Foundations for Structuring Behavioural Specifications

Răzvan Diaconescu

Simion Stoilow Institute of Mathematics of the Romanian Academy

Ionuţ Ţuţu

Department of Computer Science, Royal Holloway University of London, and
Institute of Mathematics of the Romanian Academy, Research group of the project ID-3-0439

Abstract

We develop foundations for structuring behavioural specifications based on the logic tradition of hidden algebra. This includes an analysis of a number of important technical compositional properties for behavioural signatures, such as pushouts, inclusions and unions, as well as an investigation of algebraic rules for behavioural module composition. As a particularity of behavioural specifications, some of the constructions and results arise in a partial algebraic form. This partiality aspect is one of the distinguishing features of our approach to behavioural specification modules. In addition, our study does not commit to any actual choice of structuring constructs, thus being applicable to a wide variety of structuring situations.

1. Introduction

Modern algebraic specification theory and practice has extended the traditional many-sorted algebra-based specification to several new paradigms. One of the most promising is behavioural specification, which originates from the work of Horst Reichel [31, 32] and can be found in the literature under names such as hidden algebra [24, 25], observational logic [4, 27], coherent hidden algebra [19] and hidden logic [33]. Behavioural specification characterises how objects (and systems) *behave*, not how they are implemented. This new form of abstraction can be very powerful for the specification and verification of software systems since it naturally embeds other useful paradigms such as concurrency, object-orientation, constraints, nondeterminism, etc. (see [25] for details). In the tradition of algebraic specification, the behavioural abstraction is achieved by using specification with hidden sorts and a behavioural concept of satisfaction based on the idea of indistinguishability of states that are observationally the same, which also generalizes process algebra and transition systems (see [25]).

An important effort has been undertaken to develop languages and systems supporting the behavioural extension of conventional or less conventional algebraic specification techniques; these include CafeOBJ [18, 20], CIRC [35] and BOBJ [33]. In other situations, behavioural specification, although not directly realized at the level of the language definition, is employed as a mere methodological device [5]. In all cases there is the unavoidable need of a structuring mechanism for behavioural specifications. Structuring or modularization is in fact a common aspect of any formal method that aims at assisting the development of complex systems; without it such developments are simply not possible. Due to its important role in

Email addresses: Razvan.Diaconescu@imar.ro (Răzvan Diaconescu), ittutu@gmail.com (Ionuţ Ţuţu)

formal specification development (see for example [37]), the study of modularization is supported by a rather vast scientific literature developed over the past four decades. In the case of logic-based formal methods it has become standard to approach the study of modularization of systems through the so-called institution theory of Goguen and Burstall [23]. This trend has led to a fairly uniform understanding of various systems of modularization or structuring, one that has been successfully applied to the design of modern algebraic specification languages such as CASL [2] or CafeOBJ [18, 20]. However, in spite of this, due to the inherent complexity and difficulty of the subject, the study of modularization is far from being closed as insufficiently explored aspects still exist. For instance, the recent works [17, 15] attempt to give an answer as complete as possible to the issue of instantiating multiple parameters in a sharing context. Moreover, behavioural specification itself poses a particular challenge given by some of the specificities of its underlying logic that raise serious obstacles when attempting to apply established general theories on modularization. For example, one major source of problems is given by the fact that the union or aggregation of behavioural signatures is inherently partial rather than total and moreover, by noticing that this partiality is induced by two different factors. On the one hand, one may not aggregate signatures that share a sort name that is declared visible in one of the signatures and hidden in the other. On the other hand, any aggregation of signatures has to fulfil the encapsulation condition characteristic to behavioural signature morphisms, which cannot be guaranteed in all possible situations; this property is essential for the basic *satisfaction condition* of the underlying logic to hold, which in turn is absolutely necessary with respect to modularization. The requirement of both of these hypotheses (i.e. the preservation and the reflection of sorts' visibility, together with the encapsulation condition) has been extensively discussed in the literature (e.g. [25, 27]), not only from a technical perspective, but also in terms of their practical relevance. Hence, under the current general assumptions on behavioural specification, the union of signatures arises naturally as a partial operation.

In this paper we develop foundations for structuring of behavioural specifications in the light of the most recent developments in structuring specifications in general [17, 15]. This means reliance upon well established theoretical devices used in modularization studies such as institutions, pushouts, and inclusion systems. It also means that only some of the concepts and results already available at the general level may be applied directly, while much has to be reconsidered for the specific situation of behavioural specifications, thus leading to a series of new theoretical investigations.

The structure and the contents of the paper can be briefly described as follows. After a first preliminary section that recalls a number of concepts and results from previous works on the modularization of specifications on which we rely in our work, we present our contribution in two main sections:

1. The first one is devoted to our choice of a logical system that underlies behavioural specification. We introduce the institution of hidden algebra, detail the connection with related work, and develop a series of technical properties that are required by our modularization study. This includes pushouts and inclusion systems for behavioural signatures, as well as the investigation of several basic algebraic rules for the composition of behavioural signatures that are important for our study. A characteristic of these rules is that they are given in the style of partial algebras (in the sense of [9]). In contrast to their conventional corresponding variants they are conditional and, in addition, non-trivial to obtain. For example, the associativity of the union or aggregation of signatures, which in general follows immediately as a property of suprema, here, because of the partiality of the union of signatures, follows by reliance on a series of specific technical results.

2. In the second main section we introduce our basic framework for structuring behavioural specifications, which is based on recent ideas from [15]. An immediate consequence of the abstract nature of the central definitions is that we are able to develop our study on the structuring of behavioural specifications independently of any actual choice of specification building operators; this means direct

applicability to a wide range of actual specification languages and systems. Moreover, the base institution is also considered abstractly by axiomatising some of the compositionality properties of hidden algebra; in this way, our work may be applied to other behavioural specification logics and even to other specification logics that share with hidden algebra some compositional properties.

We also develop a few important algebraic rules for structured behavioural specifications, again in a partial algebra style that is inherited from the level of signatures. This partiality aspect is unique to our development since the module algebra literature [3, 21, 37] has considered thus far only total algebraic rules. In fact, our results constitute a generalisation of those module algebra works in the sense in which partial algebra is a generalisation of total algebra.

2. Preliminaries

In this section we recall a series of well-established concepts in the specification literature that are of central importance for the mathematical and logical foundations of modularization.

2.1. Categories

Institution theory relies technically upon category theory. We assume the reader is familiar with basic notions and standard notations from category theory. With few exceptions, in general we follow the terminology and the notations of [29]. With respect to notational conventions, $|\mathbb{C}|$ denotes the class of objects of a category $\mathbb{C}$, $\mathbb{C}(A, B)$ the set of arrows (morphisms) with domain A and codomain B, and ";" the composition (in diagrammatic order). A subcategory $\mathbb{C}'$ of $\mathbb{C}$ is *broad* when $|\mathbb{C}'| = |\mathbb{C}|$. The category of sets (as objects) and functions (as arrows) is denoted by $\mathbb{S}\mathrm{et}$, and the category of all categories (as objects) and functors (as arrows)[1] is denoted by $\mathbb{CAT}$.

2.2. Institutions

Institutions have been defined by Goguen and Burstall in [10], the seminal paper [23] being printed after a delay of many years. Below we recall the concept of institution, which formalises the intuitive notion of logical system, including the syntax, semantics and the satisfaction between them.

Definition 2.1 (Institutions)**.** An *institution* $\mathcal{I} = (\mathrm{Sign}^{\mathcal{I}}, \mathrm{Sen}^{\mathcal{I}}, \mathrm{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$ consists of

1. a category $\mathrm{Sign}^{\mathcal{I}}$, whose objects are called *signatures*,
2. a functor $\mathrm{Sen}^{\mathcal{I}} : \mathrm{Sign}^{\mathcal{I}} \to \mathbb{S}\mathrm{et}$, giving for each signature a set whose elements are called *sentences* over that signature,
3. a functor $\mathrm{Mod}^{\mathcal{I}} : (\mathrm{Sign}^{\mathcal{I}})^{\mathrm{op}} \to \mathbb{CAT}$ giving for each signature Σ a category whose objects are called Σ-*models*, and whose arrows are called Σ-(*model*) *homomorphisms*, and
4. a relation $\models^{\mathcal{I}}_{\Sigma} \subseteq |\mathrm{Mod}^{\mathcal{I}}(\Sigma)| \times \mathrm{Sen}^{\mathcal{I}}(\Sigma)$ for each $\Sigma \in |\mathrm{Sign}^{\mathcal{I}}|$, called Σ-*satisfaction*,

such that for each morphism $\varphi\colon \Sigma \to \Sigma'$ in $\mathrm{Sign}^{\mathcal{I}}$, the *satisfaction condition*

$$M' \models^{\mathcal{I}}_{\Sigma'} \mathrm{Sen}^{\mathcal{I}}(\varphi)(\rho) \quad \text{if and only if} \quad \mathrm{Mod}^{\mathcal{I}}(\varphi)(M') \models^{\mathcal{I}}_{\Sigma} \rho$$

holds for each $M' \in |\mathrm{Mod}^{\mathcal{I}}(\Sigma')|$ and $\rho \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$.

[1] Strictly speaking, this is only a quasi-category living in a higher set-theoretic universe.

We denote the *reduct* functor $\mathrm{Mod}^{\mathcal{I}}(\varphi)$ by $\_\!\upharpoonright_{\varphi}$ and the sentence translation $\mathrm{Sen}^{\mathcal{I}}(\varphi)$ by $\varphi(\_)$. When $M = M'\!\upharpoonright_{\varphi}$ we say that M is a *φ-reduct of M'*, and that M' is a *φ-expansion of M*. When there is no danger of ambiguity, we may skip the superscripts from the notations of the entities of the institution; for example, $\mathrm{Sign}^{\mathcal{I}}$ may be denoted simply by Sign.

General assumption: We assume that model isomorphisms preserve the satisfaction of all sentences of the considered institutions, i.e. if M and N are isomorphic (denoted $M \cong N$) then for each sentence ρ we have that $M \models \rho$ if and only if $N \models \rho$. It is easy to see that this assumption holds in all the concrete examples of institutions of interest for specification and programming.

There is a myriad of examples of logics captured as institutions, both from logic and from computing. A few of them can be found in [13, 37]. In fact, the thesis underlying institution theory is that anything that deserves to be called logic can be captured as an institution. The following is the most traditional institution in algebraic specification.

Example 2.1 (Many sorted algebra – $\underline{MSA}$)**.** The $\underline{MSA}$*-signatures* are pairs (S, F) consisting of a set S of sort symbols and of a family $F = \{F_{w\to s} \mid w \in S^*, s \in S\}$ of sets of function symbols indexed by arities (for the arguments) and sorts (for the results). *Signature morphisms* $\varphi\colon (S, F) \to (S', F')$ consist of a function $\varphi^{\mathrm{st}}\colon S \to S'$ and a family of functions $\varphi^{\mathrm{op}} = \{\varphi^{\mathrm{op}}_{w\to s}\colon F_{w\to s} \to F'_{\varphi^{\mathrm{st}}(w)\to\varphi^{\mathrm{st}}(s)} \mid w \in S^*, s \in S\}$.

The (S, F)*-models* M, called algebras, interpret each sort symbol s as a set M_s and each function symbol $\sigma \in F_{w\to s}$ as a function M_σ from the product M_w of the interpretations of the argument sorts to the interpretation M_s of the result sort. An (S, F)*-model homomorphism* $h\colon M \to M'$ is an indexed family of functions $\{h_s\colon M_s \to M'_s \mid s \in S\}$ such that $h_s(M_\sigma(m)) = M'_\sigma(h_w(m))$ for each $\sigma \in F_{w\to s}$ and each $m \in M_w$, where $h_w\colon M_w \to M'_w$ is the canonical componentwise extension of h, i.e. $h_w(m_1, \ldots, m_n) = (h_{s_1}(m_1), \ldots, h_{s_n}(m_n))$ for $w = s_1 \ldots s_n$ and $m_i \in M_{s_i}$.

For each signature morphism $\varphi\colon (S, F) \to (S', F')$, the *reduct* $M'\!\upharpoonright_{\varphi}$ of an (S', F')-model M' is defined by $(M'\!\upharpoonright_{\varphi})_x = M'_{\varphi(x)}$ for each sort or function symbol x from the domain signature of φ.

For each signature (S, F), $T_{(S,F)} = ((T_{(S,F)})_s)_{s\in S}$ is the least family of sets such that $\sigma(t) \in (T_{(S,F)})_s$ for all $\sigma \in F_{w\to s}$ and all tuples $t \in (T_{(S,F)})_w$. The elements of $(T_{(S,F)})_s$ are called (S, F)*-terms of sort* s. For each (S, F)-algebra M, the *evaluation of an* (S, F)*-term* $\sigma(t)$ *in* M, denoted $M_{\sigma(t)}$, is defined as $M_\sigma(M_t)$, where M_t is the componentwise evaluation of the tuple of (S, F)-terms t in M.

Sentences are the usual first order sentences built from equational atoms $t = t'$, with t and t' (well-formed) terms of the same sort, by iterative application of Boolean connectives ($\wedge$, $\Rightarrow$, $\neg$, $\vee$) and quantifiers ($\forall X$, $\exists X$ – where X is a sorted set of variables). Sentence translations along signature morphisms just rename the sort and function symbols according to the respective signature morphisms. They can be formally defined by recursion on the structure of the sentences. The satisfaction of sentences by models is the usual Tarskian satisfaction defined recursively on the structure of the sentences. (As a special note for the satisfaction of the quantified sentences, defined in this formalisation by means of model reducts, we recall that $M \models_\Sigma (\forall X)\rho$ if and only if $M' \models_{\Sigma+X} \rho$ for each expansion M' of M to the signature $\Sigma + X$ that adds the variables X as new constants to Σ.)

2.3. Pushouts of signatures and model amalgamation

The crucial role of model amalgamation for the semantic studies of formal specifications comes up in a large number of works in the area, a few early cases being [6, 36, 38, 30, 21]. The model amalgamation property is a necessary condition in many institution-independent model-theoretic results (see [13]), thus being one of the most desirable properties for an institution. It can be considered even as more fundamental than the satisfaction condition since in institutions with quantifications it is used in one of its weak forms in the proof of the satisfaction condition at the induction step corresponding to quantifiers (see [13] for the details). Its importance within the context of module algebra has been first emphasized in [21]. Model amalgamation properties for institutions formalise the possibility of amalgamating models of different signatures when they are consistent on some kind of generalized 'intersection' of signatures.

Definition 2.2 (Amalgamation square)**.** A commutative square of signature morphisms

$$\begin{tikzcd}
\Sigma \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & \Sigma_1 \arrow[d, "\theta_1"] \\
\Sigma_2 \arrow[r, "\theta_2"'] & \Sigma'
\end{tikzcd}$$

is an *amalgamation square* if and only if for each Σ_1-model M_1 and Σ_2-model M_2 such that $M_1\upharpoonright_{\varphi_1} = M_2\upharpoonright_{\varphi_2}$, there exists an unique Σ'-model M', denoted $M_1 \otimes_{\varphi_1,\varphi_2} M_2$, or $M_1 \otimes M_2$ for short when there is no danger of ambiguity, such that $M'\upharpoonright_{\theta_1} = M_1$ and $M'\upharpoonright_{\theta_2} = M_2$. When we drop the uniqueness requirement we call this a *weak model amalgamation square*.

In most of the institutions formalising conventional or non-conventional logics, pushout squares of signature morphisms are model amalgamation squares [21, 13].

Definition 2.3 (Model amalgamation and semi-exactness)**.** An institution *has (weak) model amalgamation* when each pushout square of signatures is a (weak) amalgamation square. A *semi-exact institution* is an institution with the model amalgamation property extended also to model homomorphisms.

The literature considers also extensions of model amalgamation from pushouts to arbitrary colimits; however, for reasons of simplicity of presentation and because they are by far the most important case with respect to the applications, in this paper we consider model amalgamation only for pushouts.

The result below is well known in the literature ([37, 13] are two of the many relevant references).

Proposition 2.1. *$\underline{MSA}$ has all pushouts of signature morphisms and is semi-exact.*

2.4. *Inclusion systems*

Inclusion systems were introduced in [21] as a categorical device supporting an abstract general study of structuring of specification and programming modules that is independent of any underlying logic. They have been used in a series of general module algebra studies such as [21, 26, 13] but also for developing axiomatisability [34, 12, 13] and definability [1] results within the framework of the so-called institution-independent model theory [13]. Inclusion systems capture categorically the concept of set-theoretic inclusion in a way reminiscent of how the rather notorious concept of factorization system [7] captures categorically the set-theoretic injections; however, in many applications the former are more convenient than the latter. Here we recall from the literature the basics of the theory of inclusion systems.

The definition below can be found in the recent literature on inclusion systems (e.g. [13]) and differs slightly from the original one of [21].

Definition 2.4 (Inclusion systems)**.** A pair of categories $\langle \mathcal{I}, \mathcal{E} \rangle$ is an *inclusion system* for a category $\mathbb{C}$ if $\mathcal{I}$ and $\mathcal{E}$ are two broad subcategories of $\mathbb{C}$ such that

1. $\mathcal{I}$ is a partial order (with the ordering relation denoted by $\subseteq$), and
2. every arrow f in $\mathbb{C}$ can be factored uniquely as $f = e_f; i_f$ with $e_f \in \mathcal{E}$ and $i_f \in \mathcal{I}$.

The arrows of $\mathcal{I}$ are called *abstract inclusions*, and the arrows of $\mathcal{E}$ are called *abstract surjections*. The domain of the inclusion i_f in the factorization of f is called the *image of* f and is denoted as $\mathrm{Im}(f)$ or $f(A)$ when A is the domain of f. An inclusion $i \colon A \to B$ may also be denoted simply by $A \subseteq B$.

An inclusion system $\langle \mathcal{I}, \mathcal{E} \rangle$ is said to be *epic* when all abstract surjections of $\mathcal{E}$ are epimorphisms in $\mathbb{C}$.

In [11] it is shown that the class $\mathcal{I}$ of abstract inclusions determines the class $\mathcal{E}$ of abstract surjections. In this sense, [11] gives an explicit equivalent definition of inclusion systems that is based only on the class $\mathcal{I}$ of abstract inclusions. The standard example of inclusion system is that from $\mathbb{S}\mathrm{et}$, with set theoretic inclusions in the role of the abstract inclusions and surjective functions in the role of the abstract surjections. The literature contains many other examples of inclusion systems for the categories of signatures and for the categories of models of various institutions from logic or from specification theory. Due to lack of space let us recall here only a couple of them.

Example 2.2 (Inclusion systems for $\underline{MSA}$-signatures)**.** Besides the trivial inclusion system that can be defined in any category (i.e. identities as abstract inclusions and all arrows as abstract surjections) the category of $\underline{MSA}$-signatures admits also the following two non-trivial inclusion systems:

inclusion system	abstract surjections $\varphi \colon (S, F) \to (S', F')$	abstract inclusions $(S, F) \subseteq (S', F')$
closed	$\varphi^{\mathrm{st}} \colon S \to S'$ surjective	$S \subseteq S'$ $F_{w \to s} = F'_{w \to s}$ for $w \in S^*$, $s \in S$
strong	$\varphi^{\mathrm{st}} \colon S \to S'$ surjective $F'_{w' \to s'} = \bigcup_{\varphi^{\mathrm{st}}(ws) = w's'} \varphi^{\mathrm{op}}(F_{w \to s})$	$S \subseteq S'$ $F_{w \to s} \subseteq F'_{w \to s}$ for $w \in S^*$, $s \in S$

Note that the strong inclusion system of the category of $\underline{MSA}$-signatures is epic.

Definition 2.5 (Union and intersection)**.** In any inclusion system, the *union* and the *intersection* of two objects A and B are, respectively, their least upper bound $A \cup B$ and their greatest lower bound $A \cap B$, relative to the partial order $\subseteq$.

In [21] it has been shown that whenever a category equipped with an inclusion system has pullbacks, the existence of unions implies the existence of intersections, which can be obtained as pullbacks of the structural morphisms that correspond to unions.

$$\begin{tikzcd}
A \cap B \arrow[r, "\subseteq"] \arrow[d, "\subseteq"'] & A \arrow[d, "\subseteq"] \\
B \arrow[r, "\subseteq"'] & A \cup B
\end{tikzcd}$$

It is often useful that the intersection-union squares are not only pullbacks, but they are also pushouts.

Example 2.3. In the strong inclusion system for $\underline{MSA}$-signatures, any two signatures Σ_1 and Σ_2 admit the union $\Sigma_1 \cup \Sigma_2$ and the intersection $\Sigma_1 \cap \Sigma_2$ (see [21]). Moreover, the lattice of inclusions is distributive and all the corresponding intersection-union squares of signatures are pushout squares. The closed inclusion system for $\underline{MSA}$-signatures does not enjoy any of these properties, and for this reason it is completely unsuitable for the development of specifications.

The following abstract concept captures a rather common situation in practice, including of course $\underline{MSA}$; it has been introduced in [21].

Definition 2.6 (Inclusive institution)**.** An institution is *inclusive* when its category of signatures is endowed with an inclusion system such that whenever $\Sigma \subseteq \Sigma'$, we have $\mathrm{Sen}(\Sigma) \subseteq \mathrm{Sen}(\Sigma')$.

3. The institution of hidden algebra

The logic underlying behavioural specification that we consider here is an upgraded version of the so-called hidden algebra [24] (abbreviated $\underline{HA}$). This section is structured as follows:

1. We first recall the institution of hidden algebra.
2. Then we develop results about the existence of pushouts and pullbacks of signatures, and we derive the model amalgamation property for hidden algebras.
3. We define an inclusion system for behavioural signatures and clarify the necessary conditions for the existence of unions of behavioural signatures.
4. Based upon results developed in the first parts of the section, in the final part we develop a number of algebraic rules for the union and intersection of behavioural signatures. Most notably, we investigate the associativity of the union and the distributivity of the union over the intersection; both properties arise in a conditional and partial form.

3.1. Signatures, sentences, models and satisfaction

Our presentation of the main concepts of hidden algebra given below represents an upgraded variant of the so-called coherent hidden algebra [19, 20] framework employed by CafeOBJ, which also covers the hidden algebra framework of [33]. This is both a simplification and an extension of the classical hidden algebra of [24, 25] in several directions, most notably by allowing operations with multiple hidden sorts in the arity, and differs only slightly from other modern formalisations of hidden algebra in the literature [27, 33]. $\underline{HA}$ is also significantly more general than coalgebra with final semantics [28] since it integrates smoothly data types and it allows behavioural operations with multiple hidden sorts.

Definition 3.1 (Behavioural signature)**.** An $\underline{HA}$-*signature* is a tuple (H, V, F, BF) where

- $(H \cup V, F)$ is an $\underline{MSA}$-signature with $H \cap V = \emptyset$; the sorts in V are called *visible sorts* and the sorts in H are called hidden sorts; and
- $(H \cup V, BF)$ is a subsignature of $(H \cup V, F)$ such that $BF_{w \to s} = \emptyset$ when $w \in V^*$; the operations of BF are called *behavioural operations*.

For any $\underline{HA}$-signature (H, V, F, BF) let $\mathrm{msa}(H, V, F, BF)$ denote its underlying $\underline{MSA}$-signature $(H \cup V, F)$.

Definition 3.2 (Hidden algebra)**.** Given a signature (H, V, F, BF), an (H, V, F, BF)-*algebra* is just an $\underline{MSA}$-algebra for the signature $\mathrm{msa}(H, V, F, BF)$.

Definition 3.3 (Hidden congruence)**.** Given an (H, V, F, BF)-algebra A, a *hidden* (H, V, F, BF)-*congruence* $\sim$ on A is an $(H \cup V, BF)$-congruence whose components on visible sorts are all identities.

Definition 3.4 (Behavioural equivalence)**.** The largest hidden (H, V, F, BF)-congruence on an (H, V, F, BF)-algebra A, denoted $\sim_A$, is called the *behavioural equivalence on* A.

A proof of the following crucial result can be found in several variants in several places in the literature; in the form represented by our particular hidden algebra formalisation it can be found in [14]. This result generalizes the final semantics employed by the early hidden algebra frameworks [24] or by the coalgebraic approaches [28] to the situation of behavioural operations with multiple hidden sorts in the arity and of loose interpretation of the visible part of the signature.

Theorem 3.1. *The behavioural equivalence exists for any* (H, V, F, BF)*-algebra.*

Definition 3.5 ($\underline{HA}$-sentence)**.** Given a $\underline{HA}$-signature (H, V, F, BF), the (H, V, F, BF)*-sentences* are built like the $\underline{MSA}$ $(H \cup V, F)$-sentences from two kinds of atoms, behavioural equations $t \sim t'$ and strict equations $t = t'$, by iteration of Boolean connectives ($\wedge$, $\vee$, $\neg$, $\Rightarrow$, etc.) and quantifications.

Definition 3.6 ($\underline{HA}$-satisfaction)**.** The satisfaction relation between (H, V, F, BF)-algebras and (H, V, F, BF)-sentences is defined like in $\underline{MSA}$, in the Tarski style, by recursion on the structure of sentences, with the addition that a hidden algebra A satisfies a behavioural equation $t \sim t'$ if and only if $A_t \sim_A A_{t'}$.

Definition 3.7 (Quasi-morphism of $\underline{HA}$-signatures)**.** A *quasi-morphism of* $\underline{HA}$*-signatures* $\varphi\colon (H, V, F, BF) \to (H', V', F', BF')$ is just an $\underline{MSA}$-signature morphism $\varphi\colon \mathrm{msa}(H, V, F, BF) \to \mathrm{msa}(H', V', F', BF')$ such that

- $\varphi(H) \subseteq H'$, $\varphi(V) \subseteq V'$, and
- the restriction of φ to $(H \cup V, BF)$ is an $\underline{MSA}$-signature morphism $(H \cup V, BF) \to (H' \cup V', BF')$.

Fact 3.1. *The* $\underline{HA}$*-signature quasi-morphisms are closed under composition.*

Definition 3.8 ($\underline{HA}$-signature morphism)**.** A quasi-morphism $\varphi\colon (H, V, F, BF) \to (H', V', F', BF')$ is a *signature morphism* if and only if the following 'encapsulation' condition holds:

$$\text{for any } \sigma' \in BF'_{w \to s}, \text{ if } w \cap \varphi(H) \neq \emptyset \text{ (i.e. } w \text{ contains an 'old' hidden sort)}$$
$$\text{then there exists } \sigma \text{ in } BF \text{ such that } \sigma' = \varphi(\sigma).$$

Notation 3.1. Given an $\underline{MSA}$-signature (S, F) and a sort $z \in S$ we define

$$F_{[z]} = \{(\sigma, w, s) \mid w \in S^*, s \in S, \sigma \in F_{w \to s}, z \in w\}.$$

Note that for any $\underline{MSA}$-signature (S, F) and sort $z \in S$, any morphism of $\underline{MSA}$-signatures $\varphi\colon (S, F) \to (S', F')$ induces a map $\varphi_{[z]}\colon F_{[z]} \to F'_{[\varphi(z)]}$ defined by $\varphi_{[z]}(\sigma, w, s) = (\varphi(\sigma), \varphi(w), \varphi(s))$.

Fact 3.2. *Let* $\varphi\colon (H, V, F, BF) \to (H', V', F', BF')$ *be any quasi-morphism of* $\underline{HA}$*-signatures. Then*

1. *If* φ *is injective on hidden sorts then* φ *is a morphism of signatures if and only if for any* $h \in H$ *the map* $\varphi_{[h]}\colon BF_{[h]} \to BF'_{[\varphi(h)]}$ *is surjective.*
2. *If* φ *is injective then* φ *is a morphism of signatures if and only if for any* $h \in H$ *the map* $\varphi_{[h]}\colon BF_{[h]} \to BF'_{[\varphi(h)]}$ *is bijective.*

Fact 3.3. *The* $\underline{HA}$*-signature morphisms are closed under composition.*

The additional 'encapsulation' condition of Dfn. 3.8 has the flavour of class encapsulation from object-oriented programming and guarantees that the behavioural equivalence on the 'old' (hidden) sorts is not changed. The following result is a very important consequence of this and has been proved in several different variants in several places in the literature (perhaps for the first time in [22] but within a significantly more restricted hidden-algebra context).

Corollary 3.1 ($\underline{HA}$-satisfaction condition)**.** *For any $\underline{HA}$-signature morphism $\varphi\colon \Sigma \to \Sigma'$, any Σ-sentence ρ and any Σ'-algebra A' we have that*

$$A' \models \varphi(\rho) \quad \textit{if and only if} \quad A'\!\upharpoonright_{\varphi} \models \rho.$$

Hence, $\underline{HA}$ is an institution.

In both [22] and [24] the authors remark that the derivation of the encapsulation condition on the morphisms of signatures from the meta-principle of invariance of truth under change of notation (the Satisfaction Condition of institutions) seems to confirm the naturalness of each of the principles.

3.2. Pushouts of signature morphisms

Pushouts of signature morphisms constitute one of the most important technical devices employed in various kinds of module compositions such as module aggregation (or sum), and most notably in the instantiation of parameterised modules. In this section we study the existence of pushouts of signature morphisms in $\underline{HA}$ by lifting them from $\underline{MSA}$ in two steps; first to the category of quasi-morphisms and then to the category of $\underline{HA}$-signature morphisms. Colimits of $\underline{MSA}$-signatures, and pushouts in particular, are a classic of algebraic specification literature and we do not recall their construction here. For the readers keen to understand these from scratch we recommended references such as [39, 13].

Proposition 3.1. *The forgetful functor from the category of quasi-morphisms of $\underline{HA}$-signatures to* $\mathrm{Sign}^{\underline{MSA}}$ *lifts pushouts.*

Proof. Let (φ_1, φ_2) be a span of quasi-morphisms of signatures as depicted below.

$$\begin{tikzcd}
(H_0, V_0, F_0, BF_0) \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & (H_1, V_1, F_1, BF_1) \\
(H_2, V_2, F_2, BF_2) &
\end{tikzcd}$$

Assuming that (θ_1, θ_2) is the following pushout of the underlying $\underline{MSA}$-signature morphisms of φ_1 and φ_2, let H and V be the sets $\theta_1(H_1) \cup \theta_2(H_2)$ and $\theta_1(V_1) \cup \theta_2(V_2)$, respectively. Then $S = H \cup V$.

$$\begin{tikzcd}
(H_0 \cup V_0, F_0) \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & (H_1 \cup V_1, F_1) \arrow[d, "\theta_1"] \\
(H_2 \cup V_2, F_2) \arrow[r, "\theta_2"'] & (S = H \cup V, F)
\end{tikzcd}$$

Let us show that $H \cap V = \emptyset$. Since for each $k \in \{0, 1, 2\}$, $H_k \cap V_k = \emptyset$ and for each $k \in \{1, 2\}$, $\varphi_k(H_0) \subseteq H_k$ and $\varphi_k(V_0) \subseteq V_k$, there exists H', V' such that $H' \cap V' = \emptyset$, and a commutative square like below such that $g_i(H_i) \subseteq H'$ and $g_i(V_i) \subseteq V'$, for $i \in \{1, 2\}$ (these may be obtained for example by considering pushouts of the restrictions of φ_1 and φ_2 to hidden and visible sorts, respectively).

$$\begin{tikzcd}
H_0 \cup V_0 \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & H_1 \cup V_1 \arrow[d, "g_1"] \\
H_2 \cup V_2 \arrow[r, "g_2"'] & H' \cup V'
\end{tikzcd}$$

Therefore, by the pushout property of (θ_1, θ_2), there exists $f\colon S \to H' \cup V'$ such that $\theta_1; f = g_1$ and $\theta_2; f = g_2$. If there existed $h_i \in H_i$ and $v_j \in V_j$ such that $\theta_i(h_i) = \theta_j(v_j)$ then $g_i(h_i) = g_j(v_j) \in H' \cap V'$, which contradicts $H' \cap V' = \emptyset$. Hence, $H \cap V = \emptyset$.

We extend θ_1 and θ_2 to quasi-morphisms of signatures by letting σ in BF if and only if there exists $k \in \{1, 2\}$ and σ_k in BF_k such that $\sigma = \theta_k(\sigma_k)$. The resulting square is a commuting square of quasi-morphisms of signatures.

It remains to prove that for any quasi-morphisms $\zeta_k\colon (H_k, V_k, F_k, BF_k) \to (H'', V'', F'', BF'')$, where $k \in \{1, 2\}$, such that $\varphi_1; \zeta_1 = \varphi_2; \zeta_2$ there exists a unique quasi-morphism $\zeta\colon (H, V, F, BF) \to (H'', V'', F'', BF'')$ verifying the equality $\theta_k; \zeta = \zeta_k$.

$$\begin{tikzcd}
(H_0, V_0, F_0, BF_0) \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & (H_1, V_1, F_1, BF_1) \arrow[d, "\theta_1"'] \arrow[rdd, bend left, "\zeta_1"] & \\
(H_2, V_2, F_2, BF_2) \arrow[r, "\theta_2"] \arrow[rrd, bend right, "\zeta_2"'] & (H, V, F, BF) \arrow[rd, "\zeta"] & \\
 & & (H'', V'', F'', BF'')
\end{tikzcd}$$

We obtain ζ as an $\underline{MSA}$-signature morphism $(H \cup V, F) \to (H'' \cup V'', F'')$ from the universal property of pushout squares in the category of the $\underline{MSA}$-signatures. In order to prove that it is a quasi-morphism of $\underline{HA}$-signatures, let us first notice that

$$\zeta(V) = \zeta(\theta_1(V_1) \cup \theta_2(V_2)) = \zeta(\theta_1(V_1)) \cup \zeta(\theta_2(V_1)) = \zeta_1(V_1) \cup \zeta_2(V_2).$$

Since $\zeta_k(V_k) \subseteq V''$, for $k \in \{1, 2\}$ it follows that $\zeta(V) \subseteq V''$. Similarly, we obtain $\zeta(H) \subseteq H''$.

Now let $\sigma \in BF_{w \to s}$. Then there exists $k \in \{1, 2\}$, $w_k \in (H_k \cup V_k)^*$, $s_k \in H_k \cup V_k$ and $\sigma_k \in BF_{w_k \to s_k}$ such that $\sigma = \theta_k(\sigma_k)$. Hence, $\zeta(\sigma) = \zeta(\theta_k(\sigma_k)) = \zeta_k(\sigma_k)$. Given that ζ_k is a quasi-morphism of $\underline{HA}$-signatures, it follows that $\zeta_k(\sigma_k) \in BF''_{\zeta_k(w_k) \to \zeta_k(s_k)}$, which means that $\zeta(\sigma) \in BF''_{\zeta(w) \to \zeta(s)}$. □

Proposition 3.2. *Under the notations used in the proof of Prop. 3.1, if φ_2 is a morphism of $\underline{HA}$-signatures then θ_1 is a morphism of $\underline{HA}$-signatures too.*

Proof. Assume that $\sigma \in BF_{w \to s}$ satisfies $w \cap \theta_1(H_1) \neq \emptyset$. By the definition of BF, the interesting case is that in which $\sigma = \theta_2(\sigma_2)$, with σ_2 in BF_2. Let us thus consider $h \in w \cap H$ and $h_1 \in H_1$ such that $h = \theta_1(h_1)$.

Since $\sigma = \theta_2(\sigma_2)$, there exists a hidden sort h_2 in the arity w_2 of σ_2 such that $h = \theta_2(h_2)$. Moreover, since $h = \theta_1(h_1) = \theta_2(h_2)$, by the pushout construction in the category of sets, it follows that there exists $h_0 \in H_0$ such that $h_1 = \varphi_1(h_0)$ and $h_2 = \varphi_2(h_0)$. Hence

$$w_2 \cap \varphi_2(H_0) \neq \emptyset.$$

Now by the encapsulation hypothesis for φ_2, there exists σ_0 in BF_0 such that $\sigma_2 = \varphi_2(\sigma_0)$, which allows us to define $\sigma_1 = \varphi_1(\sigma_0)$; since σ_0 is in BF_0 it follows that σ_1 is in BF_1. We thus obtain

$$\begin{array}{rlllll}
\theta_1(\sigma_1) = & \theta_1(\varphi_1(\sigma_0)) & \text{(by the definition of } \sigma_1\text{)} & = & \theta_2(\varphi_2(\sigma_0)) & \text{(since } \varphi_1; \theta_1 = \varphi_2; \theta_2\text{)} \\
= & \theta_2(\sigma_2) & \text{(since } \sigma_2 = \varphi_2(\sigma_0)\text{)} & = & \sigma & \text{(since } \sigma = \theta_2(\sigma_2)\text{).}
\end{array}$$

This completes the proof. □

Corollary 3.2. *The forgetful functor from the category of $\underline{HA}$-signature morphisms to the category of quasi-morphisms of $\underline{HA}$-signatures lifts pushouts.*

Proof. The conclusion of this corollary follows from Prop. 3.2 (applied twice) if, in addition, we can show that under the notations used in the proof of Prop. 3.1, the mediating quasi-morphism ζ is a morphism whenever both ζ_1 and ζ_2 are morphisms.

Let us consider $\sigma' \in (BF')_{w' \to s'}$ such that $w' \cap \zeta(H) \neq \emptyset$. Then there exists $h \in H$ such that $\zeta(h) \in w'$. By the construction of pushouts in $\mathbb{S}\mathrm{et}$ there exists $k \in \{1, 2\}$ and $h_k \in H_k$ such that $h = \theta_k(h_k)$. Hence, there exists $k \in \{1, 2\}$ such that $w' \cap \zeta_k(H_k) \neq \emptyset$. Since ζ_k is a morphism, meaning that it satisfies the encapsulation condition, there exists σ_k in BF_k such that $\sigma' = \zeta_k(\sigma_k)$. As a result, we can define $\sigma = \theta_k(\sigma_k)$ such that

$$\zeta(\sigma) = \zeta(\theta_k(\sigma_k)) = \zeta_k(\sigma_k) = \sigma'.$$

Finally, since σ_k is in BF_k, we deduce that σ is a behavioural operation in BF. □

Corollary 3.3. *The forgetful functor from the category of $\underline{HA}$-signature morphisms to* $\mathrm{Sign}^{\underline{MSA}}$ *lifts pushouts.*

3.3. *Pullbacks of quasi-morphisms of signatures*

In specification theory and practice, the role of pullbacks of signature morphisms is, in general, minor compared with that played by pushouts. This situation is related to the fact that while pushouts constitute the main technical device underlying various forms of aggregation of specifications, the role played by pullbacks is confined mostly to the fact that they are used for the general definition of intersections from unions (within a given inclusion system). The intersections of signatures are used mainly in situations that consider some form of sharing (for example in works such as [17, 15] etc.).

In $\underline{HA}$, pullbacks of signature morphisms enjoy lesser properties than those enjoyed by pushouts; however, these are sufficient for a full development of the theory of structuring behavioural specifications.

Proposition 3.3. *The forgetful functor from the category of the quasi-morphisms of $\underline{HA}$-signatures to* $\mathrm{Sign}^{\underline{MSA}}$ *lifts pullbacks.*

Proof. Consider the following cospan of quasi-morphisms of $\underline{HA}$-signatures.

$$\begin{tikzcd}
& (H_1, V_1, F_1, BF_1) \arrow[d, "\theta_1"] \\
(H_2, V_2, F_2, BF_2) \arrow[r, "\theta_2"'] & (H, V, F, BF)
\end{tikzcd}$$

Let (φ_1, φ_2) be the pullback as depicted below of the underlying $\underline{MSA}$-signature morphisms of θ_1 and θ_2 such that $H_0 \cap V_0 = \emptyset$ (this condition can be easily ensured based on the standard pullback construction in $\mathbb{S}\mathrm{et}$ as a subset of a product, by noticing that $H_1 \cap V_1 = H_2 \cap V_2 = \emptyset$ implies $(H_1 \times H_2) \cap (V_1 \times V_2) = \emptyset$).

$$\begin{tikzcd}
(H_0 \cup V_0, F_0) \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & (H_1 \cup V_1, F_1) \arrow[d, "\theta_1"] \\
(H_2 \cup V_2, F_2) \arrow[r, "\theta_2"'] & (H \cup V, F)
\end{tikzcd}$$

We extend φ_1 and φ_2 to quasi-morphisms of signatures by defining for each $w \in (H_0 \cup V_0)^*$ and $s \in H_0 \cup V_0$:

$$(BF_0)_{w \to s} = \varphi_1^{-1}((BF_1)_{\varphi_1(w) \to \varphi_1(s)}) \cap \varphi_2^{-1}((BF_2)_{\varphi_2(w) \to \varphi_2(s)}).$$

The resulting square is a commutative diagram of quasi-morphisms of signatures.

$$\begin{tikzcd}
(H',V',F',BF') \arrow[drr, bend left, "\zeta_1"] \arrow[dr, "\zeta"] \arrow[ddr, bend right, "\zeta_2"'] & & \\
& (H_0,V_0,F_0,BF_0) \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & (H_1,V_1,F_1,BF_1) \arrow[d, "\theta_1"'] \\
& (H_2,V_2,F_2,BF_2) \arrow[r, "\theta_2"] & (H,V,F,BF)
\end{tikzcd}$$

All we need to show is that for any quasi-morphisms $\zeta_k\colon (H',V',F',BF') \to (H_k,V_k,F_k,BF_k)$, where $k \in \{1,2\}$, such that $\zeta_1;\theta_1 = \zeta_2;\theta_2$, there exists a unique $\zeta\colon (H',V',F',BF') \to (H,V,F,BF)$ such that $\zeta;\varphi_k = \zeta_k$. We first obtain ζ as an $\underline{MSA}$-signature morphism from the universal property of pullback squares in the the category of $\underline{MSA}$-signatures. To prove that it is a quasi-morphism of $\underline{HA}$-signatures, note that

$$\begin{array}{lll}
\zeta(V') \subseteq V_0 & \text{if and only if} & \zeta(V') \subseteq \varphi_k^{-1}(V_k),\ \text{for } k \in \{1,2\} \quad (\text{since} V_0 = \varphi_k^{-1}(V_k),\ \text{for } k \in \{1,2\}) \\
& \text{if and only if} & \varphi_k(\zeta(V')) \subseteq V_k,\ \text{for } k \in \{1,2\} \\
& \text{if and only if} & \zeta_k(V') \subseteq V_k,\ \text{for } k \in \{1,2\} \qquad\quad (\text{since } \zeta_k = \zeta;\varphi_k).
\end{array}$$

Since for each $k \in \{1,2\}$, ζ_k is a quasi-morphism, it follows that $\zeta_k(V') \subseteq V_k$, for $k \in \{1,2\}$; hence, $\zeta(V') \subseteq V_0$. In exactly the same way we can also prove that $\zeta(H') \subseteq H_0$.

Now let $\sigma \in BF'_{w' \to s'}$. For each $k \in \{1,2\}$ we have

$$\varphi_k(\zeta(\sigma)) = \zeta_k(\sigma) \in (BF_k)_{\zeta_k(w') \to \zeta_k(s')}.$$

Consequently, $\zeta(\sigma) \in \varphi_1^{-1}((BF_1)_{\varphi_1(\zeta(w')) \to \varphi_1(\zeta(s'))}) \cap \varphi_2^{-1}((BF_2)_{\varphi_2(\zeta(w')) \to \varphi_2(\zeta(s'))}) = (BF_0)_{\zeta(w') \to \zeta(s')}$. □

Proposition 3.4. *Under the notations of the proof of Prop. 3.3, if θ_2 is a morphism of $\underline{HA}$-signatures then φ_1 is a morphism of $\underline{HA}$-signatures as well.*

Proof. Suppose that $\sigma_1\colon w_1 \to s_1$ is a behavioural operation symbol in BF_1 such that $w_1 \cap \varphi_1(H_0) \neq \emptyset$. Then $\theta_1(\sigma_1)\colon \theta_1(w_1) \to \theta_1(s_1)$ is a behavioural operation in BF such that $\theta_1(w_1) \cap (\varphi_1;\theta_1)(H_0) \neq \emptyset$; moreover, since $\varphi_1;\theta_1 = \varphi_2;\theta_2$ and $\varphi_2(H_0) \subseteq H_2$, we have $\theta_1(w_1) \cap \theta_2(H_2) \neq \emptyset$. By the encapsulation condition of θ_2, we deduce that there exists a behavioural operation $\sigma_2\colon w_2 \to s_2$ in BF_2 such that $\theta_2(\sigma_2) = \theta_1(\sigma_1)$. Hence, since (φ_1,φ_2) is a pullback of the underlying $\underline{MSA}$-signature morphisms of θ_1 and θ_2, we know there exists an operation symbol σ_0 in F_0 such that $\varphi_1(\sigma_0) = \sigma_1$ and $\varphi_2(\sigma_0) = \sigma_2$. This allows us to conclude, based on the definition of BF_0, that σ_0 is a behavioural operation in BF_0 (such that $\varphi_1(\sigma_0) = \sigma_1$). □

At this point the common path walked in establishing pushouts and pullbacks of $\underline{HA}$-signature morphisms splits. Although by Prop. 3.3 and 3.4 we know that for every cospan of $\underline{HA}$-signature morphisms (θ_1,θ_2), its pullback (φ_1,φ_2) in the category of quasi-morphisms gives a square of $\underline{HA}$-signature morphisms, this does not mean we have a pullback in the category of $\underline{HA}$-signature morphisms. The following is a very simple counterexample. Let us instantiate the signature morphisms of Prop. 3.3 and 3.4 such that each of the considered signatures defines only one sort, which is hidden; φ_1 is the identity of the signature that has only two unary operation symbols σ and σ', declared as behavioural; ζ_2 and θ_2 are the identities on the signature containing only σ, also considered behavioural, and ζ_1, ζ are inclusions. Then the quasi-morphism ζ fails to satisfy the encapsulation condition, and thus we do not have a pullback in the category of $\underline{HA}$-signature morphisms, but only one in the category of quasi-morphisms. One may still think that there is room for a pullback of (θ_1,θ_2) to be obtained in the category of $\underline{HA}$-signature morphisms in a different way than by lifting successively from $\underline{MSA}$ and quasi-morphisms. However, the following result rules out this possibility.

Corollary 3.4. *Every pullback of $\underline{HA}$-signature morphisms can be lifted from the category of quasi-morphisms of $\underline{HA}$-signatures.*

Proof. We show that every pullback cone of $\underline{HA}$-signature morphisms, i.e. defined in $\mathrm{Sign}^{\underline{HA}}$, is isomorphic to a pullback cone defined in the category of quasi-morphisms of $\underline{HA}$-signatures. For this, let (θ_1', θ_2') be a pullback of $\underline{HA}$-signature morphisms θ_1 and θ_2 as depicted in the diagram below, and let (φ_1, φ_2) be a pullback of θ_1 and θ_2 in the category of quasi-morphisms of $\underline{HA}$-signatures.

$$\begin{tikzcd}
(H_0, V_0, F_0, BF_0) \arrow[rrd, bend left, "\varphi_1"] \arrow[rdd, bend right, "\varphi_2"'] \arrow[rd, bend left, "\zeta"] & & \\
& (H', V', F', BF') \arrow[r, "\theta_1'"] \arrow[d, "\theta_2'"'] \arrow[lu, dashed, bend left, "\xi"] & (H_1, V_1, F_1, BF_1) \arrow[d, "\theta_1"] \\
& (H_2, V_2, F_2, BF_2) \arrow[r, "\theta_2"'] & (H, V, F, BF)
\end{tikzcd}$$

By Prop. 3.4 (applied twice), we know that the quasi-morphisms φ_1 and φ_2 satisfy the encapsulation condition. This allows us to infer, based on the universality property of the pullback (θ_1', θ_2'), that there exists a unique signature morphism ζ such that $\zeta; \theta_1' = \varphi_1$ and $\zeta; \theta_2' = \varphi_2$. By a similar argument, we also obtain a quasi-morphism ξ that satisfies $\xi; \varphi_1 = \theta_1'$ and $\xi; \varphi_2 = \theta_2'$. Since $(\zeta; \xi); \varphi_1 = \zeta; \theta_1' = \varphi_1$ and $(\zeta; \xi); \varphi_2 = \zeta; \theta_2' = \varphi_2$, we further deduce by the universality property of (φ_1, φ_2) that $\zeta; \xi = 1_{(H_0, V_0, F_0, BF_0)}$. It follows that every behavioural operation $\sigma_0 \in BF_0$ is the image under ξ of the behavioural operation $\zeta(\sigma_0) \in BF'$, which means that ξ also satisfies the encapsulation condition. Therefore, by the universality property of (θ_1', θ_2'), we have $\xi; \zeta = 1_{(H', V', F', BF')}$, thus concluding the proof. □

Even though the question of establishing precisely the class of cospans (θ_1, θ_2) that admit pullbacks in $\mathrm{Sign}^{\underline{HA}}$ remains open, it should be noted that the answer to this question is irrelevant for the purposes of our work, as the pullback property provided by Prop. 3.3 is sufficient for the subsequent developments on the structuring of behavioural specifications.

3.4. Model amalgamation

Corollary 3.5. *Each pushout square of $\underline{HA}$-signature morphisms is an amalgamation square.*

Proof. By Cor. 3.3, for any pushout square of $\underline{HA}$-signature morphisms, its underlying square of $\underline{MSA}$-signature morphisms is also a pushout square, which by the amalgamation property in $\underline{MSA}$ (see Prop. 2.1) is an amalgamation square. The conclusion follows by the fact that $\underline{HA}$-models are just $\underline{MSA}$-models of their corresponding underlying $\underline{MSA}$-signatures. □

3.5. Inclusion systems for $\underline{HA}$-signatures

Proposition 3.5. *The category of quasi-morphisms of $\underline{HA}$-signatures admits an epic inclusion system that inherits the strong inclusion system of $\underline{MSA}$-signatures as follows.*

- *The abstract inclusions $(H, V, F, BF) \subseteq (H', V', F', BF')$ are defined by*
  - *$H \subseteq H'$, $V \subseteq V'$, and*
  - *for each $w \in (H \cup V)^*$ and $s \in H \cup V$, $F_{w \to s} \subseteq F'_{w \to s}$.*
- *The abstract surjections $\varphi\colon (H, V, F, BF) \to (H', V', F', BF')$ are defined by*

- *$\varphi(H) = H'$, $\varphi(V) = V'$, and*
- *for each $w' \in (H' \cup V')^*$ and $s' \in H' \cup V'$,*

  $F'_{w' \to s'} = \bigcup\{\varphi(F_{w \to s}) \mid \varphi(w) = w', \varphi(s) = s'\}$ *and* $BF'_{w' \to s'} = \bigcup\{\varphi(BF_{w \to s}) \mid \varphi(w) = w', \varphi(s) = s'\}$.

Proof. In this case, apart from the factoring axiom, all the other axioms of inclusion systems are rather straightforward to check. Therefore let us focus on factoring. Let $\varphi\colon (H, V, F, BF) \to (H', V', F', BF')$ be any quasi-morphism of $\underline{HA}$-signatures. We factor the underlying morphism of $\underline{MSA}$-signatures $\varphi\colon (H \cup V, F) \to (H' \cup V', F')$ through the strong inclusion system of the category of $\underline{MSA}$-signatures:

$$\begin{tikzcd}
(H \cup V, F) \arrow[r, "e_\varphi"'] \arrow[rr, bend left, "\varphi"] & (S'', F'') \arrow[r, "i_\varphi"'] & (H' \cup V', F')
\end{tikzcd}$$

We can thus define

- $V'' = \varphi(V)$ and $H'' = \varphi(H)$; then $H'' \cap V'' = \emptyset$ since $\varphi(H) \subseteq H'$, $\varphi(V) \subseteq V'$, and $H' \cap V' = \emptyset$;
- for each $w'' \in (H'' \cup V'')^*$ and $s'' \in H'' \cup V''$, $BF''_{w'' \to s''} = \bigcup\{\varphi(BF_{w \to s}) \mid \varphi(w) = w'', \varphi(s) = s''\}$.

This yields a factoring in which e_φ is an abstract surjection of $\underline{HA}$-signatures and i_φ an abstract inclusion of $\underline{HA}$-signatures:

$$\begin{tikzcd}
(H, V, F, BF) \arrow[r, "e_\varphi"'] \arrow[rr, bend left, "\varphi"] & (H'', V'', F'', BF'') \arrow[r, "i_\varphi"'] & (H', V', F', BF')
\end{tikzcd}$$

The uniqueness of this factoring is explained by the uniqueness of the two factorings through the strong inclusion system of $\mathrm{Sign}^{\underline{MSA}}$:

$$\begin{tikzcd}
(H \cup V, F) \arrow[r, "e_\varphi"'] \arrow[rr, bend left, "\varphi"] & (S'', F'') \arrow[r, "i_\varphi"'] & (H' \cup V', F') & (H \cup V, BF) \arrow[r, "e_{\varphi_{BF}}"'] \arrow[rr, bend left, "\varphi_{BF}"] & (S'', BF'') \arrow[r, "i_{\varphi_{BF}}"'] & (H' \cup V', BF')
\end{tikzcd}$$

where φ_{BF} is the restriction of φ to the subsignature $(H \cup V, BF)$, and by the fact that H'' and V'' are uniquely determined as $H'' = \varphi(H)$ and $V'' = \varphi(V)$. □

Straight from the definitions of inclusions and quasi-morphisms we obtain the following result.

Fact 3.4. *For any inclusion $(H, V, F, BF) \subseteq (H', V', F', BF')$, and for all $w \in (H \cup V)^*$ and $s \in (H \cup V)$, we have $BF_{w \to s} \subseteq BF'_{w \to s}$.*

Proposition 3.6. *Let φ be any morphism of $\underline{HA}$-signatures, and $\varphi = e_\varphi ; i_\varphi$ its factorisation through the inclusion system of the category of quasi-morphisms defined in Prop. 3.5. Then both the abstract surjection e_φ and the abstract inclusion i_φ are morphisms of $\underline{HA}$-signatures.*

Proof. First let us consider $\sigma' \in BF'_{w'}$ such that $w' \cap H'' \neq \emptyset$. Since $H'' = \varphi(H)$ we have $w' \cap \varphi(H) \neq \emptyset$. Furthermore, since φ enjoys the encapsulation condition of Dfn. 3.8 there exists σ in BF such that $\sigma' = \varphi(\sigma)$. By the definition of BF'', $\varphi(\sigma)$ is in BF'', which means σ' is in BF''. This shows that i_φ enjoys the encapsulation condition too.

Now let us consider $\sigma'' \in BF''_{w'' \to s''}$ such that $w'' \cap e_\varphi(H) \neq \emptyset$. This means that $\sigma'' \in BF'_{w'' \to s''}$ and $w'' \cap \varphi(H) \neq \emptyset$. By the encapsulation hypothesis for φ, there exists σ in BF such that $\sigma'' = \varphi(\sigma)$. But this implies that $\sigma'' = e_\varphi(\sigma)$, which shows that e_φ also enjoys the encapsulation condition. □

Corollary 3.6. *The category of $\underline{HA}$-signature morphisms admits an inclusion system that inherits the strong inclusion system of the category of $\underline{MSA}$-signatures.*

3.6. Unions and intersections of $\underline{HA}$-signatures

Notation 3.2. The inclusions, unions, and intersections, are denoted by $\subseteq$, $\cup$ and $\cap$, respectively, in the inclusion system of the category of quasi-morphisms of $\underline{HA}$-signatures and by $\sqsubseteq$, $\sqcup$ and $\sqcap$, respectively, in the inclusion system of the category of $\underline{HA}$-signature morphisms.

Lemma 3.1. *If $\Sigma \subseteq \Sigma' \subseteq \Sigma''$ and $\Sigma \sqsubseteq \Sigma''$ then $\Sigma \sqsubseteq \Sigma'$.*

Proof. Let $\Sigma = (H, V, F, BF)$, $\Sigma' = (H', V', F', BF')$, and $\Sigma'' = (H'', V'', F'', BF'')$. Let $h \in H$. Then $\Sigma \subseteq \Sigma' \subseteq \Sigma''$ implies $BF_{[h]} \subseteq BF'_{[h]} \subseteq BF''_{[h]}$, and by Fact 3.2, $\Sigma \sqsubseteq \Sigma''$ implies that $BF_{[h]} = BF''_{[h]}$; hence, $BF_{[h]} = BF'_{[h]}$. It follows by Fact 3.2 that $\Sigma \sqsubseteq \Sigma'$. □

The following abbreviation will be used quite often in the remaining part of our paper.

Notation 3.3. For any $\underline{MSA}$-signature (S, F) and any $S_0 \subseteq S$, we denote by $(S_0; F)$ the largest subsignature of (S, F) having S_0 as the set of sorts, i.e. (S_0, F_0) such that $(F_0)_{w\to s} = F_{w\to s}$ for $w \in S_0^*$ and $s \in S_0$.

For any $\underline{HA}$-signature (H, V, F, BF) and sets $H_0 \subseteq H$ and $V_0 \subseteq S$, we denote by $(H_0, V_0; F, BF)$ the signature (H_0, V_0, F_0, BF_0) where $(H_0 \cup V_0, F_0) = (H_0 \cup V_0; F)$ and $(H_0 \cup V_0, BF_0) = (H_0 \cup V_0; BF)$.

Proposition 3.7. *Let $(\Sigma_i)_{i\in I}$ be a non-empty family of $\underline{HA}$-signatures. Then*

1. *The intersection $\bigcap_{i\in I} \Sigma_i$ exists.*
2. *The intersection $\bigsqcap_{i\in I} \Sigma_i$ exists. Moreover, $\bigsqcap_{i\in I} \Sigma_i = \bigcap_{n\in\omega} \Sigma^{(n)}$, where*
   - *$\Sigma^{(0)} = \bigcap_{i\in I} \Sigma_i$, and*
   - *for each $n > 0$, if $\Sigma^{(n)} = \left(H^{(n)}, V, F^{(n)}, BF^{(n)}\right)$ then*
   $$\Sigma^{(n+1)} = \left(H^{(n+1)}, V, F^{(n+1)}, BF^{(n+1)}\right) = \left(H^{(n+1)}, V; F^{(n)}, BF^{(n)}\right)$$
   where $H^{(n+1)} = H^{(n)} \setminus \left\{h \in H^{(n)} \mid BF^{(n)}_{[h]} \neq (BF_i)_{[h]} \text{ for some } i \in I\right\}$.
3. *$\bigsqcap_{i\in I} \Sigma_i \sqsubseteq \bigcap_{i\in I} \Sigma_i$.*

Proof.

1. For each $i \in I$ we let $\Sigma_i = (H_i, V_i, F_i, BF_i)$. Then the intersection $\bigcap_{i\in I} \Sigma_i$ is (H, V, F, BF) where
   - $H = \bigcap_{i\in I} H_i$ and $V = \bigcap_{i\in I} V_i$, and
   - for each $w \in (H \cup V)^*$ and $s \in H \cup V$, $F_{w\to s} = \bigcap_{i\in I}(F_i)_{w\to s}$ and $BF_{w\to s} = \bigcap_{i\in I}(BF_i)_{w\to s}$.

2. Let $\bigcap_{n\in\omega} \Sigma^{(n)} = (H', V, F', BF')$. By Fact 3.2, to show that $\bigcap_{n\in\omega} \Sigma^{(n)} \sqsubseteq \Sigma_i$ for each $i \in I$, it suffices to show that for each $h \in H'$ and each $i \in I$ we have $BF'_{[h]} = (BF_i)_{[h]}$. If $BF'_{[h]} \neq (BF_i)_{[h]}$ then there exists $n \in \omega$ such that $BF^{(n)}_{[h]} \neq (BF_i)_{[h]}$. Since $BF^{(n)}_{[h]} \neq (BF_i)_{[h]}$ implies $h \notin H^{(n+1)}$, it follows that $h \notin H'$, which is a contradiction.

Now let $\Sigma'' = (H'', V'', F'', BF'')$ such that for each $i \in I$, $\Sigma'' \sqsubseteq \Sigma_i$. To show that $\Sigma'' \sqsubseteq \bigcap_{k\in\omega} \Sigma^{(n)}$, by Lemma 3.1, it suffices to show that $\Sigma'' \subseteq \bigcap_{n\in\omega} \Sigma^{(n)}$. This follows if we showed (by induction on $n \in \omega$) that $\Sigma'' \subseteq \Sigma^{(n)}$. For $n = 0$ the property is obvious because $\Sigma'' \sqsubseteq \Sigma_i$ implies $\Sigma'' \subseteq \Sigma_i$. Now we assume the property holds for n and show it for $n + 1$. By the construction of $\Sigma^{(n)}$, it is enough to show that $H'' \subseteq H^{(n+1)}$. If there exists $h \in H'' \setminus H^{(n+1)}$, since by the induction hypothesis $H'' \subseteq H^{(n)}$, it follows that $h \in H^{(n)} \setminus H^{(n+1)}$; hence, there exists $i \in I$ such that $BF^{(n)}_{[h]} \neq (BF_i)_{[h]}$. Then, by the induction hypothesis, $\Sigma'' \subseteq \Sigma^{(n)}$. This means that $BF''_{[h]} \subseteq BF^{(n)}_{[h]}$, which implies $BF''_{[h]} \neq (BF_i)_{[h]}$. Based on Fact 3.2 we obtain a second contradiction, this time of the hypothesis that $\Sigma'' \sqsubseteq \Sigma_i$.

$\boxed{3.}$ The fact that $\bigsqcap_{i \in I} \Sigma_i \sqsubseteq \bigcap_{i \in I} \Sigma_i$ follows immediately from Lemma 3.1. □

Proposition 3.8. *For any two $\underline{HA}$-signatures $\Sigma_1 = (H_1, V_1, F_1, BF_1)$ and $\Sigma_2 = (H_2, V_2, F_2, BF_2)$ the following statements are equivalent:*

1. $H_1 \cap V_2 = H_2 \cap V_1 = \emptyset$.
2. *The union* $\Sigma_1 \cup \Sigma_2$ *exists and* $\mathrm{msa}(\Sigma_1 \cup \Sigma_2) = \mathrm{msa}(\Sigma_1) \cup \mathrm{msa}(\Sigma_2)$.
3. *The union* $\Sigma_1 \cup \Sigma_2$ *exists.*
4. *There exists* Σ' *such that* $\Sigma_1, \Sigma_2 \subseteq \Sigma'$.
5. $\mathrm{msa}(\Sigma_1 \cap \Sigma_2) = \mathrm{msa}(\Sigma_1) \cap \mathrm{msa}(\Sigma_2)$ *(note that, according to Prop. 3.7, the intersection* $\Sigma_1 \cap \Sigma_2$ *exists).*

Moreover, in all these situations the corresponding intersection-union square of $\underline{HA}$-signatures is a pushout square of quasi-morphisms of $\underline{HA}$-signatures.

Proof.

$\boxed{1 \Rightarrow 2}$: We define the following unions of $\underline{MSA}$-signatures: $(S, F) = (H_1 \cup V_1, F_1) \cup (H_2 \cup V_2, F_2)$ and $(S, BF) = (H_1 \cup V_1, BF_1) \cup (H_2 \cup V_2, BF_2)$. We let $H = H_1 \cup H_2$ and $V = V_1 \cup V_2$. Note that $S = H \cup V$; also $H \cap V = \emptyset$ because of the assumption $H_1 \cap V_2 = H_2 \cap V_1 = \emptyset$ and because $H_1 \cap V_1 = H_2 \cap V_2 = \emptyset$.

Let us show that $\Sigma_1 \cup \Sigma_2 = (H, V, F, BF)$. For any behavioural signature (H', V', F', BF') that satisfies $(H_k, V_k, F_k, BF_k) \subseteq (H', V', F', BF')$, for $k \in \{1, 2\}$, we have $(H_k \cup V_k, F_k) \subseteq (H' \cup V', F')$ and $(H_k \cup V_k, BF_k) \subseteq (H' \cup V', BF')$. Therefore, $(S, F) \subseteq (H' \cup V', F')$ and $(S, BF) \subseteq (H' \cup V', BF')$, which implies $(H, V, F, BF) \subseteq (H', V', F', BF')$.

$\boxed{2 \Rightarrow 3}$ and $\boxed{3 \Rightarrow 4}$ are trivial implications.

$\boxed{4 \Rightarrow 1}$: Let H' be the set of hidden sorts and V' be the set of visible sorts of Σ'. The conditions $H_1, H_2 \subseteq H'$ and $V_1, V_2 \subseteq V'$ together with the observation that $H' \cap V' = \emptyset$ imply $H_1 \cap V_2 = H_2 \cap V_1 = \emptyset$.

$\boxed{2 \Rightarrow 5}$: According to a general result from [21] (see also [13]), by relying upon the existence of pullbacks for quasi-morphisms of signatures of Prop. 3.3, it follows that $\Sigma_1 \cap \Sigma_2$ is obtained as the following pullback:

$$\begin{tikzcd}
\Sigma_1 \cap \Sigma_2 \arrow[r, "\subseteq"] \arrow[d, "\subseteq"'] & \Sigma_1 \arrow[d, "\subseteq"] \\
\Sigma_2 \arrow[r, "\subseteq"'] & \Sigma_1 \cup \Sigma_2
\end{tikzcd}$$

Since by hypothesis the union $\Sigma_1 \cup \Sigma_2$ inherits the union of the underlying $\underline{MSA}$-signatures, and since according to Prop. 3.3 pullbacks of quasi-morphisms are also inherited from $\underline{MSA}$, it follows that the intersection $\Sigma_1 \cap \Sigma_2$ inherits the intersection of the underlying $\underline{MSA}$-signatures.

$\boxed{5 \Rightarrow 1}$: If $\Sigma_1 \cap \Sigma_2$ inherits the intersection $(H_1 \cup V_1, F_1) \cap (H_2 \cup V_2, F_2)$ then we have

$$(H_1 \cap H_2) \cup (V_1 \cap V_2) = (H_1 \cup V_1) \cap (H_2 \cup V_2).$$

This means that

(1) $(H_1 \cap H_2) \cup (V_1 \cap V_2) = (H_1 \cap H_2) \cup (H_1 \cap V_2) \cup (H_2 \cap V_1) \cup (V_1 \cap V_2)$.

Because for each $k \in \{1, 2\}$ we have $H_k \cap V_k = \emptyset$, it follows that

(2) $((H_1 \cap V_2) \cup (H_2 \cap V_1)) \cap ((H_1 \cap H_2) \cup (V_1 \cap V_2)) = \emptyset$.

This allows us to deduce from (1) and (2) that $(H_1 \cap V_2) \cup (H_2 \cap V_1) = \emptyset$.

For the second part of the result, since the intersection-union squares in $\underline{MSA}$ are pushout squares, and since the signatures of $\underline{HA}$ inherit the intersection and the union of their underlying $\underline{MSA}$-signatures, it follows by Prop. 3.1 that the intersection-union squares of $\underline{HA}$-signatures are pushout squares too. □

Proposition 3.9. *For any two $\underline{HA}$-signatures the following are equivalent:*

1. *The union $\Sigma_1 \sqcup \Sigma_2$ exists and $\Sigma_1 \sqcup \Sigma_2 = \Sigma_1 \cup \Sigma_2$.*
2. *The union $\Sigma_1 \sqcup \Sigma_2$ exists.*
3. *There exists Σ' such that $\Sigma_1, \Sigma_2 \sqsubseteq \Sigma'$.*
4. *The union $\Sigma_1 \cup \Sigma_2$ exists and $\Sigma_1, \Sigma_2 \sqsubseteq \Sigma_1 \cup \Sigma_2$.*
5. $\mathrm{msa}(\Sigma_1 \cap \Sigma_2) = \mathrm{msa}(\Sigma_1) \cap \mathrm{msa}(\Sigma_2)$ *and* $\Sigma_1 \cap \Sigma_2 \sqsubseteq \Sigma_1, \Sigma_2$.
6. $\Sigma_1 \sqcap \Sigma_2 = \Sigma_1 \cap \Sigma_2$ *and* $\mathrm{msa}(\Sigma_1 \cap \Sigma_2) = \mathrm{msa}(\Sigma_1) \cap \mathrm{msa}(\Sigma_2)$.

Moreover, in these situations the corresponding intersection-union square of $\underline{HA}$-signatures is a pushout square of morphisms of $\underline{HA}$-signatures.

Proof.
$\boxed{1 \Rightarrow 2}$ and $\boxed{2 \Rightarrow 3}$ are trivial implications.

$\boxed{3 \Rightarrow 4}$: From $\Sigma_1, \Sigma_2 \sqsubseteq \Sigma'$ it follows that $\Sigma_1, \Sigma_2 \subseteq \Sigma'$, and by Prop. 3.8 that the union $\Sigma_1 \cup \Sigma_2$ exists. Moreover, $\Sigma_1 \cup \Sigma_2 \subseteq \Sigma'$. Since $\Sigma_1, \Sigma_2 \sqsubseteq \Sigma'$, by Lemma 3.1, it follows that $\Sigma_1, \Sigma_2 \sqsubseteq \Sigma_1 \cup \Sigma_2$.

$\boxed{4 \Rightarrow 5}$: The fact that $\Sigma_1 \cap \Sigma_2$ inherits the corresponding intersection of the underlying $\underline{MSA}$-signatures of Σ_1 and Σ_2 follows from the observation that the union $\Sigma_1 \cup \Sigma_2$ exists via Prop. 3.8. Now let $\Sigma_1 = (H_1, V_1, F_1, BF_1)$, $\Sigma_2 = (H_2, V_2, F_2, BF_2)$, $\Sigma_1 \cap \Sigma_2 = (H, V, F, BF)$ and $\Sigma_1 \cup \Sigma_2 = (H', V', F', BF')$. For each $h \in H$ we know that $BF_{[h]} = (BF_1)_{[h]} \cap (BF_2)_{[h]}$ and that $BF'_{[h]} = (BF_1)_{[h]} \cup (BF_2)_{[h]}$. Then by Fact 3.2 $\Sigma_1, \Sigma_2 \sqsubseteq \Sigma_1 \cup \Sigma_2$ implies $(BF_1)_{[h]} = (BF_2)_{[h]}$, which further implies $BF_{[h]} = (BF_1)_{[h]} = (BF_2)_{[h]}$; hence, by Fact 3.2, it follows that $\Sigma_1 \cap \Sigma_2 \sqsubseteq \Sigma_1, \Sigma_2$.

$\boxed{5 \Rightarrow 6}$: The condition $\Sigma_1 \cap \Sigma_2 \sqsubseteq \Sigma_1, \Sigma_2$ implies $\Sigma_1 \cap \Sigma_2 \sqsubseteq \Sigma_1 \sqcap \Sigma_2$. Since in general (see Prop. 3.7) $\Sigma_1 \sqcap \Sigma_2 \sqsubseteq \Sigma_1 \cap \Sigma_2$, it follows that $\Sigma_1 \sqcap \Sigma_2 = \Sigma_1 \cap \Sigma_2$.

$\boxed{6 \Rightarrow 1}$: According to Prop. 3.8, the union $\Sigma_1 \cup \Sigma_2$ exists and its corresponding intersection-union square is a pushout square of quasi-morphisms. Then by Prop. 3.2, it follows that $\Sigma_1, \Sigma_2 \sqsubseteq \Sigma_1 \cup \Sigma_2$, which implies, by Cor. 3.2, that $\Sigma_1 \cup \Sigma_2 = \Sigma_1 \sqcup \Sigma_2$. □

The following property is a rather straightforward consequence of Prop. 3.8 and Prop. 3.9.

Corollary 3.7. *For any $\underline{HA}$-signatures (H_1, V_1, F_1, BF_1) and (H_2, V_2, F_2, BF_2) the union $(H_1, V_1, F_1, BF_1) \sqcup (H_2, V_2, F_2, BF_2)$ exists if and only if $H_1 \cap V_2 = H_2 \cap V_1 = \emptyset$ and $(BF_1)_{[h]} = (BF_2)_{[h]}$ for all $h \in H_1 \cap H_2$.*

3.7. *The algebra of $\underline{HA}$-signatures*

The algebraic properties of the union $\sqcup$ and intersection $\sqcap$ of $\underline{HA}$-signatures play an important role in establishing algebraic properties of behavioural specification modules. The union $\sqcup$ on the class of $\underline{HA}$-signatures is a partial rather than total operation, which gives rise to a partial algebra of signatures in the sense of [8]. In what follows we make use of two types of equalities specific to partial algebras: $t \overset{e}{=} t'$ for the *existence equality*, i.e. both t and t' are defined and their values are equal, and $t = t'$ for the *strong equality*, i.e. either both t and t' are undefined or $t \overset{e}{=} t'$.

Fact 3.5. *For any $\underline{HA}$-signatures Σ, Σ_1, Σ_2,*

(3) $\quad \Sigma \sqcup \Sigma \overset{e}{=} \Sigma$, *and*

(4) $\quad \Sigma_1 \sqcup \Sigma_2 = \Sigma_2 \sqcup \Sigma_1$.

Proposition 3.10. *For any $\underline{HA}$-signatures Σ_1, Σ_2, Σ_3,*

(5) $\quad \Sigma_1 \sqcup (\Sigma_2 \sqcup \Sigma_3) = (\Sigma_1 \sqcup \Sigma_2) \sqcup \Sigma_3$.

Proof. By the general associativity property of the unions in inclusion systems [21] it is enough to show that the left-hand and the right-hand side of the associativity equation exist simultaneously. Let us assume that the union $\Sigma_1 \sqcup (\Sigma_2 \sqcup \Sigma_3)$ exists. This implies that $\Sigma_2 \sqcup \Sigma_3$ exists and also that there exists Σ' such that $\Sigma_1, \Sigma_2, \Sigma_3 \sqsubseteq \Sigma'$ (just by taking $\Sigma_1 \sqcup (\Sigma_2 \sqcup \Sigma_3)$ in the role of Σ'). Now from Prop. 3.9 it follows both that $\Sigma_1 \sqcup \Sigma_2$ exists and, since $\Sigma_1 \sqcup \Sigma_2 \sqsubseteq \Sigma'$ and $\Sigma_3 \sqsubseteq \Sigma'$, that $(\Sigma_1 \sqcup \Sigma_2) \sqcup \Sigma_3$ exists.

In the same way we may show the opposite implication; for this reason we omit its proof here. □

Proposition 3.11. *For any $\underline{HA}$-signatures Σ, Σ_1, Σ_2,*

$$\Sigma \sqcup \Sigma_1 \sqcup \Sigma_2 \overset{e}{=} \Sigma \sqcup \Sigma_1 \sqcup \Sigma_2 \quad \textit{implies} \quad \Sigma \sqcap (\Sigma_1 \sqcup \Sigma_2) \overset{e}{=} (\Sigma \sqcap \Sigma_1) \sqcup (\Sigma \sqcap \Sigma_2).$$

Proof. On the one hand, that $\Sigma_1 \sqcup \Sigma_2$ exists follows immediately from the condition of the distributivity rule. Then Prop. 3.7 implies the existence of $\Sigma \sqcap (\Sigma_1 \sqcup \Sigma_2)$; moreover, by Prop. 3.9, from the existence of $\Sigma \sqcup (\Sigma_1 \sqcup \Sigma_2)$ it follows that $\Sigma \sqcap (\Sigma_1 \sqcup \Sigma_2)$ inherits the intersection of the underlying $\underline{MSA}$-signatures.

On the other hand, the existence of $\Sigma \sqcup \Sigma_1 \sqcup \Sigma_2$ implies the existence of both $\Sigma \sqcup \Sigma_1$ and $\Sigma \sqcup \Sigma_2$, which by Prop. 3.9 imply that both $\Sigma \sqcap \Sigma_1$ and $\Sigma \sqcap \Sigma_2$ inherit the corresponding intersections of the underlying $\underline{MSA}$-signatures of Σ, Σ_1 and Σ_2. Moreover, $\Sigma \sqcap \Sigma_1, \Sigma \sqcap \Sigma_2 \subseteq \Sigma \sqcup \Sigma_1 \sqcup \Sigma_2$, which by Prop. 3.9 guarantees that $(\Sigma \sqcap \Sigma_1) \sqcup (\Sigma \sqcap \Sigma_2)$ exists.

Since all intersections and unions considered here inherit the corresponding intersections and unions of their underlying $\underline{MSA}$-signatures, and because the distributivity of intersection over union holds for $\underline{MSA}$-signatures [21], we conclude that $\Sigma \sqcap (\Sigma_1 \sqcup \Sigma_2) = (\Sigma \sqcap \Sigma_1) \sqcup (\Sigma \sqcap \Sigma_2)$. □

Remark 3.1. In general, the distributivity rule of Prop. 3.11 does not hold unconditionally, not even in a weaker form such as

$$\Sigma \sqcap (\Sigma_1 \sqcup \Sigma_2) = (\Sigma \sqcap \Sigma_1) \sqcup (\Sigma \sqcap \Sigma_2).$$

The following is a simple counterexample. Let Σ consist of a visible sort v, Σ_1 of a visible sort s and Σ_2 of a hidden sort s. Then $\Sigma_1 \sqcup \Sigma_2$ does not exist, hence the left-hand side of the rule does not exist. On the other hand the right-hand side of the equation does exist and it is the empty signature.

4. Abstract structured behavioural specifications

We aim to study the structuring of behavioural specifications independently of any particular choice of structuring operators. In order to achieve this, we employ the recent theory of abstract structured specifications developed in [15]. The structure of this section is as follows:

1. We recall the main concepts from the theory of abstract structured specifications from [15].
2. We develop concepts of concrete structuring operators within this abstract context.
3. We develop a module algebra for abstract structured specifications that is applicable to the present behavioural setting and constitutes a generalisation of other module-algebra rules from the literature.

4.1. Basic definitions

Although an important motivation for institution theory [23] (see also [16]) had been a logic-independent approach to modularization, the pivotal paper on institution-independent structuring of specifications is [36]. The paper [21] is another influential work in this direction. All these works provide an abstract treatment of the underlying logic as an institution, but consider a fixed set of concrete structuring operators. However, works such as [17] have shown the need to consider new structuring operators, for example for treating non-protecting importation modes. Moreover, in the practice of specification languages one need not distinguish between the order of imports, an aspect that requires the consideration of structured specifications modulo some algebraic rules such as commutativity and associativity of module sums. In order to overcome the limitations of a fixed standard set of building operators, of the free construction of specifications, but also to achieve unification between Goguen-Burstall [23, 21] and Sannella-Tarlecki [36, 37] approaches to the semantics of structured specifications, the recent paper [15] introduces a second level of institution-independence by treating the class of structured specifications together with their model theory as an abstract institution. Besides these, the abstraction involved in the approach proposed in [15] means a new level of conceptual simplicity.

The relationship between the level of the structured specifications and the level of the underlying logic is axiomatized by a special kind of institution morphism. The following definition recalls from [15] the main concept of this theory.

Definition 4.1 (Structured institution)**.** Given two institutions $\mathcal{I}$ and $\mathcal{I}'$, with $\mathcal{I} = (\mathrm{Sign}, \mathrm{Sen}, \mathrm{Mod}, \models)$ and $\mathcal{I}' = (\mathrm{Sign}', \mathrm{Sen}', \mathrm{Mod}', \models')$, we say that $\mathcal{I}'$ is $(sig, \mathcal{I})$*-structured*[2] when

- $sig\colon \mathrm{Sign}' \to \mathrm{Sign}$ is a functor,
- for each $\mathcal{I}'$-signature Σ' we have $\mathrm{Sen}(sig(\Sigma')) = \mathrm{Sen}'(\Sigma')$, and for each $\mathcal{I}'$-signature morphism φ we have $\mathrm{Sen}(sig(\varphi)) = \mathrm{Sen}'(\varphi)$,
- for each $\mathcal{I}'$-signature Σ' we have that $\mathrm{Mod}'(\Sigma')$ is a full subcategory of $\mathrm{Mod}(sig(\Sigma'))$ such that for each $\mathcal{I}'$-signature morphism $\varphi\colon \Sigma'_1 \to \Sigma'_2$ the diagram below commutes,

$$\begin{tikzcd}
\mathrm{Mod}'(\Sigma'_1) \arrow[r, "\subseteq"] & \mathrm{Mod}(sig(\Sigma'_1)) \\
\mathrm{Mod}'(\Sigma'_2) \arrow[r, "\subseteq"'] \arrow[u, "\mathrm{Mod}'(\varphi)"] & \mathrm{Mod}(sig(\Sigma'_2)) \arrow[u, "\mathrm{Mod}(sig(\varphi))"']
\end{tikzcd}$$

and

[2] In [15] this is called "structured over $\mathcal{I}$ through sig".

– for each $\mathcal{I}'$-signature Σ', each Σ'-model M' and each Σ'-sentence ρ we have that

$$M' \models'_{\Sigma'} \rho \quad \text{if and only if} \quad M' \models_{sig(\Sigma')} \rho.$$

In [15] several examples are presented in some detail; here let us just mention them rather briefly.

1. The Sannella-Tarlecki approach [36, 37] is covered by considering the structured specifications formed from finite presentations of $\mathcal{I}$-theories by iteration of operators such as union, translation, derivation and free semantics, as the signatures of $\mathcal{I}'$.
2. In the case of the Goguen-Burstall approach [23, 21], instead of specifications as terms formed by a fixed set of specification building operators like in the previous example, in the role of the $\mathcal{I}'$-signatures one considers the closed theories determined by those.
3. One may also consider quotients of specifications by algebraic rules, such as commutativity and/or associativity of the union.
4. Other formalisms not necessarily rooted within specification theory may also be covered, such as the module systems for model expansion problems [40].

Moreover, in all the situations mentioned above, the particular set of structuring operators involved in defining specifications may be replaced by or extended with other sets of structuring operators.

4.2. Structuring operators in the abstract context

Although one of the main points of the theory of abstract structured specifications is the liberation from concrete structuring operators, in some situations it is useful to talk about structuring operators in an abstract context. This is especially relevant when studying the algebraic rules of module composition. Dfn. 4.2 and 4.3 below introduce a couple of the most important generic structuring operators in the literature within the context of abstract structured specifications. The following analogy with monoids may be quite helpful. The structured specifications in the traditional approach [36] would correspond to free monoids, while the concept of abstract structured specifications endowed with definitions of some (concrete) structuring operators corresponds to the class of *all* monoids.

Definition 4.2 (Unions)**.** An institution $\mathcal{I}'$ that is $(sig, \mathcal{I})$-structured with $\mathcal{I}$ inclusive (where $\sqcup$ is used for unions when they exist) *has unions* when for any $\mathcal{I}'$-signatures Σ'_1 and Σ'_2 such that $sig(\Sigma'_1) \sqcup sig(\Sigma'_2)$ exists there exists as well a designated $\mathcal{I}'$-signature, denoted $\Sigma'_1 \sqcup \Sigma'_2$, such that

– $sig(\Sigma'_1 \sqcup \Sigma'_2) = sig(\Sigma'_1) \sqcup sig(\Sigma'_2)$, and

– $|\mathrm{Mod}'(\Sigma'_1 \sqcup \Sigma'_2)| = \{M' \in |\mathrm{Mod}(sig(\Sigma'_1) \sqcup sig(\Sigma'_2))| \mid M'\upharpoonright_{sig(\Sigma'_k)} \in |\mathrm{Mod}'(\Sigma'_k)|, k \in \{1, 2\}\}$.

Definition 4.3 (Translation and Derivation)**.** For any institution $\mathcal{I}'$ that is $(sig, \mathcal{I})$-structured and any $\mathcal{I}$-signature morphism $\varphi\colon \Sigma \to \Omega$, we say that

– $\mathcal{I}'$ has *φ-translations* when for any $\mathcal{I}'$-signature Σ' such that $sig(\Sigma') = \Sigma$ there exists a designated $\mathcal{I}'$-signature, denoted $\Sigma' \star \varphi$, such that

  - $sig(\Sigma' \star \varphi) = \Omega$, and
  - $|\mathrm{Mod}'(\Sigma' \star \varphi)| = \{M' \in |\mathrm{Mod}(\Omega)| \mid M'\upharpoonright_{\varphi} \in |\mathrm{Mod}'(\Sigma')|\}$.

– $\mathcal{I}'$ has *φ-derivations* when for any $\mathcal{I}'$-signature Ω' such that $sig(\Omega') = \Omega$ there exists a designated $\mathcal{I}'$-signature, denoted $\varphi \,\square\, \Omega'$, such that

- $sig(\varphi \,\Box\, \Omega') = \Sigma$, and
- $|\mathrm{Mod}'(\varphi \,\Box\, \Omega')| = \{M' \upharpoonright_{\varphi} \mid M' \in |\mathrm{Mod}'(\Omega')|\}$.

For any class $\mathcal{D}$ of $\mathcal{I}$-signature morphisms we say that $\mathcal{I}'$ *has* $\mathcal{D}$*-translations/derivations* when it has φ-translations/derivations for each morphism φ in $\mathcal{D}$.

Notation 4.1. If $\mathcal{I}$ is inclusive (with inclusions denoted by $\sqsubseteq$) then for any $\mathcal{I}$-signature morphism $\varphi\colon \Sigma \to \Omega$ and any $\mathcal{I}'$-signatures Σ', Ω' such that $sig(\Sigma') \sqsubseteq \Sigma$ and $\Omega \sqsubseteq sig(\Omega')$, we may abbreviate $\Sigma' \star ((sig(\Sigma') \sqsubseteq \Sigma); \varphi)$ by $\Sigma' \star \varphi$ and $(\varphi; (\Omega \sqsubseteq sig(\Omega'))) \,\Box\, \Omega'$ by $\varphi \,\Box\, \Omega'$.

Let us discuss now some examples for Dfn. 4.2 and 4.3.

Example 4.1. Structured specifications in the style of Sannella and Tarlecki [36, 37] assume an abstract base institution $\mathcal{I}$ and also assume usually four structuring operators (we use here our own notations rather than theirs): binary union ($\cup$) for specifications that have the same signature, translation ($\star$), derivation ($\Box$), and a free semantics operator (we skip its details here). The terms formed from finite sets of sentences by these structuring operators constitute the signatures of $\mathcal{I}'$.

The structuring functor *sig* calculates in this case the signature of each specification by recursion on its structure. The $\mathcal{I}'$-sentences are inherited from $\mathcal{I}$ and the $\mathcal{I}'$-models are determined, just as the signatures, by recursion on the structure of the considered specification. If needed, a more detailed description of $\mathcal{I}'$ for this example may be found in [15].

Then $\mathcal{I}'$ has unions in the sense of Dfn. 4.2 by taking into account the trivial inclusion system of the category of $\mathcal{I}$-signatures, in which each signature morphism is an abstract surjection and the abstract inclusions are the identities. Note that in this case the union is a partial operation. It also has corresponding translations and derivations in the sense of Dfn. 4.3; however, this is a fairly straightforward discussion.

Example 4.2. A variant of Ex. 4.1, with total rather than partial unions, may be obtained by assuming a proper inclusion system for the signatures of $\mathcal{I}$, one that has unions for any two signatures. An important example would be $\underline{MSA}$ with the strong inclusion system for its category of signatures. This example, but endowed also with other structuring operators meant to capture non-protecting importation modes, forms the basis of the work reported in [17].

Example 4.3. An example that constitutes one of the most important motivations for our work and which corresponds to the actual practice of behavioural specification (e.g. CafeOBJ [18, 20]) is as follows. $\mathcal{I}$ is set to $\underline{HA}$ and we consider the same structuring operators as in Ex. 4.1, but with the following particularities:

- The union is defined for any two specifications SP_1 and SP_2 for which the union $sig(\mathrm{SP}_1) \sqcup sig(\mathrm{SP}_2)$ of their underlying signatures exists; moreover, we define $sig(\mathrm{SP}_1 \sqcup \mathrm{SP}_2) = sig(\mathrm{SP}_1) \sqcup sig(\mathrm{SP}_2)$.
- The initial semantics operator is defined only for signatures that have no hidden sorts.

4.3. Algebraic rules for structured behavioural specifications

In this section we consider a $(sig, \mathcal{I})$-structured institution $\mathcal{I}'$ that has unions in the sense of Dfn. 4.2. Let us call the signatures of $\mathcal{I}'$ *specifications*, and denote them by SP, SP′, etc.

Definition 4.4 (Module expression)**.** The set of *module expressions* is the least set such that

- SP is a module expression for each ‘variable’ SP denoting a specification,
- $E_1 \sqcup E_2$ is a module expression when E_1 and E_2 are module expressions,

- $E \star \varphi$ and $\varphi \Box E$ are module expressions when E is a module expression and φ is an $\mathcal{I}$-signature morphism.

Definition 4.5. For any module expressions E and E',

- $E \equiv E'$ when none of E and E' are defined or they are both defined and we have $sig(E) = sig(E')$ and $\mathrm{Mod}'(E) = \mathrm{Mod}'(E')$; and
- $E \overset{e}{\equiv} E'$ when E and E' are defined and $E \equiv E'$.

The rather straightforward proof of the following corollary is omitted.

Corollary 4.1. *If the unions of $\mathcal{I}$ satisfy the idempotence (3), commutativity (4) and associativity (5) rules then for any specifications* SP, SP′, SP″,

(6) $\quad \mathrm{SP} \sqcup \mathrm{SP} \overset{e}{\equiv} \mathrm{SP}.$

(7) $\quad \mathrm{SP} \sqcup \mathrm{SP}' \equiv \mathrm{SP}' \sqcup \mathrm{SP}.$

(8) $\quad (\mathrm{SP} \sqcup \mathrm{SP}') \sqcup \mathrm{SP}'' \equiv \mathrm{SP} \sqcup (\mathrm{SP}' \sqcup \mathrm{SP}'').$

Proposition 4.1. *If $\mathcal{I}'$ has $\mathcal{D}$-translations for a class $\mathcal{D}$ of $\mathcal{I}$-signature morphisms that is closed under composition to the left with inclusions, then for any specifications* SP_1, SP_2 *and any $\varphi \in \mathcal{D}$,*

(9) $\quad \mathrm{SP}_1 \sqcup \mathrm{SP}_2 \overset{e}{\equiv} \mathrm{SP}_1 \sqcup \mathrm{SP}_2 \quad$ *implies* $\quad (\mathrm{SP}_1 \sqcup \mathrm{SP}_2) \star \varphi \equiv (\mathrm{SP}_1 \star \varphi) \sqcup (\mathrm{SP}_2 \star \varphi).$

Proof. Let $\varphi \colon \Sigma \to \Omega$. That definedness of $\mathrm{SP}_1 \sqcup \mathrm{SP}_2$ means just that $sig(\mathrm{SP}_1) \sqcup sig(\mathrm{SP}_2)$ exists. When this happens, since $sig(\mathrm{SP}_1) \sqcup sig(\mathrm{SP}_2) \sqsubseteq \Sigma$ if and only if $sig(\mathrm{SP}_1) \sqsubseteq \Sigma$ and $sig(\mathrm{SP}_2) \sqsubseteq \Sigma$, it follows that $(\mathrm{SP}_1 \sqcup \mathrm{SP}_2) \star \varphi$ and $(\mathrm{SP}_1 \star \varphi) \sqcup (\mathrm{SP}_2 \star \varphi)$ are defined simultaneously.

When defined, both members of the conclusion of (9) have Ω as their underlying $\mathcal{I}$-signature. The fact that they have the same class of models is established by the following argument:

$$\begin{array}{llll}
M \in \mathrm{Mod}'((\mathrm{SP}_1 \sqcup \mathrm{SP}_2) \star \varphi) & \text{if and only if} & M{\upharpoonright_{\varphi}}{\upharpoonright_{sig(\mathrm{SP}_1) \sqcup sig(\mathrm{SP}_2)}} \in \mathrm{Mod}'(\mathrm{SP}_1 \sqcup \mathrm{SP}_2) & \text{(by Dfn. 4.3)} \\
 & \text{if and only if} & M{\upharpoonright_{\varphi}}{\upharpoonright_{sig(\mathrm{SP}_k)}} \in \mathrm{Mod}'(\mathrm{SP}_k),\ \text{for } k \in \{1,2\} & \text{(by Dfn. 4.2)} \\
 & \text{if and only if} & M \in \mathrm{Mod}'(\mathrm{SP}_k \star \varphi),\ \text{for } k \in \{1,2\} & \text{(by Dfn. 4.3)} \\
 & \text{if and only if} & M \in \mathrm{Mod}'((\mathrm{SP}_1 \star \varphi) \sqcup (\mathrm{SP}_2 \star \varphi)) & \text{(by Dfn. 4.2).}
\end{array}$$

□

The proof of the following result is identical in essence to a corresponding result from [17], and thus we omit it here. It should be noted however that its applicability to $\mathcal{I} = \underline{HA}$ relies upon the model amalgamation property of $\underline{HA}$ (cf. Cor. 3.5).

Proposition 4.2. *If $\mathcal{I}'$ has $\mathcal{D}$-derivations for a class $\mathcal{D}$ of $\mathcal{I}$-signature morphisms that is closed under composition then for any pushout of $\mathcal{I}$-signatures that consists of morphisms from $\mathcal{D}$ as below*

$$\begin{tikzcd}
\Sigma \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & \Sigma_1 \arrow[d, "\theta_1"] \\
\Sigma_2 \arrow[r, "\theta_2"'] & \Sigma'
\end{tikzcd}$$

and for any specifications SP_1, SP_2 *such that* $\Sigma_k = sig(\mathrm{SP}_k)$ *for* $k \in \{1,2\}$, *it holds that*

(10) $\quad (\varphi_k;\theta_k) \Box (\mathrm{SP}_1 \star \theta_1 \sqcup \mathrm{SP}_2 \star \theta_2) \overset{e}{\equiv} (\varphi_1 \Box \mathrm{SP}_1) \sqcup (\varphi_2 \Box \mathrm{SP}_2),\ $ *for* $k \in \{1,2\}$.

Corollary 4.2. *If $\mathcal{I}'$ has translations and derivations for inclusions and in $\mathcal{I}$ each intersection-union square describes a pushout then for any specifications* SP_1 *and* SP_2*,*

$$\mathrm{SP}_1 \sqcup \mathrm{SP}_2 \overset{e}{\equiv} \mathrm{SP}_1 \sqcup \mathrm{SP}_2 \quad \textit{implies} \quad \Sigma \Box (\mathrm{SP}_1 \sqcup \mathrm{SP}_2) \overset{e}{\equiv} (\Sigma \Box \mathrm{SP}_1) \sqcup (\Sigma \Box \mathrm{SP}_2), \tag{11}$$

where $\Sigma = \mathit{sig}(\mathrm{SP}_1) \sqcap \mathit{sig}(\mathrm{SP}_2)$.

Proof. Let us assume that $\mathrm{SP}_1 \sqcup \mathrm{SP}_2$ is defined. This means that $\mathit{sig}(\mathrm{SP}_1) \sqcup \mathit{sig}(\mathrm{SP}_2)$ exists. From the hypotheses, it follows that the square of inclusions depicted below is a pushout square.

$$\begin{tikzcd}
\Sigma \arrow[r, "\sqsubseteq"] \arrow[d, "\sqsubseteq"'] & \mathit{sig}(\mathrm{SP}_1) \arrow[d, "i_1"', "\sqsubseteq"] \\
\mathit{sig}(\mathrm{SP}_2) \arrow[r, "i_2", "\sqsubseteq"'] & \mathit{sig}(\mathrm{SP}_1) \sqcup \mathit{sig}(\mathrm{SP}_2)
\end{tikzcd}$$

Hence, by Prop. 4.2, we deduce that

$$\Sigma \Box (\mathrm{SP}_1 \star i_1 \sqcup \mathrm{SP}_2 \star i_2) \overset{e}{\equiv} (\Sigma \Box \mathrm{SP}_1) \sqcup (\Sigma \Box \mathrm{SP}_2). \tag{12}$$

By applying Prop. 4.1 with φ as the identity of $\mathit{sig}(\mathrm{SP}_1) \sqcup \mathit{sig}(\mathrm{SP}_2)$ we further deduce that

$$\mathrm{SP}_1 \star i_1 \sqcup \mathrm{SP}_2 \star i_2 \equiv \mathrm{SP}_1 \sqcup \mathrm{SP}_2,$$

which, since $\mathrm{SP}_1 \sqcup \mathrm{SP}_2$ is defined, may be obtained in its stronger form as

$$\mathrm{SP}_1 \star i_1 \sqcup \mathrm{SP}_2 \star i_2 \overset{e}{\equiv} \mathrm{SP}_1 \sqcup \mathrm{SP}_2. \tag{13}$$

The conclusion now follows from the relations (13) and (12). □

Cor. 4.1, Prop. 4.1, Prop. 4.2, and Cor. 4.2 can be easily instantiated to the case when $\mathcal{I} = \underline{\mathit{HA}}$ based on the results discussed in Sect. 3. They can also be very easily applied to other frameworks with partial unions of signatures, such as those considered in Ex. 4.1. Moreover, they generalise corresponding module-algebra properties that have been previously proved in the literature, in which the unions are assumed to be total.

For a specific set of specification structuring operators for equational logic, a corresponding variant of the distributivity rule (11) has been stated as an exercise in [37] and has been proved in an abstract institution-independent setting in [17]. Its property-oriented variant has been a cornerstone in [3] (for the special case of many sorted first order logic) and in [21] this gets a general institution-independent treatment and proof. The property-oriented variants of [3, 21] required not only significantly more difficult proofs but also significantly harder conditions, namely an interpolation property for the underlying institution. Since derivation gets here (□ in Dfn. 4.3) a model-oriented definition, the result of Cor. 4.2 shares with the corresponding result from [37] freedom from interpolation.[3] However, a big difference between these related results is that in our framework the union of specifications is a partial operation, hence the conditional form of the rule (11) and a more sophisticated proof, which in the case of $\underline{\mathit{HA}}$ relies upon the series of results about the existence of unions of $\underline{\mathit{HA}}$-signatures (detailed within Prop. 3.9).

[3] Which is quite important since interpolation in general is difficult to establish, and in the particular case of $\underline{\mathit{HA}}$ it has not been studied yet, at least up to our knowledge.

Acknowledgements

We thank both referees for their careful study of our work and for giving us useful feedback, which helped in improving the paper. The work of both authors has been supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE-2011-3-0439.

5. Conclusions

In this paper we have studied a number of compositionality properties for behavioural signatures; we have established the existence of pushouts and pullbacks, of model amalgamation, and also of an inclusion system for signatures that is suitable for modularization. A particular characteristic of this inclusion system is that it has only partial unions, which reflects further into partial rather than total algebraic rules for behavioural module compositions. Moreover, these rules may arise also in a conditional rather than in an unconditional form (like in the common situations).

Our definition of behavioural module is abstract in two ways: at the upper level, it is independent from any choice of actual structuring constructs, while at the base level, is independent of any choice of an actual behavioural-specification formalism. This is achieved by reliance upon the recent work on abstract structured specifications developed in [15].

Our work sets the ground for investigations of more specific aspects of structuring systems for behavioural specifications such as multiple parametrization, including several degrees of sharing. This can be achieved, for example, by adapting the theory of parameterisation developed in [17, 41] to the current framework of abstract structured specifications with partial structuring operators. In this way it would be possible to define distinct techniques for instantiating the parameters, as well as to study basic algebraic properties about the results of the instantiations.

Bibliography

[1] Marc Aiguier and Fabrice Barbier. An institution-independent proof of the Beth definability theorem. *Studia Logica*, 85(3):333–359, 2007.

[2] Edigio Astesiano, Michel Bidoit, Hélène Kirchner, Berndt Krieg-Brückner, Peter Mosses, Don Sannella, and Andrzej Tarlecki. CASL: The common algebraic specification language. *Theoretical Computer Science*, 286(2):153–196, 2002.

[3] Jan Bergstra, Jan Heering, and Paul Klint. Module algebra. *Journal of the Association for Computing Machinery*, 37(2):335–372, 1990.

[4] Michel Bidoit, Rolf Hennicker, and Martin Wirsing. Behavioural and abstractor specifications. *Sci. Comput. Program.*, 25(2-3):149–186, 1995.

[5] Michel Bidoit, Donald Sannella, and Andrzej Tarlecki. Observational interpretation of CASL specifications. *Mathematical Structures in Computer Science*, 18(2):325–371, 2008.

[6] Edward K. Blum and Francesco Parisi-Presicce. The semantics of shared submodules specifications. In Hartmut Ehrig, Christiane Floyd, Maurice Nivat, and James W. Thatcher, editors, *TAPSOFT 1*, volume 185 of *Lecture Notes in Computer Science*, pages 359–373, 1985.

[7] Francis Borceux. *Handbook of Categorical Algebra*. Cambridge University Press, 1994.

[8] Peter Burmeister. Partial algebra - an introductory survey. *Algebra Universalis*, 15:306–358, 1982.

[9] Peter Burmeister. *A Model Theoretic Oriented Approach to Partial Algebras*. Akademie-Verlag, Berlin, 1986.

[10] Rod Burstall and Joseph Goguen. The semantics of Clear, a specification language. In Dines Bjorner, editor, *1979 Copenhagen Winter School on Abstract Software Specification*, volume 86 of *Lecture Notes in Computer Science*, pages 292–332. Springer, 1980.

[11] Virgil Emil Căzănescu and Grigore Roşu. Weak inclusion systems. *Mathematical Structures in Computer Science*, 7(2):195–206, 1997.

[12] Răzvan Diaconescu. Elementary diagrams in institutions. *Journal of Logic and Computation*, 14(5):651–674, 2004.

[13] Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.

[14] Răzvan Diaconescu. Coinduction for preordered algebras. *Information and Computation*, 209(2):108–117, 2011.

[15] Răzvan Diaconescu. An axiomatic approach to structuring specifications. *Theoretical Computer Science*, 433:20–42, 2012.

[16] Răzvan Diaconescu. Three decades of institution theory. In Jean-Yves Béziau, editor, *Universal Logic: an Anthology*, pages 309–322. Springer Basel, 2012.

[17] Răzvan Diaconescu and Ionuţ Ţuţu. On the algebra of structured specifications. *Theoretical Computer Science*, 412(28):3145–3174, 2011.

[18] Răzvan Diaconescu and Kokichi Futatsugi. *CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification*, volume 6 of *AMAST Series in Computing*. World Scientific, 1998.

[19] Răzvan Diaconescu and Kokichi Futatsugi. Behavioural coherence in object-oriented algebraic specification. *Universal Computer Science*, 6(1):74–96, 2000. First version appeared as JAIST Technical Report IS-RR-98-0017F, June 1998.

[20] Răzvan Diaconescu and Kokichi Futatsugi. Logical foundations of CafeOBJ. *Theoretical Computer Science*, 285:289–318, 2002.

[21] Răzvan Diaconescu, Joseph Goguen, and Petros Stefaneas. Logical support for modularisation. In Gerard Huet and Gordon Plotkin, editors, *Logical Environments*, pages 83–130. Cambridge, 1993. Proceedings of a Workshop held in Edinburgh, Scotland, May 1991.

[22] Joseph Goguen. Types as theories. In George Michael Reed, Andrew William Roscoe, and Ralph F. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 357–390. Oxford, 1991. Proceedings of a Conference held at Oxford, June 1989.

[23] Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[24] Joseph Goguen and Răzvan Diaconescu. Towards an algebraic semantics for the object paradigm. In Hartmut Ehrig and Fernando Orejas, editors, *Recent Trends in Data Type Specification*, volume 785 of *Lecture Notes in Computer Science*, pages 1–34. Springer, 1994.

[25] Joseph Goguen and Grant Malcolm. A hidden agenda. *Theoretical Computer Science*, 245(1):55–101, 2000.

[26] Joseph Goguen and Grigore Roşu. Composing hidden information modules over inclusive institutions. In Olaf Owe, Stein Krogdahl, and Tom Lyche, editors, *From Object-Orientation to Formal Methods*, volume 2635 of *Lecture Notes in Computer Science*, pages 96–123. Springer, 2004.

[27] Rolf Hennicker and Michel Bidoit. Observational logic. In A. M. Haeberer, editor, *Algebraic Methodology and Software Technology*, number 1584 in LNCS, pages 263–277. Springer, 1999. Proc. AMAST'99.

[28] B. Jacobs and J.M. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of EATCS*, 62:222–259, 1997.

[29] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.

[30] José Meseguer. General logics. In H.-D. Ebbinghaus et al., editors, *Proceedings, Logic Colloquium, 1987*, pages 275–329. North-Holland, 1989.

[31] Horst Reichel. Behavioural equivalence – a unifying concept for initial and final specifications. In *Proceedings, Third Hungarian Computer Science Conference*. Akademiai Kiado, 1981. Budapest.

[32] Horst Reichel. *Initial Computability, Algebraic Specifications, and Partial Algebras*. Clarendon, 1987.

[33] Grigore Roşu. *Hidden Logic*. PhD thesis, University of California at San Diego, 2000.

[34] Grigore Roşu. Axiomatisability in inclusive equational logic. *Mathematical Structures in Computer Science*, 12(5):541–563, 2002.

[35] Grigore Roşu and Dorel Lucanu. Circular coinduction: A proof theoretical foundation. In Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki, editors, *Algebra and Coalgebra in Computer Science*, volume 5728 of *Lecture Notes in Computer Science*, pages 127–144, 2009.

[36] Donald Sannella and Andrzej Tarlecki. Specifications in an arbitrary institution. *Information and Control*, 76:165–210, 1988.

[37] Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specifications and Formal Software Development*. Springer, 2012.

[38] Andrzej Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoretical Computer Science*, 37:269–304, 1986.

[39] Andrzej Tarlecki, Rod Burstall, and Joseph Goguen. Some fundamental algebraic tools for the semantics of computation, part 3: Indexed categories. *Theoretical Computer Science*, 91:239–264, 1991.

[40] Shahab Tasharrofi and Eugenia Ternovska. A semantic account for modularity in multi-language modelling of search problems. In *Frontiers of combining systems*, volume 6989 of *Lecture Notes in Computer Science*, pages 259–274, 2011.

[41] Ionuţ Ţuţu. Parameterisation for abstract structured specifications. *Theoretical Computer Science*, 517:102–142, 2014.