ELSEVIER Science of Computer Programming 25 (1995) 149–186

Science of Computer Programming

Behavioural and abstractor specifications

Michel Bidoit[a], Rolf Hennicker[b,*], Martin Wirsing[b]

[a]*LIENS, CNRS & Ecole Normale Supérieure, 45, Rue d'Ulm, 75230 Paris Cedex, France*
[b]*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 the specifications and, on the other hand, one can define observational semantics 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 the observational case. The distinction between the different observational concepts is reflected by our notions of behavioural specification and abstractor specification. We provide necessary and sufficient conditions for the semantical equivalence of both kinds of specifications and we show that behavioural specifications can be characterized by an abstractor construction and, vice versa, abstractor specifications can be characterized in terms of behavioural specifications. Hence, there exists a duality between both concepts which allows to express each one by the other. We also study the relationships to fully abstract algebras which can be used for a further characterization of behavioural semantics. Finally, we provide proof-theoretic results which show that behavioural theories of specifications can be reduced to standard theories of some classes of algebras.

Keywords: Algebraic specification; Behaviour; Abstraction; Partial congruence; Factorizable equivalence; Fully abstract algebra; Behavioural theory

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.

* Corresponding author.

SSDI 0167-6423(95)00014-3

Since the beginning of the 1980s, 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. [3,12,19,22]. 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 between algebras (cf. e.g. [21,23,24,26,27]. In [22] (and similarly in [19]) both semantical views are considered and it is shown that they are equivalent if the axioms of the 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 between the two semantical concepts in a more general framework which allows us to "abstract" from technical details (like e.g. observable contexts) appearing in observational approaches. For this purpose, we generalize the two concepts of observational semantics in the following way: instead of the observational equivalence of elements (in the following simply called observational equality) we use an arbitrary partial congruence relation for the interpretation of equations. This leads to our notion of *flat behavioural specification* which admits as models all algebras satisfying the axioms of a specification with respect to a given congruence relation. In order to be general enough for capturing the different views about which "inputs" should be allowed for "observable" experiments (e.g. arbitrary inputs as in [22] or only observable inputs as in [19]) we use *partial* congruences. As a first result, we show that the model class of a flat behavioural specification can be characterized by the class of all algebras whose "behavioural quotient" is a (standard) model of the underlying specification. This characterization leads to a straightforward extension of behavioural semantics to arbitrary structured specifications (of an ASL-like specification language). On the other hand, following the notion of an "abstractor" in [24], 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 between algebras. In order to establish the connection between behavioural and abstractor specifications, we consider only those equivalences on algebras which are "factorizable" (by a partial congruence relation between the elements of the algebras). As an example, we show that all observational equivalences of algebras w.r.t. a set of observable sorts are factorizable (for any choice of the input sorts).

As a central result of our approach, we obtain necessary and sufficient conditions for the semantical equivalence of behavioural and abstractor specifications. For instance, behavioural semantics coincides with abstractor semantics if and only if the (standard) model class of the underlying specification is closed under the "behavioural quotient" construction. Particular instantiations of this condition lead to the theorems of [19,22] (cf. above). Moreover, we show that in general behavioural semantics is included (i.e. is more restrictive) than abstractor semantics and we prove that

behavioural specifications can be characterized in terms of abstractor specifications and, conversely, abstractor specifications can be characterized in terms of behavioural specifications as well. Hence, there exists a duality between both kinds of specifications. Each one can be expressed by the other one. An important further characterization of behavioural semantics is provided using fully abstract algebras. It says that the model class of a behavioural specification can be characterized by the class of all algebras which are equivalent to a fully abstract (standard) model of the underlying specification. Hence, a behavioural specification has a model if and only if there exists a fully abstract (standard) model of the specification.

For the analysis of behavioural properties of specifications, we consider their *behavioural theories*. According to the generalized satisfaction relation with respect to a partial congruence, the behavioural theory of a specification is defined as the set of all formulas which are behaviourally satisfied (w.r.t. the given congruence) by all models of the specification. Since it is usually difficult to prove behavioural theorems, we need techniques which allow to reduce behavioural proofs to standard ones. For this purpose, we 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 show that the behavioural theory of an abstractor specification can be reduced to the standard theory of the class of the "behavioural quotients" of the (standard) models of the specification. These results provide the basis for the investigation of concepts which allow one to prove behavioural properties of specifications by standard proof techniques (cf. [4,6,7]).

The paper is organized as follows: Section 2 provides the underlying notions of our approach. In Section 3, behavioural specifications are introduced, and in Section 4, we consider abstractor specifications. The relationships between behavioural and abstractor specifications are studied in Section 5. In Section 6, we focus on fully abstract algebras and on the characterization of behavioural semantics by fully abstract algebras. In Section 7, behavioural theories are studied and finally, in Section 8, we end with some concluding remarks.

2. Basic concepts

2.1. Algebraic preliminaries

In this section, the basic notions of algebraic specifications which will be used hereafter are briefly summarized (for more details see e.g. [9,29].

A (many sorted) *signature* Σ is a pair (S, F), where S is a set of *sorts* and F is a set of *function symbols*. To each function symbol $f \in F$, a functionality $s_1, \ldots, s_n \to s$ with $s_1, \ldots, s_n, s \in S$, is associated. If $n = 0$, then f is called *constant* of sort s. A (*total*) Σ-*algebra* $A = ((A_s)_{s\in S}, (f^A)_{f\in F})$ over a signature $\Sigma = (S, F)$ consists of a family of carrier sets $(A_s)_{s\in S}$ and a family of functions $(f^A)_{f\in F}$ such that, if f has functionality $s_1, \ldots, s_n \to s$, then f^A is a (total) function from $A_{s_1} \times \cdots \times A_{s_n}$ to A_s (if $n = 0$, then f^A

denotes a constant object of A_s). A Σ-algebra A_0 is called *subalgebra* of a Σ-algebra A if $(A_0)_s \subseteq A_s$ for all $s \in S$, and if for all $f \in F$, the restriction of f^A to A_0 is the function f^{A_0}. Throughout this paper, we always assume that the carrier sets A_s of a Σ-algebra A are not empty for all $s \in S$. The category of all Σ-algebras with the usual notion of Σ-homomorphism is denoted by $\mathrm{Alg}(\Sigma)$. For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras, Iso(C) denotes the closure of C under Σ-isomorphism, i.e. $\mathrm{Iso}(C) =_{\mathrm{def}} \{A \in \mathrm{Alg}(\Sigma) \mid A$ is isomorphic to some $B \in C\}$.

Given an arbitrary S-sorted family $X = (X_s)_{s \in S}$ of sets X_s, $T(\Sigma, X)$ denotes the *Σ-term algebra* freely generated by X. An element $t \in T(\Sigma, X)_s$ is called *term* of sort s with variables in X. In several occasions, we will consider a subset $\mathrm{In} \subseteq S$ and we will choose $X_s = \emptyset$ for all $s \in S \backslash \mathrm{In}$ (and $X_s \neq \emptyset$ for all $s \in \mathrm{In}$). In that case, due to the requirement of nonempty carrier sets from above, we will always assume that the signature Σ is *sensible w.r.t.* In which means that for all $s \in S \backslash \mathrm{In}$ (and hence for all $s \in S$) there exists a term t of sort s which is built by function symbols of Σ and by the variables of the nonempty sets X_s with $s \in \mathrm{In}$. A term t without variable is called *ground term.* Given a Σ-algebra A, a *valuation* $\alpha: X \to A$ is a family of mappings $(\alpha_s: X_s \to A_s)_{s \in S}$. Any valuation $\alpha: X \to A$ uniquely extends to a Σ-homomorphism $I_\alpha: T(\Sigma, X) \to A$, called the *interpretation* associated to α and defined by:

(1) $I_{\alpha_s}(x) =_{\mathrm{def}} \alpha_s(x)$ if $x \in X_s$,

(2) $I_{\alpha_s}(f(t_1, \ldots, t_n)) =_{\mathrm{def}} f^A(I_{\alpha_s}(t_1), \ldots, I_{\alpha_{s_n}}(t_n))$ if f has functionality $s_1, \ldots, s_n \to s$.

2.2. Partial congruences

A *partial Σ-congruence* on a Σ-algebra A is a family $\approx_A = (\approx_{A,s})_{s \in S}$ of nonempty, partial equivalence relations (i.e. symmetric and transitive relations) $\approx_{A,s}$ on A_s compatible with the signature Σ, i.e. for all $f \in F$ with functionality $s_1, \ldots, s_n \to s$ and for all $a_i, b_i \in A_{s_i}$, if $a_i \approx_{A,s_i} b_i$, then $f^A(a_1, \ldots, a_n) \approx_{A,s} f^A(b_1, \ldots, b_n)$. In particular, if f is a constant of sort s, then $f^A \approx_{A,s} f^A$ holds.[1] A *Σ-congruence* $\approx_A$ is *total* if $a \approx_A a$ for all $a \in A$, i.e. if the relations $\approx_{A,s}$ are reflexive. The "definition domain" of a partial Σ-congruence $\approx_A$, denoted by $\mathrm{Dom}(\approx_A)$, is defined by $\mathrm{Dom}(\approx_A)_s =_{\mathrm{def}} \{a \in A_s \mid a \approx_A a\}$ for all $s \in S$.

Fact 2.1. *Let A be a Σ-algebra and $\approx_A$ be a partial Σ-congruence on A. Then:*

(1) $\mathrm{Dom}(\approx_A)$ *is a subalgebra of* A;

(2) *the restriction of* $\approx_A$ *to* $\mathrm{Dom}(\approx_A)$ *is a total Σ-congruence on* $\mathrm{Dom}(\approx_A)$.

Proof. (1) Let $f \in F$ with functionality $s_1, \ldots, s_n \to s$ and let $a_i \in \mathrm{Dom}(\approx_A)_{s_i}$ for $i = 1, \ldots, n$. Then $a_i \approx_A a_i$ for $i = 1, \ldots, n$. Since $\approx_A$ is compatible with the signature Σ, we have $f^A(a_1, \ldots, a_n) \approx_A f^A(a_1, \ldots, a_n)$, i.e. $f^A(a_1, \ldots, a_n) \in \mathrm{Dom}(\approx_A)_s$. Hence, the

[1] In the sequel, we will often omit the index s and write $a \approx_A b$ instead of $a \approx_{A,s} b$.

carrier sets $\mathrm{Dom}(\approx_A)_S$ together with the restrictions of the functions f^A to $\mathrm{Dom}(\approx_A)_{s_1} \times \cdots \times \mathrm{Dom}(\approx_A)_{s_n}$ (for any $f \in F$ with functionality $s_1 \times \cdots \times s_n \leftarrow s$) constitute a subalgebra $\mathrm{Dom}(\approx_A)$ of A. In particular, $\mathrm{Dom}(\approx_A)_s \neq \emptyset$ for all $s \in S$ since it is assumed that $\approx_{A,s} \neq \emptyset$ for all $s \in S$.

(2) is obvious. □

Notation. For any Σ-algebra A and partial Σ-congruence $\approx_A$ on A, $A/\approx_A$ denotes the quotient algebra of $\mathrm{Dom}(\approx_A)$ by $\approx_A$.

2.3. Formulas

In the sequel of this paper, we assume given an arbitrary but fixed family $X = (X_s)_{s \in S}$ of countably infinite sets X_s of variables of sort $s \in S$. 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 $\neg\phi$ and $\phi \wedge \psi$ are Σ-formulas.
(2) if ϕ is a Σ-formula, then $\forall x{:}s.\,\phi$ is a Σ-formula.
(3) If $\{\phi_i | i \in I\}$ is a countable family of Σ-formulas, then $\bigwedge_{i \in I} \phi_i$ is a Σ-formula.

All other logical operators such as finitary and infinitary disjunction " $\vee$ ", implication " $\Rightarrow$ ", and the existential quantifier "$\exists$" are defined as usual.[2] A Σ-formula is a (*finitary*) *first-order formula* if it is built only by the rules (0)–(2). A *Σ-sentence* is a Σ-formula which contains no free variable.

The (*standard*) *satisfaction relation*, denoted by $\models$, is inductively defined as follows: Let A be a Σ-algebra, $t, r \in T(\Sigma, X)_s$ be two terms of sort s, ϕ, ψ be two Σ-formulas, $\{\phi_i | i \in I\}$ be a countable family of Σ-formulas and $\alpha: X \to A$ be a valuation.

(0) $A, \alpha \models t = r$ holds if $I_\alpha(t) = I_\alpha(r)$.
(1) $A, \alpha \models \neg\phi$ holds if $A, \alpha \models_\approx \phi$ does not hold and $A, \alpha \models \phi \wedge \psi$ holds if both $A, \alpha \models \phi$ and $A, \alpha \models \psi$ hold.
(2) A, $\alpha \models \forall x{:}s.\phi$ holds if for all valuations $\beta: X \to A$ with $\beta(y) = \alpha(y)$ for all $y \neq x$, $A, \beta \models \phi$ holds.
(3) $A, \alpha \models \bigwedge_{i \in I} \phi_i$ holds if for all $i \in I$, $A, \alpha \models \phi_i$ holds.
(4) $A \models \phi$ holds if $A, \alpha \models \phi$ holds for all valuations $\alpha: X \to A$.

This definition is a straightforward extension of the satisfaction relation of the (many sorted) first-order predicate calculus given in [17,29] to infinitary formulas. Note that due to the assumption that algebras have no empty carrier sets, no pathological situation can occur.[3] Other solutions avoiding the nonempty carrier set

[2] In fact, our language of Σ-formulas coincides with (many-sorted) infinitary logic $L_{\omega_1\omega}$.

[3] Otherwise, according to the above definition, a Σ-algebra with an empty carrier set would satisfy any Σ-formula ϕ.

requirement are possible (for instance, considering only valuations from the set of the free variables of the given formula). But then one has to handle the problem with empty carrier sets when using proof systems as discussed for the equational calculus e.g in [11].

2.4. Specifications

A *basic (algebraic) specification* $\mathrm{SP} = \langle \Sigma, \mathrm{Ax} \rangle$ consists of a signature Σ and a set Ax of (possibly infinitary) Σ-sentences, called *axioms* of SP. The signature Σ is called the signature of SP and the *model class* of SP is defined by $\mathrm{Mod}(\mathrm{SP}) =_{\mathrm{def}} \{A \in \mathrm{Alg}(\Sigma) \mid A \models \phi \text{ for all } \phi \in \mathrm{Ax}\}$. We assume given a specification language for constructing large, structured specifications which has the following properties:

(1) To any specification SP is associated a signature, denoted by Sig(SP), and a class of models, denoted by Mod(SP), such that $\mathrm{Mod}(\mathrm{SP}) \subseteq \mathrm{Alg}(\mathrm{Sig}(\mathrm{SP}))$ and Mod(SP) is closed under Σ-isomorphism, i.e. $\mathrm{Mod}(\mathrm{SP}) = \mathrm{Iso}(\mathrm{Mod}(\mathrm{SP}))$.

(2) The language contains among arbitrary other specification building operators (cf. e.g. [24] or the operators of ASL [28]) the following two constructs:

(i) Basic specifications (cf. above).

(ii) An operator $+$ for the combination of specifications SP and SP′ such that

$$\mathrm{Sig}(\mathrm{SP} + \mathrm{SP}') = \mathrm{Sig}(\mathrm{SP}) \cup \mathrm{Sig}(\mathrm{SP}'),$$

$$\mathrm{Mod}(\mathrm{SP} + \mathrm{SP}') = \{A \in \mathrm{Alg}(\mathrm{Sig}(\mathrm{SP} + \mathrm{SP}')) \mid A|_{\mathrm{Sig}(\mathrm{SP})} \in \mathrm{Mod}(\mathrm{SP}) \text{ and } A|_{\mathrm{Sig}(\mathrm{SP}')} \in \mathrm{Mod}(\mathrm{SP}')\},$$

where $A|_{\mathrm{Sig}(\mathrm{SP})}$ ($A|_{\mathrm{Sig}(\mathrm{SP}')}$ resp.) denotes the restriction of A to Sig(SP) (Sig(SP′) resp.). If $\mathrm{Sig}(\mathrm{SP}) = \mathrm{Sig}(\mathrm{SP}')$, then $\mathrm{Mod}(\mathrm{SP} + \mathrm{SP}') = \mathrm{Mod}(\mathrm{SP}) \cap \mathrm{Mod}(\mathrm{SP}')$.

(Note that the model class of any basic specification and the model class $\mathrm{Mod}(\mathrm{SP} + \mathrm{SP}')$ of any combination of specifications SP and SP′ is closed under isomorphism (since it is assumed that Mod(SP) and Mod(SP′) are isomorphically closed). Two specifications SP and SP′ are *semantically equivalent*, denoted by $\mathrm{SP} = \mathrm{SP}'$, if $\mathrm{Sig}(\mathrm{SP}) = \mathrm{Sig}(\mathrm{SP}')$ and $\mathrm{Mod}(\mathrm{SP}) = \mathrm{Mod}(\mathrm{SP}')$.

2.5. Reachability constraints

In the definition of basic specifications, we have allowed infinitary formulas to be used as axioms. However, in practice we consider basic specifications with finitary axioms together with a reachability constraint that allows to express a generation principle for the elements of a Σ-algebra. We will see below that a reachability constraint is equivalent to a particular set of infinitary formulas and hence all results developed in this paper apply also to specifications with reachability constraints. Formally, we need the following definitions:

(1) A *reachability constraint* over a signature $\Sigma = (S, F)$ is a pair $\mathscr{R} = (S_{\mathscr{R}}, F_{\mathscr{R}})$ such that $S_{\mathscr{R}} \subseteq S$, $F_{\mathscr{R}} \subseteq F$ and for any $f \in F_{\mathscr{R}}$ with functionality $s_1, \ldots, s_n \to s$, the sort s belongs to $S_{\mathscr{R}}$. A sort $s \in S_{\mathscr{R}}$ is called *constrained sort* and a function symbol $f \in F_{\mathscr{R}}$ is

called *constructor symbol* (or briefly *constructor*). We assume also that for each constrained sort $s \in S_{\mathscr{R}}$, there exists at least one constructor in $F_{\mathscr{R}}$ with range s.

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

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

The notion of reachability constraint corresponds to the "reachable" construct of ASL (cf. [28]) and to the concept of generation constraint in [10]. In particular, requiring reachability induces a structural induction proof principle. The following fact shows that reachability constraints can be expressed by infinitary sentences.

Fact 2.2. *Let A be a Σ-algebra and $\mathscr{R} = (S_{\mathscr{R}}, F_{\mathscr{R}})$ be a reachability constraint over Σ. Then $A \models \mathscr{R}$ if and only if $A \models GEN_s$ for all $s \in S_{\mathscr{R}}$ where GEN_s is the following infinitary Σ-sentence:*

$$\mathrm{GEN}_s =_{\mathrm{def}} \forall x{:}s.\ \bigvee_{t \in (T_{\mathscr{R}})_s} \exists \mathrm{Var}(t).\, x = t.$$

Here $\exists \mathrm{Var}(t).x = t$ is an abbreviation for $\exists x_1 : s_1 \ldots \exists x_n : s_n.\ x = t$ where $x_1, \ldots, x_n$ are the variables occurring in t of (nonconstrained) sorts $s_1, \ldots, s_n$.

According to this fact, specifications with finitary axioms and reachability constraints can be defined as a particular kind of basic specifications with infinitary axioms as follows. Let Σ be a signature, $\mathscr{R} = (S_{\mathscr{R}}, F_{\mathscr{R}})$ be a reachability constraint over Σ and Ax be a set of finitary Σ-sentences. Then the triple $\mathrm{SP} = \langle \Sigma, \mathscr{R}, \mathrm{Ax} \rangle$ is, by definition, the basic specification $\langle \Sigma, \mathrm{Ax} \cup \{\mathrm{GEN}_s \mid s \in S_{\mathscr{R}}\} \rangle$.

2.6. Examples

Example 2.3. The following specification SET is a usual specification of finite sets over arbitrary elements. It introduces a reachability constraint which says that the constants "true" and "false" are constructors for the boolean values and the operations "empty" and 'add" are constructors for sets. The operation "iselem" defines the membership test on sets.

spec SET =
  sorts {bool, elem, set}
  functs {true: → bool, false: → bool, empty: → set, add: elem, set → set,
    iselem: elem, set → bool}
  constrained sorts {bool, set}

constructors {true, false, empty, add}
axioms {$\forall x, y$: elem, s: set.
$\text{iselem}(x, \text{empty}) = \text{false} \wedge \text{iselem}(x, \text{add}(x, s)) = \text{true} \wedge$
$[x \neq y \Rightarrow \text{iselem}(x, \text{add}(y, s)) = \text{iselem}(x, s)] \wedge$
$\text{add}(x, \text{add}(y, s)) = \text{add}(y, \text{add}(x, s)) \wedge \text{add}(x, \text{add}(x, s)) = \text{add}(x, s)\}$
endspec

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

Example 2.4. 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 "$\cdot$" of actions and processes and a nondeterministic choice operator "+". The operational semantics is given by a one-step (ternary) transition function where $(p \xrightarrow{a} p') = \text{true}$ 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}
functs {true: $\rightarrow$ bool, false: $\rightarrow$ bool,
nil: $\rightarrow$ process, $\ldots$: action, process $\rightarrow$ process, . + .: process,
process $\rightarrow$ process, $\rightarrow$:process, action, process $\rightarrow$ bool}
constrained sorts {bool, process}
constructors {true, false, nil, $\ldots$, . + .}
axioms {$\forall a$: action, p, p', q: process.
$(a.p \xrightarrow{a} p) = \text{true} \wedge$
$[(p \xrightarrow{a} p') = \text{true} \Rightarrow ((p + q \xrightarrow{a} p') = \text{true} \wedge (q + p \xrightarrow{a} p') = \text{true})]\}$
endspec

3. Behavioural specifications

3.1. Behavioural satisfaction relation

Behavioural specifications are a generalization of standard specifications which allow to describe the behaviour of data structures (or programs) using a behavioural equality. Formally, a behavioural equality is represented by a family $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ of (partial) Σ-congruences on the algebras of $\text{Alg}(\Sigma)$ where for any two elements a, b of a Σ-algebra A, $a \approx_A b$ holds whenever a and b are considered to be behaviourally indistinguishable. The underlying idea of behavioural specifications is to interpret the axioms of a specification according to the given behavioural equality. For this purpose, we generalize the standard satisfaction relation (cf. Section 2.3) to

the *behavioural satisfaction relation.* The difference to the standard case is twofold: First, we interpret the variables occurring in a Σ-formula not by all values of an algebra A but only by the values in the definition domain $\mathrm{Dom}(\approx_A)$ of the partial congruence relation $\approx_A$. (Remember that $\mathrm{Dom}(\approx_A)$ is a subalgebra of A, cf. Fact 2.1.) Secondly, the equality symbol " = " is not interpreted by the set-theoretic equality but by the given congruence relation $\approx_A$ (cf. also the notion of observational Σ-algebra in [16]).

Definition 3.1 (*Behavioural satisfaction relation*). Let $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ be a family of partial Σ-congruences and let A be a Σ-algebra. The *behavioural satisfaction relation w.r.t.* $\approx$, denoted by $\models_{\approx}$, is defined as follows: Let $t, r \in T(\Sigma, X)_s$ be two terms of sort s, ϕ, ψ be two Σ-formulas, $\{\phi_i | i \in I\}$ be a countable family of Σ-formulas and $\alpha: X \to \mathrm{Dom}(\approx_A)$ be a valuation. Then:

(0) $A, \alpha \models_{\approx} t = r$ holds if $I_\alpha(t) \approx_A I_\alpha(r)$.

(1) A, $\alpha \models_{\approx} \neg \phi$ holds if $A, \alpha \models_{\approx} \phi$ does not hold and $A, \alpha \models_{\approx} \phi \wedge \psi$ holds if both $A, \alpha \models_{\approx} \psi$ hold.

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

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

(4) $A \models_{\approx} \phi$ holds if $A, \alpha \models_{\approx} \phi$ holds for all valuations $\alpha: X \to \mathrm{Dom}(\approx_A)$.

A connection between the generalized and the standard satisfaction relation will be established in Section 3.3. The following fact shows how the standard satisfaction of reachability constraints translates to behavioural satisfaction using the characterization of reachability constraints by infinitary sentences in Fact 2.2.

Fact 3.2. *Let A be a Σ-algebra and $\mathscr{R} = (S_{\mathscr{R}}, F_{\mathscr{R}})$ be a reachability constraint over Σ. We say that A behaviourally satisfies $\mathscr{R}$, denoted by $A \models_{\approx} \mathscr{R}$, if and only if $A \models_{\approx} \mathrm{GEN}_s$ for all $s \in S_{\mathscr{R}}$ (cf. Fact 2.2) which is equivalent to the fact that for any $s \in S_{\mathscr{R}}$ and $a \in \mathrm{Dom}(\approx_A)_s$, there exists a constructor term $t \in T_{\mathscr{R}}$ of sort s and a valuation $\alpha: X' \to \mathrm{Dom}(\approx_A)$ such that $I_\alpha(t) \approx_A a$. This coincides exactly with our intuition that in the behavioural case, a generation principle should be interpreted up to (the given) behavioural equality.*

Example 3.3 (*Observational equalities*). The most important examples of partial congruences are *observational equalities* between the elements of an algebra. Formally, we assume given a signature $\Sigma = (S, F)$ and a distinguished set $\mathrm{Obs} \subseteq S$ of "observable" sorts which denote the carrier sets of observable values. Moreover, we assume given a set $\mathrm{In} \subseteq S$ of "input" sorts such that Σ is sensible w.r.t. In (cf. Section 2.1). All values of an input sort can be used as inputs for observable computations. Then two objects of an algebra are considered to be observationally equal 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 $c \in T(\Sigma, X_{\mathrm{In}} \cup Z)$ of observable sort that contains (besides input variables) exactly one variable $z_s \in Z$. Thereby, the S-sorted family X_{In} of sets of input variables is defined by $(X_{\mathrm{In}})_s =_{\mathrm{def}} \emptyset$ if $s \notin \mathrm{In}$, $(X_{\mathrm{In}})_s =_{\mathrm{def}} X_s$ if $x \in \mathrm{In}$ (where $X = (X_s)_{s \in S}$ is the generally assumed family of countably infinite sets X_s of variables of sort s) and $Z = (\{z_s\})_{s \in S}$ is an S-sorted family of singleton sets $\{z_s\}$ where z_s is a variable of sort s not occurring in $(X_{\mathrm{In}})_s$ for all $s \in S$. Now for any Σ-algebra $A \in \mathrm{Alg}(\Sigma)$, the *observational equality* of objects w.r.t. the observable sorts Obs and the input sorts In is the partial Σ-congruence $\approx_{\mathrm{Obs,In},A}$ defined as follows:

Let $A[X_{\mathrm{In}}]$ be the smallest subalgebra of A generated by Σ and X_{In}. The carrier sets of $A[X_{\mathrm{In}}]$ are defined by $(A[X_{\mathrm{In}}])_s =_{\mathrm{def}} \{a \in A_s \mid$ there exists a term $t \in T(\Sigma, X_{\mathrm{In}})_s$ and a valuation $\alpha: X_{\mathrm{In}} \to A$ such that $I_\alpha(t) = a\}$ and for any $f \in F, f^{A[X_{\mathrm{In}}]}$ is the restriction of f^A to $A[X_{\mathrm{In}}]$. Obviously, $A[X_{\mathrm{In}}]$ is a subalgebra of A (with nonempty carrier sets since Σ is sensible w.r.t. In). $A[X_{\mathrm{In}}]$ is the smallest subalgebra of A which is generated by (the interpretations of) the operations F over the values of input sorts. In other words, $A[X_{\mathrm{In}}] = R_{S \setminus \mathrm{In}}(A)$, where $R_{S \setminus \mathrm{In}}$ denotes the restriction functor considered e.g. in [10,24]. In particular, if $\mathrm{In} = \emptyset$ then $A[X_{\mathrm{In}}]$ is the finitely generated, smallest subalgebra of A.

Two elements $a, b \in A_s$ are observationally equal, i.e. $a \approx_{\mathrm{Obs,In},A} b$ if and only if both a and b belong to $A[X_{\mathrm{In}}]$ and for all observable Σ-contexts $c \in T(\Sigma, X_{\mathrm{In}} \cup Z)$ containing z_s and for all valuations $\alpha: X_{\mathrm{In}} \to A$, we have $I_{\alpha_a}(c) = I_{\alpha_b}(c)$ where $\alpha_a, \alpha_b: X_{\mathrm{In}} \cup \{z_s\} \to A$ are the unique extensions of α defined by $\alpha_a(z_s) =_{\mathrm{def}} a$, $\alpha_b(z_s) =_{\mathrm{def}} b$. Obviously, if s is an observable sort, then for all $a, b \in A[X_{\mathrm{In}}]$, $a \approx_{\mathrm{Obs,In},A} b$ is equivalent to $a = b$.

It is easy to show that $\approx_{\mathrm{Obs,In},A}$ is a partial Σ-congruence on A with $\mathrm{Dom}(\approx_{\mathrm{Obs,In},A}) = A[X_{\mathrm{In}}]$. The family $(\approx_{\mathrm{Obs,In},A})_{A \in \mathrm{Alg}(\Sigma)}$ of observational equalities will be denoted by $\approx_{\mathrm{Obs,In}}$. The behavioural satisfaction relation w.r.t. $\approx_{\mathrm{Obs,In}}$ is often called *observational satisfaction relation.* In particular, if $t = r$ is an equation, then from the definition it follows that $A \models_{\mathrm{Obs,In}} t = r$ if and only if $A \models c[\sigma(t)] = c[\sigma(r)]$ for all observable contexts $c \in T(\Sigma, X_{\mathrm{In}} \cup Z)$ and for all substitutions σ which replace the variables occurring in t and r by arbitrary terms of $T(\Sigma, X_{\mathrm{In}})$. (Here $c[\sigma(t)]$ and $c[\sigma(r)]$ denote the application of the context c to $\sigma(t)$ and $\sigma(r)$ which is simply defined by replacing the variable z_s in c by the terms $\sigma(t)$ and $\sigma(r)$, respectively.) In Example 3.5, we will discuss three different kinds of observational satisfaction relations according to different choices of the set In of input sorts.

The following fact provides a useful alternative definition for the observational equality:

Fact 3.4. *The observational equality of objects defined in Example* 3.3 *could be equivalently defined using instead of the countably infinite sets X_s of variables for any Σ-algebra A the carrier sets A_s themselves as variable sets and the interpretation $I_{\mathrm{id}}: T(\Sigma, A_{\mathrm{In}}) \to A$, where $\mathrm{id}_s: A_s \to A_s$ is the identity on A_s for all $s \in \mathrm{In}$. For any $s \in S$,*

let $(A[A_{\mathrm{In}}])_s =_{\mathrm{def}} \{a \in A_s \mid$ *there exists a term* $t \in T(\Sigma, A_{\mathrm{In}})_s$ *such that* $I_{\mathrm{id}}(t) = a\}$. *Then for all* $a, b \in A_s$, $a \approx_{\mathrm{Obs,In},A} b$ *holds if and only if both* a *and* b *belong to* $A[A_{\mathrm{In}}]$ *and for all observable* Σ*-contexts* $c \in T(\Sigma, A_{\mathrm{In}} \cup Z)$ *containing* z_s, $I_{\mathrm{id}_a}(c) = I_{\mathrm{id}_b}(c)$, *where* id_a, $\mathrm{id}_b : A_{\mathrm{In}} \cup \{z_s\} \leftarrow A$ *are the unique extensions of* id *defined by* $\mathrm{id}_a(z_s) =_{\mathrm{def}} a$, $\mathrm{id}_b(z_s) =_{\mathrm{def}} b$. *Obviously,* $\mathrm{Dom}(\approx_{\mathrm{Obs,In},A}) = A[A_{\mathrm{In}}]$.

Example 3.5 (*Particular observational equalities*). Let $\Sigma = (S, F)$ be a signature and $\mathrm{Obs} \subseteq S$ be a set of observable sorts. For any Σ-algebra A, we distinguish the following important cases of observational equalities for the elements of A:

(1) If we choose $\mathrm{In} = S$, i.e. the elements of all carrier sets of the algebra A can be used as input for observable computations, then $\mathrm{Dom}(\approx_{\mathrm{Obs},S,A}) = A$ and $\approx_{\mathrm{Obs},S,A}$ is a total Σ-congruence which is the behavioural equality used e.g. in [4,22]. In particular, the behavioural satisfaction relation w.r.t. $\approx_{\mathrm{Obs},S}$ coincides with the behavioural satisfaction relation used in [4] and, if we restrict to conditional equations, with the notion of behavioural validity in [22].

(2) If we choose $\mathrm{In} = \mathrm{Obs}$, i.e. only the observable elements of A can be used as input for observable computations, then $\mathrm{Dom}(\approx_{\mathrm{Obs,Obs},A})$ consists of all values that can be generated over the observable elements of A by the operations F and $\approx_{\mathrm{Obs,Obs},A}$ is a partial Σ-congruence. The behavioural satisfaction relation w.r.t. $\approx_{\mathrm{Obs,Obs}}$ is the one used in [19] for the behavioural satisfaction of equations. The advantage here is that nonobservable junk (i.e. values which are not reachable from the observable ones) will not be considered for the satisfaction of formulas and hence cannot cause problems, for instance, with respect to the correctness of implementations (cf. e.g. [20]).

(3) If we choose $\mathrm{In} = \emptyset$, then observable computations are always represented by ground terms of observable sort. In this case $\approx_{\mathrm{Obs},\emptyset,A}$ is a partial Σ-congruence whose definition domain $\mathrm{Dom}(\approx_{\mathrm{Obs},\emptyset,A})$ is the finitely generated, smallest subalgebra of A. To our knowledge, the corresponding behavioural satisfaction relation w.r.t. $\approx_{\mathrm{Obs},\emptyset}$ is not used in the literature for arbitrary (not necessarily finitely generated) Σ-algebras although it provides an interesting candidate for further applications because it eliminates not only problems with nonobservable junk but also problems that can occur with respect to observable junk (cf. Example 3.9(3)).

Example 3.6 (*Congruences generated by a set of equations*). Let $\Sigma = (S, F)$ be a signature and $\mathrm{In} \subseteq S$ be a set of input sorts such that Σ is sensible w.r.t. In. Moreover, let E be a set of equations between Σ-terms. The set E generates on any Σ-algebra A a partial Σ-congruence $\approx_{E,\mathrm{In},A}$ (relative to the input sorts In) as follows:

Let $A[X_{\mathrm{In}}]$ be (as in Example 3.3) the reachable part of A which is generated over the values of input sorts. Then $\approx_{E,\mathrm{In},A}$ is the partial Σ-congruence on A with definition domain $\mathrm{Dom}(\approx_{E,\mathrm{In},A}) = A[X_{\mathrm{In}}]$ such that the restriction of $\approx_{E,\mathrm{In},A}$ to $A[X_{\mathrm{In}}]$ is the (total) Σ-congruence on $A[X_{\mathrm{In}}]$ generated by the equations E (in the usual way).

The family $(\approx_{E,\mathrm{In},A})_{A \in \mathrm{Alg}(\Sigma)}$ will be denoted by $\approx_{E,\mathrm{In}}$. A Σ-algebra A behaviourally satisfies w.r.t. $\approx_{E,\mathrm{In}}$ a Σ-formula ϕ if it satisfies ϕ up to junk elements (i.e. elements

which are not generated over the input values) and up to the identification of elements w.r.t. the equations E.

Example 3.7 (*Strong bisimulation*). The notion of strong bisimulation is an example of a congruence on CSO (cf. Example 2.4). For any algebra A over the signature of CSO, a *simulation equivalence* $\approx_{\text{sim},A}$ is a relation such that for all $p, q \in A_{\text{process}}$:

$$p \approx_{\text{sim},A} q \text{ if for all } p' \in A_{\text{process}},\ s \in (A_{\text{action}})^*,$$

$$[(p\ {}^s\!\rightarrow^{*A} p') = \text{true}^A \Leftrightarrow \exists q' \in A_{\text{process}},\ p' \approx_{\text{sim},A} q' \text{ and } (q\ {}^s\!\rightarrow^{*A} q') = \text{true}^A]$$

and vice versa for all $q' \in A_{\text{process}}$, $s \in (A_{\text{action}})^*$,

$$[(q\ {}^s\!\rightarrow^{*A} q') = \text{true}^A \Leftrightarrow \exists p' \in A_{\text{process}},\ q' \approx_{\text{sim},A} p' \text{ and } (p\ {}^s\!\rightarrow^{*A} p') = \text{true}^A].$$

Thereby, $\rightarrow^{*A}$ denotes the reflexive and transitive closure of $\rightarrow^A$ which is the least relation on $A_{\text{process}} \times (A_{\text{action}})^* \times A_{\text{process}}$ with the following properties:

(1) $p\ {}^{\varepsilon}\!\rightarrow^{*A} p$,
(2) If $p\ {}^a\!\rightarrow^A q$, then $p\ {}^{\langle a\rangle}\!\rightarrow^{*A} q$,
(3) If $p\ {}^{s1}\!\rightarrow^{*A} q$ and $q\ {}^{s2}\!\rightarrow^{*A} r$, then $p\ {}^{s1 \circ s2}\!\rightarrow^{*A} r$,

where $\langle \cdot \rangle$ denotes the construction of singleton sequences and $\circ$ denotes the concatenation of sequences. For the carrier sets A_{bool} and A_{action}, $\approx_{\text{sim},A}$ is defined as the (standard) equality of elements.

According to [2] (see also [1]) we have the following fact: Let A be a (standard) model of CSO. Then any simulation equivalence is a (total) congruence (w.r.t. the signature of CSO) and the simulation equivalences on A form a complete lattice. For any model A of CSO, the coarsest simulation congruence on A is called *strong bisimulation* on A and is denoted by $\approx_{\text{bisim},A}$. If for A we choose the Herbrand model $H(\text{CSO})$ of CSO with the set $T_{\mathscr{R}}$ of constructor terms as carrier sets and the following interpretation of " $\rightarrow$ " by valid transitions:

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

then $\approx_{\text{bisim},H(\text{CSO})}$ is Milner's strong bisimulation congruence.

The following equations are behaviourally satisfied by $H(\text{CSO})$ (but are obviously not satisfied in the standard sense):

$$H(\text{CSO}) \models_{\approx_{\text{bisim}}} \forall p, q, r\text{: process.}$$
$$p + (q + r) = (p + q) + r \wedge p + q = q + p \wedge p + p = p \wedge p + \text{nil} = p.$$

Here $\approx_{\text{bisim}}$ is the family of total Σ-congruences where $\approx_{\text{bisim},A}$ is defined as above if A is a model of CSO and $\approx_{\text{bisim},A}$ is defined as an arbitrary congruence (for instance as the set-theoretic equality $=_A$) otherwise.

3.2. Flat behavioural specifications

For any basic specification $\text{SP} = \langle \Sigma, \text{Ax} \rangle$ (cf. Section 2.4) and any family $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ of partial Σ-congruences, one can construct a (flat) behavioural

specification where instead of the standard satisfaction relation the behavioural satisfaction relation w.r.t. $\approx$ is used for the interpretation of the axioms.

Definition 3.8 (*Flat behavioural specifications*). Let $SP = \langle \Sigma, Ax \rangle$ be a basic specification and let $\approx = (\approx_A)_{A \in Alg(\Sigma)}$ be a family of partial Σ-congruences. Then:

(1) The expression **behaviour** SP **wrt** $\approx$ is a *flat behavioural specification.*

(2) The signature of a behavioural specification is given by Σ and its model class is defined by $Mod(\textbf{behaviour}\ SP\ \textbf{wrt} \approx) =_{def} \{A \in Alg(\Sigma) | A \models_{\approx} \phi \text{ for all } \phi \in Ax\}$.

Example 3.9 (*Observable behaviour specifications*). Let $\Sigma = (S, F)$ be a signature, $Obs \subseteq S$ be a set of observable sorts and $In \subseteq S$ be a set of input sorts such that Σ is sensible w.r.t. In. Let $\approx_{Obs,In}$ be the family of observational equalities induced by Obs and In (cf. Example 3.3). Then the specification **behaviour** SP **wrt** $\approx_{Obs,In}$ specifies the *observable behaviour* of a data structure (or a program) by means of the corresponding observational satisfaction relation. Again we can distinguish three cases for the choice of the input sorts In:

(1) If $In = S$, then the semantics of an observable behaviour specification coincides with the behavioural semantics of specifications used in [4,22]. Note that if $Obs = In = S$, there is no difference between standard semantics and behavioural semantics of specifications.

(2) If $In = Obs$, then the semantics of an observable behaviour specification coincides with the behavioural semantics of specifications in the sense of [19]. In this case, the variables occurring in the axioms of a specification will not be interpreted by nonobservable junk. As a concrete example, we can construct the behavioural specification

behaviour SET **wrt** $\approx_{Obs,Obs}$

on top of the standard specification SET of sets (cf. Example 2.3) where $Obs = \{bool, elem\}$ is the set of observable sorts and the set of input sorts as well. Since the sort "set" is not observable, sets can only be observed via the membership test "iselem". For instance, the algebra $\mathbb{N}^*$ of finite sequences of natural numbers is a model of this behavioural specification of sets. In particular, $\mathbb{N}^*$ 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 $\mathbb{N}^*$ does not satisfy in the standard sense the last two SET axioms and hence is not a model of the (basic) specification SET.

(3) If $In = \emptyset$, then the variables occurring in the axioms of a specification will only be interpreted by values which are reachable by the operations F, i.e. are represented by ground terms over the signature Σ. This means that neither nonobservable nor observable junk will be considered for the satisfaction of the axioms of a specification.

As an example, consider a specification NAT of natural numbers that contains an axiom $\forall x : nat.\ equal(0, succ(x)) = false$ for specifying the equality of natural numbers and where all values are defined as being observable. The algebra $\mathbb{Z}$ of the integers is

a model of **behaviour** NAT **wrt** $\approx_{\{\text{bool, nat}\},\emptyset}$. Indeed since $\text{In} = \emptyset$, we have $\text{Dom}(\approx_{\{\text{bool, nat}\},\emptyset,\mathbb{Z}}) = \mathbb{N}$. Therefore, the equation $\forall x : \text{nat}.\ \text{equal}(0, \text{succ}(x)) = \text{false}$ is satisfied w.r.t. $\approx_{\{\text{bool, nat}\},\emptyset}$ since the universal quantifier ranges over values of $\mathbb{N}$; thus the variable x cannot be interpreted by the (observable junk) value -1 (if it could, then the above axiom would be violated and hence the integers would not be admitted as an implementation of the natural numbers).

Example 3.10. Let CSO1 be the specification CSO (cf. Example 2.4) enriched by the following axiom:

$$\forall p, q, r\text{: process.}\quad p + (q + r) = (p + q) + r \wedge p + q = q + p \wedge p + p = p \wedge p + \text{nil} = p.$$

Then the behavioural specification **behaviour** CSO1 **wrt** $\approx_{\text{bisim}}$ describes all algebras (over the signature of CSO) that behaviourally satisfy the axioms of CSO1 w.r.t. the strong bisimulation congruence. For instance, the Herbrand model $H(\text{CSO})$ described in Example 3.7 is a model of the behavioural specification **behaviour** CSO1 **wrt** $\approx_{\text{bisim}}$.

3.3. Relating behavioural satisfaction and standard satisfaction

The following theorem establishes an important connection between the behavioural satisfaction w.r.t. $\approx$ and the standard satisfaction of Σ-formulas. The theorem and its various consequences have recently been extended to higher-order logic in [14].

Theorem 3.11. *Let $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ be a family of partial Σ-congruences. For any Σ-algebra A and any Σ-formula ϕ the following holds:*

$$A \models_{\approx} \phi \quad \textit{if and only if} \quad A/\approx_A \models \phi,$$

where $A/\approx_A$ denotes the quotient algebra of $\text{Dom}(\approx_A)$ *by* $\approx_A$ (cf. *Section* 2.2). *In particular, for any reachability constraint $\mathscr{R}$ over Σ, $A \models_{\approx} \mathscr{R}$ if and only if $A/\approx_A \models \mathscr{R}$.*

Proof. Let A be a Σ-algebra. For the proof the the theorem we use the following lemma (*):

(*) *For all Σ-formulas ϕ and for all valuations $\alpha : X \to \text{Dom}(\approx_A)$: $A, \alpha \models_{\approx} \phi$ if and only if $A/\approx_A, \pi \circ \models \phi$ where $\pi : \text{Dom}(\approx_A) \to A/\approx_A$ is the canonical epimorphism.*

Let us first show how to prove the theorem using lemma (*): Assume that $A \models_{\approx} \phi$ holds. We have to show that $A/\approx_A \models \text{L}\phi$, i.e. $A/\approx_A, \beta \models \phi$ for all valuations $\beta : X \to A/\approx_A$. Let $\beta : X \to A/\approx_A$ be an arbitrary valuation. Then there exists a valuation $\alpha : X \to \text{Dom}(\approx_A)$ such that $\beta = \pi \circ \alpha$. .Since we have assumed $A \models_{\approx} \phi$, in particular $A, \alpha \models_{\approx} \phi$ holds. Then using (*) we obtain as desired that $A/\approx_A, \beta \models \phi$ holds. The converse direction can be shown similarly.

It remains to prove the above lemma (*) by induction on the form of ϕ:

Case 0: Let $t = r$ be an equation with $t, r \in T(\Sigma, X)$ and let $\alpha: X \to \mathrm{Dom}(\approx_A)$ be an arbitrary valuation. Then $A, \alpha \models_{\approx} t = r$ iff $I_\alpha(t) \approx_A I_\alpha$ iff $[I_\alpha(t)] = [I_\alpha(r)]$ iff $I_{\pi\circ\alpha}(t) = I_{\pi\circ\alpha}(r)$ iff $A/\approx_A, \pi \circ \alpha \models t = r$.

Case 1: For formulas of the form $\neg\phi$ and $\phi \wedge \psi$, the desired result follows immediately from the induction hypothesis.

Case 2: Consider an arbitrary formula of the form $\forall x{:}s.\phi$ and an arbitrary valuation $\alpha: X \to \mathrm{Dom}(\approx_A)$. Then $A, \alpha \models_{\approx} \forall x{:}s.\phi$ iff for all valuations $\beta: X \to \mathrm{Dom}(\approx_A)$ with $\beta(y) = \alpha(y)$ for $y \neq x$, $A, \beta \models_{\approx} \phi$ and $A/\approx_A, \pi \circ \alpha \models \forall x{:}s.\phi$ iff for all valuations $\gamma: X \to A/\approx_A$ with $\gamma(y) = (\pi \circ \alpha)(y)$ for $y \neq x$, $A/\approx_A, \gamma \models \phi$. Hence, we have to show:

For all valuations $\beta: X \to \mathrm{Dom}(\approx_A)$ with $\beta(y) = \alpha(y)$ for $y \neq x$, $A, \beta \models_{\approx} \phi$ iff for all valuations $\gamma: X \to A/\approx_A$ with $\gamma(y) = (\pi \circ \alpha)(y)$ for $y \neq x$, $A/\approx_A, \gamma \models \phi$.

Proof of "$\Rightarrow$": Let $\gamma: X \to A/\approx_A$ be a valuation with $\gamma(y) = (\pi \circ \alpha)(y)$ for $y \neq x$. Then define $\beta: X \to \mathrm{Dom}(\approx_A)$ by $\beta(x) =_{\mathrm{def}} a$ with $[a] = \gamma(x)$ and $\beta(y) = \alpha(y)$ for $y \neq x$. Hence, $\gamma = \pi \circ \beta$. By assumption, $A, \beta \models_{\approx} \phi$. Then, by induction hypothesis, we have $A/\approx_A, \pi \circ \beta \models \phi$. Thus, $A/\approx_A, \gamma \models \phi$.

Proof of "$\Leftarrow$": Let $\beta: X \to \mathrm{Dom}(\approx_A)$ be a valuation with $\beta(y) = \alpha(y)$ for $y \neq x$. Then define $\gamma: X \to A/\approx_A$ by $\gamma =_{\mathrm{def}} \pi \circ \beta$. Hence, $\gamma(y) = (\pi \circ \alpha)(y)$ for $y \neq x$. By assumption, $A/\approx_A, \gamma \models \phi$. Then, by induction hypothesis, we have $A, \beta \models_{\approx} \phi$.

Case 3: For formulas of the form $\bigwedge_{i \in I} \phi_i$, the desired result follows immediately from the induction hypothesis.

This completes the proof. □

3.4. Behavioural specifications: the general case

Up to now, behavioural specifications are always built on top of basic specifications. In this section, we extend this concept to the arbitrary structured specifications of the generally assumed specification language (cf. Section 2.4). The underlying idea for the extension is provided by the following characterization of the model class of a flat behavioural specification (which is an immediate consequence of Theorem 3.11):

Corollary 3.12. *Let* $\mathrm{SP} = \langle \Sigma, \mathrm{Ax} \rangle$ *be a basic specification and let* $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ *be a family of partial* Σ*-congruences. Then*:

$$\mathrm{Mod}(\textbf{behaviour SP wrt} \approx) = \{A \in \mathrm{Alg}(\Sigma) \mid A/\approx_A \in \mathrm{Mod}(\mathrm{SP})\}.$$

Proof. By Theorem 3.11, for any Σ-algebra A and any axiom $\phi \in \mathrm{Ax}$, $A \models_{\approx} \phi$ if and only if $A/\approx_A \models \phi$. Hence, $A \in \mathrm{Mod}(\textbf{behaviour}\ \mathrm{SP}\ \textbf{wrt}\ \approx)$ if and only if $A/\approx_A \in \mathrm{Mod}(\mathrm{SP})$. □

Corollary 3.12 says that a Σ-algebra A is a model of a behavioural specification if and only if its quotient $A/\approx_A$ is a model of the underlying specification SP. This result points out the crucial role of quotients in the behavioural framework. Indeed, since the quotient of a Σ-algebra A identifies all elements of A which are behaviourally equal, i.e. are indistinguishable "from the outside", it can be considered as the "black box view" or as the "behaviour" of A.

Definition 3.13 (*Behaviour*). Let $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ be a family of partial Σ-congruences. For any Σ-algebra A, the *behaviour of A w.r.t.* $\approx$ is the quotient algebra $A/\approx_A$.

The following definition describes two simple properties which should be satisfied by any reasonable behavioural equality $\approx$. First, we expect that two isomorphic algebras have (up to isomorphism) the same behaviour w.r.t. $\approx$. Secondly, the construction of the behaviour of an algebra should be idempotent which means that the behaviour of the behaviour of a Σ-algebra A is (up to isomorphism) the same as the behaviour of A.

Definition 3.14 (*Isomorphism compatibility and weak regularity*). A family $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ of partial Σ-congruences is called:

(1) *isomorphism compatible* if for any two isomorphic Σ-algebras A and B, the behaviours $A/\approx_A$ and $B/\approx_B$ are isomorphic;

(2) *weakly regular* if for any Σ-algebra A, its behaviour $A/\approx_A$ is isomorphic to $(A/\approx_A)/\approx_{(A/\approx_A)}$.[4]

Example 3.15. Any family $\approx_{\mathrm{Obs,In}}$ which is generated by a set Obs of observable sorts and a set In of input sorts (cf. Example 3.3) and any family $\approx_{\mathrm{E,In}}$ which is generated by a set E of equations and a set In of input sorts (cf. Example 3.6) is isomorphism compatible and weakly regular. (The only nonobvious proof is the weak regularity of $\approx_{\mathrm{Obs,In}}$ which will be provided in Example 6.7 where it is shown that $\approx_{\mathrm{Obs,In}}$ is even regular.)

General Assumption 1. In the following, we assume that $\approx$ always denotes an isomorphism compatible family of partial Σ-congruences. (Weak regularity will technically not be needed before Section 5.)

[4] The notion of regularity will be introduced in Section 6.

The characterization of Corollary 3.12 gives rise to the definition of a semantical behaviour operator, denoted by $\mathrm{Beh}_{\approx}$, which constructs for any class C of Σ-algebras, the class of all Σ-algebras whose behaviour belongs to C.

Definition 3.16 (*Behaviour operator*). For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras, $\mathrm{Beh}_{\approx}(C) =_{\mathrm{def}} \{A \in \mathrm{Alg}(\Sigma) \mid A/\approx_A \in C\}$.

We are now able to extend the given specification language by a construct for behavioural specifications built on top of any arbitrary (structured) specification.

Definition 3.17 (*Behavioural specification*). Let SP be a specification with signature Σ.

(1) The expression **behaviour** SP **wrt** $\approx$ is a *behavioural specification.*

(2) The signature and the model class of a behavioural specification are given by:
$\mathrm{Sig}(\textbf{behaviour SP wrt} \approx) =_{\mathrm{def}} \mathrm{Sig}(\mathrm{SP})$,
$\mathrm{Mod}(\textbf{behaviour SP wrt} \approx) =_{\mathrm{def}} \mathrm{Beh}_{\approx}(\mathrm{Mod}(\mathrm{SP}))$.

The model class of a behavioural specification consists, by definition, of all algebras whose behaviour belongs to Mod(SP) and hence fulfills the requirements of SP. Thus, a behavioural specification describes all algebras which can be considered as "behaviourally correct realizations" of the models of SP (cf. [8]). Note that the model class of a behavioural specification is closed under isomorphism since $\approx$ is assumed to be isomorphism compatible and that, by Corollary 3.12, the above definition is consistent with Definition 3.8.

The following example shows that it may happen that some models of a specification SP are not models of a behavioural specification **behaviour** SP **wrt** $\approx$.

Example 3.18. Consider the following specification DEMO:

spec DEMO =
  sorts $\{s\}$
  functs $\{a, b: \rightarrow s\}$
  axioms $\{a \neq b\}$
endspec

The specification DEMO has a model where a and b are interpreted as different objects. Now consider the observational equality $\approx_{\emptyset,\{s\}}$ generated by the empty set of observable sorts and by the input sort s (the choice of the input sorts is not relevant here). Then there is no observation that allows to distinguish elements; hence all elements (in particular a and b) are observationally equal. Therefore, there is no algebra whose behaviour w.r.t. $\approx_{\emptyset,\{s\}}$ satisfies $a \neq b$, i.e. is a model of DEMO. Hence, $\mathrm{Mod}(\textbf{behaviour DEMO wrt} \approx_{\emptyset,\{s\}}) = \emptyset$ while $\mathrm{Mod}(\mathrm{DEMO}) \neq \emptyset$. In this case, we say that the specification DEMO is behaviourally inconsistent w.r.t. $\approx_{\emptyset,\{s\}}$. Intuitively, the reason for this behavioural inconsistency is that the requirements of the specification DEMO, asking for a and b to be different, contradict the chosen observational equality which does not provide any observational computation for distinguishing

elements. For obtaining a behaviourally consistent specification, one has either to omit the axiom $a \neq b$ or one has to consider the sort s to be observable.

In general, a class C of Σ-algebras is called behaviourally consistent w.r.t $\approx$ if the behaviour $A/\approx_A$ of any algebra A in C is also an algebra of C, i.e. $C \subseteq \mathrm{Beh}_{\approx}(C)$. In particular, a specification SP is behaviourally consistent if the behaviour of any model of SP is also a model of SP (and hence fulfills the requirements of SP). This means that the properties required by SP are compatible with the chosen behavioural equality $\approx$.

Definition 3.19 (*Behavioural consistency*). Let $C \subseteq \mathrm{Alg}(\Sigma)$ be a class of Σ-algebras and SP be a specification with signature Σ. Then:

(1) C is *behaviourally consistent w.r.t.* $\approx$ if $C \subseteq \mathrm{Beh}_{\approx}(C)$.

(2) SP is *behaviourally consistent w.r.t.* $\approx$ if $\mathrm{Mod}(\mathrm{SP}) \subseteq \mathrm{Mod}(\textbf{behaviour}\ \mathrm{SP}\ \textbf{wrt}\ \approx)$.

Using the following quotient operator which extends the construction of the behavioural quotient of a Σ-algebra A to classes C of Σ-algebras, we can formulate an obvious equivalent condition for behavioural consistency.

Definition 3.20. For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras, $C/\approx =_{\mathrm{def}} \{A/\approx_A \mid A \in C\}$.

Proposition 3.21. *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *of* Σ*-algebras, C is behaviourally consistent w.r.t.* $\approx$ *if and only if* $C/\approx \subseteq C$.

The semantical quotient operator for classes of Σ-algebras gives rise to a quotient operator for specifications which will be added in the following to the given specification language.

Definition 3.22 (*Behavioural quotient*). Let SP be a specification with signature Σ.

(1) The expression $\mathrm{SP}/\approx$ is a specification, called the *behavioural quotient* of SP.

(2) The signature and the model class of a behavioural quotient specification are given by:

$$\mathrm{Sig}(\mathrm{SP}/\approx) =_{\mathrm{def}} \mathrm{Sig}(\mathrm{SP}),$$
$$\mathrm{Mod}(\mathrm{SP}/\approx) =_{\mathrm{def}} \mathrm{Iso}(\mathrm{Mod}(\mathrm{SP})/\approx).$$

Note that for fulfilling the requirement that model classes have to be closed under isomorphism one has explicitly to construct in (2) the isomorphic closure to $\mathrm{Mod}(\mathrm{SP})/\approx$ (cf. Section 2.1). Obviously we have the following equivalent conditions for the behavioural consistency of specifications:

Proposition 3.23. *Let* SP *be a specification with signature* Σ. *The following conditions are equivalent*:

(1) SP *is behaviourally consistent w.r.t.* $\approx$,

(2) $\mathrm{Mod}(\mathrm{SP})/\approx \subseteq \mathrm{Mod}(\mathrm{SP})$,

(3) $\mathrm{Mod}(\mathrm{SP}/\approx) \subseteq \mathrm{Mod}(\mathrm{SP})$.

Example 3.24. In the case of (partial) observational equalities $\approx_{\mathrm{Obs,In}}$, one can show that the model class of a basic specification SP is closed under the behavioural quotient construction if the axioms of SP are conditional equations with observable premises. Hence, in this case SP is behaviourally consistent w.r.t. $\approx_{\mathrm{Obs,In}}$ (cf. also Example 5.12). More generally, we have provided in [8] a proof-theoretic characterization of the behavioural consistency of basic specifications if the given behavioural equality $\approx$ is axiomatizable and we have developed conditions which allow to derive the behavioural consistency of structured specifications.

Example 3.25 (*Connection to the forget–restrict–identity approach*). (1) Let **behaviour** SP **wrt** $\approx_{\mathrm{Obs,In}}$ be an observational behaviour specification (cf. Example 3.9). Then the model class Mod(**behaviour** SP **wrt** $\approx_{\mathrm{Obs,In}}$) is the class of all Σ-algebras A which, due to the definition of $A/\approx_{\mathrm{Obs,In},A}$, have the following property: If we first restrict A to its subalgebra $\mathrm{Dom}(\approx_{\mathrm{Obs,In},A})$ generated over the values of input sorts and if we then identify all observationally equal elements of $\mathrm{Dom}(\approx_{\mathrm{Obs,In},A})$, then we obtain a model of SP. Hence, an observational behaviour specification describes all Σ-algebras which after restriction and observational identification are models of the underlying specification SP. Note that in the case where all sorts are observable an observational behaviour specification describes all algebras which after restriction are standard models of SP.

(2) Let $\mathrm{SP} = \langle \Sigma, E \rangle$ be an equational specification, i.e. the axioms E consist of universally quantified equations, such that the signature Σ is sensible w.r.t. the empty set of input sorts. Assume that **init** SP is a specification such that the model class Mod(**init** SP) is defined as the isomorphism class of the initial model of SP. Let $\approx_{E,\emptyset}$ be the family of partial Σ-congruences generated by the equations E relative to the empty set of input sorts (cf. Example 3.6). Then the model class of the behavioural specification **behaviour (init** SP) **wrt** $\approx_{E,\emptyset}$ consists of all Σ-algebras A such that if we first restrict A to its finitely generated subalgebra (which is just $\mathrm{Dom}(\approx_{E,\emptyset,A})$) and if we then identify all elements of this subalgebra w.r.t. the congruence relation generated by the equations E, we obtain an algebra of Mod(**init** SP), i.e. an initial model of SP. This means that **behaviour (init** SP) **wrt** $\approx_{E,\emptyset}$ describes all Σ-algebras which can be considered as forget–restrict–identify implementations of the initial model of SP (with a trivial forget step).

4. Abstractor specifications

The notion of "abstractor" was introduced in [24] 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. Intuitively, the equivalence relation is used for expressing that two algebras have "the same behaviour".

Definition 4.1. Let $\equiv \subseteq \mathrm{Alg}(\Sigma) \times \mathrm{Alg}(\Sigma)$ be an equivalence relation on $\mathrm{Alg}(\Sigma)$. For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras, $\mathrm{Abs}_{\equiv}(C)$ denotes the closure of C under $\equiv$, i.e.

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

Definition 4.2 (*Isomorphism protection*). An equivalence relation $\equiv$ on $\mathrm{Alg}(\Sigma)$ is called *isomorphism protecting* if for any two Σ-algebras $A, B \in \mathrm{Alg}(\Sigma)$, the following holds: If A and B are isomorphic then $A \equiv B$.

General Assumption 2. In the following, we always assume that $\equiv$ is an isomorphism protecting equivalence relation between Σ-algebras.

We now extend our specification language by the concept of abstractor specification which allows to construct the closure of (the model class of) a given specification under $\equiv$.

Definition 4.3 (*Abstractor specification*). Let SP be a specification with signature Σ.

(1) The expression **abstract SP wrt** $\equiv$ is an *abstractor specification.*
(2) The signature and the model class of an abstractor specification are given by:
$\mathrm{Sig}(\textbf{abstract SP wrt} \equiv) =_{\mathrm{def}} \mathrm{Sig}(\mathrm{SP})$,
$\mathrm{Mod}(\textbf{abstract SP wrt} \equiv) =_{\mathrm{def}} \mathrm{Abs}_{\equiv}(\mathrm{Mod}(\mathrm{SP}))$.

Note that from our hypothesis that $\equiv$ is isomorphism protecting the model class of an abstractor specification is closed under isomorphism.

Example 4.4 (*Observational abstractions*). Important examples for abstractor specifications are *observational abstractions* which are determined by observational equivalence relations between algebras. The basic idea behind such relations is that two algebras are considered to be observationally equivalent if they cannot be distinguished by a predefined set of observations. In [23] such observations are represented by formulas while (more specifically) in the algebraic specification language ASL (cf. [26]), the admissible observations are defined by a set W of terms. In this case, two algebras are called W-equivalent if they satisfy the same equations between terms of W. For W, we will consider here all terms (over a given signature) of observable sort which may contain variables of some given input sorts. More precisely, we assume again given a signature $\Sigma = (S, F)$, a distinguished set $\mathrm{Obs} \subseteq S$ of observable sorts and a set $\mathrm{In} \subseteq S$ of input sorts such that Σ is sensible w.r.t. In. Then two Σ-algebras A and B are called *observationally equivalent* w.r.t. Obs and In, denoted by $A \equiv_{\mathrm{Obs,In}} B$, if there exists an S-sorted family Y_{In} of sets $(Y_{\mathrm{In}})_s$ of variables of sort s with $(Y_{\mathrm{In}})_s = \emptyset$ for all $s \notin \mathrm{In}$, $(Y_{\mathrm{In}})_s \neq \emptyset$ for all $s \in \mathrm{In}$ and if there exist two valuations $\alpha 1 : Y_{\mathrm{In}} \to A$ and $\beta 1 : Y_{\mathrm{In}} \to B$ with surjective mappings $\alpha 1_s : (Y_{\mathrm{In}})_s \to A_s$ and $\beta 1_s : (Y_{\mathrm{In}})_s \to B_s$ for all $s \in \mathrm{In}$ such that for all terms $t, r \in T(\Sigma, Y_{\mathrm{In}})_s$ of observable sort

$s \in \mathrm{Obs}$ the following holds:

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

Obviously, $\equiv_{\mathrm{Obs,In}}$ is isomorphism protecting. In the following we consider three important cases of observational equivalence relations between algebras:

(1) If we choose $\mathrm{In} = S$, then the relation $\equiv_{\mathrm{Obs},S}$ coincides with the behavioural equivalence relation of Reichel (cf. e.g. [22]). We will give a sketch of the proof that $\equiv_{\mathrm{Obs},S}$ is indeed Reichel's equivalence: First assume that A and B are Σ-algebras which are behaviourally equivalent in the sense of [22]. Then there exists a Σ-algebra C and surjective Σ-homomorphisms $r_A : C \to A$ and $r_B : C \to B$ (called reductions) such that the mappings $(r_A)_s : C_s \to A_s$ and $(r_B)_s : C_s \to B_s$ are bijective for all $s \in \mathrm{Obs}$. One can now prove that $A \equiv_{\mathrm{Obs},S} B$ holds by choosing $Y_{\mathrm{In}} = C$ and $\alpha 1 = r_A$, $\beta 1 = r_B$. Conversely, assume that $A \equiv_{\mathrm{Obs},S} B$ holds w.r.t. some Y_{In} and some surjective valuations $\alpha 1 : Y_{\mathrm{In}} \to A$ and $\beta 1 : Y_{\mathrm{In}} \to B$. The interpretations $I_{\alpha 1} : T(\Sigma, Y_{\mathrm{In}}) \to A$ and $I_{\beta 1} : T(\Sigma, Y_{\mathrm{In}}) \to B$ induce corresponding Σ-congruences on the term algebra $T(\Sigma, Y_{\mathrm{In}})$ denoted by $\sim I_{\alpha 1}$ and $\sim I_{\beta 1}$. Then the intersection $\sim I_{\alpha 1} \cap \sim I_{\beta 1}$ is also a Σ-congruence on $T(\Sigma, Y_{\mathrm{In}})$. We can now prove that A and B are equivalent in the sense of Reichel by choosing for C the quotient algebra $T(\Sigma, Y_{\mathrm{In}})/(\sim I_{\alpha 1} \cap \sim I_{\beta 1})$ and by defining $r_A : C \to A$ by $r_A([t]) =_{\mathrm{def}} I_{\alpha 1}(t)$ and $r_B : C \to B$ by $r_B([t]) =_{\mathrm{def}} I_{\beta 1}(t)$.

(2) If we choose $\mathrm{In} = \mathrm{Obs}$, then we can show that the relation $\equiv_{\mathrm{Obs,Obs}}$ coincides with the behavioural equivalence of algebras in the sense of [19] and, if algebras with the same observable carrier sets are considered, also with the equivalence relation defined in [27].

(3) If we choose $\mathrm{In} = \emptyset$, then two algebras are equivalent w.r.t. $\equiv_{\mathrm{Obs},\emptyset}$ if they satisfy the same equations between observable ground terms. Hence, in this case the equivalence relation $\equiv_{\mathrm{Obs},\emptyset}$ determines a behavioural abstraction in the sense of [24].

5. Relating behavioural and abstractor specifications

Behavioural specifications 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 [24] or behavioural implementations in [8], for a survey on implementation concepts and observability see [20]). 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.

Remark 5.1. If we consider the particular case of observable behaviour specifications (cf. Example 3.9) and observational abstractions (cf. Example 4.4) then we can

conclude from a result in [22] that (if all sorts are input sorts) both specifications have the same semantics if the axioms of the specification are conditional equations with observable premises. Analogously a result in [19] shows that if only the observable sorts are used as input sorts and if the specification is equational, then again behavioural semantics and abstractor semantics coincide. However, this is in general not true if the axioms are arbitrary first-order formulas. For instance, the specification DEMO of Example 3.18 has a standard model which, by definition, is also a model of the abstractor specification "**abstract** DEMO **wrt** $\equiv_{\mathrm{Obs,In}}$" for any choice of Obs and In. However, if we choose Obs $= \emptyset$, then we have seen that the behavioural specification "**behaviour** DEMO **wrt** $\approx_{\emptyset,\{s\}}$" has no model.

5.1. *Factorizability*

The underlying idea of the behavioural approach is to consider the behaviour $A/\approx_A$ of any Σ-algebra A w.r.t. a given behavioural equality of objects $\approx$ while the abstractor approach is based on an equivalence relation $\equiv$ between algebras where intuitively two algebras A and B are equivalent if they have the same behaviour. Hence, to relate both approaches an obvious preliminary requirement is that in both cases "behaviour" has the same meaning. More formally, this means that for any two Σ-algebras A and B, $A \equiv B$ holds if and only if A and B "have the same behaviour" if and only if "the behaviour of A" and "the behaviour of B" are "the same" if and only if $A/\approx_A$ and $B/\approx_B$ are the same (up to isomorphism). This consideration leads to the following definition:

Definition 5.2 (*Factorizability*). Let $\equiv \subseteq \mathrm{Alg}(\Sigma) \times \mathrm{Alg}(\Sigma)$ be an equivalence relation and let $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ be a family of partial Σ-congruences. $\equiv$ is called *factorizable by* $\approx$ if for all Σ-algebras $A, B \in \mathrm{Alg}(\Sigma)$, the following holds:

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

The equivalence $\equiv$ is called factorizable if there exists a family $\approx$ such that $\equiv$ is factorizable by $\approx$.

Remark 5.3. Given a family $\approx$ of partial Σ-congruences, one can always construct an associated equivalence relation between Σ-algebras, denoted by $\equiv_{\approx}$, which is factorizable by $\approx$ in the following way: For any $A, B \in \mathrm{Alg}(\Sigma)$, $A \equiv_{\approx} B$ holds, by definition, if $A/\approx_A$ and $B/\approx_B$ are isomorphic (cf. [30]). On the other hand, if we are given an equivalence relation $\equiv$ on $\mathrm{Alg}(\Sigma)$, we can find an associated family of partial Σ-congruences only if the equivalence is factorizable. Indeed, it is usually not a simple task to prove factorizability.

The following example shows that observational equivalences between algebras as defined in Example 4.4 are factorizable. As a consequence, we obtain that the equivalences of Reichel (cf. Example 4.4(1)), of Nivela and Orejas (cf. Example 4.4(2))

and the equivalence of Sannella and Tarlecki w.r.t. all equations between observable ground terms (cf. Example 4.4(3)) are factorizable.

Example 5.4 (*Observational equivalences of algebras are factorizable*). For any set Obs of observable sorts and any set In of input sorts, the equivalence $\equiv_{\mathrm{Obs,In}}$ of Example 4.4 is factorizable by the family $\approx_{\mathrm{Obs,In}}$ of partial Σ-congruences defined in Example 3.3.

In order to prove this, we have to show that for all Σ-algebras A and B with partial Σ-congruences $\approx_{\mathrm{Obs,In},A}$ and $\approx_{\mathrm{Obs,In},B}$, the following holds: $A \equiv_{\mathrm{Obs,In}} B$ if and only if $A/\approx_{\mathrm{Obs,In},A}$ and $B/\approx_{\mathrm{Obs,In},B}$ are isomorphic. Thereby, we will use the variant of the definition of the observational equality of elements given in Fact 3.4 and we will write shortly $\approx_A$ instead of $\approx_{\mathrm{Obs,In},A}$ (and similarly for B).

"$\Rightarrow$": Let $A \equiv_{\mathrm{Obs,In}} B$ w.r.t. Y_{In}, $\alpha 1 : Y_{\mathrm{In}} \leftarrow A$ and $\beta 1 : Y_{\mathrm{In}} \rightarrow B$. In a first step we will show that the following holds:

(1) For all terms $t, r \in T(\Sigma, Y_{\mathrm{In}})$: $I_{\alpha 1}(t), I_{\alpha 1}(r) \in \mathrm{Dom}(\approx_A)$, $I_{\beta 1}(t), I_{\beta 1}(r) \in \mathrm{Dom}(\approx_B)$ and $I_{\alpha 1}(t) \approx_A I_{\alpha 1}(r)$ iff $I_{\beta 1}(t) \approx_B I_{\beta 1}(r)$.

Using (1) we will show in a second step that the following holds:

(2) $h : A/\approx_A \rightarrow B/\approx_B$, $h([a]) =_{\mathrm{def}} [I_{\beta 1}(t)]$ if $t \in T(\Sigma, Y_{\mathrm{In}})$ with $[a] = [I_{\alpha 1}(t)]$ defines a Σ-isomorphism.

Proof of (1): By construction of $\approx_A$ and $\approx_B$ and since $(Y_{\mathrm{In}})_s = \emptyset$ for all $s \notin \mathrm{In}$, it is obvious that for any terms $t, r \in T(\Sigma, Y_{\mathrm{In}})$, $I_{\alpha 1}(t), I_{\alpha 1}(r) \in \mathrm{Dom}(\approx_A)$ and $I_{\beta 1}(t), I_{\beta 1}(r) \in \mathrm{Dom}(\approx_A)$. Now let $I_{\alpha 1}(t) \approx_A I_{\alpha 1}(r)$ with $t, r \in T(\Sigma, Y_{\mathrm{In}})_s$. We have to show that then $I_{\beta 1}(t) \approx_B I_{\beta 1}(r)$ holds, i.e. that for any observable Σ-context $c \in T(\Sigma, B_{\mathrm{In}} \cup Z)$ containing z_s, $I_\tau(c) = I_\rho(c)$ where $\tau(z_s) I_{\beta 1}(t)$, $\rho(z_s) = I_{\beta 1}(r)$ and $\tau(x) = \rho(x) = x$ for all $x \in B_{\mathrm{In}}$. W.l.o.g. assume that c contains besides z_s exactly one variable $x \in B_{\mathrm{In}}$. Since $\beta 1_s : (Y_{\mathrm{In}})_s \rightarrow B_s$ is surjective for all $s \in \mathrm{In}$, there exists $y \in Y_{\mathrm{In}}$ with $\beta 1(y) = x$. Then, if we replace z_s by t and x by y in c we have $I_{\beta 1}(c[y/x, t/z_s]) = I_\tau(c)$ and analogously $I_{\beta 1}(c[y/x, r/z_s]) = I_\rho(c)$. Hence, it remains to show that $I_{\beta 1}(c[y/x, t/z_s]) = I_{\beta 1}(c[y/x, r/z_s])$. Since we have assumed $I_{\alpha 1}(t) \approx_A I_{\alpha 1}(r)$, it is easy to derive that $I_{\alpha 1}(c[y/x, t/z_s]) = I_{\alpha 1}(c[y/x, r/z_s])$ holds. Because $c[y/x, t/z_s]$ and $c[y/x, r/z_s]$ are observable terms of $T(\Sigma, Y_{\mathrm{In}})$, we then obtain by the assumption $A \equiv_{\mathrm{Obs,In}} B$ the desired result $I_{\beta 1}(c[y/x, t/z_s]) = I_{\beta 1}(c[y/x, r/z_s])$. Analogously, one can show that $I_{\beta 1}(t) \approx_B I_{\beta 1}(r)$ implies $I_{\alpha 1}(t) \approx_A I_{\alpha 1}(r)$.

Proof of (2): h is well-defined because, first, due to the construction of $\approx_A$ and to the surjectivity of $\alpha 1_s : (Y_{\mathrm{In}})_s \rightarrow A_s$ for all $s \in \mathrm{In}$ there exists for any $a \in \mathrm{Dom}(\approx_A)$ a term $t \in T(\Sigma, Y_{\mathrm{In}})$ with $[a] = [I_{\alpha 1}(t)]$ and, secondly, if $[a] = [I_{\alpha 1}(t)]$ and $[a] = [I_{\alpha 1}(r)]$, then $I_{\alpha 1}(t) \approx_A I_{\alpha 1}(r)$ and therefore, using (1), $I_{\beta 1}(t) \approx_B I_{\beta 1}(r)$, i.e. $[I_{\beta 1}(t)] = [I_{\beta 1}(r)]$. It is easy to show that h is a Σ-homomorphism.

In order to prove that h is a Σ-isomorphism we consider $h' : B/\approx_B \rightarrow A/\approx_A$, $h'([b]) =_{\mathrm{def}} [I_{\alpha 1}(t)]$ if $t \in T(\Sigma, Y_{\mathrm{In}})$ with $[b] = [I_{\beta 1}(t)]$. Analogously to h, one can

show that h' is a well-defined Σ-homomorphism. Moreover, we obtain as follows that $h' \circ h$ is the identity on $A/\approx_A$:

Let $[a] \in A/\approx_A$ with $[a] = [I_{\alpha 1}(t)]$. Then $h([a]) = [I_{\beta 1}(t)]$. Hence $h'(h([a])) = h'([I_{\beta 1}(t)]) = [I_{\alpha 1}(t) = [a]$ (by definition of h'). Analogously, we obtain that $h \circ h'$ is the identity on $B/\approx_B$. Hence, h and h' are Σ-isomorphisms.

"$\Leftarrow$": Let $h: A/\approx_A \to B/\approx_B$ be a Σ-isomorphism. W.l.o.g. we assume that A and B are disjoint. Then let $(Y_{\text{In}})_s =_{\text{def}} \emptyset$ for all $s \notin \text{In}$ and $(Y_{\text{In}})_s =_{\text{def}} A_s \cup B_s$ for all $s \in \text{In}$ (note that $A_s = \text{Dom}(\approx_A)_s$ and $B_s = \text{Dom}(\approx_B)_s$ for all $s \in \text{In}$). Moreover, let for all $s \in \text{In}$, $\alpha 1_s: (Y_{\text{In}})_s \to A_s$ be defined by $\alpha 1_s(y) =_{\text{def}} y$ if $y \in A_s$, $\alpha 1_s(y) =_{\text{def}} a$ if $y \in B_s$ and $h([a]) = [y]$ and let $\beta 1_s: (Y_{\text{In}})_s \to B_s$ be defined by $\beta 1_s(y) =_{\text{def}} y$ if $y \in B_s$, $\beta 1(y) =_{\text{def}} b$ if $y \in A_s$ and $h([y]) = [b]$. Using definition of Y_{In}, $\alpha 1$ and $\beta 1$, we can prove by structural induction on t that for all terms $t \in T(\Sigma, Y_{\text{In}})$, $h([I_{\alpha 1}(t)]) = [I_{\beta 1}(t)]$ holds. Now, in order to prove $A \equiv_{\text{Obs.In}} B$ we have to show that for all terms $t, r \in T(\Sigma, Y_{\text{In}})_s$ of observable sort $s \in \text{Obs}$, $I_{\alpha 1}(t) = I_{\alpha 1}(r)$ iff $I_{\beta 1}(t) = I_{\beta 1}(r)$. Assume $I_{\alpha 1}(t) = I_{\alpha 1}(r)$ holds. Since $I_{\alpha 1}(t), I_{\alpha 1}(r) \in \text{Dom}(\approx_A)$ then $[I_{\alpha 1}(t)] = [I_{\alpha 1}(r)]$ holds. 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. $I_{\beta 1}(t) \approx_B I_{\beta 1}(r)$. Then $I_{\beta 1}(t) = I_{\beta 1}(r)$ holds, by definition of $\approx_B$, since t and r are of observable sort and $I_{\beta 1}(t), I_{\beta 1}(r) \in \text{Dom}(\approx_B)$. Analogously, one can show that for observable terms t and r, $I_{\beta 1}(t) = I_{\beta 1}(r)$ implies $I_{\alpha 1}(t) = I_{\alpha 1}(r)$. Hence, $A \equiv_{\text{Obs.In}} B$.

According to the correspondence between a family $\approx$ of partial Σ-congruences and a factorizable equivalence relation $\equiv$ we can disregard whether we start from one point of view or from the other one.

General Assumption 3. In the sequel of this paper, we consider arbitrary pairs $(\approx, \equiv)$ consisting of a family $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ of partial Σ-congruences and an equivalence relation $\equiv$ between Σ-algebras such that $\equiv$ is factorizable by $\approx$. (We still assume that $\approx$ is isomorphism compatible and that $\equiv$ is isomorphism protecting which are equivalent assumptions since $\equiv$ is factorizable by $\approx$.)

5.2. *Behavioural generalization of Scott's theorem*

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

Proposition 5.5. *Let A, B be two Σ-algebras. If $A \equiv B$ then for all Σ-formulas ϕ, $A \models_{\approx} \phi$ if and only if $B \models_{\approx} \phi$.*

(In particular, A and B are elementary equivalent w.r.t. the behavioural satisfaction relation).

Proof. Since $A \equiv B$ and since $\equiv$ is factorizable by $\approx$, $A/\approx_A$ and $B/\approx_B$ are isomorphic. By Theorem 3.11, we know that $A \models_{\approx} \phi$ iff $A/\approx_A \models \phi$. Since isomorphic

Σ-algebras satisfy the same Σ-formulas, we have $A/\approx_A \models \phi$ iff $B/\approx_B \models \phi$ and again by Theorem 3.11 we obtain $B/\approx_B \models \phi$ iff $B \models_\approx \phi$. □

Since Σ-formulas may be infinitary formulas of $L_{\omega_1\omega}$ (cf. Section 2.3) we can state as a consequence of Proposition 5.5 a "behavioural" version of Scott's theorem (cf. e.g. [15]):

Theorem 5.6. *Let A, B be two Σ-algebras such that $A/\approx_A$ and $B/\approx_B$ are countable. Then $A \equiv B$ holds if and only if A and B behaviourally satisfy w.r.t. $\approx$ the same Σ-formulas ϕ (i.e. for all Σ-formulas ϕ, $A \models_\approx \phi$ if and only if $B \models_\approx \phi$.)*

Proof. "⇒": Follows from Proposition 5.5.

"⇐": Assume that for all Σ-formulas ϕ, $A \models_\approx \phi$ iff $B \models_\approx \phi$. Then, by Theorem 3.11, for all Σ-formulas ϕ, $A/\approx_A \models \phi$ iff $B/\approx_B \models \phi$. Since $A/\approx_A$ and $B/\approx_B$ are countable, we can now apply Scott's theorem (cf. e.g. [15]) and we obtain that $A/\approx_A$ and $B/\approx_B$ are isomorphic. Then $A \equiv B$ holds since $\equiv$ is factorizable by $\approx$. □

Remark 5.7. Theorem 5.6 can be generalized to arbitrary Σ-algebras A and B if we use formulas ϕ of the more powerful logic $L_{\infty\infty}$. This follows from a straightforward extension of the behavioural satisfaction relation and of Theorem 3.11 to formulas of $L_{\infty\infty}$ and from the fact that $L_{\infty\infty}$ allows us to identify arbitrary algebras up to isomorphism (cf. [23]).

5.3. Semantical equivalence of behavioural and abstractor specifications

According to our intuition, two Σ-algebras are equivalent if they have the same behaviour and the behaviour of a Σ-algebra A is given by the algebra $A/\approx_A$. Hence, we expect that any algebra A is equivalent to its behaviour, i.e. $A \equiv A/\approx_A$. The following lemma shows that this requirement is equivalent to the weak regularity of $\approx$ (cf. Definition 3.14 (2):

Lemma 5.8. *The following conditions are equivalent:*

(1) *For any Σ-algebra A, $A \equiv A/\approx_A$.*

(2) $\approx$ is weakly regular.

Proof. Since $\equiv$ is factorizable by $\approx$, (1) is equivalent to the fact that for any Σ-algebra A, $A/\approx_A$ is isomorphic to $(A/\approx_A)/\approx_{(A/\approx_A)}$ which means, by definition, that $\approx$ is weakly regular. □

General Assumption 4. In the following of this paper, we assume that $\approx$ is a weakly regular family of partial Σ-congruences.

Then we know by Lemma 5.8 that $A \equiv A/\approx_A$ for any Σ-algebra A which is the crucial fact needed to prove the relationships between behavioural and abstractor

specifications. As a first result, we show that the model class of a behavioural specification is included in the model class of the corresponding abstractor specification. Hence, behavioural specifications are in general more restrictive than abstractor specifications.

Theorem 5.9. *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *of* Σ*-algebras and any specification* SP *of signature* Σ,

(1) $\mathrm{Beh}_{\approx}(C) \subseteq \mathrm{Abs}_{\equiv}(C)$,

(2) Mod(**behaviour** SP **wrt** $\approx$) $\subseteq$ Mod(**abstract** SP **wrt** $\equiv$).

Proof. (1) Let $A \in \mathrm{Beh}_{\approx}(C)$. Then $A/\approx_A \in C$. Since $\approx$ is weakly regular we have, by Lemma 5.8, $A \equiv A/\approx_A$. Hence, $A \in \mathrm{Abs}_{\equiv}(C)$.

(2) Follows from (1). □

In the next step, we will provide a characterization of the semantical equivalence of behavioural and abstractor specifications. For this purpose, we use the following lemma which shows that for behaviourally consistent specifications (cf. Definition 3.19), the model class of an abstractor specification is included in the model class of the corresponding behavioural specification.

Lemma 5.10. *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *of* Σ*-algebras and any specification* SP *with signature* Σ,

(1) *if* C *is closed under isomorphism then* $\mathrm{Abs}_{\equiv}(C) \subseteq \mathrm{Beh}_{\approx}(C)$ *if and only if* C *is behaviourally consistent w.r.t.* $\approx$,

(2) Mod(**abstract** SP **wrt** $\equiv$) $\subseteq$ Mod(**behaviour** SP **wrt** $\approx$) *if and only if* SP *is behaviourally consistent w.r.t.* $\approx$.

Proof. (1) "$\Rightarrow$": Obvious, since $C \subseteq \mathrm{Abs}_{\equiv}(C)$ always holds and hence $C \subseteq \mathrm{Beh}_{\approx}(C)$.

"$\Leftarrow$": Let $A \in \mathrm{Abs}_{\equiv}(C)$. Then $A \equiv B$ for some $B \in C$. By factorizability, $A/\approx_A$ is isomorphic to $B/\approx_B$. Since C is behaviourally consistent w.r.t. $\approx$, $B/\approx_B \in C$ (cf. Proposition 3.21) and since C is isomorphically closed also $A/\approx_A \in C$. Hence, $A \in \mathrm{Beh}_{\approx}(C)$.

(2) Follows from (1) since model classes are closed under isomorphism.

(Note that we have not used weak regularity here.) □

As a direct consequence of Theorem 5.9 and Lemma 5.10, we obtain that behavioural and abstractor specifications are semantically equivalent if and only if the underlying specification SP is behaviourally consistent.

Theorem 5.11. *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *of* Σ*-algebras and any specification* SP *with signature* Σ,

(1) *if* C *is closed under isomorphism then* $\mathrm{Beh}_{\approx}(C) = \mathrm{Abs}_{\equiv}(C)$ *if and only if* C *is behaviourally consistent w.r.t.* $\approx$,

(2) **behaviour** SP **wrt** $\approx$ = **abstract** SP **wrt** $\equiv$ *if and only if* SP *is behaviourally consistent w.r.t.* $\approx$,
where " = " *stands for the semantical equivalence of specifications* (*cf. Section* 2.4).

According to the characterization of behavioural consistency in Proposition 3.23, we see that for proving the equivalence of behavioural and abstractor specifications, it is enough to check whether the model class of the underlying specification SP is closed under the behavioural quotient construction.

Example 5.12. We have pointed out in Example 3.24 that a basic specification SP is behaviourally consistent w.r.t the observational equality $\approx_{\mathrm{Obs,In}}$ if the axioms of SP are conditional equations with observable premises. Hence, in this case Theorem 5.11 tells us that the observational behaviour specification **behaviour** SP **wrt** $\approx_{\mathrm{Obs,In}}$ is semantically equivalent to the observational abstractor specification **abstract** SP **wrt** $\equiv_{\mathrm{Obs,In}}$ (since $\approx_{\mathrm{Obs,In}}$ is weakly regular and the observational equivalence $\equiv_{\mathrm{Obs,In}}$ is factorizable by $\approx_{\mathrm{Obs,In}}$, cf. Example 5.4).

In particular, if we use the observational equivalence $\equiv_{\mathrm{Obs},S}$ (cf. Example 4.4 (1)) which is factorizable by the family $\approx_{\mathrm{Obs},S}$ (cf. Example 3.5 (1)), then we obtain, as one application of Theorem 5.11, the theorem of [22] which says that observational behaviour semantics is the same as observational abstractor semantics if the axioms of the specification SP are conditional equations with observable premises. Analogously, as a further application of Theorem 5.11, we obtain the theorem of [19] which says that in their approach (where the observable sorts are the input sorts and the axioms are equations) behavioural semantics and abstractor semantics coincide as well.

Example 5.13. Let $\approx_{E,\mathrm{In}}$ be the family of partial Σ-congruences generated by a set E of equations and a set In of input sorts (cf. Example 3.6) and let $\equiv_{\approx_{E,\mathrm{In}}}$ be the equivalence relation associated to $\approx_{E,\mathrm{In}}$ (cf. Remark 5.3). By definition, for any two Σ-algebras A and B, $A \equiv_{\approx_{E,\mathrm{In}}} B$ holds if $A/\approx_{E,\mathrm{In},A}$ and $B/\approx_{E,\mathrm{In},B}$ are isomorphic. It is not obvious to find an interpretation for this equivalence relation. An intuition may be provided if we use the following notion of simulation of one algebra by another algebra (which was similarly defined in [25]):

A Σ-algebra A *simulates w.r.t. In a* Σ-algebra B if there exists a surjective Σ-homomorphism $h: A[X_{\mathrm{In}}] \to B$. (Remember that $A[X_{\mathrm{In}}]$ is the reachable part of A generated over the values of input sorts and that $\mathrm{Dom}(\approx_{E,\mathrm{In},A}) = A[X_{\mathrm{In}}]$.) The surjectivity requirement ensures that any element of B has a representation in A generated over the values of input sorts.

Then one can prove that $A \equiv_{\approx_{E,\mathrm{In}}} B$ holds if and only if A and B simulate w.r.t. In a Σ-algebra C which satisfies the equations E and which satisfies the following universal properties (U_A) and (U_B). The universal property (U_A) ((U_B) resp.) says that C is an initial object among the algebras which satisfy E and are simulated w.r.t. In by A (by B resp.)

(U_A): Let $h_A: A[X_{\text{In}}] \to C$ be the surjective Σ-homomorphism used for the simulation of C by A. For any other Σ-algebra D which satisfies the equations E and which is simulated w.r.t. In by A with surjective Σ-homomorphism $h_D: A[X_{\text{In}}] \to D$ there exists a unique Σ-homomorphism $f: C \to D$ such that the composition of h_A and f yields h_D.

(U_B): Analogous to (U_A) using the simulation of C by B.

Now assume that the specification **init** SP and the family $\approx_{E,\emptyset}$ are given as in Example 3.25(2). The specification **init** SP is behaviourally consistent w.r.t. $\approx_{E,\emptyset}$ since the quotient algebra $I/\approx_{E,\emptyset,I}$ of an initial model I of SP is also an initial model of SP (cf. the characterization of behavioural consistency in Proposition 3.23). Then, by Theorem 5.11, the specifications **behaviour (init** SP) **wrt** $\approx_{E,\emptyset}$ and **abstract (init** SP) **wrt** $\equiv_{\approx_{E,\emptyset}}$ are semantically equivalent. Hence, **abstract (init** SP) **wrt** $\equiv_{\approx_{E,\emptyset}}$ describes also all Σ-algebras which can be considered as forget–restrict–identify implementations of the initial model of SP (cf. Example 3.25(2)).

Example 5.14 (*Connection to terminal algebras*). Assume that SP is a specification with signature $\Sigma = (S, F)$ such that all models of SP are finitely generated, i.e. satisfy the reachability constraint $\mathscr{R} = (S, F)$. As usual, a model T of SP is called *terminal model* if for any model $A \in \text{Mod(SP)}$, there exists a unique Σ-homomorphism $h_A: A \to T$. The existence of a terminal model of SP can be characterized by the following equivalent conditions:

(1) SP has a terminal model,

(2) there exists a family $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ of total Σ-congruences such that SP is behaviourally consistent w.r.t. $\approx$ and all models $A, B \in \text{Mod(SP)}$ have isomorphic behaviours, i.e. $A/\approx_A$ and $B/\approx_B$ are isomorphic,

(3) there exists a family $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ of total Σ-congruences such that **behaviour** SP **wrt** $\approx$ = **abstract** SP **wrt** $\equiv_{\approx}$ and all models $A, B \in \text{Mod(SP)}$ are equivalent, i.e. $A \equiv_{\approx} B$ (where $\equiv_{\approx}$ is the equivalence relation associated to $\approx$, cf. Remark 5.3).

Note that for the equivalence of (1) and (2), it is not necessary to assume that $\approx$ is weakly regular and isomorphism compatible.

Proof of the equivalence of (1)–(3):

(1) $\Rightarrow$ (2): Let T be a terminal model of SP with unique Σ-homorphisms $h_A: A \to T$ for all $A \in \text{Mod(SP)}$. For any $A \in \text{Mod(SP)}$ we define $\approx_A$ by $a \approx_A b$ iff $h_A(a) = h_A(b)$ for $a, b \in A$. If A is a Σ-algebra not belonging to Mod(SP) then we define $\approx_A$ as the set-theoretic equality $=_A$ on A. Obviously, $\approx = (\approx_A)_{A \in \text{Alg}(\Sigma)}$ is a family of total Σ-congruences. By construction of $\approx$, for any model $A \in \text{Mod(SP)}$, $A/\approx_A$ is isomorphic to T. Hence, $A/\approx_A \in \text{Mod(SP)}$ and therefore, by Proposition 3.23, SP is behaviourally consistent w.r.t $\approx$. It is also clear that for any two models $A, B \in \text{Mod(SP)}$, $A/\approx_A$ and $B/\approx_B$ are isomorphic (because both quotient algebras are isomorphic to T).

(1) $\Leftarrow$ (2): Assume that (2) holds for $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$. Now choose an arbitrary model $A \in \mathrm{Mod}(\mathrm{SP})$ and let $T =_{\mathrm{def}} A/\approx_A$. Since SP is behaviourally consistent w.r.t $\approx$, $T \in \mathrm{Mod}(\mathrm{SP})$. Let B be an arbitrary model of SP. Then, by assumption, there exists an isomorphism $g: B/\approx_B \to T$ and the composition of the canonical epimorphism $\pi_B: B \to B/\approx_B$ with g is a Σ-homomorphism $h_B: B \to T$. h_B is unique since all models of SP are finitely generated.

(2) $\Leftrightarrow$ (3): Follows from Theorem 5.11 and from the definition of $\equiv_{\approx}$.

5.4. *Characterization of behavioural and abstractor specifications*

The following theorem shows that any behavioural specification can be expressed in terms of an abstractor specification.

Theorem 5.15 (Characterization of behavioural specifications). *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *of* Σ*-algebras and any specification* SP *of signature* Σ,

(1) *if* C *is closed under isomorphism, then* $\mathrm{Beh}_{\approx}(C) = \mathrm{Abs}_{\equiv}(C \cap \mathrm{Iso}(C/\approx))$,

(2) **behaviour** SP **wrt** $\approx$ = **abstract** (SP + SP/$\approx$) **wrt** $\equiv$.

Proof. (1) "$\subseteq$": Let $A \in \mathrm{Beh}_{\approx}(C)$. Then $A/\approx_A \in C$. Since $\approx$ is weakly regular, $A/\approx_A$ and $(A/\approx_A)/\approx_{(A/\approx_A)}$ are isomorphic. Hence, $A/\approx_A \in \mathrm{Iso}(C/\approx)$ and therefore $A/\approx_A \in C \cap \mathrm{Iso}(C/\approx)$. Moreover, by Lemma 5.8, $A \equiv A/\approx_A$. Thus, $A \in \mathrm{Abs}_{\equiv}(C \cap \mathrm{Iso}(C/\approx))$.

"$\supseteq$": Let $A \in \mathrm{Abs}_{\equiv}(C \cap \mathrm{Iso}(C/\approx))$. Then $A \equiv B$ for some $B \in C \cap \mathrm{Iso}(C/\approx)$. Hence, $B \in C$ and B is isomorphic to some quotient $D/\approx_D$. By factorizability, $A/\approx_A$ is isomorphic to $B/\approx_B$ and, by isomorphism compatibility of $\approx$, $B/\approx_B$ is isomorphic to $(D/\approx_D)/\approx_{(D/\approx_D)}$ which, by weak regularity, is isomorphic to $D/\approx_D$ which in turn was isomorphic to $B \in C$. Hence, $B/\approx_B \in C$ and then, since C is closed under isomorphism, $A/\approx_A \in C$, i.e. $A \in \mathrm{Beh}_{\approx}(C)$.

(2) Obviously both specifications have the same signature. Moreover, the semantical definitions of + and behavioural quotient imply that $\mathrm{Mod}(\mathrm{SP} + \mathrm{SP}/\approx) = \mathrm{Mod}(\mathrm{SP}) \cap \mathrm{Iso}(\mathrm{Mod}(\mathrm{SP})/\approx)$. Then we obtain the desired result from (1). □

The next theorem shows that vice versa abstractor specifications can be expressed in terms of behavioural specifications. Hence, there exists a duality between behavioural and abstractor specifications. Each one can be expressed by the other one.

Theorem 5.16 (*Characterization of abstractor specifications*). *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *of* Σ*-algebras and any s-pecification* SP *with signature* Σ,

(1) $\mathrm{Abs}_{\equiv}(C) = \mathrm{Beh}_{\approx}(\mathrm{Iso}(C/\approx))$,

(2) **abstract** SP **wrt** $\equiv$ = **behaviour** (SP/$\approx$) **wrt** $\approx$.

Proof. (1) "$\subseteq$": Let $A \in \mathrm{Abs}_{\equiv}(C)$. Then $A \equiv B$ for some $B \in C$. By factorizability, $A/\approx_A$ and $B/\approx_B$ are isomorphic. Hence, $A/\approx_A$ belongs to $\mathrm{Iso}(C/\approx)$ and therefore $A \in \mathrm{Beh}_{\approx}(\mathrm{Iso}(C/\approx))$.

"$\supseteq$": Let $A \in \mathrm{Beh}_{\approx}(\mathrm{Iso}(C/\approx))$. Then $A/\approx_A$ is isomorphic to some $B/\approx_B$ with $B \in C$. By factorizability, $A \equiv B$ and therefore $A \in \mathrm{Abs}_{\equiv}(C)$.

(2) Follows from (1). □

6. Fully abstract algebras

A further characterization of (the model class of) behavioural specifications can be obtained using fully abstract algebras.[5] Following Milner's notion (cf. [18]), we define full abstractness with respect to a given family $\approx$ of partial Σ-congruences in the following way.

Definition 6.1 (*Fully abstract algebra*). (1) A Σ-algebra A is called *fully abstract with respect to* $\approx$ (or briefly fully abstract) if $\approx_A$ coincides with the set-theoretic equality over the carrier sets of A. (In particular $\approx_A$ is total.)

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

Note that if A is fully abstract then the notation $A/\approx_A$ denotes $A/=_A$.

Example 6.2. (1) In the observational framework, fully abstract algebras w.r.t $\approx_{\mathrm{Obs,In}}$ are generated over the values of input sorts; two elements are equal if and only if they are observationally equal. For instance, if we consider the signature of sets and the family of congruences $\approx_{\mathrm{Obs,Obs}}$ used in Example 3.9(2) then the algebra $P_{\mathrm{fin}}(\mathbb{N})$ is fully abstract while $\mathbb{N}^*$ is not. Note that the powerset algebra $P(\mathbb{N})$ which contains not only finite but also infinite subsets of $\mathbb{N}$ is also not fully abstract w.r.t. $\approx_{\mathrm{Obs,Obs}}$, because infinite sets cannot be generated by the set operations. However, if we choose the family of congruences $\approx_{\mathrm{Obs},S}$ where $S = \{\mathrm{bool}, \mathrm{elem}, \mathrm{set}\}$ then $P(\mathbb{N})$ is fully abstract w.r.t. $\approx_{\mathrm{Obs},S}$ because all sets can be used as input values and two sets are equal if and only if they are observationally equal.

(2) If E is a set of Σ-equations and In is a set of input sorts then a Σ-algebra A is fully abstract w.r.t. $\approx_{E,\mathrm{In}}$ (cf. Example 3.6) if and only if it is generated over the values of input sorts and satisfies the equations E (in the standard sense).

As an obvious consequence of the property of full abstractness, we obtain that for fully abstract algebras there is no difference between standard satisfaction and behavioural satisfaction of formulas.

[5] Indeed, the investigation of the relationships between behavioural semantics and fully abstract algebras was originally the starting point of our study (cf. [5]).

Lemma 6.3. *For any fully abstract Σ-algebra A and any Σ-formula ϕ,*

$$A \models_{\approx} \phi \quad \text{if and only if} \quad A \models \phi.$$

In particular, for any reachability constraint $\mathscr{R}$ over Σ, $A \models_{\approx} \mathscr{R}$ if and only if $A \models \mathscr{R}$.

Example 6.4. In Example 3.3, we have noted that for arbitrary Σ-algebras A and equations $t = r$, $A \models_{\approx_{\mathrm{Obs,In}}} t = r$ if and only if $A \models c[\sigma(t)] = c[\sigma(r)]$ for all observable contexts $c \in T(\Sigma, X_{\mathrm{In}} \cup Z)$ and for all substitutions σ which replace the variables in t and r by terms of $T(\Sigma, X_{\mathrm{In}})$. In the case of fully abstract algebras, behavioural satisfaction is the same as standard satisfaction and one can get rid of the substitutions σ since fully abstract algebras are generated over the values of input sorts. Hence, for any fully abstract algebra A and equation $t = r$, we have $A \models t = r$ if and only if $A \models c[t] = c[r]$ for all observable contexts $c \in T(\Sigma, X_{\mathrm{In}} \cup Z)$.

Definition 6.5 (*Regularity*). A family $\approx = (\approx_A)_{A \in \mathrm{Alg}(\Sigma)}$ of partial Σ-congruences is called *regular* if for any Σ-algebra A, the quotient algebra $A/\approx_A$ is fully abstract.

Lemma 6.6. *If $\approx$ is regular then $\approx$ is weakly regular.*

Proof. Let $A \in \mathrm{Alg}(\Sigma)$.Since $\approx$ is regular, $A/\approx_A$ is fully abstract. Therefore, $(A/\approx_A)/\approx_{(A/\approx_A)}$ is the same as $(A/\approx_A)/=_{(A/\approx_A)}$ which is obviously isomorphic to $A/\approx_A$. □

Example 6.7. Any family $\approx_{\mathrm{Obs,In}}$ which is generated by a set obs of observable sorts and a set In of input sorts (cf. Example 3.3) and any family $\approx_{E,\mathrm{In}}$ generated by a set E of equations and a set In of input sorts (cf. Example 3.6) is regular. In fact, it seems that all reasonable examples of families of partial Σ-congruences are regular. Examples of congruences which are weakly regular but not regular exist but are constructed in a rather artificial, nonuniform way. We will now prove the regularity of $\approx_{\mathrm{Obs,In}}$ (the regularity of $\approx_{E,\mathrm{In}}$ is obvious since any quotient algebra $A/\approx_{E,\mathrm{In},A}$ satisfies the equations E and is generated over the input values).

Proof of the regularity of $\approx_{\mathrm{Obs,In}}$:[6] Let A be an arbitrary Σ-algebra and let $B = A/\approx_{\mathrm{Obs,In},A}$. We will briefly write $\approx_A$ instead of $\approx_{\mathrm{Obs,In},A}$ (and similarly for B). We have to show that for all elements $b, b' \in B_s$, $b \approx_B b'$ iff $b = b'$ (for all $s \in S$). By definition of B, there exist for any $b, b' \in B_s$ elements $a, a' \in \mathrm{Dom}(\approx_A)_s$ with $b = [a]$, $b' = [a']$. Hence, it is enough to show that for all $a, a' \in \mathrm{Dom}(\approx_A)_s$, $[a] \approx_B [a']$ iff $[a] = [a']$. One direction is trivial. For the other direction, assume $[a] \approx_B [a']$. We have to show $a \approx_A a'$. Let $c \in T(\Sigma, X_{\mathrm{In}} \cup Z)$ be an observable Σ-context containing z_s, let $\alpha : X_{\mathrm{In}} \to A$ be an arbitrary valuation and let $\alpha_a, \alpha_{a'} : X_{\mathrm{In}} \cup \{z_s\} \to A$ be the unique

[6] In [6], we have pointed out that the regularity of $\approx_{\mathrm{Obs,In}}$ is also a consequence of a given invariant axiomatization of the observational equality.

extensions of α defined by $\alpha_a(z_s) =_{\text{def}} a$, $\alpha_{a'}(z_s) =_{\text{def}} a'$. We have to show $I_{\alpha_a}(c) = I_{\alpha_{a'}}(c)$. Now define $\beta: X_{\text{In}} \to B$ by $\beta(x) =_{\text{def}} [\alpha(x)]$ for all $x \in X_{\text{In}}$. By assumption, $I_{\beta_{[a]}}(c) = I_{\beta_{[a']}}(c)$ where $\beta_{[a]}, \beta_{[a']}: X_{\text{In}} \cup \{z_s\} \to B$ are the unique extensions of β defined by $\beta_{[a]}(z_s) =_{\text{def}} [a]$, $\beta_{[a']}(z_s) =_{\text{def}} [a']$. Obviously, $I_{\beta_{[a]}}(c) = [I_{\alpha_a}(c)]$ and $I_{\beta_{[a']}}(c) = [I_{\alpha_{a'}}(c)]$. Hence, $[I_{\alpha_a}(c)] = [I_{\alpha_{a'}}(c)]$, i.e. $I_{\alpha_a}(c) \approx_A I_{\alpha_{a'}}(c)$. Since the sort of c is observable, we then know $I_{\alpha_a}(c) = I_{\alpha_{a'}}(c)$. Thus, $a \approx_A a'$.

We are now prepared to prove a further important characterization of behavioural semantics which says that if $\approx$ is a regular family of partial Σ-congruences, then the model class of a behavioural specification "**behaviour** SP **wrt** $\approx$" coincides with the closure of the class of the fully abstract models of SP under $\equiv$. This result will be useful when considering behavioural theories in the next section. (Note that our general assumption that $\equiv$ is factorizable by $\approx$ is still valid.)

Theorem 6.8. *Let $\approx$ be a regular family of partial Σ-congruences. Then for any class $C \subseteq \text{Alg}(\Sigma)$ of Σ-algebras and any specification* SP *of signature Σ the following holds:*

(1) *If C is closed under isomorphism, then* $\text{Beh}_{\approx}(C) = \text{Abs}_{\equiv}(\text{FA}_{\approx}(C))$,

(2) $\text{Mod}(\textbf{behaviour SP wrt} \approx) = \text{Abs}_{\equiv}(\text{FA}_{\approx}(\text{Mod}(\text{SP})))$.

Proof. (1) "$\subseteq$": Let $A \in \text{Beh}_{\approx}(C)$. Then $A/\approx_A \in C$ and, since $\approx$ is regular, $A/\approx_A$ is fully abstract, i.e. $A/\approx_A \in \text{FA}_{\approx}(C)$. Moreover, since regularity implies weak regularity, we have, by Lemma 5.8, $A \equiv A/\approx_A$. Hence, $A \in \text{Abs}_{\equiv}(\text{FA}_{\approx}(C))$.

"$\supseteq$": Let $A \in \text{Abs}_{\equiv}(\text{FA}_{\approx}(C))$. Then $A \equiv B$ for some $B \in \text{FA}_{\approx}(C)$. By factorizability, $A/\approx_A$ and $B/\approx_B$ are isomorphic and since B is fully abstract, $B/\approx_B$ is the same as $B/\approx_B$ which is isomorphic to B. Hence, $A/\approx_A$ is isomorphic to $B \in \text{FA}_{\approx}(C)$. Since C is closed under isomorphism, we obtain $A/\approx_A \in C$ and therefore $A \in \text{Beh}_{\approx}(C)$.

(2) Follows from (1). □

Theorem 6.8 again shows that $\text{Mod}(\textbf{behaviour SP wrt} \approx) \subseteq \text{Mod}(\textbf{abstract SP wrt} \equiv)$ (cf. Theorem 5.9) since $\text{Abs}_{\equiv}(\text{FA}_{\approx}(\text{Mod}(\text{SP}))) \subseteq \text{Abs}_{\equiv}(\text{Mod}(\text{SP})) = \text{Mod}(\textbf{abstract SP wrt} \equiv)$. The following corollary summarizes some further immediate consequences of Theorem 6.8. We see that the fully abstract models of an underlying specification SP are also models of the corresponding behavioural specification (cf. (1)) and that a behavioural specification has a model if and only if there exists a fully abstract model of the underlying specification SP (cf. (2)). In order to point out the analogy to the abstractor case, the corresponding properties of abstractor specifications are given in (3) and (4).

Corollary 6.9. *Let $\approx$ be a regular family of partial Σ-congruences. For any specification* SP *with signature Σ,*

(1) $\text{FA}_{\approx}(\text{Mod}(\text{SP})) \subseteq \text{Mod}(\textbf{behaviour SP wrt} \approx)$,

(2) $\text{Mod}(\textbf{behaviour SP wrt} \approx) \neq \emptyset$ *if and only if* $\text{FA}_{\approx}(\text{Mod}(\text{SP})) \neq \emptyset$,

(3) $\text{Mod}(\text{SP}) \subseteq \text{Mod}(\textbf{abstract SP wrt} \equiv)$,

(4) $\text{Mod}(\textbf{abstract SP wrt} \equiv) \neq \emptyset$ *if and only if* $\text{Mod}(\text{SP}) \neq \emptyset$.

We conclude this section by pointing out the connection between the characterization of behavioural specifications given in Theorem 5.15 (where we have proved that **behaviour** SP **wrt** $\approx$ is semantically equivalent to **abstract** (SP + SP/$\approx$) **wrt** $\equiv$) and the characterization of behavioural semantics in Theorem 6.8. The following proposition shows that the model class of (SP + SP/$\approx$) is just the isomorphic closure of the fully abstract models of SP. Hence, it is easy to see that the model class of **abstract** (SP + SP/$\approx$) **wrt** $\equiv$ is just the class $\mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(\mathrm{Mod}(\mathrm{SP})))$ (using the assumption that $\equiv$ is isomorphism protecting). Thus, the results of Theorems 5.15 and 6.8 are compatible with each other.

Proposition 6.10. *Let $\approx$ be a regular family of partial Σ-congruences. For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras and any specification* SP *with signature Σ, the following holds:*

(1) *If C is closed under isomorphism, then* $C \cap \mathrm{Iso}(C/\approx) = \mathrm{Iso}(\mathrm{FA}_{\approx}(C))$,

(2) $\mathrm{Mod}(\mathrm{SP} + \mathrm{SP}/\approx) = \mathrm{Iso}(\mathrm{FA}_{\approx}(\mathrm{Mod}(\mathrm{SP})))$.

Proof. (1) "$\subseteq$": Let $A \in C \cap \mathrm{Iso}(C/\approx)$. Then $A \in C$ and A is isomorphic to some quotient $B/\approx_B$ with $B \in C$. Since $A \in C$ and C is closed under isomorphism, $B/\approx_B \in C$. Moreover, since $\approx$ is regular, $B/\approx_B$ is fully abstract, i.e. $B/\approx_B \in \mathrm{FA}_{\approx}(C)$. Then, since A is isomorphic to $B/\approx_B$, we have $A \in \mathrm{Iso}(\mathrm{FA}_{\approx}(\mathrm{C}))$.

"$\supseteq$": Let $A \in \mathrm{Iso}(\mathrm{FA}_{\approx}(C))$. Then A is isomorphic to some $B \in \mathrm{FA}_{\approx}(C)$. Since $B \in C$ and since C is assumed to be closed under isomorphism we have $A \in C$. It remains to show that $A \in \mathrm{Iso}(C/\approx)$. A is isomorphic to B which in turn is isomorphic to the trivial quotient $B/=_B$. Since B is fully abstract, $B/=_B$ is the same as $B/\approx_B$ which belongs to the class $C/\approx$. Hence, $A \in \mathrm{Iso}(C/\approx)$.

(2) Follows from (1) by definition of $\mathrm{Mod}(\mathrm{SP} + \mathrm{SP}/\approx)$. □

7. Behavioural theories of behavioural and abstractor specifications

According to the generalization of the standard satisfaction relation to the behavioural satisfaction relation (cf. Section 3.1) we will consider here for any class C of Σ-algebras the behavioural theory of C, i.e. the set of all Σ-formulas which are behaviourally satisfied w.r.t. $\approx$ by all algebras in C. We recall that we still assume that $(\approx, \equiv)$ denotes a pair consisting of an isomorphism compatible and weakly regular family $\approx$ of partial Σ-congruences and an equivalence relation $\equiv$ on $\mathrm{Alg}(\Sigma)$ which is factorizable by $\approx$.

General Assumption 5. Whenever we consider fully abstract algebras in the following, we assume that $\approx$ is regular.

We will consider here theories consisting of arbitrary Σ-formulas (finitary or not, cf. Section 2.3). However, all results remain valid if we restrict to first-order theories (i.e. theories consisting only of finitary Σ-formulas) because first-order theories are the intersection of infinitary theories with the set of finitary Σ-formulas.

Definition 7.1 (*Behavioural theory*). For any class $C \subseteq \mathrm{Alg}(\Sigma)$ of Σ-algebras, $\mathrm{Th}_{\approx}(C)$ denotes the set of all Σ-formulas ϕ which are behaviourally satisfied w.r.t. $\approx$ by all algebras of C, i.e.

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

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

Lemma 7.2. *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *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) $\mathrm{Th}_{\approx}(\mathrm{FA}_{\approx}(C)) = \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C))$,

(4) *If* C *is closed under isomorphism and behaviourally consistent w.r.t.* $\approx$, *then* $\mathrm{Th}_{\approx}(C) = \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C))$.

Proof. (1) follows from Theorem 3.11, (2) follows from Proposition 5.5 and (3) is a consequence of Lemma 6.3. For proving (4), assume that C is isomorphically closed and behaviourally consistent w.r.t. $\approx$. Then, by Theorem 5.11(1), $\mathrm{Beh}_{\approx}(C) = \mathrm{Abs}_{\equiv}(C)$. Moreover, by Theorem 6.8, $\mathrm{Beh}_{\approx}(C) = \mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(C))$. Hence, $\mathrm{Abs}_{=}(C) = \mathrm{Abs}_{=}(\mathrm{FA}_{\approx}(\mathrm{C}))$. Then we have

$$\begin{aligned}
\mathrm{Th}_{\approx}(C) &= \mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(C)) \quad \text{(by (2))} \\
&= \mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(C))) \\
&= \mathrm{Th}_{\approx}(\mathrm{FA}_{\approx}(C) \quad \text{(again by (2))} \\
&= \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C)) \quad \text{(by (3))}. \qquad \square
\end{aligned}$$

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

Proposition 7.3. *For any class* $C \subseteq \mathrm{Alg}(\Sigma)$ *the following holds*:

(1) *If* C *is closed under isomorphism, then* $\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(C)) = \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C))$,

(2) $\mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(C)) = \mathrm{Th}_{=}(C/\approx)$.

Proof. (1) We have

$$\begin{aligned}
\mathrm{Th}_{\approx}(\mathrm{Beh}_{\approx}(C)) &= \mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(\mathrm{FA}_{\approx}(C))) \quad \text{(by Theorem 6.8)} \\
&= \mathrm{Th}_{\approx}(\mathrm{FA}(C)) \quad \text{(by Lemma 7.2(2))} \\
&= \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(C)) \quad \text{(by Lemma 7.2(3))}.
\end{aligned}$$

(2) We have

$$\begin{aligned}
\mathrm{Th}_{\approx}(\mathrm{Abs}_{\equiv}(C)) &= \mathrm{Th}_{\approx}(C) \quad \text{(by Lemma 7.2(2))} \\
&= \mathrm{Th}_{=}(C/\approx) \quad \text{(by Lemma 7.2(1))}.
\end{aligned}$$

This completes the proof. $\square$

Proposition 7.3 leads immediately to the following theorem which shows that the behavioural theories of behavioural and abstractor specifications can be characterized by standard theories. In particular, the first part of Theorem 7.4 says that the theory of a behavioural specification which is built on top of a 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 behavioural 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 which are built on top of a basic specification SP is studied in [4]. More elaborated proof techniques for behavioural theories of arbitrary specifications are developed in [7].

Theorem 7.4. *For any specification* SP *with signature* Σ *the following holds*:

(1) $\mathrm{Th}_{\approx}(\mathrm{Mod}(\textbf{behaviour SP wrt } \approx)) = \mathrm{Th}_{=}(\mathrm{FA}_{\approx}(\mathrm{Mod}(\mathrm{SP})))$,

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

Proof. Since $\mathrm{Mod}(\textbf{behaviour SP wrt } \approx) = \mathrm{Beh}_{\approx}(\mathrm{Mod}(\mathrm{SP}))$ and $\mathrm{Mod}(\textbf{abstract SP wrt } \equiv) = \mathrm{Abs}_{\equiv}(\mathrm{Mod}(\mathrm{SP}))$, the theorem is a consequence of Proposition 7.3 if we choose $C = \mathrm{Mod}(\mathrm{SP})$, which is closed under isomorphism. In particular, $\mathrm{Th}_{=}(\mathrm{Mod}(\mathrm{SP})/\approx) = \mathrm{Th}_{=}(\mathrm{Mod}(\mathrm{SP}/\approx))$ holds since $\mathrm{Mod}(\mathrm{SP}/\approx) = \mathrm{Iso}(\mathrm{Mod}(\mathrm{SP})/\approx)$ and since isomorphic algebras satisfy the same Σ-formulas. □

Example 7.5. Let **behaviour** SP **wrt** $\approx_{\mathrm{Obs,In}}$ be an observational behaviour specification. Then $\mathrm{Th}_{\approx_{\mathrm{Obs,In}}}(\mathrm{Mod}(\textbf{behaviour SP wrt } \approx_{\mathrm{Obs,In}}))$ consists of all Σ-formulas which are observationally satisfied by the models of the observational behaviour specification. By Theorem 7.4 (1), this is the same as the standard theory of the fully abstract models of SP. In Example 6.4, we have shown that a fully abstract algebra satisfies (in the standard sense) an equation $t = r$ if and only if it satisfies all equations $c[t] = c[r]$ for all observable contexts c. Hence, for proving that an equation $t = r$ is an observational theorem over an observational behaviour specification, it is enough to prove that for any observable context c, the equation $c[t] = c[r]$ belongs to the standard theory of SP (and hence in particular is valid in all fully abstract models of SP). For this, one can use, for instance, the context induction proof technique (cf. [12]). In [4], it is shown how an explicit use of context induction can be avoided. The idea is that under a particular condition for SP (called *observability kernel*), the characterization of the equality $x = y$ in the fully abstract models by the (usually) infinite set of equations $c[x] - c[y]$ for all observable contexts c can be replaced by a finite set of equations $c_0[x] = c_0[y]$ where c_0 ranges over a finite set C_0 of observable contexts. Then observational theorems are just standard theorems of the specification SP enriched by the finitary axiom

$$\left(\bigwedge_{c_0 \in C_0} \forall \mathrm{Var}(c_0).c_0[x] = c_0[y] \right) \Rightarrow x = y.$$

As a concrete example consider the last two axioms $\mathrm{add}(x, \mathrm{add}(y, s)) = \mathrm{add}(y, \mathrm{add}(x, s))$ and $\mathrm{add}(x, \mathrm{add}(x, s)) = \mathrm{add}(x, s)$ of the SET specification and assume that SET1 is obtained from SET by omitting these two equations and that SET2 = **behaviour** SET1 **wrt** $\approx_{\mathrm{Obs,In}}$ where Obs = {bool, elem} and In is arbitrary. According to [4] the equality of sets in the fully abstract models of SET1 can be characterized by the following finitary axiom:

$$\forall s1, s2 : \mathrm{set}.\ [(\forall x : \mathrm{elem}.\ \mathrm{iselem}(x, s1) = \mathrm{iselem}(x, s2)) \Rightarrow s1 = s2].$$

Then observational theorems over SET2 are just standard theorems over the specification SET1 enriched by the above axiom for full abstractness. For instance, using the axiom for full abstractness, it is now easy to prove that the omitted equations $\mathrm{add}(x, \mathrm{add}(y, s)) = \mathrm{add}(y, \mathrm{add}(x, s))$ and $\mathrm{add}(x, \mathrm{add}(x, s)) = \mathrm{add}(x, s)$ are valid in the fully abstract models of SET1 and hence are observational theorems over SET2. A detailed study of proof techniques for observational theorems over arbitrary (not only behavioural) specifications is given in [7].

Example 7.6. Consider the specification **init** SP and the family $\approx_{E,\emptyset}$ of partial Σ-congruences of Example 3.25(2). Since Mod(**init** SP) consists of the isomorphism class of the initial models of SP, all models of **init** SP are fully abstract (cf. Example 6.2(2)). Moreover, we know that **behaviour** (**init** SP) **wrt** $\approx_{E,\emptyset}$ and **abstract** (**init** SP) **wrt** $\equiv_{\approx_{E,\emptyset}}$ are semantically equivalent (cf. Example 5.13). Hence, we have $\mathrm{Th}_{\approx_{E,\emptyset}}$(**abstract** (**init** SP) **wrt** $\equiv_{\approx_{E,\emptyset}}$) = $\mathrm{Th}_{\approx_{E,\emptyset}}$(**behaviour** (**init** SP) **wrt** $\approx_{E,\emptyset}$) = (by Theorem 7.4(1)) $\mathrm{Th}_{=}(\mathrm{FA}_{\approx_{E,\emptyset}}(\mathrm{Mod}(\textbf{init SP})))$ = $\mathrm{Th}_{=}(\mathrm{Mod}(\textbf{init SP}))$. In particular, an equation $t = r$ belongs to the behavioural theory of **behaviour** (**init** SP) **wrt** $\approx_{E,\emptyset}$ iff $t = r$ belongs to the standard theory of **init** SP.

8. 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 express each one by the other (provided that the underlying equivalence on algebras is factorizable). We have given necessary and sufficient conditions for the semantical equivalence of behavioural and abstractor specifications which subsume the theorems of Reichel and of Nivela and Orejas. An extension of our characterization theorem to higher-order logic was recently presented in [14]. As examples of factorizable equivalences, we have considered the observational equivalences of algebras w.r.t. a set of observable sorts and w.r.t. a set of input sorts for the observable computations.

Our semantical results lead to proof-theoretic considerations which show that behavioural theories of specifications can be reduced to standard theories of some classes of algebras. An elaborated study of proof techniques for behavioural theorems over arbitrary (structured) specifications, based on axiomatizations of behavioural equalities, is given in [7]. In [14], it is shown that a (finite) axiomatization of the observational equality exists using higher-order logical formulas. A different approach providing a sound and complete proof system for structured specifications with observability operators, based on infinitary proof rules, is presented in [13].

An important application of behavioural or abstractor semantics is the formalization of correctness concepts in program development (cf. e.g. [24]). Using our concept of behavioural semantics, we have studied in [8] behavioural implementations and we have investigated proof rules that allow us to establish the correctness of behavioural implementations of structured specifications in a modular way.

Acknowledgements

This work was partially sponsored by the French–German cooperation programme PROCOPE, by the ESPRIT Working Group COMPASS and by the German DFG project SPECTRUM. Special thanks are due to Andrzej Tarlecki, Don Sannella and Fernando Orejas for stimulating discussions. In particular, we would like to thank Andrzej Tarlecki for various suggestions that helped us to improve previous versions of this paper.

References

[1] E. Astesiano, A. Giovini and G. Reggio, Observational structures and their logic, *Theoret. Comput. Sci.* **96** (1992) 249–283.

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

[3] G. Bernot and M. Bidoit, Proving the correctness of algebraically specified software: modularity and observability issues, in: *Proc. AMAST '91*, Workshops in Computing Series (Springer, Berlin, 1992) 216–242.

[4] M. Bidoit and R. Hennicker, Proving behavioural theorems with standard first-order logic, in: *Proc. ALP '94, 4th Internat. Conf. on Algebraic and Logic Programming*, Lecture Notes in Computer Science, Vol. 850 (Springer, Berlin, 1994) 41–58.

[5] M. Bidoit, R. Hennicker and M. Wirsing, Behavioural and abstractor specifications, Report LIENS-94-10, Ecole Normale Supéricure, 1994; a short version appeared as: Characterizing behavioural semantics and abstractor semantics, in: *Proc. ESOP '94, 5th European Symp. on Programming*, Lecture Notes in Computer Science, Vol. 788 (Springer, Berlin, 1994) 105–119.

[6] M. Bidoit and R. Hennicker, Behavioural theories, in: E. Astesiano, G. Reggio and A. Tarlecki, eds., *Recent Trends in Data Type Specification, 10th Workshop on Specification of Abstract Data Types joint with the 5th COMPASS Workshop*, 1994, Selected Papers, Lecture Notes in Computer Science, Vol. 906 (Springer, Berlin, 1995) 153–169.

[7] M. Bidoit and R. Hennicker, Behavioural theories and the proof of behavioural properties, Theoret. Comput. Sci, to appear; preliminary version in: Report LIENS-95-5, Ecole Normale Supérieure, 1995.

[8] M. Bidoit and R. Hennicker, Proving the correctness of behavioural implementations, in: *Proc. AMAST '95*, Lecture Notes in Computer Science, Vol. 936 (Springer, Berlin, 1995) 152–168.
[9] H. Ehrig and B. Mahr, Fundamentals of algebraic specification 1, EATCS Monographs on Theoretical Computer Science. Vol. 6 (Springer, Berlin, 1985).
[10] H. Ehrig and B. Mahr, Fundamentals of algebraic specification 2, EATCS Monographs on Theoretical Computer Science, Vol. 21 (Springer, Berlin, 1990).
[11] J.A. Goguen and J. Meseguer, Completeness of many-sorted equational logic, *Houston J. Math.* **11** (3) (1985) 307–334; preliminary version appeared in: *SIGPLAN Notices* **16** (7) (1991), 24–32.
[12] R. Hennicker, Context induction: a proof principle for behavioural abstractions and algebraic implementations, *Formal Aspects Comput.* **4** (3) (1991) 326–345.
[13] R. Hennicker, M. Wirsing and M. Bidoit, Proof systems for structured specifications with observability operators, *Theoret. Comput. Sci.*, to appear.
[14] M. Hofmann and D.T. Sannella, On behavioural abstraction and behavioural satisfaction in higher-order logic, Report ECS-LFCS-95-318, University of Edinburgh, 1995; a short version will appear in: *Proc. TAPSOFT '95*, Lecture Notes in Computer Science, Vol. 915 (Springer, Berlin, 1995) 247–261.
[15] H.J. Keisler, *Model theory for Infinitary Logic* (North-Holland, Amsterdam, 1971).
[16] T. Knapik, Specifications with observable formulae and observational satisfaction relation, in: M. Bidoit and C. Choppy, eds., *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 655 (Springer, Berlin, 1993) 271–291.
[17] G. Kreisel and J.L. Krivine, *Eléments de Logique Mathématique* (Dunod, Paris, 1967).
[18] R. Milner, Fully abstract models of typed λ-calculi, *Theoret. Comput. Sci.* **4** (1977) 1–22.
[19] P. Nivela and F. Orejas, Initial behaviour semantics for algebraic specifications, in: D.T. Sannella and A. Tarlecki, eds., *Recent Trends in Data Type Specification, 5th Workshop on Specification of Abstract Data Types*, 1987, Lecture Notes in Computer Science, Vol. 332 (Springer, Berlin, 1988) 184–207.
[20] F. Orejas, M. Navarro and A. Sanchez, Implementation and behavioural equivalence: a survey, in: M. Bidoit and C. Choppy, eds., *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 655 (Springer, Berlin, 1993) 93–125.
[21] H. Reichel, Behavioural equivalence — a unifying concept for initial and final specification methods, in: M. Arato and L. Varga, eds., *Math. Models in Comp. Systems, Proc. 3rd Hungarian Computer Science Conf.* Budapest (1981) 27–39.
[22] H. Reichel, Initial computability, algebraic specifications, and partial algebras, International Series of Monographs in Computer Science, No. 2 (Clarendon Press, Oxford, 1987).
[23] D.T. Sannella and A. Tarlecki, On observational equivalence and algebraic specification, *J. Comput. System Sci.* **34** (1987) 150–178.
[24] D.T. Sannella and A. Tarlecki, Toward formal development of programs from algebraic specifications: implementation revisited, *Acta Inform.* **25** (1988) 233–281.
[25] D.T. Sannella and M. Wirsing, Implementation of parameterised specifications, in: *Proc. ICALP '82, 9th Colloq. on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 140 (Springer, Berlin, 1982) 473–488.
[26] D.T. Sannella and M. Wirsing, A kernel language for algebraic specification and implementation, in: M. Karpinski, ed., *Colloq. on Foundations of Computation Theory*, Lecture Notes in Computer Science, Vol. 158 (Springer, Berlin, 1983) 413–427.
[27] O. Schoett, Data abstraction and correctness of modular programming, Ph. D. Thesis, CST-42-87, University of Edinburgh, 1987.
[28] M. Wirsing, Structured algebraic specifications: a kernel language. *Theoret. Comput. Sci.* **42** (1986) 123–249.
[29] M. Wirsing, Algebraic specification, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science* (Elsevier, Amsterdam, 1990) 675–788.
[30] H.-D. Ehrlich, M. Gogolla and U.W. Lipeck, *Algebraische Spezifikation abstrakter Datentypen* (Teubner, Stuttgart, 1989)