Characterizing Behavioural Semantics and Abstractor Semantics

Michel Bidoit*, Rolf Hennicker**, Martin Wirsing**

*LIENS, CNRS & Ecole Normale Supérieure
45, Rue d'Ulm, 75230 Paris Cedex, FRANCE
**Institut für Informatik, Ludwig-Maximilians-Universität München
Leopoldstr. 11B, D-80802 München, GERMANY

Abstract. In the literature one can distinguish two main approaches to the definition of observational semantics of algebraic specifications. On one hand, observational semantics is defined using a notion of observational satisfaction for the axioms of a specification and on the other hand one can define observational semantics of a specification by abstraction with respect to an observational equivalence relation between algebras. In this paper we present an analysis and a comparative study of the different approaches in a more general framework which subsumes not only the observational case but also other examples like the bisimulation congruence of concurrent processes. Thereby the distinction between the different concepts of observational semantics is reflected by our notions of behavioural semantics and abstractor semantics. Our main results show that behavioural semantics can be characterized by an abstractor construction and, vice versa, abstractor semantics can be characterized in terms of behavioural semantics. Hence there exists a duality between both concepts which allows to express each one by each other. As a consequence we obtain a sufficient and necessary condition under which behavioural and abstractor semantics coincide. Moreover, the semantical characterizations lead to proof-theoretic results which show that behavioural theories of specifications can be reduced to standard theories (of some classes of algebras).

1 Introduction

Observability plays an important role in program development. For instance, formal implementation notions can be based on this concept. Other applications are the notion of equivalence between concurrent processes and the abstraction from single step transitions to input-output operational semantics.

Since the beginning of the eighties observational frameworks have found continuous interest in the area of algebraic specifications. In the literature one can distinguish two main possibilities for the definition of observational semantics. One is based on the so-called observational satisfaction relation where equations are not interpreted as identities but as observational equivalences of objects (cf. e.g. [Nivela, Orejas 88], [Bernot, Bidoit 91], [Hennicker 91]). In this case the observational semantics of a specification is given by the class of all algebras that observationally satisfy the axioms of the specification. Other approaches define observational semantics by constructing the closure of the (standard) model class of a specification with respect to an observational equivalence relation on algebras (cf. e.g. [Reichel 81], [Sannella, Tarlecki 85, 88], [Wirsing 86], [Schoett 87]). In [Reichel 85] both semantical views are considered and it is shown that they are equivalent if the axioms of a specification are conditional equations with observable premises. However, this is in general not true for specifications with arbitrary first-order formulas as axioms.

In this paper we study the relationships (and differences) between the two semantical concepts in a more general framework which allows to apply our results not only to the observational case but also to other examples like e.g. the bisimulation congruence of concurrent processes. Technically, we generalize the two views of observational semantics in the following way: Instead of the observational equivalence of elements we use an arbitrary congruence relation for the interpretation of equations. This leads to our notion of *behavioural specification* which admits as models all algebras which satisfy the axioms of a specification with respect to a given congruence relation. On the other hand, following the notion of an "abstractor" in [Sannella, Tarlecki 88], we define *abstractor specifications* which describe all algebras that are equivalent to a (standard) model of a specification w.r.t. a given equivalence relation on algebras. In order to establish the connection between behavioural and abstractor semantics we consider only those equivalences on algebras which are "factorizable" (by a congruence relation between the elements of the algebras). An example of a factorizable equivalence is the observational equivalence of algebras w.r.t. a fixed set of observable sorts where the observable "experiments" can take arbitrary inputs. Equivalences which allow only observable inputs (cf. e.g. [Nivela, Orejas 88]) can be captured by generalizing our approach to partial congruences which, however, are not considered in this paper.

As a first central result of our approach we obtain that behavioural semantics can be characterized by the class of all algebras which are equivalent to a fully abstract (standard) model of the specification. This result implies, for instance, that behavioural semantics is more restrictive than abstractor semantics and that a behavioural specification is consistent if and only if there exists a fully abstract (standard) model of the specification. Conversely, we show that abstractor semantics can be characterized in terms of behavioural semantics as well. Hence there exists a nice duality between both kinds of semantics. Each one can be expressed by each other. As a consequence we obtain a necessary and sufficient condition for the equivalence of behavioural and abstractor specifications which says that behavioural semantics coincides with abstractor semantics if and only if the (standard) model class of a specification is closed under the "behavioural quotient" construction.

For the analysis of behavioural properties of specifications we consider their *behavioural theories.* According to the generalized satisfaction relation with respect to a congruence, the behavioural theory of a specification is defined as the set of all formulas which are satisfied w.r.t. the given congruence by all models of the specification. (In the observational framework this is exactly the set of all formulas which are observationally satisfied by all observational models of a specification.) Since it is usually difficult to prove behavioural theorems we need techniques which allow to reduce behavioural proofs to standard ones. Using our characterization of behavioural semantics we can show that the behavioural theory of a behavioural specification is the same as the standard theory of the class of the fully abstract (standard) models of the specification. Similarly we can use the characterization of abstractor semantics for showing that the behavioural theory of an abstractor specification can be reduced to the standard theory of the class of "behavioural quotients" of the (standard) models of the specification. These results provide the basis for the investigation of concepts which allow to prove behavioural properties of specifications by standard proof techniques (cf. [Bidoit, Hennicker 94]).

In this paper we consider flat specifications with first-order axioms. A generalization to structured specifications is possible but requires some more technical assumptions.

2 Algebraic Preliminaries

We assume the reader to be familiar with the basic notions of algebraic specifications (cf. e.g. [Ehrig, Mahr 85]) like *signature* $\Sigma = (S, F)$ (where S is a set of *sorts* and F is a set of *function symbols*), *total* Σ*-algebra* $A = ((A_s)_{s \in S}, (f^A)_{f \in F})$ (consisting of a

family of carrier sets $(A_s)_{s \in S}$ and a family of (total) functions $(f^A)_{f \in F}$), *term algebra* $T(\Sigma, X)$ over an S-sorted family X of variables of sort s.
If A is a total Σ-algebra then a *valuation* α: $X \rightarrow A$ is a family of mappings $(\alpha_s: X_s \rightarrow A_s)_{s \in S}$. The *interpretation w.r.t.* α of a term $t \in T(\Sigma, X)$ is denoted by $I_\alpha(t)$. A Σ-congruence on A is a family of equivalence relations $\approx_A = (\approx_{A,s} \subseteq A_s \times A_s)_{s \in S}$ such that for all $f \in F$, f^A is compatible with $\approx_A$. The class of all Σ-algebras is denoted by *Alg(Σ)*.

3 Behavioural Specifications and Abstractor Specifications

In this section we will define the syntax and semantics of behavioural and abstractor specifications which both are built on top of standard specifications.

3.1 Standard Specifications

A standard specification is a flat, first-order specification where a distinguished set of function symbols is declared as constructors. The constructor declaration restricts the class of admissible models to those algebras which are finitely generated by the constructor symbols, i.e. all elements can be denoted by a constructor term (which is built only by constructor symbols and by variables of those sorts for which no constructor is defined).

Definition 3.1 Let $\Sigma = (S, F)$ be a signature, $Cons \subseteq F$ be a distinguished set of *constructors* and $X = (X_s)_{s \in S}$ a family of countably infinite sets X_s of variables of sort $s \in S$. Then a term t is called *constructor term* if $t \in T(\Sigma', X')$ where $\Sigma' = (S, Cons)$, $X' = (X_s)_{s \in S \backslash range(Cons)}$ and $range(Cons) = \{s \in S \mid \text{there exists } f \in Cons \text{ with functionality } s_1 \times \ldots \times s_n \rightarrow s\}$. The set of constructor terms is denoted by T_{Cons}.
A Σ-algebra A is called *finitely generated by Cons* if for any $a \in A$ there exists a constructor term $t \in T_{Cons}$ and a valuation α such that $I_\alpha(t) = a$.
(Note that the definition of the generation principle is independent from the choice of X as long as X is countably infinite which is generally assumed here.) ♦

The axioms of a specification are Σ-formulas which are arbitrary first-order formulas over a multi-sorted signature Σ where the only predicate symbol is equality "=".

Definition 3.2 Let Σ be a signature. The set of (well-formed) Σ-formulas is inductively defined by:
(0) If $t, r \in T(\Sigma, X)_s$ are terms of sort s, then $t = r$ is a Σ-formula (called equation).
(1) If ϕ , ψ are Σ-formulas then $\neg\phi$, $\phi \wedge \psi$ and $\forall x{:}s.\phi$ are Σ-formulas.
All other logical operators (like $\vee$, $\Rightarrow$, $\exists$) are defined as usual. ♦

A Σ-algebra A satisfies a Σ-formula ϕ (denoted by $A \models \phi$) if A satisfies ϕ for all valuations α: $X \rightarrow A$ (denoted by $A, \alpha \models \phi$) in the usual sense of first-order logic.

Definition 3.3
(1) A *standard specification* $SP = (\Sigma, Cons, E)$ consists of a signature $\Sigma = (S, F)$, a distinguished subset $Cons \subseteq F$ of constructors and a set E of Σ-formulas, called *axioms* of SP.
(2) The model class Mod(SP) of SP is defined by $Mod(SP) =_{def} \{A \in Alg(\Sigma) \mid A \models \phi$ for all $\phi \in E$ and A is finitely generated by Cons$\}$. ♦

Remark Note that if Cons = $\emptyset$ then any algebra $A \in Alg(\Sigma)$ is finitely generated by Cons. This means that in this case Mod(SP) is simply the class of all Σ-algebras satisfying the axioms of SP. Hence our assumption that any specification has a set of constructors is not a restriction but, on the contrary, it allows to apply our results also to specifications with reachability constraints. ♦

Example 3.4
1. The following specification SET is a usual specification of finite sets over arbitrary elements with constructors "true", "false" for the boolean values and "empty", "add" for sets. The operation "iselem" defines the membership test on sets.

spec SET =

sorts {bool, elem, set}

cons {true: → bool, false: → bool, empty: → set, add: elem, set → set}

opns {iselem: elem, set → bool}

axioms $\{\forall\, x, y{:}elem, s{:}set.$

$iselem(x, empty) = false \wedge iselem(x, add(x, s)) = true \wedge$

$[x \neq y \Rightarrow iselem(x, add(y, s)) = iselem(x, s)] \wedge$

$add(x, add(y, s)) = add(y, add(x, s)) \wedge add(x, add(x, s)) = add(x, s)\ \}$

endspec

For instance the algebra $P_{fin}(\mathbf{N})$ of finite subsets of the set **N** of natural numbers is a model of SET.

2. The following specification CSO describes the operational semantics of a trivial nondeterministic sublanguage of CCS. It defines a sort "process" of processes containing a constant "nil", a semantical composition "." of actions and processes and a nondeterministic choice operator "+". The operational semantics is given by a one-step transition function where $(p \xrightarrow{a} p') = true$ indicates that there is a transition from process p to process p' when executing the action a. All known equivalences on processes induce models of CSO.

spec CSO =

sorts {bool, action, process}

cons {true: → bool, false: → bool,

nil : → process, . . . : action, process → process,

. + . : process, process → process}

opns {→ : process, action, process → bool}

axioms $\{\forall\, a{:}\ action, p, p', q{:}\ process.$

$(a.p \xrightarrow{a} p) = true \wedge$

$[(p \xrightarrow{a} p') = true \Rightarrow ((p{+}q \xrightarrow{a} p') = true \wedge (q{+}p \xrightarrow{a} p') = true)]\ \}$

endspec ♦

3.2 Behavioural Specifications

Behavioural specifications are a generalization of standard specifications which allow to describe the behaviour of data structures (or programs) with respect to a given congruence relation. For this purpose we first generalize the standard satisfaction relation. The only difference between this generalization and the standard satisfaction relation of the first-order predicate calculus is that the predicate symbol = is not interpreted by the set-theoretic equality over the carrier sets of an algebra but by a given congruence relation (a connection between both satisfaction relations will be established in Proposition 3.10.1).

Definition 3.5 Let A be a Σ-algebra and $\approx_A$ be a Σ-congruence on A. Moreover, let $t, r \in T(\Sigma, X)_s$ be terms of sort s. The *satisfaction relation w.r.t.* $\approx_A$ (denoted by $\models_{\approx A}$) is defined as follows:

(0) For any valuation $\alpha: X \rightarrow A$: $A, \alpha \models_{\approx A} t = r$ if $I_\alpha(t) \approx_A I_\alpha(r)$.

(1) The satisfaction relation w.r.t. $\approx_A$ for arbitrary Σ-formulas (cf. Definition 3.2) is defined (as usual) by induction over the structure of the formulas.

(2) For any Σ-formula ϕ: $A \models_{\approx A} \phi$ if $A, \alpha \models_{\approx A} \phi$ for all valuations $\alpha: X \rightarrow A$. ♦

In a similar way we also generalize the generation principle of algebras (by a set of constructors) with respect to a congruence:

Definition 3.6 Let $\Sigma = (S, F)$ be a signature, Cons $\subseteq$ F a distinguished set of constructors, A a Σ-algebra and $\approx_A$ a Σ-congruence on A. A is called $\approx_A$*-finitely generated by Cons* if for any $a \in A$ there exists a constructor term $t \in T_{Cons}$ and a valuation α such that $I_\alpha(t) \approx_A a$. In particular, if Cons = $\emptyset$ then any algebra $A \in Alg(\Sigma)$ is $\approx_A$-finitely generated by Cons. (Proposition 3.10.2 provides a connection between the standard generation principle and the generation principle w.r.t. $\approx$.) ♦

Example 3.7

1. Observational equivalence: An important example of a congruence is the observational equivalence of objects which is used for the interpretation of equations in several observational approaches in the literature (cf. above). Formally, we assume given a signature $\Sigma = (S, F)$ and a distinguished set Obs $\subseteq$ S of observable sorts (which denote the carrier sets of observable values). Then two objects of a Σ-algebra A are considered to be observationally equivalent, if they cannot be distinguished by "experiments" with observable result. This can be formally expressed using the notion of *observable context*, which is any term $c \in T(\Sigma, X \cup Z)$ of observable sort which contains (besides variables in X) exactly one variable $z_s \in Z$. (Thereby $Z = \{z_s \mid s \in S\}$ is an S-sorted set of variables such that $z_s \notin X_s$ for all $s \in S$.) The observational equivalence of objects can now be defined in the following way:
For any $s \in S$, two elements $a, b \in A_s$ are called *observationally equivalent* (denoted by $a \approx_{Obs,A} b$) if for all observable Σ-contexts c containing z_s and for all valuations $\alpha: X \rightarrow A$, $I_{\alpha 1}(c) = I_{\alpha 2}(c)$ holds where $\alpha 1(z_s) = a$, $\alpha 2(z_s) = b$ and $\alpha 1(x) = \alpha 2(x) = \alpha(x)$ for all $x \in X$.
It is easy to show that $\approx_{Obs,A}$ defines a congruence relation on A. The satisfaction relation w.r.t. $\approx_{Obs,A}$ is often called *observational* (or *behavioural*) *satisfaction relation.* (Note that for observable equations t = r where t and r are of observable sort the observational satisfaction relation coincides with the standard one, i.e. $A \models_{\approx Obs,A} t = r$ if and only if $A \models t = r$.)

2. Strong bisimulation: The notion of strong bisimulation is an example of a congruence on CSO. For any algebra A over the signature of CSO we define the notion of *simulation equivalence* as follows: Let $p, q \in A_{process}$. Then $p \approx_{sim,A} q$ if for all $p' \in A_{process}$, $s \in (A_{action})^*$,
$[(p \;{}^{s}{\rightarrow}^{*A}\; p') = true^A \Leftrightarrow \exists q' \in A_{process}, p' \approx_{sim,A} q'$ and $(q \;{}^{s}{\rightarrow}^{*A}\; q') = true^A]$ and vice versa for all $q' \in A_{process}$, $s \in (A_{action})^*$,
$[(q \;{}^{s}{\rightarrow}^{*A}\; q') = true^A \Leftrightarrow \exists p' \in A_{process}, q' \approx_{sim,A} p'$ and $(p \;{}^{s}{\rightarrow}^{*A}\; p') = true^A]$.
Thereby $\rightarrow^{*A}$ denotes the reflexive and transitive closure of $\rightarrow^A$.
According to [Astesiano, Wirsing 89] we have the following fact: Let A be a (standard) model of CSO. Then any simulation equivalence is a congruence (w.r.t. the signature of CSO) and the simulation congruences on A form a complete lattice. For

any algebra A, the coarsest simulation congruence on A is called strong bisimulation on A and is denoted by $\approx_{\mathrm{bisim},A}$. If for A we choose the Herbrand model H(CSO) of CSO with the set T_{Cons} of constructor terms as carrier sets and the following interpretation of "$\rightarrow$" by provable transitions

$$H(CSO) \models (p \xrightarrow{a} q) = true \text{ iff } CSO \models (p \xrightarrow{a} q) = true,$$

then $\approx_{\mathrm{bisim},H(CSO)}$ is Milner's strong bisimulation congruence.
As an example for $\models_{\approx_{\mathrm{bisim},H(CSO)}}$ consider the processes $p1 =_{\mathrm{def}} a.(c.nil + d.nil)$, $p2 =_{\mathrm{def}} a.c.nil + a.d.nil$, and $p3 =_{\mathrm{def}} a.d.nil + a.c.nil$ (with pairwise different a, c, d). Then $H(CSO) \models_{\approx_{\mathrm{bisim},H(CSO)}} \neg(p1 = p2) \wedge p2 = p3$, but for the standard satisfaction we have obviously $H(CSO) \models \neg(p1 = p2) \wedge \neg(p2 = p3)$.♦

Behavioural specifications can be built on top of standard specifications for any given family $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ of Σ-congruences on the algebras $A \in \mathrm{Alg}(\Sigma)$ (cf. also the notion of observational Σ-algebra in [Knapik 91]). The essential difference to standard specifications is that instead of the standard satisfaction relation the satisfaction relation w.r.t. ≈ is used for the interpretation of the axioms.

Definition 3.8 Let SP = (Σ, Cons, E) be a standard specification and let $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ be a family of Σ-congruences. Then:

(1) **behaviour SP wrt ≈** is a *behavioural specification.*

(2) The model class of a behavioural specification consists of all Σ-algebras A which satisfy the axioms of SP w.r.t. $\approx_A$ and which are $\approx_A$-finitely generated by the constructors Cons, i.e.

Mod(**behaviour SP wrt** ≈) $=_{\mathrm{def}}$ $\{A \in \mathrm{Alg}(\Sigma) \mid A \models_{\approx_A} \phi$ for all $\phi \in E$ and A is $\approx_A$-finitely generated by Cons$\}$. ♦

Example 3.9

1. Let Σ = (S, F) be a signature and Obs ⊆ S be a set of observable sorts. The set Obs induces a family $\approx_{\mathrm{Obs}} = (\approx_{\mathrm{Obs},A})_{A \in \mathrm{Alg}(\Sigma)}$ of Σ-congruences where for any $A \in \mathrm{Alg}(\Sigma)$, $\approx_{\mathrm{Obs},A}$ is the observational equivalence defined in Example 3.7.1. Then a specification **behaviour SP wrt** $\approx_{\mathrm{Obs}}$ specifies the *observable behaviour* of a data structure (or a program) by means of the observational satisfaction relation.
As a concrete example we can construct the behavioural specification

behaviour SET wrt $\approx_{\{\mathrm{bool},\ \mathrm{elem}\}}$

on top of the standard specification SET of sets (cf. Example 3.4.1). Here Obs = {bool, elem}, i.e. the sorts "bool" and "elem" are considered as observable. Since the sort "set" is not observable, sets can only be observed via the membership test "iselem". For instance, the algebra N* of finite sequences of natural numbers is a model of the behavioural specification of sets. In particular N* satisfies observationally the last two axioms of SET, because one cannot distinguish the order of the elements and the number of occurrences of elements in a sequence by the allowed observations. But note that N* does not satisfy the last two SET axioms w.r.t. the standard satisfaction relation and hence is not a model of the standard specification SET.

2. The behavioural specification **behaviour CSO wrt** $\approx_{\mathrm{bisim}}$ describes all algebras (over the signature of CSO) that satisfy the axioms of CSO w.r.t. the strong bisimulation congruence. But since CSO does not require any equations between processes we have Mod(CSO) = Mod(**behaviour CSO wrt** $\approx_{\mathrm{bisim}}$). ♦

The following proposition establishes an important connection between the generalized satisfaction w.r.t. $\approx$ and the standard satisfaction of formulas and analogously between both kinds of generation principles:

Proposition 3.10 Let $\Sigma = (S, F)$ be a signature, $\approx\ = (\approx_A)_{A \in Alg(\Sigma)}$ be a family of Σ-congruences and $Cons \subseteq F$ be a set of constructors. Then for all Σ-algebras A the following holds:

(1) For all Σ-formulas ϕ, $A \models_{\approx_A} \phi$ if and only if $A/{\approx_A} \models \phi$,
 ($A/{\approx_A}$ denotes the quotient algebra of A w.r.t. $\approx_A$)

(2) A is $\approx_A$-finitely generated by Cons if and only if $A/{\approx_A}$ is finitely generated by Cons.

Sketch of the proof: (1) The proof follows from the following lemma which can be proved by induction on the form of ϕ: For any valuation $\alpha: X \to A$: $A, \alpha \models_{\approx_A} \phi$ if and only if $A/{\approx_A}, \pi \circ \alpha \models \phi$ where $\pi: A \to A/{\approx_A}$ is the canonical epimorphism.
(2) The proof is straightforward. ♦

As a first consequence of Proposition 3.10 we obtain:

Corollary 3.11 For any Σ-algebra A the following holds:

$A \in$ Mod(**behaviour SP wrt** $\approx$) if and only if $A/{\approx_A} \in$ Mod(SP).

3.3 Abstractor Specifications

The notion of "abstractor" was introduced in [Sannella, Tarlecki 88] for describing a specification building operation which allows to abstract from the model class of a specification with respect to a given equivalence relation on the class of all Σ-algebras.

Definition 3.12 Let $\equiv\ \subseteq Alg(\Sigma) \times Alg(\Sigma)$ be an equivalence relation on $Alg(\Sigma)$. For any class $C \subseteq Alg(\Sigma)$ of Σ-algebras, $Abs_{\equiv}(C)$ denotes the closure of C under $\equiv$, i.e.

$$Abs_{\equiv}(C) =_{def} \{B \in Alg(\Sigma) \mid B \equiv A \text{ for,some } A \in C\}.$$ ♦

The syntax and semantics of abstractor specifications is defined by:

Definition 3.13 Let $SP = (\Sigma, Cons, E)$ be a standard specification and let $\equiv\ \subseteq Alg(\Sigma) \times Alg(\Sigma)$ be an equivalence relation. Then:

(1) **abstract SP wrt** $\equiv$ is an *abstractor specification.*

(2) The model class of an abstractor specification is the closure of Mod(SP) under $\equiv$:

Mod(**abstract SP wrt** $\approx$) $=_{def} Abs_{\equiv}$(Mod(SP)). ♦

3.4 Relating Congruences of Elements and Equivalences of Algebras

In this paper we are interested in a relationship between behavioural specifications and abstractor specifications. For this purpose we have to find a connection between congruences between elements of algebras (which are used to define behavioural specifications) and equivalences between algebras themselves (which are used to define abstractor specifications). If we start from a family of Σ-congruences then it is an easy task to construct an associated equivalence relation between algebras in the following way:

Definition 3.14 Let $\approx \; = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ be a family of Σ-congruences. Then $\equiv_{\approx} \subseteq \mathrm{Alg}(\Sigma) \times \mathrm{Alg}(\Sigma)$ is the following equivalence relation on $\mathrm{Alg}(\Sigma)$:

For any $A, B \in \mathrm{Alg}(\Sigma)$, $A \equiv_{\approx} B$ if $A/{\approx_A}$ and $B/{\approx_B}$ are isomorphic. ♦

If we start from an equivalence relation on $\mathrm{Alg}(\Sigma)$ we find an associated family of Σ-congruences only if the equivalence is "factorizable":

Definition 3.15 An equivalence relation $\equiv \; \subseteq \mathrm{Alg}(\Sigma) \times \mathrm{Alg}(\Sigma)$ is called *factorizable* if there exists a family $\approx \; = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ of Σ-congruences such that for all $A, B \in \mathrm{Alg}(\Sigma)$ the following holds:

$A \equiv B$ if and only if $A/{\approx_A}$ and $B/{\approx_B}$ are isomorphic.

In this case we say that $\equiv$ is *factorizable by* $\approx$. ♦

It is usually not a simple task to prove that an equivalence is factorizable. The following example shows how this can be done for the W-equivalence of ASL (cf. [Wirsing 86]) if W is the set of all observable terms with arbitrary input variables. For dealing with observable ground terms or with observable terms that allow only observable inputs our approach can be generalized by considering "partial" congruences (cf. the remarks in Section 6).

Example 3.16 Let $\Sigma = (S, F)$ be a signature and $Obs \subseteq S$ be a set of observable sorts. According to the W-equivalence relation of ASL we say that two Σ-algebras A and B are W-equivalent, denoted by $A \equiv_W B$, if there exists a family $Y = (Y_s)_{s \in S}$ of sets Y_s of variables of sort s and two surjective valuations $\alpha 1: Y \to A$ and $\beta 1: Y \to B$ such that for all terms $t, r \in T(\Sigma, Y)_s$ of observable sort $s \in Obs$ the following holds:

$I_{\alpha 1}(t) = I_{\alpha 1}(r)$ if and only if $I_{\beta 1}(t) = I_{\beta 1}(r)$.

We will now show that $\equiv_W$ is factorizable by the family $\approx_{Obs}$ of congruences defined in Example 3.9.1. For this purpose we have to prove that $A \equiv_W B$ holds if and only if $A/{\approx_{Obs,A}}$ and $B/{\approx_{Obs,B}}$ are isomorphic.

Let us first assume $A \equiv_W B$. Then using the definition of $\equiv_W$ one can prove that for all terms $t, r \in T(\Sigma, Y)$ the following holds: $I_{\alpha 1}(t) \approx_{Obs,A} I_{\alpha 1}(r)$ if and only if $I_{\beta 1}(t) \approx_{Obs,B} I_{\beta 1}(r)$. Using this fact one can show that $h: A/{\approx_{Obs,A}} \to B/{\approx_{Obs,B}}$, $h([a]) =_{def} [I_{\beta 1}(t)]$ if $t \in T(\Sigma, Y)$ with $[a] = [I_{\alpha 1}(t)]$ defines a Σ-isomorphism between the quotient algebras.

Conversely let $h: A/{\approx_{Obs,A}} \to B/{\approx_{Obs,B}}$ be a Σ-isomorphism. W.l.o.g we assume that the carrier sets of A and B are disjoint. Then let $Y =_{def} A \cup B$, let $\alpha 1: Y \to A$ be defined by $\alpha 1(y) =_{def} y$ if $y \in A$, $\alpha 1(y) =_{def} a$ if $y \in B$ and $h([a]) = [y]$ and let $\beta 1: Y \to B$ be defined by $\beta 1(y) =_{def} y$ if $y \in B$, $\beta 1(y) =_{def} b$ if $y \in A$ and $h([y]) = [b]$. Using the definition of Y, $\alpha 1$ and $\beta 1$ we can prove by structural induction that for all terms $t \in T(\Sigma, Y)$, $h([I_{\alpha 1}(t)]) = [I_{\beta 1}(t)]$ holds. Now assume $I_{\alpha 1}(t) = I_{\alpha 1}(r)$ for two observable terms $t, r \in T(\Sigma, Y)_s$ with $s \in Obs$. Then $[I_{\alpha 1}(t)] = [I_{\alpha 1}(r)]$. Since $h([I_{\alpha 1}(t)]) = [I_{\beta 1}(t)]$ and $h([I_{\alpha 1}(r)]) = [I_{\beta 1}(r)]$ we obtain $[I_{\beta 1}(t)] = [I_{\beta 1}(r)]$, i.e. $I_{\beta 1}(t) \approx_{Obs,B} I_{\beta 1}(r)$. Then $I_{\beta 1}(t) = I_{\beta 1}(r)$ holds since t and r are of observable sort. Symmetrically one shows that for any observable terms t and r, $I_{\beta 1}(t) = I_{\beta 1}(r)$ implies $I_{\alpha 1}(t) = I_{\alpha 1}(r)$. Hence A and B are W-equivalent.

As a consequence we obtain that if Obs is a set of observable sorts then any abstractor specification **abstract** SP **wrt** $\equiv_W$ defines an *observational abstraction* which has the same models as the specification **abstract** SP **wrt** $\equiv_{\approx_{Obs}}$ where $\equiv_{\approx_{Obs}}$ is the equivalence generated by the congruence $\approx_{Obs}$. ♦

Behavioural specifications (as introduced in Section 3.2) and abstractor specifications are based on the same intention, namely to allow a more general view of the semantics of specifications. In particular this is useful for formal implementation definitions where implementations may relax (some of) the properties of a given requirement specification (cf. e.g. abstractor implementations in [Sannella, Tarlecki 88] or behavioural implementations in [Hennicker 91], for a survey on implementation concepts and observability see [Orejas et al. 91]). However, the semantical definitions of behavioural specifications and abstractor specifications are quite different. Therefore it is an important issue to compare both approaches carefully and to figure out precisely the relationships and the differences between the two concepts. This is the topic of the next sections.

4 Characterization of Behavioural and Abstractor Semantics

If we consider the particular case of observable behaviour specifications (cf. Example 3.9.1) and observational abstractions (cf. Example 3.16) then we can conclude from a result in [Reichel 85] that both specifications have the same semantics if the axioms of the specification are conditional equations with observable premises. However, in the observational framework this is in general not true if the axioms are arbitrary first-order formulas. For instance, the following specification DEMO has a standard model which, by definition, is also a model of the abstractor specification "**abstract** DEMO **wrt** $\equiv_{\approx_{Obs}}$" but which is not a model of the behavioural specification "**behaviour** DEMO **wrt** $\approx_{Obs}$" if Obs = $\{s\}$, i.e. if s is the only observable sort.

```
spec DEMO =
    sorts { s, s' }
    opns  { a, b: → s,  c, d : → s', f: s' → s }
    axioms  { f(c) = f(d) ∧ [c = d  ⇒ a = b] }
endspec
```

Any Σ-algebra A where the constants a, b, c, d are interpreted as pairwise different objects and where f takes the same value for c and d is a standard model of the specification and hence also a model of "**abstract** DEMO **wrt** $\equiv_{\approx_{Obs}}$". But such an algebra A is not a model of "**behaviour** DEMO **wrt** $\approx_{Obs}$" because it does not observationally satisfy (cf. Example 3.7.1) the second axiom: Since the only observable context for c and d is $f(z_{s'})$, the equation c = d is observationally satisfied by A but a = b is not observationally satisfied by A since a and b are different objects of observable sort.

Let us now come back to the more general situation. According to the relationship between a family $\approx = (\approx_A)_{A\in Alg(\Sigma)}$ of Σ-congruences and a factorizable equivalence relation ≡ established in Section 3.4, we can disregard whether we start from one point of view or from the other one. Hence, in the sequel we assume given a signature Σ, a set Cons of constructors and a couple (≈, ≡) consisting of a family $\approx = (\approx_A)_{A\in Alg(\Sigma)}$ of Σ-congruences and an equivalence relation ≡ between Σ-algebras such that ≡ is factorizable by ≈.

In order to obtain our central theorems we need some compatibility properties:

4.1 Compatibility Properties

The following proposition shows that an equivalence $\equiv$ which is factorizable by a congruence $\approx$ is compatible with the satisfaction relation w.r.t. $\approx$ and with the generation principle w.r.t. $\approx$.

Proposition 4.1 If $A \equiv B$ are equivalent Σ-algebras, then the following holds:

(1) For all Σ-formulas ϕ, $A \models_{\approx_A} \phi$ if and only if $B \models_{\approx_B} \phi$,

(2) A is $\approx_A$-finitely generated by Cons if and only if B is $\approx_B$-finitely generated by Cons.

Proof: (1) Since $A \equiv B$ are equivalent and since $\equiv$ is factorizable by $\approx$, $A/\approx_A$ and $B/\approx_B$ are isomporphic. By Proposition 3.10.1 we know that $A \models_{\approx_A} \phi$ iff $A/\approx_A \models \phi$. Since isomorphic algebras satisfy the same Σ-formulas we have $A/\approx_A \models \phi$ iff $B/\approx_B \models \phi$ and again by Proposition 3.10.1 we obtain $B/\approx_A \models \phi$ iff $B \models_{\approx_A} \phi$.
(2) The proof is done analogously to (1) using Proposition 3.10.2. ♦

4.2 Fully Abstract Algebras

An important role for the characterization of behavioural and abstractor semantics is played by fully abstract algebras. Following Milner's notion (cf. [Milner 77]) we define full abstractness with respect to a given family $\approx$ of Σ-congruences in the following way:

Definition 4.2

(1) A Σ-algebra A is called *fully abstract with respect to* $\approx$ (or briefly $\approx$-fully abstract) if $\approx_A$ is the equality on A, i.e. $\approx_A = =_A$.

(2) For any class $C \subseteq \text{Alg}(\Sigma)$ of Σ-algebras, $FA_{\approx}(C)$ denotes the subclass of $\approx$-fully abstract algebras of C, i.e. $FA_{\approx}(C) =_{def} \{A \in C \mid A \text{ is } \approx\text{-fully abstract}\}$. ♦

Example 4.3 If we consider the observational framework then elements of fully abstract algebras w.r.t. $\approx_{Obs}$ are equal if and only if they are observationally equivalent. For instance, in the SET example the algebra $P_{fin}(N)$ is fully abstract while N^* is not. ♦

Lemma 4.4 If $A \in \text{Alg}(\Sigma)$ is $\approx$-fully abstract, then the following holds:

(1) For all Σ-formulas ϕ, $A \models_{\approx_A} \phi$ if and only if $A \models \phi$,

(2) A is $\approx_A$-finitely generated by Cons if and only if A is finitely generated by Cons.

Proof: Since A is $\approx$-fully abstract, the congruence $\approx_A$ is the equality $=_A$ on A and therefore (1) and (2) are trivially satisfied. ♦

Definition 4.5 A family $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ of Σ-congruences is called *regular* if for any $A \in \text{Alg}(\Sigma)$ the quotient algebra $A/\approx_A$ is $\approx$-fully abstract (i.e. $\approx_{(A/\approx_A)} = =_{(A/\approx_A)}$). ♦

Example 4.6 Any family $\approx_{Obs}$ which is generated by a set Obs of observable sorts (cf. Example 3.9.1) is regular. (In fact it seems that all reasonable examples of families of Σ-congruences are regular.) ♦

Lemma 4.7 If $\approx$ is regular, then for any $A \in \mathrm{Alg}(\Sigma)$, $A \equiv A/{\approx_A}$.

Proof: Since $\approx$ is regular, $\approx_{A/\approx_A}$ is the equality $=_{A/\approx_A}$. Hence the algebras $A/{\approx_A}$ and $(A/{\approx_A})/{\approx_{A/\approx_A}}$ are isomorphic. Since $\equiv$ is factorizable by $\approx$ this means that $A \equiv A/{\approx_A}$ holds. ♦

4.3 Characterization Theorems

We are now prepared to prove our first central theorem which says that for any regular congruence $\approx$ the model class of a behavioural specification "**behaviour** SP **wrt** $\approx$ " coincides with the closure of the class of fully abstract models of the standard specification SP under any equivalence relation which is factorizable by $\approx$. In the sequel of this paper we assume that $\approx$ is regular.

Theorem 4.8 Let SP = (Σ, Cons, E) be a standard specification. Then:

Mod(**behaviour** SP **wrt** $\approx$) = $\mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(\mathrm{Mod(SP)}))$.

Proof: $\subseteq$: Let $A \in$ Mod(**behaviour** SP **wrt** $\approx$). Then, by Corollary 3.11, $A/{\approx_A} \in$ Mod(SP). Since $\approx$ is regular, $A/{\approx_A}$ is $\approx$-fully abstract and, by Lemma 4.7, $A \equiv A/{\approx_A}$ are equivalent. Hence $A \in \mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(\mathrm{Mod(SP)}))$.
$\supseteq$: Let $A \in \mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(\mathrm{Mod(SP)}))$. Then $A \equiv B$ for some $B \in \mathrm{FA}_{\approx}(\mathrm{Mod(SP)})$. For proving that $A \in$ Mod(**behaviour** SP **wrt** $\approx$) we have to show that $A \models_{\approx_A} \phi$ for any axiom $\phi \in E$ and that A is $\approx_A$-finitely generated by Cons. Now let $\phi \in E$ be an arbitrary axiom. Since B is a model of SP, $B \models \phi$ and since B is $\approx$-fully abstract, $B \models_{\approx_B} \phi$ (by Lemma 4.4.1). Then, since $A \equiv B$, Proposition 4.1.1 shows that $A \models_{\approx_A} \phi$. Analogously one can prove that A is $\approx_A$-finitely generated by Cons, using Lemma 4.4.2 and Proposition 4.1.2. ♦

The characterization of Theorem 4.8 gives rise to the definition of a semantical operator, called $\mathrm{Beh}_{\approx}$, which can be applied to any class C of Σ-algebras and which corresponds on the semantical level to the syntactic behaviour operator (in the same way as the semantic $\mathrm{Abs}_{\equiv}$ operator corresponds to the syntactic abstract operator):

Definition 4.9 For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras,

$\mathrm{Beh}_{\approx}(C) =_{\mathrm{def}} \mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(C))$. ♦

Using this definition we obtain: Mod(**behaviour** SP **wrt** $\approx$) = $\mathrm{Beh}_{\approx}(\mathrm{Mod(SP)})$ for any standard specification SP.
In Theorem 4.8 the semantics of behavioural specifications is characterized in terms of the semantical abstractor operator $\mathrm{Abs}_{\equiv}$. In the following we will show that vice versa the semantics of abstractor specifications can be characterized using the semantical behaviour operator $\mathrm{Beh}_{\approx}$. Hence there exists a duality between both kinds of semantics. Each one can be expressed by each other.
For the characterization of abstractor semantics given in Theorem 4.11 we use the following quotient construction:

Definition 4.10 For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras, $C/{\approx}$ denotes the class of all $\approx$-quotients of algebras of C, i.e. $C/{\approx} =_{\mathrm{def}} \{A/{\approx_A} \mid A \in C\}$. ♦

Theorem 4.11 Let SP = (Σ, Cons, E) be a standard specification. Then:

$$\mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \equiv) = \mathrm{Beh}_{\approx}(\mathrm{Mod}(SP)/\approx).$$

Proof: By definition, $\mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \equiv) = \mathrm{Abs}_{\equiv}(\mathrm{Mod}(SP))$. Since $\approx$ is regular, Lemma 4.7 shows that for any $A \in \mathrm{Mod}(SP)$, $A \equiv A/\!\approx_A$. Hence $\mathrm{Abs}_{\equiv}(\mathrm{Mod}(SP)) = \mathrm{Abs}_{\equiv}(\mathrm{Mod}(SP)/\approx)$. Moreover, $A/\!\approx_A$ is $\approx$-fully abstract and therefore $\mathrm{Mod}(SP)/\approx\ = \mathrm{FA}_{\approx}(\mathrm{Mod}(SP)/\approx)$. In summary we have $\mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \approx) = \mathrm{Abs}_{\equiv}(\mathrm{Mod}(SP)) = \mathrm{Abs}_{\equiv}(\mathrm{Mod}(SP)/\approx) = \mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(\mathrm{Mod}(SP)/\approx)) = \mathrm{Beh}_{\approx}(\mathrm{Mod}(SP)/\approx)$. ♦

Remark 4.12 Since $\mathrm{Beh}_{\approx}$ is defined for any class C of Σ-algebras one can extend the construction of behavioural specifications to arbitrary structured specifications SP of some ASL-like specification language by defining $\mathrm{Mod}(\textbf{behaviour}\ SP\ \textbf{wrt}\ \approx) =_{\mathrm{def}} \mathrm{Beh}_{\approx}(\mathrm{Mod}(SP))$. (Obviously, "abstract" can be extended as well.) Then, if we assume that the specification language contains the ASL-operator .+. for the combination of specifications and a quotient operator $./\approx$ with $\mathrm{Mod}(SP/\approx) =_{\mathrm{def}} \mathrm{Mod}(SP)/\approx$ (and if we assume that any algebra A is identified with its trivial quotient $A/\!=_A$) we can prove using the above theorems the following equations which show that behavioural specifications can be expressed by abstractor specifications and vice versa:

$$\mathrm{Mod}(\textbf{behaviour}\ SP\ \textbf{wrt}\ \approx) = \mathrm{Mod}(\textbf{abstract}\ (SP/\approx + SP)\ \textbf{wrt}\ \equiv) \text{ and}$$
$$\mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \equiv) = \mathrm{Mod}(\textbf{behaviour}\ (SP/\approx)\ \textbf{wrt}\ \approx). \text{ ♦}$$

4.4 Consequences of the Characterization Theorems

Theorem 4.8 has various consequences. It shows that behavioural semantics is a subclass of abstractor semantics, that fully abstract models of a standard specification are behavioural models and that a behavioural specification is consistent if and only if there exists a fully abstract model of the underlying standard specification. The following corollary states also corresponding properties of abstractor specifications which immediately follow from the definition of abstractor semantics.

Corollary 4.13 Let SP = (S, Cons, E) be a standard specification. Then:

(1) $\mathrm{Mod}(\textbf{behaviour}\ SP\ \textbf{wrt}\ \approx) \subseteq \mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \equiv)$,
(2) $\mathrm{FA}_{\approx}(\mathrm{Mod}(SP)) \subseteq \mathrm{Mod}(\textbf{behaviour}\ SP\ \textbf{wrt}\ \approx)$,
(3) $\mathrm{Mod}(\textbf{behaviour}\ SP\ \textbf{wrt}\ \approx) \neq \emptyset$ if and only if $\mathrm{FA}_{\approx}(\mathrm{Mod}(SP)) \neq \emptyset$,
(4) $\mathrm{Mod}(SP) \subseteq \mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \equiv)$,
(5) $\mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \equiv) \neq \emptyset$ if and only if $\mathrm{Mod}(SP) \neq \emptyset$.

As a further important consequence of the characterization theorems we obtain the following necessary and sufficient conditions under which behavioural semantics and abstractor semantics coincide. Note that a particular application of condition (3) leads to the theorem of [Reichel 85] since in the case of conditional equational axioms with observable premises the model class of a standard specification is closed under the observational quotient construction.

Corollary 4.14 Let SP = (Σ, Cons, E) be a standard specification. The following conditions are equivalent:

(1) $\mathrm{Mod}(\textbf{behaviour}\ SP\ \textbf{wrt}\ \approx) = \mathrm{Mod}(\textbf{abstract}\ SP\ \textbf{wrt}\ \equiv)$,
(2) $\mathrm{Mod}(SP) \subseteq \mathrm{Mod}(\textbf{behaviour}\ SP\ \textbf{wrt}\ \approx)$,
(3) $\mathrm{Mod}(SP)/\approx\ \subseteq \mathrm{Mod}(SP)$.

Proof: (1) $\Rightarrow$ (2): Since Mod(SP) $\subseteq$ Mod(**abstract** SP **wrt** $\equiv$) (2) follows immediately from (1).
(2) $\Rightarrow$ (3): We have to show that for any A $\in$ Mod(SP), $A/{\approx_A} \in$ Mod(SP) holds. Let A $\in$ Mod(SP) be an arbitrary model. (2) implies that A $\in$ Mod(**behaviour** SP **wrt** $\approx$). Then, by Corollary 3.11, we obtain $A/{\approx_A} \in$ Mod(SP).
(3) $\Rightarrow$ (1): $\subseteq$: By Corollary 4.14.1.
$\supseteq$: By Theorem 4.11, Mod(**abstract** SP **wrt** $\equiv$) = $\mathrm{Beh}_{\approx}$(Mod(SP)/$\approx$). Now (3) implies $\mathrm{Beh}_{\approx}$(Mod(SP)/$\approx$) $\subseteq$ $\mathrm{Beh}_{\approx}$(Mod(SP)) and, by Theorem 4.8 together with the definition of $\mathrm{Beh}_{\approx}$, we obtain $\mathrm{Beh}_{\approx}$(Mod(SP)) = Mod(**behaviour** SP **wrt** $\approx$). ♦

5 Theories of Behavioural and Abstractor Specifications

According to the generalization of the standard satisfaction relation to the satisfaction relation with respect to a congruence $\approx$ we will consider here for any behavioural or abstractor specification the theory with respect to $\approx$, i.e. the set of all Σ-formulas which are satisfied w.r.t. $\approx$ by all models of a behavioural or abstractor specification. We recall that we assume given also in this section a pair ($\approx$, $\equiv$) consisting of a regular family $\approx$ of Σ-congruences and an equivalence relation $\equiv$ on Alg(Σ) which is factorizable by $\approx$.

Definition 5.1 For any class C $\subseteq$ Alg(Σ) of Σ-algebras, $\mathrm{Th}_{\approx}$(C) denotes the set of all Σ-formulas ϕ which are satisfied w.r.t. $\approx$ by all algebras of C, i.e.

$$\mathrm{Th}_{\approx}(C) =_{\mathrm{def}} \{\phi \mid A \models_{\approx_A} \phi \text{ for all } A \in C\}.$$

$\mathrm{Th}_{\approx}$(C) is called *$\approx$-theory* or *behavioural theory* of C. In particular $\mathrm{Th}_{=}$(C) denotes the *standard theory* of C. ♦

Lemma 5.2 For any class C $\subseteq$ Alg(Σ) the following holds:
(1) $\mathrm{Th}_{\approx}(C) = \mathrm{Th}_{=}(C/{\approx})$,
(2) $\mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(C)) = \mathrm{Th}_{\approx}(C)$,
(3) $\mathrm{Th}_{\approx}(\mathrm{FA}_{\approx}(C)) = \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C))$.

Proof: (1) follows from Proposition 3.10.1, (2) follows from Proposition 4.1.1 and (3) is a consequence of Lemma 4.4.1. ♦

The next proposition shows that for classes of algebras which are constructed by the behaviour operator $\mathrm{Beh}_{\approx}$ or by the abstractor operator $\mathrm{Abs}_{\equiv}$, $\approx$-theories can be reduced to standard theories.

Proposition 5.3 For any class C $\subseteq$ Alg(Σ) the following holds:
(1) $\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(C)) = \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C))$,
(2) $\mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(C)) = \mathrm{Th}_{=}(C/{\approx})$.

Proof: (1): $\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(C))$ = (by definition of $\mathrm{Beh}_{\approx}$) $\mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(C)))$ = (by Lemma 5.2.2) $\mathrm{Th}_{\approx}(\mathrm{FA}_{\approx}(C))$ = (by Lemma 5.2.3) $\mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C))$.
(2): $\mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(C))$ = (by Lemma 5.2.2) $\mathrm{Th}_{\approx}(C)$ = (by Lemma 5.2.1) $\mathrm{Th}_{=}(C/{\approx})$ ♦

Proposition 5.3 leads immediately to the following theorem which shows that the $\approx$-theories of behavioural and abstractor specifications can be characterized by standard theories. In particular, the first part of Theorem 5.4 shows that the theory of a behavioural specification which is built on top of a standard specification SP is the same as the standard theory of the class of the fully abstract models of SP. Hence we can apply standard proof calculi for proving $\approx$-theorems over a behavioural specification as soon as we have a (standard) finite axiomatization of the class of the fully abstract models of SP. How such finite axiomatizations can be derived in the case of observable behaviour specifications is studied in [Bidoit, Hennicker 94].

Theorem 5.4 Let SP = (Σ, Cons, E) be a standard specification. Then for ($\approx$, $\equiv$) the following holds:
(1) $\mathrm{Th}_{\approx}$(Mod(**behaviour** SP **wrt** $\approx$)) = $\mathrm{Th}_{=}(\mathrm{FA}_{\approx}(\mathrm{Mod(SP)}))$,
(2) $\mathrm{Th}_{\approx}$(Mod(**abstract** SP **wrt** $\equiv$)) = $\mathrm{Th}_{=}(\mathrm{Mod(SP)}/\approx)$.

Proof: Follows from Theorem 4.8 and Proposition 5.3. ♦

Example 5.5 Let **behaviour** SP **wrt** $\approx_{\mathrm{Obs}}$ be an observational behaviour specification. Then $\mathrm{Th}_{\approx}$(Mod(**behaviour** SP **wrt** $\approx_{\mathrm{Obs}}$)) is called *observational theory of SP* because it consists of all formulas which are observationally satisfied by the observational models of the specification. Since in this case the fully abstract models satisfy an equation t = r if and only if they satisfy all equations c[t] = c[r] for all observable contexts c (cf. Example 3.7.1), one can prove observational theorems by using the standard theory of SP together with the context induction proof technique (cf. [Hennicker 91]). In [Bidoit, Hennicker 94] it is shown how an explicit use of context induction can be avoided.
As a concrete example consider the last two axioms add(x, add(y, s)) = add(y, add(x, s)) and add(x, add(x, s)) = add(x, s) of the SET specification. Even if these equations were omitted from the specification they would still be observational theorems w.r.t. the observable sorts "bool" and "elem" because for all observable contexts c the equations c[add(x, add(y, s))] = c[add(y, add(x, s))] and c[add(x, add(x, s))] = c[add(x, s)] can be derived already from the remaining SET axioms. ♦

6 Conclusion

We have presented a framework which clarifies the relationships between the two main approaches to observational semantics. In order to be applicable not only to the observational case but also to other specification formalisms we have introduced a general notion of behavioural specification and abstractor specification and we have seen that there exists a duality between both concepts which allows to characterize behavioural semantics in terms of an abstractor construction and vice versa provided that the underlying equivalence on algebras is factorizable. As an example of a factorizable equivalence we have considered the observational equivalence of algebras w.r.t. a fixed set of observable sorts where arbitrary inputs are allowed for the observable experiments. If we want to deal with a larger class of equivalences including the one of [Nivela, Orejas 88] where only observable inputs are considered all results of this paper can be generalized if we use instead of a family of congruences a family of partial congruences. Thereby a partial congruence over an algebra A $\in$ Alg(Σ) is given by a pair (A_0, $\approx_{A0}$) consisting of a subalgebra A_0 of A and a congruence $\approx_{A0}$ on A_0. The generalized satisfaction relation is then defined w.r.t. valuations that map variables to elements of A_0 and an algebra A is said to be fully abstract if A_0 =A and $\approx_{A0} = =_A$.

Our semantical characterization theorems lead to proof-theoretic considerations which show that behavioural theories of specifications can be reduced to standard theories of some classes of algebras. In particular, the behavioural theory of a behavioural specification is the same as the standard theory of the class of the fully abstract (standard) models of a specification. Hence we can prove behavioural properties of specifications using standard proof calculi if a finite axiomatization of the class of the fully abstract (standard) models of a specification is provided (see [Bidoit, Hennicker 94]).

Acknowledgement We would like to thank the referees of this paper for valuable remarks. This work is partially sponsored by the French-German cooperation programme PROCOPE, by the ESPRIT Working Group COMPASS and by the German DFG project SPECTRUM.

References

[Astesiano, Wirsing 89] E. Astesiano, M. Wirsing: Bisimulation in algebraic specifications. In: H. Ait-Kaci, M. Nivat (eds.): Resolution of Equations in Algebraic Structures, Vol. 1, Algebraic Techniques, London, Academic Press, 1-32, 1989.

[Bernot, Bidoit 91] G. Bernot, M. Bidoit: Proving the correctness of algebraically specified software: modularity and observability issues. Proc. AMAST '91, Techn. Report of the University of Iowa, 139-161, 1991.

[Bidoit, Hennicker 94] M. Bidoit, R. Hennicker: Proving behavioural theorems with standard first-order logic. Submitted for publication, 1994.

[Ehrig, Mahr 85] H. Ehrig, B. Mahr: Fundamentals of algebraic specification 1, EATCS Monographs on Theoretical Computer Science 6, Springer, Berlin, 1985.

[Hennicker 91] R. Hennicker: Context induction: a proof principle for behavioural abstractions and algebraic implementations. Formal Aspects of Computing 4 (3), 326-345, 1991.

[Knapik 91] T. Knapik: Specifications with observable formulae and observational satisfaction relation. In: M. Bidoit, C. Choppy (eds.): Recent Trends in Data Type Specification, Springer LNCS 655, 271-291, 1991.

[Milner 77] R. Milner: Fully abstract models of typed λ-calculi. Theoretical Computer Science 4, 1-22, 1977.

[Nivela, Orejas 88] P. Nivela, F. Orejas: Initial behaviour semantics for algebraic specifications. In: D. T. Sannella, A. Tarlecki (eds.): Proc. 5th Workshop on Algebraic Specifications of Abstract Data Types, Springer LNCS 332, 184-207, 1988.

[Orejas et al. 91] F. Orejas, M. Navarro, A. Sanchez: Implementation and behavioural equivalence: a survey. In: M. Bidoit, C. Choppy (eds.): Recent Trends in Data Type Specification, Springer LNCS 655, 93-125, 1991.

[Reichel 81] H. Reichel: Behavioural equivalence -- a unifying concept for initial and final specification methods. In: M. Arato, L. Varga (eds.): Math. Models in Comp. Systems, Proc. 3rd Hungarian Computer Science Conference, Budapest, 27-39, 1981.

[Reichel 85] H. Reichel: Initial restrictions of behaviour. IFIP Working Conference, The Role of Abstract Models in Information Processing, 1985.

[Sannella, Tarlecki 85] D. T. Sannella, A. Tarlecki: On observational equivalence and algebraic specification. Proc. TAPSOFT '85, Springer LNCS 185, 308-322, 1985.

[Sannella, Tarlecki 88] D. T. Sannella, A. Tarlecki: Toward formal development of programs from algebraic specifications: implementation revisited. Acta Informatica 25, 233-281, 1988.

[Schoett 87] O. Schoett: Data abstraction and correctness of modular programming. Ph. D. thesis, CST-42-87, University of Edinburgh, 1987.

[Wirsing 86] M. Wirsing: Structured algebraic specifications: a kernel language. Theoretical Computer Science 42, 123-249, 1986.