consequence of this result we obtain that whenever $SP \models BK$ we can use any proof system for ASL-like specifications (cf. e.g. [12]) for proving behavioural theorems over *SP*.

## 7 Example: A Behavioural Theory of Multisets

Let us consider the following specification of multisets:

**spec** : `MULTISET`
 **use** : `ELEM, NAT, BOOL`
 **sort** : `Multiset`
 **operations** :
  `∅ : → Multiset`
  `insert : Elem Multiset → Multiset`
  `_ ∈ _ : Elem Multiset → Bool`
  `occurs : Elem Multiset → Nat`
 **axioms :**
   `∀ M : Multiset, e,e' : Elem .`
  `e ∈ ∅ = false`
  `e ∈ insert(e,M) = true`
  `e ≠ e' ⇒ e ∈ insert(e',M) = e ∈ M`
  `occurs(e,∅) = 0`
  `occurs(e,insert(e,M)) = occurs(e,M) + 1`
  `e ≠ e' ⇒ occurs(e,insert(e',M)) = occurs(e,M)`
  `insert(e,insert(e',M)) = insert(e',insert(e,M))`
**end** `MULTISET`.

The last axiom says that the order of inserting elements into a multiset is irrelevant. However, note that two multisets can be distinguished by the number of occurrences of elements. Hence `MULTISET` $\not\models$ `insert(e,insert(e,M)) = insert(e,M)`. In fact adding this equation to the `MULTISET` specification would lead to an inconsistency (because, e.g., `occurs(e,insert(e,insert(e,∅))) = 2` while `occurs(e,insert(e,∅)) = 1`).

We are now interested to study the behaviour of multisets if we forget the operation `occurs`. For instance, if one would like to use multisets for an implementation of sets then no `occurs` operation would be required. Formally, we consider the signature $\Sigma$-`Set` which consists of all sorts and operations of the `MULTISET` signature but the function symbol `occurs` and we consider the specification $\texttt{MULTISET}|_{\Sigma-\mathrm{Set}}$ which is the restriction of `MULTISET` to the signature

$\Sigma$-`Set` and whose model class is defined (as usual) by:
$Mod(\texttt{MULTISET}|_{\Sigma-\mathrm{Set}}) \stackrel{\mathrm{def}}{=} \{A|_{\Sigma-\mathrm{Set}} \mid A \in Mod(\texttt{MULTISET})\}$.

The behavioural theory of $\texttt{MULTISET}|_{\Sigma-\mathrm{Set}}$ will now be studied w.r.t. the total observational equality $\approx_{Obs,S}$ induced on $\Sigma$-`Set`-algebras by $Obs = \{$ `Elem, Nat, Bool`$\}$ and $S = Obs \cup \{\texttt{Multiset}\}$. This observational equality is a family of total $\Sigma$-`Set`-congruences (cf. Example 1).

To apply Theorem 29, we must provide a conditional axiomatization of the observational equality. But before we note that we can apply some obvious simplifications. First, the observational equality we consider is a total one, hence the predicates $D_s$ are not necessary. Moreover, on observable sorts the observational equality coincides with the set-theoretic equality, hence the predicates $\sim_s$ are not necessary either when $s$ is an observable sort. This means that we can considerably simplify the lifting operator (for algebras and formulas), the axiomatization of the observational equality and the induced formula *BEH* by using just one predicate $\sim_{\texttt{Ms}}$ (for sake of clarity we abbreviate the sort `Multiset` by `Ms`). In the following we will use this simplification without further notice.

Now, to find a conditional axiomatization we apply a result of [1] which leads to:

- The behavioural kernel (called here observability kernel) is the following formula $BK$:
  $\forall$ `M,M':Ms.` $[$ ($\forall$ `e:Elem.` `e` $\in$ `M` $=$ `e` $\in$ `M'`) $\Rightarrow$
  ($\forall$ `e,e':Elem.` `e` $\in$ `insert(e',M)` $=$ `e` $\in$ `insert(e',M')` ) $]$.
- $Beh^F_{\texttt{Ms}}(\texttt{M},\texttt{M}') \stackrel{\mathrm{def}}{=}$ ($\forall$ `e:Elem.` `e` $\in$ `M` $=$ `e` $\in$ `M'`).

It is not difficult to prove that $\texttt{MULTISET}|_{\Sigma-\mathrm{Set}} \models BK$. Therefore, following Example 7, we can conclude that, for any $\Sigma$-`Set` formula $\phi$:
$\texttt{MULTISET}|_{\Sigma-\mathrm{Set}} \models_{\approx_{Obs,S}} \phi$ iff
$(\texttt{MULTISET}|_{\Sigma-\mathrm{Set}} + \langle \mathcal{L}(\Sigma\text{-}\texttt{Set}), BEH\text{-}\texttt{Set}\rangle) \models \mathcal{L}(\phi)$ iff
$\texttt{MULTISET}^* \models \mathcal{L}(\phi)$, where:

- $BEH$-`Set` $\stackrel{\mathrm{def}}{=}$ $\forall$ `M,M':Ms.` $[$($\forall$ `e:Elem.` `e` $\in$ `M` $=$ `e` $\in$ `M'`) $\Leftrightarrow$ `M` $\sim_{\texttt{Ms}}$ `M'`$]$.
- $\texttt{MULTISET}^*$ is the enrichment of the `MULTISET` specification by the predicate $\sim_{\texttt{Ms}}$ and the axiom $BEH$-`Set`.

For instance, we can prove that
$\texttt{MULTISET}|_{\Sigma-\mathrm{Set}} \models_{\approx_{Obs,S}}$ `insert(e,insert(e,M)) = insert(e,M)`
by proving $\texttt{MULTISET}^* \models$ `insert(e,insert(e,M))` $\sim_{\texttt{Ms}}$ `insert(e,M)`
which is not difficult to show according to the axioms of $\texttt{MULTISET}^*$.

## 8 Conclusion

In this paper we have studied how to prove behavioural theorems over different kinds of specifications using standard proof techniques. Our approach is abstract because we have considered behavioural equalities which are determined by arbitrary partial congruence relations. Thus our results can be instantiated by different kinds of behavioural equalities.

Important examples are the total observational congruence considered e.g. in [9] and the partial observational congruences of e.g. [8, 2]. In [1] we have shown how to build a conditional axiomatization for total observational congruences. How to construct a conditional axiomatization for partial observational congruences has still to be investigated.

**Acknowledgements:** This work is partially supported by the French-German cooperation programme PROCOPE and by the ESPRIT Working Group COMPASS II.

## References

1. 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.
2. M. Bidoit, R. Hennicker, and M. Wirsing. Behavioural and abstractor specifications. Technical Report LIENS–94–10, 1994. 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.
3. 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.
4. 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.
5. R. Hennicker and M. Wirsing. *Behavioural specifications.* Proof and Computation, International Summer School Marktoberdorf 1993. Springer-Verlag, 1995.
6. M. Hofmann and D. Sannella. On behavioural abstraction and behavioural satisfaction in higher-order logic. In *Proc. of TAPSOFT'95*. Springer-Verlag L.N.C.S., 1995. To appear.
7. R. Milner. Fully abstract models of typed $\lambda$-calculi. *Theoretical Computer Science*, 4:1–22, 1977.
8. 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.
9. H. Reichel. Initial restrictions of behaviour. In *Proc. of IFIP Working Conference, The Role of Abstract Models in Information Processing*, 1985.
10. D. Sannella and A. Tarlecki. Toward formal development of programs from algebraic specification: implementation revisited. *Acta Informatica*, 25:233–281, 1988.
11. O. Schoett. Behavioural correctness of data representation. *Science of Computer Programming*, 14:43–57, 1990.
12. M. Wirsing. *Structured specifications: syntax, semantics and proof calculus.* Logic and Algebra of Specification, International Summer School Marktoberdorf 1991. Springer-Verlag, 1993.