An Axiomatic Approach to Structuring Specifications

Răzvan Diaconescu

Simion Stoilow Institute of Mathematics of the Romanian Academy

Abstract

In this paper we develop an axiomatic approach to structured specifications in which both the underlying logical system and corresponding institution of the structured specifications are treated as abstract institutions, which means two levels of institution independence. This abstract axiomatic approach provides a uniform framework for the study of structured specifications independently from any actual choice of specification building operators, and moreover it unifies the theory and the model oriented approaches. Within this framework we develop concepts and results about 'abstract structured specifications' such as co-limits, model amalgamation, compactness, interpolation, sound and complete proof theory, and pushout-style parameterization with sharing, all of them in a top down manner dictated by the upper level of institution independence.

1. Introduction

The crucial role played by structuring or modularization in the software development, including specification development, is so well known that it does not need here any explanations. However, while there is usually much emphasis on the role (without alternative) played for managing the problems generated by the high complexity of software systems both at the development and at the maintenance or evolution stages, there is less awareness about the superior specification power of structuring over specification in-the-small. From the many examples from the literature or from the folklore of algebraic specification illustrating this latter point let us recall here the example of fields (see [15]) and that of higher order programming (see [21]).

Consequently the study of structuring or modularization has been an important research topic within the formal specification community, the modern trend being that of theoretical developments that are independent of the logical systems underlying actual specification languages, e.g. [17, 18, 20, 37]. This is achieved through abstracting away the actual logical systems to abstract institutions (in the sense of [22]). That is what we consider here to be the *lower level of institution independence*. We may distinguish two major trends within the institution independent studies of structuring or modularization: the 'theory oriented' or 'property oriented' one (represented by [17, 23]) and the 'model oriented' one (represented by [37, 38]), that have been ideologically quite irreconcilable. Given a base institution $\mathcal{I}$, in the former approach the semantics of specifications is given by $\mathcal{I}$-theories, while in the latter approach it is given by classes of models indexed by corresponding signatures. In both cases specifications are freely built from the finite sets of $\mathcal{I}$-sentences by using fixed specific sets of specification building operators; the set of the building operators may vary according to the intended applications. In all above described situations one may consider

Email address: Razvan.Diaconescu@imar.ro (Răzvan Diaconescu)

Preprint submitted to Theoretical Computer Science *February 27, 2012*

an 'upper' institution whose signatures are either theories (in the theory oriented approach) or structured specifications (in the model oriented approach) that has the following couple of properties:

- there is a 'forgetful' functor Φ to the signatures of the base institution,
- both the 'upper' and the base institution share the same sentences modulo Φ, and
- the models of the 'upper' institution are (modulo Φ) a sub-class of the models of the base institution.

The main idea underlying our approach is to consider an abstract institution in the role of this 'upper' institution together with some properties relating it to the base institution. This is what we call the *upper level of institution independence*. Technically speaking, the whole situation can be condensed in a special form of an institution morphism (in the sense of [22]), and this is taken as the axiomatic basis for developing the theory of structured or modular specifications, without reference to theories or to specification building operators. The benefits of this approach are as follows:

1. From the point of view of the model oriented approach to structured specifications, our axiomatic approach achieves independence from the commitment to any specific set of specification building operators, in other words we achieve a general uniform theory of structured specifications that can be used for any particular set of specification building operators. This is very important when we consider the richness of possible specification building operators (the book [38] gives a hint about this) with new ones being proposed very recently (in [15] for dealing with non-protecting importation modes). Moreover, one may want also to consider quotienting structured specifications under various module algebra rules (in the sense of [2, 17, 38]), a situation which is also captured naturally by our approach. Our approach may cover also structuring contexts that are beyond conventional formal specification, such as the modular approach to the model expansion problems [44].
2. It unifies the theory and the model oriented approaches to modularization, many concepts or results that seemed to bear high similarity can be now seen precisely as being both instances of the same concept or result. A basic familiar example may be given by the lifting of co-limits from signatures to specifications that can be found in [22] for the theory oriented approach and in [38] for the model oriented approach. Moreover all the concepts or results developed here can be easily reflected down to either the theory or the model oriented approach.
3. The theory is developed in a top down manner, with the hypotheses introduced on a by-need basis with the benefit of understanding clearly the causality relationships between the various aspects of specification structuring and modularization.

The structure and the contents of the paper.

1. The first section surveys briefly concepts from institution theory that are used in this work.
2. The second section is dedicated to the introduction of the main concept underlying our approach, namely that of the 'upper' layer of institution independence in which the 'structured specifications' are treated abstractly as signatures of a an (abstract) institution $\mathcal{I}'$ sitting above the base institution $\mathcal{I}$ (that abstracts the underlying logical system). Inspired by the examples and the main motivation for this work, the $\mathcal{I}'$-signatures may also be called 'abstract structured specifications'.
3. In the next section we study co-limits and model amalgamation for the 'abstract structured specifications', which are two properties that play a fundamental role in the modularization studies. The examples discussed show that some of the concepts and results of this section may be seen as generalizing and unifying corresponding results from the theory and the model oriented approaches to structuring of specifications.

4. The section dedicated to normal forms introduces a semantic concept of 'normal form' for 'abstract structured specifications' that reflects abstractly the substance of the result on existence of normal forms for structured specifications developed in various forms in works such as [2, 5, 9]. We show that the existence of 'normal forms' is the main sufficient condition for lifting a series of important logical properties from the base institution to that of the 'abstract structured specifications', including (semantic) compactness, interpolation, and a complete proof system, the other conditions being rather technical and straightforward.
5. In the last technical section we define pushout-style parameterization with sharing within the framework of our 'abstract structured specifications'. This development relies crucially upon the category theoretic concept of *inclusion system* introduced in [17] and represents an abstract upgrading of similar ideas developed recently in [15] for (concretely) structured specifications.

2. Institution theoretic Preliminaries

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 [27]. 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 composition is denoted by ";" and in diagrammatic order. A sub-category $\mathbb{C}'$ of $\mathbb{C}$ is *broad* when $|\mathbb{C}'| = |\mathbb{C}|$. The category of sets (as objects) and functions (as arrows) is denoted by **Set**, and **CAT** is the category of all categories.[1]

2.2. Institutions

Institutions have been defined by Goguen and Burstall in [8], the seminal paper [22] being printed after a delay of many years. Below we recall the concept of institution which formalizes the intuitive notion of logical system, including syntax, semantics, and the satisfaction between them.

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

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

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

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

holds for each $M' \in |\mathsf{Mod}^{\mathcal{I}}(\Sigma')|$ *and* $\rho \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$*. We denote the* reduct *functor* $\mathsf{Mod}^{\mathcal{I}}(\varphi)$ *by* $\_\!\upharpoonright_{\varphi}$ *and the sentence translation* $\mathsf{Sen}^{\mathcal{I}}(\varphi)$ *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* $\mathsf{Sig}^{\mathcal{I}}$ *may be simply denoted* Sig.

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

General assumption: We assume that model isomorphisms preserve the satisfaction of all sentences of the 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 computing. A few of them can be found in [14, 38]. In fact the thesis underlying institution theory is that anything that deserves to be called logic can be captured as institution. Due to lack of space here let us very briefly present only the following one, of great relevance to computing science in general and to algebraic specification in particular.

Example 2.1 (Many sorted algebra (*MSA*)). The *MSA signatures* are pairs (S, F) consisting of a set of sort symbols S 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. A (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 component-wise 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 φ, the *reduct* $M'\restriction_\varphi$ of a model M' is defined by $(M'\restriction_\varphi)_x = M'_{\varphi(x)}$ for each sort or function symbol x from the domain signature of φ.

Sentences are the usual first order sentences built from equational and 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 ($\forall X$, $\exists X$). Sentence translations along signature morphisms just rename the sorts, function, and relation 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.

Notation 2.1. In any institution, for any set E of Σ-sentences

- for any Σ-model M, $M \models E$ denotes $M \models e$ for each $e \in E$,
- for each signature Σ and set E of Σ-sentences $\mathsf{Mod}(\Sigma, E)$ denotes the full sub-category of $\mathsf{Mod}(\Sigma)$ consisting of the models M such that $M \models E$,
- for any Σ-sentence ρ, $E \models \rho$ denotes that for each Σ-model M we have that $M \models E$ implies $M \models \rho$, and
- $E^\bullet$ denotes $\{\rho \mid E \models \rho\}$, i.e. the set of the semantic consequences of E.

Definition 2.2 (Compactness). *An institution is* compact *when for each set E of Σ-sentences and each Σ-sentence ρ if $E \models \rho$ then $E_0 \models \rho$ for some finite subset $E_0 \subseteq E$.*

2.3. Model amalgamation

The crucial role of model amalgamation for the semantics studies of formal specifications comes up in very many works in the area, a few early examples being [17, 30, 37, 41]. The model amalgamation property is a necessary condition in many institution-independent model theoretic results (see [14]), 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 [14] for the details). Its importance within the context of module algebra has been first emphasized in [17]. Model amalgamation properties for institutions formalize the possibility of amalgamating models of different signatures when they are consistent on some kind of generalized 'intersection' of signatures.

Definition 2.3 (Amalgamation square). *A commutative square of signature morphisms*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi_1} & \Sigma_1 \\ {\scriptstyle \varphi_2}\downarrow & & \downarrow{\scriptstyle \theta_1} \\ \Sigma_2 & \xrightarrow[\theta_2]{} & \Sigma' \end{array}$$

is an amalgamation square *if and only if for each Σ_1-model M_1 and a 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 off the uniqueness requirement we call this a* weak model amalgamation square.

In most of the institutions formalizing conventional or non-conventional logics, pushout squares of signature morphisms are model amalgamation squares [14, 17]. These of course include our benchmark *MSA* example.

Definition 2.4 (Model amalgamation; 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 co-limits, however for reasons of simplicity of presentation and because they are by far the most important case in the applications, in this paper we consider model amalgamation only for pushouts.

2.4. Institution independent interpolation

In the algebraic specification literature there are several institution-independent formulations of interpolation, all of them being strongly related. For example [40] is one of the first work introducing the concept of interpolation at the level of abstract institutions. The common feature of these formulations is that they generalize the conventional intersection-union (of signatures) framework to commutative squares of signature morphisms. In most cases these commutative squares are required to be pushouts (like in [4, 5, 18, 41]), in other case the signature morphisms are required to be (abstract) inclusions (like in [17]). However in [12] it has been noticed that the mere formulation of interpolation does not require any extra technical assumptions besides a commuting square of signature morphisms, the role of such additional assumptions having more to do with the proof of interpolation properties rather than with its formulation.

Definition 2.5 (Craig-Robinson interpolation). *In any institution we say that a commutative square of signature morphisms*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi_1} & \Sigma_1 \\ {\scriptstyle \varphi_2}\downarrow & & \downarrow{\scriptstyle \theta_1} \\ \Sigma_2 & \xrightarrow[\theta_2]{} & \Sigma' \end{array}$$

is a Craig-Robinson Interpolation square *(abbreviated* CRI *square) when for each set E_1 of Σ_1-sentences and each sets E_2 and Γ_2 of Σ_2-sentences, if $\theta_1(E_1) \cup \theta_2(\Gamma_2) \models_{\Sigma'} \theta_2(E_2)$, then there exists a set E of Σ-sentences such that $E_1 \models_{\Sigma_1} \varphi_1(E)$ and $\Gamma_2 \cup \varphi_2(E) \models_{\Sigma_2} E_2$.*

The particular case of Craig-Robinson interpolation for Γ_2 empty is called Craig interpolation.

In logic this case is usually more studied than Craig-Robinson interpolation. Craig-Robinson form of interpolation seems to have been first introduced in first order logic by [28]. Several works [2, 14, 17, 18] show that Craig-Robinson rather than Craig may be the appropriate interpolation concept for formal specification studies. Particular examples in this sense are the interdependency relationship between Craig-Robinson interpolation and important modularization property [18] and Borzyszkowski's complete calculus for structured specifications [5] which in reality relies upon the Craig-Robinson form of interpolation (this was shown in [14] which corrects the rather restricted original result of [5] relying upon Craig interpolation plus additional conditions for the base institutions, the latter narrowing significantly the range of the applications of this important and beautiful result). Moreover even in model theory sometimes [34] Craig-Robinson seems to be the appropriate form of interpolation. This is one of the reasons we adopt here this form of interpolation, another one being just technical. The name 'Craig-Robinson' has been used for instances of the corresponding interpolation property in [18, 39, 45], 'Maehara interpolation' in sentential logic studies, while 'strong Craig interpolation' has been used in [17]. We mention that Craig-Robinson and Craig forms of interpolation can be shown equivalent under some additional conditions on the institution [14]. For example this applies to classical first order logic (and to *MSA* as well) (which perhaps is the main reason why in conventional logic Craig-Robinson formulation of interpolation is shadowed by the simpler Craig formulation), but not for example to Horn clause or equational logic.

Another important aspect of Def. 2.5 is that it uses sets of sentences rather than single sentences, as is common in conventional logic. The works [36] and [17] argue successfully that the formulation of interpolation in terms of sets of sentences is more natural than the more traditional formulations in terms of single sentences. First, on the one hand, the applications of interpolation do not require the single sentence formulation, and on the other hand the single sentence formulation excludes important examples such as equational or Horn logics. Then, in traditional works on or using interpolation, under the assumption of compactness the two formulations can be shown equivalent [14].

The definition below formulates interpolation as a property of institutions. In its current form it has been introduced in [13] as a simplified variant of the original definition of [5].

Definition 2.6 (($\mathcal{L}, \mathcal{R}$)-interpolation). *For any classes of signature morphisms $\mathcal{L}, \mathcal{R} \subseteq \mathsf{Sig}$ in any institution, we say that the institution has the* Craig-Robinson ($\mathcal{L}, \mathcal{R}$)-interpolation *if each pushout square of signature morphisms of the form*

$$\begin{array}{ccc} \bullet & \xrightarrow{\mathcal{L}} & \bullet \\ {\scriptstyle \mathcal{R}}\downarrow & & \downarrow \\ \bullet & \longrightarrow & \bullet \end{array}$$

is a CRI square.

Example 2.2. According to [6, 25], *MSA* has Craig-Robinson $(\mathcal{L}, \mathcal{R})$-interpolation when $\mathcal{L}$ is the class of all sort injective signature morphisms and $\mathcal{R}$ the class of all signature morphisms or the other way around. Interestingly, this result which stayed as a conjecture for several years has received an elegant proof in [25] using an institution independent method; in fact the result proved there is institution independent and thus much more general than classical first order logic interpolation.

Example 2.3. According to [14], the Horn clause and the equational logic sub-institutions of *MSA* have Craig $(\mathcal{L}, \mathcal{R})$-interpolation for $\mathcal{L}$ the class of all signature morphisms and $\mathcal{R}$ the class of the injective signature morphisms, and it has Craig-Robinson $(\mathcal{L}, \mathcal{R})$-interpolation for $\mathcal{R}$ the class of all signature morphisms and $\mathcal{L}$ the class of the signature morphisms that are injective on the sorts and such that no operation symbol outside the image of the signature morphism is allowed to have the sort in the image of the signature morphism (in other words if $\varphi\colon (S, F) \to (S', F')$ and $\sigma' \in F'_{w' \to s'}$ with $s' \in \varphi(s)$ then there exists $\sigma \in F_{w \to s}$ such that $\varphi(\sigma) = \sigma'$). The proof of this result given in [14] involves the interpolation result for Grothendieck institutions of [13].

2.5. Proof theory for institutions

The enhancement of institution theory with a proof theoretic side is motivated by the important need to address the verification aspect of formal specifications. In the following we recall from the literature the standard way to do this, which is essentially an upgrading of Tarski's notion of 'consequence operator' [43] to the institution theoretic framework that is indexed by signatures. In the institution theory literature this is called π-*institution* in [20, 29] or *entailment system* in [14, 26, 30, 32, 33].

Definition 2.7. *An* entailment system $\vdash$ *consists of*

1. *a functor* $\mathsf{Sen}\colon \mathsf{Sig} \to \mathbf{Set}$*; the objects of* Sig *are called* signatures *and the elements of each* $\mathsf{Sen}(\Sigma)$ *are called* Σ-sentences*, and*
2. *a relation*[2] $\vdash_\Sigma \subseteq \mathcal{P}(\mathsf{Sen}(\Sigma)) \times \mathsf{Sen}(\Sigma)$ *for each* $\Sigma \in |\mathsf{Sig}|$*, called* Σ-consequence*,*

such that the following conditions hold:

A. reflexivity: $\{e\} \vdash_\Sigma e$ *for each* $e \in \mathsf{Sen}(\Sigma)$*;*

B. monotonicity: *if* $E \vdash_\Sigma e$ *and* $E \subseteq E'$ *then* $E' \vdash_\Sigma e$*;*

C. transitivity: *if* $E \vdash_\Sigma e'$ *for each* $e' \in E'$ *and if* $(E \cup E') \vdash_\Sigma e$*, then* $E \vdash_\Sigma e$*;*

D. translation: *if* $E \vdash_\Sigma e$ *and if* $\varphi\colon \Sigma \to \Sigma'$ *in* Sig*, then* $\mathsf{Sen}(\varphi)(E) \vdash_{\Sigma'} \mathsf{Sen}(\varphi)(e)$*.*

Note that each institution appears canonically as an entailment system by considering the semantic consequence relations $\models_\Sigma$ in the role of the consequence relations $\vdash_\Sigma$. Conversely, each entailment system can be given a rather artificial model theory by a comma category construction on theories [30].

Definition 2.8. *Given an institution* $\mathcal{I} = (\mathsf{Sig}, \mathsf{Sen}, \mathsf{Mod}, \models)$*, an* entailment system $\vdash$ for $\mathcal{I}$ *is just an entailment system* $\vdash$ *that shares with* $\mathcal{I}$ *the sentence functor (and implicitly the category of the signatures). Then* $(\mathcal{I}, \vdash)$ *is* sound *when* $E \vdash_\Sigma \rho$ *implies* $E \models_\Sigma \rho$ *and it is* complete *when* $E \models_\Sigma \rho$ *implies* $E \vdash_\Sigma \rho$ *(for any signature* Σ*, any set of* Σ*-sentences* E *and any* Σ*-sentence* ρ *of the same signature).*

The pair $(\mathcal{I}, \vdash)$ of Dfn. 2.8 appears in the literature under various names, such as 'logic' in [32].

[2] Here $\mathcal{P}$ denotes the power set function.

3. Structured institutions

In this section we introduce the main technical concept underlying our novel approach to structured specifications involving two levels of institution independence, and illustrates it with a series of examples from the theory of structured specifications. These examples will serve as a benchmark to our abstract developments.

Definition 3.1 (Structured institutions). *Given two institutions* $\mathcal{I} = (\mathsf{Sig}, \mathsf{Sen}, \mathsf{Mod}, \models)$ *and* $\mathcal{I}' = (\mathsf{Sig}', \mathsf{Sen}', \mathsf{Mod}', \models'$ *) we say that* $\mathcal{I}'$ *is* structured over $\mathcal{I}$ through Φ *when*

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

$$\begin{array}{ccc} \mathsf{Mod}'(\Sigma'_1) & \xrightarrow{\subseteq} & \mathsf{Mod}(\Phi(\Sigma'_1)) \\ {\scriptstyle \mathsf{Mod}'(\varphi)} \uparrow & & \uparrow {\scriptstyle \mathsf{Mod}(\Phi(\varphi))} \\ \mathsf{Mod}'(\Sigma'_2) & \xrightarrow[\subseteq]{} & \mathsf{Mod}(\Phi(\Sigma'_2)) \end{array}$$

  and

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

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

Within the framework of Dfn. 3.1, the examples given below in the section, especially Ex. 3.3 and 3.4, support the idea of the following nickname for the $\mathcal{I}'$-signatures: *abstract structured specifications*.

The readers familiar with the concept of *institution morphism* introduced by [22] may understand the concept of structured institution in the following way:

Fact 3.1. $\mathcal{I}'$ *is structured over* $\mathcal{I}$ *if and only if there exists an institution morphism* $(\Phi, \alpha, \beta) \colon \mathcal{I}' \to \mathcal{I}$ *such that* α *is identity and the components of* β *are full subcategory inclusions.*

Example 3.1 (Trivial structuring). Each institution $\mathcal{I}$ is trivially structured over itself through the identity functor on the signature category.

Example 3.2 (Theories). According to [22], in any institution $\mathcal{I}$ a *theory* is a pair (Σ, E) that consists of a signature Σ and a set E of Σ-sentences closed under semantic consequence, i.e. if $E \models \rho$ then $\rho \in E$. A *theory morphism* $\varphi \colon (\Sigma, E) \to (\Sigma', E')$ is a signature morphism $\varphi \colon \Sigma \to \Sigma'$ such that $\varphi(E) \subseteq E'$. Note that $\mathcal{I}$-theory morphisms form a category under the composition induced by the composition of signature morphisms.

The institution of $\mathcal{I}$-theories, denoted $\mathcal{I}^{\text{th}}$, is defined as follows:

- the category of the signatures of $\mathcal{I}^{\text{th}}$ is the category of theories of $\mathcal{I}$, and

– for each theory (Σ, E), the (Σ, E)-sentences are the Σ-sentences and the category of the (Σ, E)-models is the full subcateory of the Σ-models that satisfy E.

Then $\mathcal{I}^{\text{th}}$ is structured over $\mathcal{I}$ through the forgetful functor from the category of $\mathcal{I}$-theories to the category of $\mathcal{I}$-signatures, i.e. that maps each theory (Σ, E) to its underlying signature Σ. The institution $\mathcal{I}^{\text{th}}$ constitutes the implicit framework for the modularization studies in works such as [17].

Example 3.3 (Structured specifications). Given any institution $\mathcal{I} = (\mathsf{Sig}, \mathsf{Sen}, \mathsf{Mod}, \models)$ with designated classes of signature morphisms $\mathcal{T}$ and $\mathcal{D}$ the class of the $(\mathcal{T}, \mathcal{D})$-*structured specifications* [5] is the least class such that

– it contains all finite *presentations*, i.e. pairs (Σ, E) with Σ signature and E finite set of Σ-sentences; we also define $\Phi(\Sigma, E) = \Sigma$,

– if SP_1 and SP_2 are structured specifications such that $\Phi(\mathrm{SP}_1) = \Phi(\mathrm{SP}_2)$ then $\mathrm{SP}_1 \cup \mathrm{SP}_2$ is also a structured specification and we define $\Phi(\mathrm{SP}_1 \cup \mathrm{SP}_2) = \Phi(\mathrm{SP}_i)$,

– if SP is a structured specification and $(\varphi\colon \Phi(\mathrm{SP}) \to \Sigma') \in \mathcal{T}$ then $\mathrm{SP} \star \varphi$ is structured specification and $\Phi(\mathrm{SP} \star \varphi) = \Sigma'$, and

– if SP′ is a structured specification and $(\varphi\colon \Sigma \to \Phi(\mathrm{SP}')) \in \mathcal{D}$ then $\varphi \mid \mathrm{SP}'$ is structured specification and $\Phi(\varphi \mid \mathrm{SP}') = \Sigma$.

For each structured specification SP its category of models $\mathsf{Mod}(\mathrm{SP})$ is the full subcategory of $\mathsf{Mod}(\Phi(\mathrm{SP}))$ determined as follows:

– $M \in |\mathsf{Mod}(\Sigma, E)|$ if and only if $M \models E$,

– $|\mathsf{Mod}(\mathrm{SP}_1 \cup \mathrm{SP}_2)| = |\mathsf{Mod}(\mathrm{SP}_1) \cap \mathsf{Mod}(\mathrm{SP}_2)|$,

– $|\mathsf{Mod}(\mathrm{SP} \star \varphi)| = \{M' \mid M'\restriction_\varphi \in \mathsf{Mod}(\mathrm{SP})\}$, and

– $|\mathsf{Mod}(\varphi \mid \mathrm{SP}')| = \{M'\restriction_\varphi \mid M' \in \mathsf{Mod}(\mathrm{SP})\}$.

According to [15, 38] a *morphism of specifications* $\varphi\colon \mathrm{SP} \to \mathrm{SP}'$ is an $\mathcal{I}$-signature morphism $\Phi(\mathrm{SP}) \to \Phi(\mathrm{SP}')$ such that for each $M' \in \mathsf{Mod}(\mathrm{SP}')$ we have that $M'\restriction_\varphi \in \mathsf{Mod}(\mathrm{SP})$. Note that structured specifications and their morphisms form a category Spec and $\Phi\colon \mathsf{Spec} \to \mathsf{Sig}$ is a functor.

These data are enough to define the *institution of the* $(\mathcal{T}, \mathcal{D})$-*structured specifications* as an institution which is structured over $\mathcal{I}$ through Φ.

In the literature one may find several other examples of primitive specification building operators besides the ones presented in the example, most notably initial semantics operators. In fact, each specification formalism may be based upon its own specific set of primitive specification building operators, and in this respect there may be significant differences across various specification languages. From this perspective, this example may be replicated for other sets of specification building operators. Besides of course the choice of the base institution $\mathcal{I}$, another parameter is given by is the choice of $\mathcal{T}$ and $\mathcal{D}$, which is also specific to the particularities of actual specification formalisms.

Example 3.4 (Quotienting the structuring).

Definition 3.2 (Structuring congruence). *Given an institution $\mathcal{I}'$ structured over $\mathcal{I}$ through Φ a congruence relation $\equiv$ on* Sig' *is a* structuring congruence *when*

– *if two signatures are equivalent, i.e.* $\Sigma'_1 \equiv \Sigma'_2$ *then* $\Phi(\Sigma'_1) = \Phi(\Sigma'_2)$ *and* $\mathsf{Mod}'(\Sigma'_1) = \mathsf{Mod}'(\Sigma'_2)$*, and*

– *if two signature morphisms are equivalent, i.e.* $\varphi'_1 \equiv \varphi'_2$*, then* $\Phi(\varphi'_1) = \Phi(\varphi'_2)$*.*

Note that the latter condition implies also $\mathsf{Mod}'(\varphi'_1) = \mathsf{Mod}'(\varphi'_2)$.

For any structuring congruence we may build the quotient of $\mathcal{I}'/_{\equiv} = (\mathsf{Sig}'', \mathsf{Sen}'', \mathsf{Mod}'', \models'')$ which has the quotient category $\mathsf{Sig}'/_{\equiv}$ as its category of signatures Sig'', and the sentence and the model functors and the satisfaction relation defined canonically those of $\mathcal{I}'$, i.e. $\mathsf{Sen}''(\Sigma'/_{\equiv}) = \mathsf{Sen}'(\Sigma') = \mathsf{Sen}(\Phi(\Sigma'))$, $\mathsf{Mod}''(\Sigma'/_{\equiv}) = \mathsf{Mod}'(\Sigma')$ and $M \models''_{\Sigma'/_{\equiv}} \rho$ if and only if $M \models'_{\Sigma'} \rho$ (if and only if $M \models_{\Phi(\Sigma)} \rho$). Note that this institution is structured over $\mathcal{I}$ through $\Phi/_{\equiv} : \mathsf{Sig}'/_{\equiv} \to \mathsf{Sig}$ defined by $\Phi/_{\equiv}(\Sigma'/_{\equiv}) = \Phi(\Sigma')$.

This quotienting has relevance within the context of the classes of examples described by Ex. 3.3 above. Then $\equiv$ may be defined as the equivalence generated by some of the module algebra rules on the structured specifications [2, 15, 17, 38] such as for example the associativity of union of specifications ($\cup$), which essentially means that we do not distinguish between $(\mathrm{SP} \cup \mathrm{SP}') \cup \mathrm{SP}''$ and $\mathrm{SP} \cup (\mathrm{SP}' \cup \mathrm{SP}'')$. The largest case for $\equiv$ is to define it as semantical equivalence $\models\mid$, i.e. $\mathrm{SP} \models\mid \mathrm{SP}'$ if and only if $\Phi(\mathrm{SP}) = \Phi(\mathrm{SP}')$ and $\mathsf{Mod}'(\mathrm{SP}) = \mathsf{Mod}'(\mathrm{SP}')$.

Yet another example may be given by the modular structure of model expansion problems [44]; we omit its presentation here.

The following result is straightforward and shows that the process of structuring institutions is compositional (in a category theoretic sense), or in other words the nesting of structuring of institutions yields a structuring of institutions. This result is relevant (within the context of Ex. 3.3) to situations when one wants to add new specification building operators in a way that does not interfere with the existing ones.

Corollary 3.1. *If* $\mathcal{I}'$ *is structured over* $\mathcal{I}$ *through* Φ *and* $\mathcal{I}''$ *is structured over* $\mathcal{I}'$ *through* Φ' *then* $\mathcal{I}''$ *is structured over* $\mathcal{I}$ *through* $\Phi';\Phi$*.*

4. General properties

In this section we address two properties of the abstract structured specifications that are considered of fundamental importance in the algebraic specification approaches to modularization:

1. The existence of co-limits and their relationship to the co-limits of signatures in the base institution.
2. The model amalgamation properties. We will show that these rely upon three factors that can be established quite naturally in the applications: the corresponding amalgamation property at the level of the base institution, a pushout preservation property, and a model compositionality property of the structuring.

4.1. Co-limits

Definition 4.1 (Lifting co-limits). *Given an institution* $\mathcal{I}'$ *structured over* $\mathcal{I}$ *through* Φ*, we say that* Φ lifts co-limits *when for each diagram* D *in* Sig' *each co-limit* μ *of* $D;\Phi$*, i.e. the image in* Sig *of* D *through* Φ*, can be lifted to a co-limit* μ' *of* D *such that* $\mu'\Phi = \mu$*.*

The following consequence of lifting co-limits is rather straightforward.

Fact 4.1. *If* Φ *lifts co-limits in the sense of Dfn. 4.1 then it also preserves co-limits.*

Example 4.1. The seminal paper [22] shows the lifting of co-limits for the structuring of Ex. 3.2. This result constitutes the foundations for Clear style module systems [8] and is one of the most important results in the institution-independent development of the theory of the algebraic specifications. Let us recall here the case of pushouts. Given two theory morphisms $\varphi\colon (\Sigma, E) \to (\Sigma_1, E_1)$ and $\theta\colon (\Sigma, E) \to (\Sigma_2, E_2)$ for any pushout of the underlying signature morphisms as follows

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi} & \Sigma_1 \\ {\scriptstyle\theta}\downarrow & & \downarrow{\scriptstyle\theta'} \\ \Sigma_2 & \xrightarrow[\varphi']{} & \Sigma' \end{array}$$

we let $E' = (\theta'(E_1) \cup \varphi'(E_2))^\bullet$. Then the following is a pushout square of signature morphisms.

$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\varphi} & (\Sigma_1, E_1) \\ {\scriptstyle\theta}\downarrow & & \downarrow{\scriptstyle\theta'} \\ (\Sigma_2, E_2) & \xrightarrow[\varphi']{} & (\Sigma', E') \end{array}$$

Example 4.2. The book [38] shows the lifting of finite co-limits for the structuring of Ex. 3.3 provided that the set of the specification building operators contains union ($\cup$) and translation ($\star$). However in this case some special technical conditions are required with respect to the class $\mathcal{T}$, in the sense that the components of the co-limit co-cone need to belong to $\mathcal{T}$. In fact this may restrict the class of co-limits that can be lifted, fortunately without leading to real restrictions at the level of the applications. Let us see how this works for pushouts. Given specification morphisms $\varphi\colon \mathrm{SP} \to \mathrm{SP}_1$ and $\theta\colon \mathrm{SP} \to \mathrm{SP}_2$ and a pushout of the underlying signature morphisms

$$\begin{array}{ccc} \Phi(\mathrm{SP}) & \xrightarrow{\varphi} & \Phi(\mathrm{SP}_1) \\ {\scriptstyle\theta}\downarrow & & \downarrow{\scriptstyle\theta'} \\ \Phi(\mathrm{SP}_2) & \xrightarrow[\varphi']{} & \Sigma' \end{array}$$

Then the following is a pushout square of specification morphisms.

$$\begin{array}{ccc} \mathrm{SP} & \xrightarrow{\varphi} & \mathrm{SP}_1 \\ {\scriptstyle\theta}\downarrow & & \downarrow{\scriptstyle\theta'} \\ \mathrm{SP}_2 & \xrightarrow[\varphi']{} & (\mathrm{SP}_1 \star \theta') \cup (\mathrm{SP}_2 \star \varphi') \end{array}$$

Proposition 4.1. *Given an institution $\mathcal{I}'$ structured over $\mathcal{I}$ through Φ and a structuring congruence $\equiv$ such that*

1. Φ *is faithful, and*
2. *if $\Sigma'_1 \equiv \Sigma'_2$ then there exists $i\colon \Sigma'_1 \to \Sigma'_2$ such that $i/_{\equiv} = 1_{\Sigma'_k/_{\equiv}}$, $k \in \{1, 2\}$,*

if Φ lifts co-limits then $\Phi/_{\equiv}$ lifts co-limits too.

Proof. For reasons of clarity of the presentation let us do this proof for the case of pushouts. Thus let us consider a couple of morphisms $\varphi'' : \Sigma'' \to \Sigma''_1$ and $\theta'' : \Sigma'' \to \Sigma''_2$ in $\mathsf{Sig}'/_{\equiv}$ and consider a pushout square in Sig as follows:

$$\begin{array}{ccc} \Sigma = \Phi/_{\equiv}(\Sigma'') & \xrightarrow{\varphi = \Phi/_{\equiv}(\varphi'')} & \Sigma_1 = \Phi/_{\equiv}(\Sigma''_1) \\ {\scriptstyle \theta = \Phi/_{\equiv}(\theta'')}\downarrow & & \downarrow{\scriptstyle \alpha} \\ \Sigma_2 = \Phi/_{\equiv}(\Sigma''_2) & \xrightarrow[\beta]{} & \Omega \end{array}$$

We have that there exists $\varphi' : \Sigma' \to \Sigma'_1$ and $\theta' : \overline{\Sigma'} \to \Sigma'_2$ in Sig' such that $\varphi'/_{\equiv} = \varphi''$ and $\theta'/_{\equiv} = \theta''$. It follows that $\Sigma' \equiv \overline{\Sigma'}$; let $i : \Sigma' \to \overline{\Sigma'}$ with $i/_{\equiv} = 1_{\Sigma''}$. Since $\Phi(\varphi') = \varphi$ and $\Phi(i;\theta') = \Phi(i);\Phi(\theta') = \Phi/_{\equiv}(i/_{\equiv});\theta = 1;\theta = \theta$ let us consider a lifting of the above pushout square as follows:

$$\begin{array}{ccc} \Sigma' & \xrightarrow{\varphi'} & \Sigma'_1 \\ {\scriptstyle i;\theta'}\downarrow & & \downarrow{\scriptstyle \alpha'} \\ \Sigma'_2 & \xrightarrow[\beta']{} & \Omega' \end{array}$$

We define $\alpha'' = \alpha'/_{\equiv}$ and $\beta'' = \beta'/_{\equiv}$. The square below commutes

$$\begin{array}{ccc} \Sigma'' & \xrightarrow{\varphi''} & \Sigma''_1 \\ {\scriptstyle \theta''}\downarrow & & \downarrow{\scriptstyle \alpha''} \\ \Sigma''_2 & \xrightarrow[\beta'']{} & \Omega'' = \Omega/_{\equiv} \end{array}$$

because it represents an application of the quotienting functor $\_/_{\equiv}$ to the previous square, that obviously commutes since it is a pushout square.

Now we show that the above square is a pushout. For this let us consider $f'' : \Sigma''_1 \to \Gamma''$ and $g'' : \Sigma''_2 \to \Gamma''$ such that $\varphi'';f'' = \theta'';\beta''$. We have to show that there exists a unique $h'' : \Omega'' \to \Gamma''$ such that $\alpha'';h'' = f''$ and $\beta'';h'' = g''$. Let $f' : \overline{\Sigma'_1} \to \Gamma'$ and $g' : \overline{\Sigma'_2} \to \overline{\Gamma'}$ such that $f'/_{\equiv} = f''$ and $g'/_{\equiv} = g''$. These imply the existence of $i_1 : \Sigma'_1 \to \overline{\Sigma'_1}$, $i_2 : \Sigma'_2 \to \overline{\Sigma'_2}$ and $j : \Gamma' \to \overline{\Gamma'}$ that are mapped by the quotienting functor $\_/_{\equiv}$ to identities. It follows that

$$\Phi(i;\theta';i_2;g') = \Phi(i;\theta');\Phi(i_2;g') = \Phi/_{\equiv}(\theta'');\Phi/_{\equiv}(g'') = \Phi/_{\equiv}(\theta'';g'') = \Phi/_{\equiv}(\varphi'';f'') =$$
$$\Phi/_{\equiv}(\varphi'');\Phi/_{\equiv}(f'') = \Phi(\varphi';i_1);\Phi(f';j) = \Phi(\varphi';i_1;f';j).$$

Since Φ is faithful we obtain that $i;\theta';i_2;g' = \varphi';i_1;f';j$.

$$\begin{array}{ccccc} \Sigma' & \xrightarrow{\varphi'} & \Sigma'_1 & \xrightarrow{i_1} & \overline{\Sigma'_1} \\ {\scriptstyle i;\theta'}\downarrow & & \downarrow{\scriptstyle \alpha'} & & \downarrow{\scriptstyle f'} \\ \Sigma'_2 & \xrightarrow[\beta']{} & \Omega' & & \Gamma' \\ {\scriptstyle i_2}\downarrow & & & \searrow^{h'} & \downarrow{\scriptstyle j} \\ \overline{\Sigma'_2} & & \xrightarrow[g']{} & & \overline{\Gamma'} \end{array}$$

By the pushout property of the left upper corner square in the diagram above there exists a unique h' such that

$$\alpha'; h' = i_1; f'; j \text{ and } \beta'; h' = i_2; g'. \tag{1}$$

We define $h'' = h'/_{\equiv}$. By applying the quotienting functor $\_/_{\equiv}$ to the equations (1) we get that

$$\alpha''; h'' = f'' \text{ and } \beta''; h'' = g''. \tag{2}$$

For showing the uniqueness of h'' that satisfies the equations (2) let us assume a k'' such that $\alpha''; k'' = f''$ and $\beta''; k'' = g''$. By applying $\Phi/_{\equiv}$ to this equalities we obtain that $\alpha; \Phi/_{\equiv}(k'') = \Phi(f')$ and $\beta; \Phi/_{\equiv}(k'') = \Phi(g')$. By the uniqueness aspect of the pushout property of

$$\begin{array}{ccc}
\Sigma = \Phi/_{\equiv}(\Sigma'') & \xrightarrow{\varphi = \Phi/_{\equiv}(\varphi'')} & \Sigma_1 = \Phi/_{\equiv}(\Sigma''_1) \\
{\scriptstyle \theta = \Phi/_{\equiv}(\theta'')} \downarrow & & \downarrow {\scriptstyle \alpha} \\
\Sigma_2 = \Phi/_{\equiv}(\Sigma''_2) & \xrightarrow[\beta]{} & \Omega
\end{array}$$

we get that $\Phi/_{\equiv}(k'') = \Phi/_{\equiv}(h'')$. The desired conclusion followed if we transferred the faithfulness property from Φ to $\Phi/_{\equiv}$. For this it is enough to consider $k' : \Omega'_0 \to \Gamma'_0$ such that $k'/_{\equiv} = k''$ and note that there exists $j_1 : \Omega' \to \Omega'_0$ and $j_2 : \overline{\Gamma'} \to \Gamma'_0$ such that $j_1/_{\equiv}$ and $j_2/_{\equiv}$ are identities. From $\Phi/_{\equiv}(k'') = \Phi/_{\equiv}(h'')$ it follows that $\Phi(h') = \Phi(j_1; k'; j_2)$ and from the faithfulness of Φ that $h' = j_1; k'; j_2$. By applying the quotienting functor $\_/_{\equiv}$ to this equality we get $h'' = k''$. □

Example 4.3. It is easy to note that the structuring congruences on the structured specifications discussed in Ex. 3.4 satisfy the conditions of Prop. 4.1, hence the forgetful functors on the signatures of those quotienting of structured specifications lift co-limits.

4.2. Model amalgamation

Definition 4.2 (Compositionality). *An institution $\mathcal{I}'$ structured over $\mathcal{I}$ through Φ is* compositional *when for each pushout in* Sig'

$$\begin{array}{ccc}
\Sigma' & \xrightarrow{\varphi_1} & \Sigma'_1 \\
{\scriptstyle \varphi_2} \downarrow & & \downarrow {\scriptstyle \theta_1} \\
\Sigma'_2 & \xrightarrow[\theta_2]{} & \Omega'
\end{array}$$

for any model $M' \in \mathsf{Mod}(\Phi(\Omega'))$, $M' \upharpoonright_{\Phi(\theta_k)} \in \mathsf{Mod}'(\Sigma'_k)$, $k \in \{1, 2\}$, implies $M' \in \mathsf{Mod}'(\Omega')$.

Example 4.4. The structuring of Ex. 3.2 is compositional; let us see how this works. Let the following be a pushout of $\mathcal{I}$-theory morphisms (see Ex. 4.1).

$$\begin{array}{ccc}
(\Sigma, E) & \xrightarrow{\varphi_1} & (\Sigma_1, E_1) \\
{\scriptstyle \varphi_2} \downarrow & & \downarrow {\scriptstyle \theta_1} \\
(\Sigma_2, E_2) & \xrightarrow[\theta_2]{} & (\Sigma', E')
\end{array}$$

Let $M' \in \mathsf{Mod}(\Sigma')$ such that $M' \upharpoonright_{\theta_1} \models E_1$ and $M' \upharpoonright_{\theta_2} \models E_2$. Then by the Satisfaction Condition $M' \models \theta_1(E_1)$ and $M' \models \theta_2(E_2)$. Hence $M' \models \theta_1(E_1) \cup \theta_2(E_2)$. Since E' is the semantic closure of $\theta_1(E_1) \cup \theta_2(E_2)$ (see Ex. 4.1) it follows that $M' \models E'$.

Example 4.5. The structuring of Ex. 3.3 given by the structured specifications that include union (∪) and translation (⋆) building operators among others enjoys the compositionality property as follows. Let the following be a pushout of structured specifications morphisms (see Ex. 4.2).

$$\begin{array}{ccc} \mathrm{SP} & \xrightarrow{\varphi_1} & \mathrm{SP}_1 \\ {\scriptstyle \varphi_2}\downarrow & & \downarrow{\scriptstyle \theta_1} \\ \mathrm{SP}_2 & \xrightarrow[\theta_2]{} & \mathrm{SP}' \end{array}$$

Let $M' \in \mathsf{Mod}(\Phi(\mathrm{SP}'))$ such that $M'\!\upharpoonright_{\theta_k} \in \mathsf{Mod}'(\mathrm{SP}_k)$ for $k \in \{1,2\}$. It follows that $M' \in \mathsf{Mod}'(\mathrm{SP}_k \star \theta_k)$ for $k \in \{1,2\}$ hence $M' \in \mathsf{Mod}'(\mathrm{SP}_1 \star \theta_1 \cup \mathrm{SP}_2 \star \theta_2)$. Since according to Ex. 4.2 it is easy to note that $\mathrm{SP}' \models\!| \mathrm{SP}_1 \star \theta_1 \cup \mathrm{SP}_2 \star \theta_2$ we obtain that $M' \in \mathsf{Mod}'(\mathrm{SP}')$.

Proposition 4.2. *Under the conditions of Prop. 4.1 if the structuring of $\mathcal{I}'$ is compositional then the structuring of the quotient $\mathcal{I}'/_{\equiv}$ is compositional too.*

Proof. Let the following be a pushout of $\mathcal{I}'/_{\equiv}$-signature morphisms

$$\begin{array}{ccc} \Sigma'' & \xrightarrow{\varphi''_1} & \Sigma''_1 \\ {\scriptstyle \varphi''_2}\downarrow & & \downarrow{\scriptstyle \theta''_1} \\ \Sigma''_2 & \xrightarrow[\theta''_2]{} & \Omega'' \end{array}$$

and let $M'' \in \mathsf{Mod}(\Phi/_{\equiv}(\Omega''))$ such that $M''\!\upharpoonright_{\Phi/_{\equiv}(\theta''_k)} \in \mathsf{Mod}''(\Sigma''_k)$ for $k \in \{1,2\}$. From the proof of Prop. 4.1 we have that there exists a pushout of $\mathcal{I}'$-signature morphisms

$$\begin{array}{ccc} \Sigma' & \xrightarrow{\varphi'_1} & \Sigma'_1 \\ {\scriptstyle \varphi'_2}\downarrow & & \downarrow{\scriptstyle \theta'_1} \\ \Sigma'_2 & \xrightarrow[\theta'_2]{} & \Omega' \end{array}$$

that gets mapped by the quotienting functor $\_/_{\equiv}$ to the above pushout of $\mathcal{I}'/_{\equiv}$-signature morphisms. Hence $M'' \in \mathsf{Mod}(\Phi(\Omega'))$ and $M''\!\upharpoonright_{\Phi(\theta'_k)} \in \mathsf{Mod}'(\Sigma'_k)$ for $k \in \{1,2\}$. By the compositionality property for the structuring of $\mathcal{I}'$ we have that $M'' \in \mathsf{Mod}'(\Omega')$. But $\mathsf{Mod}'(\Omega') = \mathsf{Mod}''(\Omega'')$, thus $M'' \in \mathsf{Mod}''(\Omega'')$. □

Example 4.6. By Prop. 4.2 we obtain that the quotients of the institutions of structured specifications modulo module algebra rules (described in Ex. 3.4; see also Ex. 4.3) have the compositionality property of Dfn. 4.2.

Proposition 4.3. *Let $\mathcal{I}'$ be an institution structured over $\mathcal{I}$ through Φ such that*

1. *Φ preserves pushouts, and*
2. *$\mathcal{I}'$ structured over $\mathcal{I}$ through Φ is compositional.*

If $\mathcal{I}$ has model amalgamation (resp. weak model amalgamation, semi-exactness) then $\mathcal{I}'$ has model amalgamation (resp. weak model amalgamation, semi-exactness).

Proof. Let us assume the model amalgamation property for $\mathcal{I}$ and prove it for $\mathcal{I}'$. Let the square below be a pushout square of $\mathcal{I}'$-signature morphisms.

$$\begin{array}{ccc} \Sigma' & \xrightarrow{\varphi_1'} & \Sigma_1' \\ {\scriptstyle \varphi_2'}\downarrow & & \downarrow{\scriptstyle \theta_1'} \\ \Sigma_2' & \xrightarrow[\theta_2']{} & \Omega' \end{array}$$

Let $M_k' \in \mathsf{Mod}'(\Sigma_k')$ for $k \in \{1,2\}$ such that $M_1' \upharpoonright_{\varphi_1'} = M_2' \upharpoonright_{\varphi_2'}$. By the condition on the preservation of pushouts we have that the following square is a pushout of $\mathcal{I}$-signature morphisms.

$$\begin{array}{ccc} \Phi(\Sigma') & \xrightarrow{\Phi(\varphi_1')} & \Phi(\Sigma_1') \\ {\scriptstyle \Phi(\varphi_2')}\downarrow & & \downarrow{\scriptstyle \Phi(\theta_1')} \\ \Phi(\Sigma_2') & \xrightarrow[\Phi(\theta_2')]{} & \Phi(\Omega') \end{array}$$

By the model amalgamation property of the institution there exists an unique amalgamation $M' \in \mathsf{Mod}(\Phi(\Omega'))$ of M_1' and M_2'. Since $M_k' \in \mathsf{Mod}'(\Sigma_k')$ for $k \in \{1,2\}$ by the compositionality condition we obtain that $M' \in \mathsf{Mod}'(\Omega')$. The uniqueness of the amalgamation in $\mathcal{I}'$ follows directly from the corresponding property in $\mathcal{I}$.

Similar arguments may be employed for establishing the weak model amalgamation and semi-exactness properties, resp. □

Example 4.7. As corollaries to Prop. 4.3 we obtain that the institutions of the theories (Ex. 3.2), of the structured specifications (Ex. 3.3), and of the structured specifications modulo module algebra rules (Ex. 3.4) enjoy the amalgamation properties of the base institution.

5. Normal Forms and their consequences

In this section we introduce a concept of normal form for abstract structured specifications that captures abstractly the normal forms from the model oriented approach to structured specifications (see [2, 5, 9]). We show that in the presence of normal forms it is possible to lift a series of important logical properties from the base institution to the upper institution of the abstract structured specifications. These properties well known for their relevance to specification, include compactness, and interpolation. Another important property studied is the closure of (the class of) models of an abstract structured specification under isomorphisms. Moreover, like in the work [5], we use normal forms for lifting a sound and complete proof system from the base institution to the institution of the abstract structured specifications (however this is done differently from [5]).

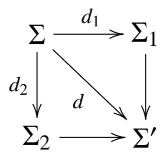
5.1. Normal forms

Definition 5.1 (Normal form). *Given an institution $\mathcal{I}'$ structured over $\mathcal{I}$ through Φ and a class $\mathcal{D}$ of $\mathcal{I}$-signature morphisms, a pair (φ, E) where $(\varphi\colon \Phi(\Sigma') \to \Sigma) \in \mathcal{D}$ and $E \subseteq \mathsf{Sen}(\Sigma)$ is a $\mathcal{D}$-*normal form *for an $\mathcal{I}'$-signature Σ' when $\mathsf{Mod}'(\Sigma') = \mathsf{Mod}(\Sigma, E)\upharpoonright_{\varphi}$. When E is finite we say that the normal form is* finitary. *We say that $\mathcal{I}'$ admits (finitary) $\mathcal{D}$-normal forms when each $\mathcal{I}'$-signature has at least a (finitary) $\mathcal{D}$-normal form.*

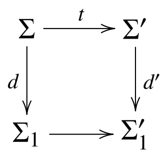
Example 5.1. The institution $\mathcal{I}^{\text{th}}$ of theories over an institution $\mathcal{I}$ (Ex. 3.2) trivially has $\mathcal{D}$-normal forms for any $\mathcal{D}$ that contains the identities.

Example 5.2. The institution of the $(\mathcal{T}, \mathcal{D})$-structured specifications built over an institution $\mathcal{I}$ that has model amalgamation (Ex. 3.3) has finitary $\mathcal{D}$-normal forms when

1. $\mathcal{D}$ is a broad subcategory of Sig (the category of the $\mathcal{I}$-signatures),
2. for each $(d_1 : \Sigma \to \Sigma_1) \in \mathcal{D}$ and $(d_2 : \Sigma \to \Sigma_2) \in \mathcal{D}$ there exists a pushout square as below such that $d \in \mathcal{D}$



3. for each $(t\colon \Sigma \to \Sigma') \in \mathcal{T}$ and $(d\colon \Sigma \to \Sigma_1) \in \mathcal{D}$ there exists a pushout square as below such that $d' \in \mathcal{D}$



The proof of this result can be found in the literature [14, 38]. Note that this result depends upon the choice of the specific set of specification building operators of Ex. 3.3, for example if we added an initial semantics operator the normal form property is lost.

The normal form property transfers rather straightforwardly from the above example of the $(\mathcal{T}, \mathcal{D})$-structured specifications to any of its quotients described in Ex. 3.4.

The following is an important technical property that comes with the existence of normal forms and that will be used several times in the proofs of results of this section.

Proposition 5.1. *Let $(\varphi\colon\ \Phi(\Sigma') \to \Sigma, E)$ be a $\mathcal{D}$-normal form for an $\mathcal{I}'$-signature Σ'. Then for each set $\Gamma' \subseteq \mathsf{Sen}'(\Sigma')$ and $\rho \in \mathsf{Sen}'(\Sigma')$*

$$\Gamma' \models'_{\Sigma'} \rho \ \text{ if and only if } \ E \cup \varphi(\Gamma') \models_{\Sigma} \varphi(\rho).$$

Proof. For the implication from the left to the right let us consider a Σ-model M such that $M \models_\Sigma E \cup \varphi(\Gamma')$. By the Satisfaction Condition for $\mathcal{I}$ it follows that $M{\upharpoonright_\varphi} \models_{\Phi(\Sigma')} \Gamma'$. But $M \models_\Sigma E$ means that $M{\upharpoonright_\varphi} \in \mathsf{Mod}'(\Sigma')$ hence $M{\upharpoonright_\varphi} \models'_{\Sigma'} \Gamma'$. Since (by the hypothesis) $\Gamma' \models'_{\Sigma'} \rho$ it follows that $M{\upharpoonright_\varphi} \models'_{\Sigma'} \rho$ which means $M{\upharpoonright_\varphi} \models_{\Phi(\Sigma')} \rho$. By the Satisfaction Condition for $\mathcal{I}$ it follows that $M \models_\Sigma \varphi(\rho)$.

For the implication from the right to the left we let $M' \in \mathsf{Mod}'(\Sigma')$ such that $M' \models'_{\Sigma'} \Gamma'$ which means $M' \models_{\Phi(\Sigma')} \Gamma'$. Since $\mathsf{Mod}'(\Sigma') = \mathsf{Mod}(\Sigma, E){\upharpoonright_\varphi}$ there exists $M \in \mathsf{Mod}(\Sigma, E)$ such that $M{\upharpoonright_\varphi} = M'$. By the Satisfaction Condition for $\mathcal{I}$ it follows that $M \models_\Sigma \varphi(\Gamma')$. Since $M \models_\Sigma E$ it follows that $M \models_\Sigma E \cup \varphi(\Gamma')$. By the hypothesis it now follows that $M \models_\Sigma \varphi(\rho)$. By the Satisfaction Condition for $\mathcal{I}$ we obtain $M' \models_{\Phi(\Sigma')} \rho$ which means $M' \models'_{\Sigma'} \rho$. □

Definition 5.2. *A signature morphism $\varphi\colon\ \Sigma \to \Sigma'$ in an institution* lifts isomorphisms *(of models) if and only if for any two isomorphic Σ-models $M \cong N$ and any φ-expansion M' of M there exists a φ-expansion N' of N such that $M' \cong N'$.*

Example 5.3. It is known from the literature (e.g. [14]) that in *MSA* a signature morphism lifts isomorphisms if and only if it is injective on the sorts. Moreover, this property holds as well for other many sorted logical systems.

Proposition 5.2. *Let $\mathcal{I}'$ be an institution structured over $\mathcal{I}$ through Φ such that $\mathcal{I}'$ admits $\mathcal{D}$-normal forms for some class $\mathcal{D}$ of $\mathcal{I}$-signature morphisms. If each morphism from $\mathcal{D}$ lifts isomorphisms then for each $\mathcal{I}'$-signature Σ' we have that $\mathsf{Mod}'(\Sigma')$ is closed under ($\mathcal{I}$-model) isomorphisms.*

Proof. Let $M' \in \mathsf{Mod}'(\Sigma)$ and let $N' \in \mathsf{Mod}(\Phi(\Sigma'))$ such that $M' \cong N'$. We have to prove that $N' \in \mathsf{Mod}'(\Sigma)$.

By the normal form assumption there exists $(\varphi\colon\ \Phi(\Sigma') \to \Sigma) \in \mathcal{D}$ such that $\mathsf{Mod}'(\Sigma') = \mathsf{Mod}(\Sigma, E)\restriction_\varphi$. Let $M \in \mathsf{Mod}(\Sigma, E)$ such that $M\restriction_\varphi = M'$. By the lifting of isomorphisms assumption there exists a Σ-model N such that $N\restriction_\varphi = N'$ and $N \cong M$. Since in all our institutions isomorphisms of models preserve the satisfaction relation we have that $N \models E$, hence $N \in \mathsf{Mod}(\Sigma, E)$. It follows that $N' = N\restriction_\varphi \in \mathsf{Mod}(\Sigma, E)\restriction_\varphi = \mathsf{Mod}'(\Sigma')$. □

5.2. Compactness

Proposition 5.3. *Let $\mathcal{I}'$ be an institution structured over $\mathcal{I}$ through Φ such that $\mathcal{I}'$ admits $\mathcal{D}$-normal forms for some class $\mathcal{D}$ of $\mathcal{I}$-signature morphisms. If $\mathcal{I}$ is compact then $\mathcal{I}'$ is compact too.*

Proof. Let Σ' be any $\mathcal{I}'$-signature and let $E' \models'_{\Sigma'} \rho$ for $E' \subseteq \mathsf{Sen}'(\Sigma')$ and $\rho \in \mathsf{Sen}'(\Sigma')$. We have to show that there exists finite $E'_0 \subseteq E'$ such that $E'_0 \models'_{\Sigma'} \rho$. By the normal form condition there exists $(\varphi\colon\ \Phi(\Sigma') \to \Sigma) \in \mathcal{D}$ and $E \subseteq \mathsf{Sen}(\Sigma)$ such that $\mathsf{Mod}'(\Sigma') = \mathsf{Mod}(\Sigma, E)\restriction_\varphi$. By Prop. 5.1 we have

$$E \cup \varphi(E') \models_\Sigma \varphi(\rho). \tag{3}$$

By the compactness assumption on $\mathcal{I}$, from (3) there exists $E'_0 \subseteq E'$ and $E_0 \subseteq E$, both sets finite, such that

$$E_0 \cup \varphi(E'_0) \models_\Sigma \varphi(\rho). \tag{4}$$

By Prop. 5.1 the relation (4) implies $E'_0 \models'_{\Sigma'} \rho$. □

Example 5.4. Applications of the general compactness result given by Prop. 5.3 include institutions of theories $\mathcal{I}^{\mathrm{th}}$ (cf. Ex. 5.1) and institutions of structured specifications and their quotients (cf. Ex. 5.2).

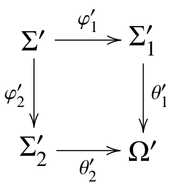
5.3. Interpolation

Theorem 5.1. *Let $\mathcal{I}'$ be an institution structured over $\mathcal{I}$ through Φ and $\mathcal{L}'$ and $\mathcal{R}'$ classes of signature morphisms such that*

1. *Φ preserves pushouts,*
2. *the structuring of $\mathcal{I}'$ is compositional,*
3. *$\mathcal{I}'$ admits $\mathcal{D}$-normal forms for some class $\mathcal{D}$ of $\mathcal{I}$-signature morphisms,*
4. *$\mathcal{I}$ has Craig-Robinson $(\mathcal{L}, \mathcal{R})$-interpolation, and*
5. *$\Phi(\mathcal{L}'); \mathcal{D} \subseteq \mathcal{L}$ and $\Phi(\mathcal{R}'); \mathcal{D} \subseteq \mathcal{R}$.*

Then $\mathcal{I}'$ has Craig-Robinson $(\mathcal{L}', \mathcal{R}')$-interpolation.

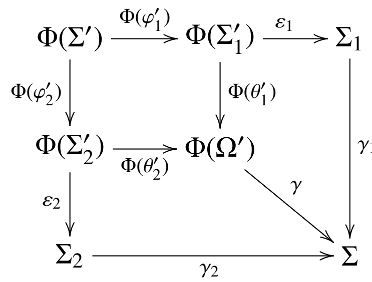
Proof. Let us consider a pushout of $\mathcal{I}'$-signature morphisms with $\varphi'_1 \in \mathcal{L}'$ and $\varphi'_2 \in \mathcal{R}'$ as follows



and $E'_1 \subseteq \mathsf{Sen}'(\Sigma'_1)$ and $E'_2, \Gamma'_2 \subseteq \mathsf{Sen}'(\Sigma'_2)$ such that

$$\theta'_1(E'_1) \cup \theta'_2(\Gamma'_2) \models'_{\Omega'} \theta'_2(E'_2). \tag{5}$$

By the normal forms assumption let $(\varepsilon_k : \Phi(\Sigma'_k) \to \Sigma_k) \in \mathcal{D}$ and $E_k \subseteq \mathsf{Sen}(\Sigma_k)$ for each $k \in \{1, 2\}$ such that $\mathsf{Mod}'(\Sigma'_k) = \mathsf{Mod}(\Sigma_k, E_k)\restriction_{\varepsilon_k}$ for each $k \in \{1, 2\}$. In the commutative diagram below we let the outer square to be a pushout and we let γ to be the unique signature morphism which makes the diagram commutative (γ exists because the left-upper corner square of the diagram is a pushout by the assumption that Φ preserves pushouts).



Let us show that

$$\gamma_1(E_1) \cup \gamma_2(E_2) \cup \gamma(\Phi(\theta'_1)(E'_1)) \cup \gamma(\Phi(\theta'_2)(\Gamma'_2)) \models_\Sigma \gamma(\Phi(\theta'_2)(E'_2)). \tag{6}$$

For this we consider a Σ-model M such that

$$M \models_\Sigma \gamma_1(E_1) \cup \gamma_2(E_2) \cup \gamma(\Phi(\theta'_1)(E'_1)) \cup \gamma(\Phi(\theta'_2)(\Gamma'_2)). \tag{7}$$

For each $k \in \{1, 2\}$ by the Satisfaction Condition of $\mathcal{I}$ it follows that $M\restriction_{\gamma_k} \models_{\Sigma_k} E_k$ which means that $M\restriction_{\gamma_k}\restriction_{\varepsilon_k} \in \mathsf{Mod}'(\Sigma'_k)$. Since $\varepsilon_k; \gamma_k = \Phi(\theta'_k); \gamma$ it follows that $M\restriction_\gamma\restriction_{\Phi(\theta'_k)} \in \mathsf{Mod}'(\Sigma'_k)$ which by the compositionality assumption on $\mathcal{I}'$ implies $M\restriction_\gamma \in \mathsf{Mod}'(\Omega')$. By the Satisfaction Condition on $\mathcal{I}$ from (7) it follows that $M\restriction_\gamma \models_{\Phi(\Omega')} \Phi(\theta'_1)(E'_1) \cup \Phi(\theta'_2)(\Gamma'_2)$ which because $M\restriction_\gamma \in \mathsf{Mod}'(\Omega')$ means $M\restriction_\gamma \models'_{\Omega'} \theta'_1(E'_1) \cup \theta'_2(\Gamma'_2)$. From (5) it follows that $M\restriction_\gamma \models'_{\Omega'} \theta'_2(E'_2)$ which means $M\restriction_\gamma \models_{\Phi(\Omega')} \Phi(\theta'_2)(E'_2)$. By the Satisfaction Condition on $\mathcal{I}$ this is equivalent to $M \models_\Sigma \gamma(\Phi(\theta'_2)(E'_2))$.

Note that (6) means

$$\gamma_1(E_1) \cup \gamma_2(E_2) \cup \gamma_1(\varepsilon_1(E'_1)) \cup \gamma_2(\varepsilon_2(\Gamma'_2)) \models_\Sigma \gamma_2(\varepsilon_2(E'_2)). \tag{8}$$

Note also that by the hypothesis that $\Phi(\mathcal{L}'); \mathcal{D} \subseteq \mathcal{L}$ and $\Phi(\mathcal{R}'); \mathcal{D} \subseteq \mathcal{R}$ it follows that $\Phi(\varphi'_1); \varepsilon_1 \in \mathcal{L}$ and $\Phi(\varphi'_2); \varepsilon_2 \in \mathcal{R}$. Since $\mathcal{I}$ has Craig-Robinson $(\mathcal{L}, \mathcal{R})$-interpolation there exists $E' \subseteq \mathsf{Sen}(\Phi(\Sigma'))$ such that

$$E_1 \cup \varepsilon_1(E'_1) \models_{\Sigma_1} \varepsilon_1(\Phi(\varphi'_1)(E')) \text{ and } E_2 \cup \varepsilon_2(\Gamma'_2) \cup \varepsilon_2(\Phi(\varphi'_2)(E')) \models_{\Sigma_2} \varepsilon_2(E'_2). \tag{9}$$

Now by Prop. 5.1 the relations (9) imply

$$E'_1 \models'_{\Sigma'_1} \varphi'_1(E') \text{ and } \varphi'_2(E') \cup \Gamma_2 \models'_{\Sigma'_2} E'_2. \tag{10}$$

which show the Craig-Robinson $(\mathcal{L}', \mathcal{R}')$-interpolation property for $\mathcal{I}'$. □

Example 5.5. Let $\mathcal{T}$ be the class of all *MSA* signature morphisms, $\mathcal{D}$ the class of those that are injective on the sorts, and $\mathcal{D}'$ the subclass of $\mathcal{D}$ of those morphisms for which no operation symbol outside the image of the signature morphism is allowed to have the sort in the image of the signature morphism (see Ex. 2.3). Then we consider the $(\mathcal{T}, \mathcal{D})$-structured specifications (see Ex. 3.3) over *MSA* as base institution and the $(\mathcal{T}, \mathcal{D}')$-structured specifications over the Horn clause sub-institution of *MSA*. Then in both situations we have the following:

– From Ex. 4.2 we have that Φ preserves pushouts.

– From Ex. 4.5 we have the required compositionality property for the structuring.

– From Ex. 5.2 we know that $\mathcal{I}'$ admits $\mathcal{D}$-normal forms.

Then from Thm. 5.1, through the interpolation results recalled in Ex. 2.2 and 2.3 we obtain the following interpolation results for our examples:

1. The $(\mathcal{T}, \mathcal{D})$-structured specifications over *MSA* have the Craig-Robinson $(\mathcal{T}, \mathcal{D})$ and $(\mathcal{D}, \mathcal{T})$-interpolation.
2. The $(\mathcal{T}, \mathcal{D}')$-structured specifications over the Horn cause sub-institution of *MSA* have the Craig-Robinson $(\mathcal{D}', \mathcal{T})$-interpolation.

5.4. Proof theory

Definition 5.3. *Let $\mathcal{I}'$ be an institution structured over $\mathcal{I}$ through Φ and let $\mathcal{D}$ be a designated class of $\mathcal{I}$-signature morphisms. We let $\vdash'$ be the least entailment system for $\mathcal{I}'$ such that for each $\mathcal{I}'$-signature Σ', for each of its $\mathcal{D}$-normal forms $(\varphi\colon \Phi(\Sigma') \to \Sigma, E)$, for each $E' \subseteq \mathsf{Sen}'(\Sigma')$ and each $\rho \in \mathsf{Sen}'(\Sigma')$*

$$E' \vdash'_{\Sigma'} \rho \ \text{ if } \ E \cup \varphi(E') \models_{\Sigma} \varphi(\rho).$$

The following is an immediate consequence of Prop. 5.1.

Corollary 5.1. $(\mathcal{I}', \vdash')$ *is sound. Moreover, if $\mathcal{I}'$ admits $\mathcal{D}$-normal forms then $(\mathcal{I}', \vdash')$ is complete too.*

Dfn. 5.3 together with Cor. 5.1 constitute the basis for a rather simple lifting of a sound and complete proof theory from the base institution $\mathcal{I}$ to the abstract structured specifications (the institution $\mathcal{I}'$). This goes as follows. Assuming that $\mathcal{I}'$ admits $\mathcal{D}$-normal forms, if we are interested to prove that ρ is a property of an abstract structured specification Σ', i.e. that

$$\models'_{\Sigma'} \rho$$

then we have to do the following:

1. Compute a $\mathcal{D}$-normal form $(\varphi\colon \Phi(\Sigma') \to \Sigma, E)$ for Σ'. For example, for the $(\mathcal{T}, \mathcal{D})$-structured specifications of Ex. 3.3, the literature (e.g. [14]) gives a simple algorithm for this, the result being a finitary $\mathcal{D}$-normal form.
2. Prove

   $$E \models_{\Sigma} \varphi(\rho)$$

   by using a sound and complete proof theory of the base institution.

Note that the computed normal form may be any since according to Prop. 5.1 any normal form has the same effect. The important thing here is to have at least one normal form. This procedure corresponds to (some of the) actual formal verification practices, for example implementations of the OBJ family of languages (e.g. CafeOBJ [16]) compute tacitly such normal forms as flattenings of actual loose semantics modules. When performing formal verifications, the users of these languages often invoke the `open` command which (among other things) makes available for the proof process the set E of sentences of the normal form. In such methodologies the reuse of proofs comes in forms of lemmas, which may be properties proved for component parts of the specification and which are being brought to the actual context via the 'translation' property of entailment systems.

The core verification methodology for structured specifications discussed here is simpler than that emerging from the fundamental work of [5], for example it does not require interpolation. The drawback here is that the correspondence between the modular structure of proofs and that of specifications is lost. The existence of simple proof systems via normal forms has been known in the literature and is explicitly stated in [38]. Note though that all these rely upon a common important requirement: the existence of normal forms, which is explicit in our approach and in [5] and implicit in [31].

6. Pushout-style Parameterization with Sharing

Pushout-style parameterization originate from work on Clear [7] and constitutes the basis of parameterized specification for the whole OBJ family of languages (i.e. OBJ3 [24], CafeOBJ [16], etc.) but also for ACT TWO [19] and other languages. In [15] we have developed a semantics for pushout-style parameterization that refines the existing one by considering the possible sharing between the body of the parameterized module and the instance of the parameter, a situation that is more realistic in practice than the current approaches based upon an assumption of non-sharing. That have been done within the framework of concretely structured specifications á la [37], meaning one level of institution independence, i.e. only for the underlying logical system. Here we upgrade the concept of parameterization from [15] to our axiomatic framework of abstract structured specifications. This upgrade is non-trivial as it involves several novel technical developments required by the higher level of abstraction, the end result being a highly general and definitive theory of pushout-style parameterization that covers a wide range of concrete specification frameworks.

The section is structured as follows:

1. It starts with a brief survey of some necessary concepts and results about inclusion systems.
2. It introduces the concepts of parameterized module and of instantiation of parameters.
3. Under some technical conditions it develops an alternative definition for the instantiation of parameters.

6.1. Inclusion systems

Inclusion systems were introduced in [17] 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 [14, 17, 23] but also for developing axiomatizability [11, 14, 35] and definability [1] results within the framework of the so-called 'institution-independent model theory' [14]. Inclusion systems capture categorically the concept of set-theoretic inclusion in a way reminiscent of how the rather notorious concept of factorization system [3] captures categorically the set-theoretic injections; however in many applications the former are more convenient than the latter. Here we first recall from the literature the basics of the theory of inclusion systems together with a series of new concepts and results developed recently [15] and needed here.

The definition below can be found in the recent literature on inclusion systems (e.g. [14]) and differs slightly from the original one of [17].

Definition 6.1 (Inclusion systems). *$\langle \mathcal{I}, \mathcal{E} \rangle$ is an* inclusion system *for a category $\mathbb{C}$ if $\mathcal{I}$ and $\mathcal{E}$ are two broad sub-categories 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 a domain of f. An inclusion $i : A \to B$ may be also denoted simply by $A \subseteq B$.*

The inclusion system

- *is* epic *when all abstract surjections are epis,*
- *has* unions *when $\mathcal{I}$ has finite least upper bounds (denoted $\cup$),*
- *has* intersections *when $\mathcal{I}$ has greatest lower bounds (denoted $\cap$), and*
- *is* distributive *when it has unions and intersections that satisfy the usual distributivity rules.*

In [10] it is shown that the class $\mathcal{I}$ of the abstract inclusions determines the class $\mathcal{E}$ of the abstract surjections. In this sense, [10] gives an explicit equivalent definition of inclusion systems which uses only the class $\mathcal{I}$ of the abstract inclusions. In [17] it has been shown that whenever a category with an inclusion system has pullbacks the existence of unions implies the existence of the intersections that are obtained as the pullback of the union.

$$\begin{array}{ccc} A \cap B & \xrightarrow{\subseteq} & A \\ {\scriptstyle\subseteq}\downarrow & & \downarrow{\scriptstyle\subseteq} \\ B & \xrightarrow[\subseteq]{} & A \cup B \end{array}$$

It is often useful that the intersection-union squares are not only pullbacks, but they are also pushouts. Although this property is widely spread among inclusion systems of interest, it does not hold in general and therefore at the level of abstract inclusion systems it has to be assumed when necessary.

The standard example of inclusion system is that from **Set**, with set theoretic inclusions in the role of the abstract inclusions and the surjective functions in the role of the abstract surjections. It is easy to note that this has all properties introduced by Dfn. 6.1 above. The literature contains many examples of inclusion systems for the categories of the signatures and for the categories of models of various institutions from logic or from specification theory. Due to lack of space let us here recall only a couple of them.

Example 6.1 (Inclusion systems for *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 the *MSA* signatures admits also the following non-trivial inclusion systems:

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

Note that the strong inclusion systems for the *MSA* signatures is epic and distributive while the closed one has none of these properties. Therefore, the inclusion system for *MSA* signatures that is relevant for specification is the strong one.

The following abstract concept that captures a rather common situation in practice, including of course *MSA*, has been introduced in [17].

Definition 6.2 (Inclusive institutions). *An institution is inclusive when its category of signatures is endowed with an inclusion system such that whenever* $\Sigma \subseteq \Sigma'$ *we have* $\mathsf{Sen}(\Sigma) \subseteq \mathsf{Sen}(\Sigma')$.

In the following we recall some concepts and results about inclusion systems that are necessary for our work here and that have been developed in [15].

Definition 6.3 (Disjoint objects). *In a category with pullbacks and a designated inclusion system we say that two objects A and B are* disjoint *if and only if the intersection-union square*

$$\begin{array}{ccc} A \cap B & \xrightarrow{\subseteq} & A \\ {\scriptstyle\subseteq}\downarrow & & \downarrow{\scriptstyle\subseteq} \\ B & \xrightarrow[\subseteq]{} & A \cup B \end{array}$$

is pushout and $A \cap B$ is an initial object in the category.

Example 6.2. Note that disjoint objects in **Set** just mean ordinary disjoint sets, while two signatures (S_1, F_1) and (S_2, F_2) are disjoint (with respect to the strong inclusion system for the *MSA* signatures) if and only if $S_1 \cap S_2 = \emptyset$. If we considered *single sorted* signatures then disjointness of signatures F_1 and F_2 means $(F_1)_n \cap (F_2)_n = \emptyset$ for each arity $n \in \omega$.

Corollary 6.1. *If A and B are disjoint then $A \cup B$ is the coproduct of A and B.*

Proposition 6.1. *If $B' \subseteq B$ and A and B are disjoint, then A and B' are disjoint too.*

Proposition 6.2. *In a category with an epic inclusion system we consider a pushout square as below*

$$\begin{array}{ccc} A & \xrightarrow{\subseteq} & B \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ A & \xrightarrow[\subseteq]{} & B \end{array}$$

such that $f; f = f$. Let $f = e_f; (f(A) \subseteq A)$ and $g = e_g; (g(B) \subseteq B)$ with e_f, e_g abstract surjections. Then $f(A) \subseteq g(B)$ and the commutative squares below are pushout squares

$$\begin{array}{ccccc} A & \xrightarrow{e_f} & f(A) & \xrightarrow{\subseteq} & A \\ {\scriptstyle\subseteq}\downarrow & & \downarrow{\scriptstyle\subseteq} & & \downarrow{\scriptstyle\subseteq} \\ B & \xrightarrow[e_g]{} & g(B) & \xrightarrow[\subseteq]{} & B \end{array}$$

Definition 6.4 (Preservation of objects). *In any category endowed with an inclusion system with intersections we say that an arrow $f\colon A \to B$* preserves *an object C when $(A \cap C \subseteq A); f$ is an inclusion.*

Definition 6.5 (Free extensions along inclusions). *In any category endowed with an inclusion system with signatures we say that an arrow $f\colon A \to A_1$* admits free extensions along an inclusion *$A \subseteq A'$ when there exist pushout squares of the form*

$$\begin{array}{ccc} A & \xrightarrow{\subseteq} & A' \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f'} \\ A_1 & \xrightarrow[\subseteq]{} & A'_1 \end{array}$$

such that every object preserved by f is also preserved by f'.

The following is an important example from [15].

Proposition 6.3. *In MSA every signature morphism $\varphi\colon (S, F) \to (S, F)$ admits free extensions along any inclusion of signatures $(S, F) \subseteq (S', F')$.*

Definition 6.6 (Idempotent-by-extension). *In a category with pullbacks and endowed with an epic inclusion system, an arrow $f\colon A \to A$ is called* idempotent-by-extension *when it is idempotent, i.e. $f; f = f$, and there exists an object B such that $A = B \cup f(A)$ and B and $f(A)$ are disjoint.*

6.2. Parameters and their instantiations

Definition 6.7 (Parameterized $\mathcal{I}'$-signatures). *Let $\mathcal{I}'$ be an institution structured over an inclusive institution $\mathcal{I}$ through Φ. A* parameterized $\mathcal{I}'$-signature*, denoted $\Sigma'(\iota)$, consists of an $\mathcal{I}'$-signature morphism $\iota\colon P \to \Sigma'$ such that $\Phi(\iota)$ is inclusion $\Phi(P) \subseteq \Phi(\Sigma')$. Then P is called the* parameter *of the $\mathcal{I}'$-signature and Σ' the* body *of the parameterized $\mathcal{I}'$-signature.*

In practice, the parameter P is an (isomorphic) renaming of a specification P_0 such that $\Phi(P_0)$ and $\Phi(P)$ are disjoint. If we denote by p the corresponding isomorphism $\Phi(P_0) \to \Phi(P)$, then under the notations from Ex. 3.3 $P = P_0 \star p$. The readers familiar with the OBJ family of languages may find that our $\Sigma'(\iota)$ here corresponds there to $\Sigma'(p :: P_0)$. The reason for such isomorphic renamings is that while usually we specify P_0, we also need to make sure the parameter does not share with other parts of our specifications, such as other parameters or specifications used for instantiations. A practical way to achieve this, which is realized in some implementations of actual specification languages, is to rename the entities of P_0 by qualifying them by p. For example a sort s of P_0 would appear in P as $s.p$. This ideology about what is a parameterized specification module has been explicitly defined also in [23] within the context of the theory oriented approach.

In the literature (e.g. [38]) parameterized specifications are sometimes defined just as specification morphisms $P \to \Sigma'$. We think that this is much too general and does not capture precisely enough the realities of parameterized specifications (especially does not support considering sharing), our additional condition that $\Phi(P) \subseteq \Phi(\Sigma')$ filling this conceptual gap. Below we will see that one of the consequences of our inclusion systems based approach is the possibility to consider sharing in a rather natural and clean way.

Example 6.3. Let us consider the extension of the $(\mathcal{T}, \mathcal{D})$-structured specifications over *MSA* (see Ex. 3.3) with another specification building operator for initial semantics, called 'free' in [14, 15, 38] (we skip here the details of this operator). Let us fix $\mathcal{T}$ to be the class of all signature morphisms and $\mathcal{D}$ the class of the identities (meaning that we actually eliminate the building operator $\_ \mid \_$). We use the CafeOBJ [16] notation for writing our specifications.

Below, in the parameterized specification of semigroups 'with powers', namely `SG^`, the parameter consists of the renaming of the specification `SG` of semigroups by `S`. In the CafeOBJ notation this is denoted `(S :: SG)`.

```
mod* SG {
  [ Elt ]
  op _+_ : Elt Elt -> Elt { assoc }
}
mod! PNAT {
  [ PNat ]
  op 0 : -> PNat
  op s_ : PNat -> PNat
}
mod* SG^ (S :: SG) {
  protecting(PNAT)
  op _^_ : Elt PNat -> Elt
  eq E:Elt ^ s(N:PNat) = E + (E ^ N) .
}
```

In the parameterized specification `SG^`, the sort of `SG` ⋆ `S` is `Elt.S`. In this example the specification `SG^` is defined as $(\texttt{SG} \star \texttt{S}) \cup \texttt{PNAT} \cup (\Sigma', E')$ where Σ' is $\Phi(\texttt{SG} \star \texttt{S}) \cup \Phi(\texttt{PNAT})$ (meaning the union of the *MSA* signatures in the strong inclusion system) plus the operation $\_\hat{}\_$ and E' consists of the only equation specified by `SG^`. The modules `SG` and `SG^` are specified with loose semantics (`mod*`) while `PNAT` is specified with initial semantics (`mod!`). This means the denotation of `SG` consists of all semigroups and the denotation of `PNAT` consists of the models that are isomorphic to the Peano model of the natural numbers. Also the denotation of `SG^` consists of the amalgamation of the semigroups with the Peano model of the natural numbers and with the models that satisfy E'.

Definition 6.8 (Instantiation of parameters). *Let $\mathcal{I}'$ be an institution structured over an inclusive institution $\mathcal{I}$ through Φ such that*

1. Φ *lifts co-products,*
2. Φ *has a left adjoint $\overline{\Phi}$ such that the units of the adjunctions are identities, and*
3. *the inclusion system of $\mathcal{I}$-signatures has unions and intersections.*

Given a parameterized $\mathcal{I}'$-signature $\iota\colon P \to \Sigma'$ and an $\mathcal{I}'$-signature morphism $v\colon P \to \Sigma'_1$ such that $\Phi(P)$ and $\Phi(\Sigma'_1)$ are disjoint an instance $\Sigma'(\iota \Leftarrow v)$ of $\Sigma'(\iota)$ through v *is defined as a pushout of $\mathcal{I}'$-signature morphisms as follows:*

$$\begin{array}{ccc} P + (\Sigma' \Cap \Sigma'_1) & \xrightarrow{\iota + i} & \Sigma' \\ {\scriptstyle v+i_1}\downarrow & & \downarrow \\ \Sigma'_1 & \longrightarrow & \Sigma'(\iota \Leftarrow v) \end{array}$$

where

- $\Sigma' \Cap \Sigma'_1$ *denotes* $\overline{\Phi}(\Phi(\Sigma') \cap \Phi(\Sigma'_1))$,
- $P+(\Sigma' \Cap \Sigma'_1)$ *is a co-product of P and $\Sigma' \Cap \Sigma'_1$ obtained as a lifting of the disjoint union* $\Phi(P) \cup (\Phi(\Sigma') \cap \Phi(\Sigma'_1))$ *(that this is a disjoint union follows by Prop. 6.1; on the other hand Cor. 6.1 gives that this disjoint union is co-product of $\mathcal{I}$-signatures),*
- $i\colon \Sigma' \Cap \Sigma'_1 \to \Sigma'$ *and* $i_1\colon \Sigma' \Cap \Sigma'_1 \to \Sigma'_1$, *resp., denote the $\mathcal{I}'$-signature morphisms* $\overline{\Phi}(\Phi(\Sigma') \cap \Phi(\Sigma'_1) \subseteq \Phi(\Sigma')); \epsilon_{\Sigma'}$ *and* $\overline{\Phi}(\Phi(\Sigma') \cap \Phi(\Sigma'_1) \subseteq \Phi(\Sigma'_1)); \epsilon_{\Sigma'_1}$, *resp., where ϵ denotes the co-unit of the adjunction between $\mathcal{I}$-signatures and $\mathcal{I}'$-signatures, and*
- $\iota + i$ *and* $v + i_1$, *resp., are the corresponding unique morphisms given by the co-product property of* $P + (\Sigma' \Cap \Sigma'_1)$.

In the applications the existence of instances of parameterized $\mathcal{I}'$-signatures may be guaranteed by the existence of pushouts of $\mathcal{I}$-signatures and by the lifting of pushouts by Φ. The uniqueness of co-limits up to isomorphisms imply that instances of parameterized $\mathcal{I}'$-signatures are also unique up to isomorphisms. The condition about $\overline{\Phi}$ of Dfn. 6.8 holds naturally in the applications as suggested by the example below.

Example 6.4. For the example of theories (Ex. 3.2) the left adjoint $\overline{\Phi}$ maps any $\mathcal{I}$-signature Σ to the theory of the Σ-tautologies, i.e. $(\Sigma, \emptyset^\bullet)$.

For the example of the $(\mathcal{T}, \mathcal{D})$-structured specifications (Ex. 3.3) $\overline{\Phi}$ maps any $\mathcal{I}$-signature Σ to the empty presentation $(\Sigma, \emptyset)$. This may be also extended to Ex. 3.4 of quotients of $(\mathcal{T}, \mathcal{D})$-structured specifications.

In the actual situations when P is the renaming via an isomorphism p of another specification P_0 (i.e. $P = P_0 \star p$) we specify a specification morphism $v_0\colon P_0 \to \Sigma'_1$, usually called *view* in the literature. In this case of course the specification morphism v above is just $p^{-1}; v_0$ and the result $\Sigma'(\iota \Leftarrow v)$ of the instantiation may be denoted by $\Sigma'(p \Leftarrow v_0)$; a convention that is used by the OBJ family of languages.

Example 6.5. In the continuation of Ex. 6.3 let us obtain the multiplication of the natural numbers from the addition of the natural numbers by instantiating `SG^` by the signature morphism `pnat-as-sg`. Below is the CafeOBJ code for this.

```
mod! PNAT+ {
  protecting(PNAT)
  op _+_ : PNat PNat -> PNat
  vars M N : PNat
  eq M + 0 = M .
  eq M + s(N) = s(M + N) .
```

```
}
view pnat-as-sg from SG to PNAT+ {
  sort Elt -> PNat,
  op _+_ -> _+_
}
mod* PNAT* {
  protecting(SG^ (S <= pnat-as-sg) * {op _^_ -> _*_})
  eq M:PNat * 0 = 0 .
}
```

First let us note that for the case of our $(\mathcal{T},\mathcal{D})$-structured specifications over *MSA* all technical conditions of Dfn. 6.8 may be checked quite easily:

- that the category of the *MSA* signatures has all finite co-limits is well known from the literature (e.g. [14, 42]) and Ex. 6.2 shows how these are lifted to co-limits of $(\mathcal{T},\mathcal{D})$-structured specifications,
- we have already noted in Ex. 6.1 that the strong inclusion system for the *MSA* signatures has unions and intersections, and
- Ex. 6.4 gives the left adjoint $\overline{\Phi}$.

Our example gets fitted to the notations of Dfn. 6.8 as follows:

- P is $\texttt{SG} \star \texttt{S}$,
- Σ' is `SG^` and Σ'_1 is `PNAT+`,
- v is $\texttt{S}^{-1}$;`pnat-as-sg`, and
- $\Sigma' \Cap \Sigma'_1$ is the empty presentation $(\Phi(\texttt{PNAT}),\emptyset)$ and therefore $P+(\Sigma' \Cap \Sigma'_1)$ is $(\Phi(\texttt{SG}\star\texttt{S})\cup\Phi(\texttt{PNAT}),\emptyset)$.

Then according to Dfn. 6.8 the instance `SG^`(S$\Leftarrow$ `pnat-as-sig`) is obtained by a pushout of specification shown below:

$$\begin{array}{ccc} \texttt{SG} & \qquad & (\Phi(\texttt{SG} \star \texttt{S}) \cup \Phi(\texttt{PNAT}),\emptyset) & \xrightarrow{\subseteq} & \texttt{SG} \\ \downarrow{\scriptstyle \texttt{pnat-as-sg}} & & \downarrow{\scriptstyle (\texttt{S}^{-1};\texttt{pnat-as-sg})+i_{PNAT}} & & \downarrow \\ \texttt{PNAT+} & & \texttt{PNAT+} & \longrightarrow & \texttt{SG}(\texttt{S} \Leftarrow \texttt{pnat-as-sg}) \end{array}$$

where i_{PNAT} denotes the specification 'inclusion' $\texttt{PNAT} \to \texttt{PNAT+}$. Note how `PNAT` is shared between the body `SG^` of the parameterized specification and the instance `PNAT+` of the parameter.

6.3. An alternative definition for parameter instantiation

In the following we provide another definition for parameter instantiations that under some technical conditions on the structured institution is equivalent to Dfn. 6.8 but that may be technically more convenient than Dfn. 6.8 in some situations (such as dealing with multiple parameters; see the last result of [15]).

Notation 6.1. Let $\mathcal{I}'$ be an institution structured over an inclusive institution $\mathcal{I}$ through Φ such that

1. Φ lifts pushouts,
2. Φ has a left adjoint $\overline{\Phi}$ such that the units of the adjunctions are identities, and
3. each intersection-union square of $\mathcal{I}$-signatures is pushout.

For any $\mathcal{I}'$-signatures Σ' and Σ'_1 by $\Sigma' \Cup \Sigma'_1$ we denote a lifting of the intersection-union square determined by $\Phi(\Sigma')$ and $\Phi(\Sigma'_1)$ to a pushout of $\mathcal{I}'$-signature morphisms as shown by the following diagram:

$$\begin{array}{ccc} \Phi(\Sigma') \cap \Phi(\Sigma'_1) & \xrightarrow{\subseteq} & \Phi(\Sigma') \\ {\scriptstyle\subseteq}\downarrow & & \downarrow{\scriptstyle\subseteq} \\ \Phi(\Sigma'_1) & \xrightarrow[\subseteq]{} & \Phi(\Sigma') \cup \Phi(\Sigma'_1) \end{array} \qquad \begin{array}{ccc} \Sigma' \Cap \Sigma'_1 & \xrightarrow{i} & \Sigma' \\ {\scriptstyle i_1}\downarrow & & \downarrow{\scriptstyle i'} \\ \Sigma'_1 & \xrightarrow[i'_1]{} & \Sigma' \Cup \Sigma'_1 \end{array}$$

Note that $\Sigma' \Cup \Sigma'_1$ in general is not unique, but rather denotes a class of isomorphic $\mathcal{I}'$-signatures. However we are going to be lax about this and when there is not danger of error $\Sigma' \Cup \Sigma'_1$ will mean whatever member of this class of $\mathcal{I}'$-signatures.

Example 6.6. It is rather easy to check that within the framework of Ex. 3.2 the 'union' of theories introduced by Prop. 6.4 is

$$(\Sigma, E) \Cup (\Sigma_1, E_1) = (\Sigma \cup \Sigma_1, (E \cup E_1)^\bullet)$$

and that within the framework of Ex. 3.3 the 'union' of $(\mathcal{T}, \mathcal{D})$-structured specifications introduced by Prop. 6.4 is

$$\Sigma' \Cup \Sigma'_1 = (\Sigma' \star (\Phi(\Sigma') \subseteq \Phi(\Sigma') \cup \Phi(\Sigma'_1))) \cup (\Sigma'_1 \star (\Phi(\Sigma'_1) \subseteq \Phi(\Sigma') \cup \Phi(\Sigma'_1))).$$

The following generalizes a corresponding result from [15] to our structured institutions framework.

Proposition 6.4. *Let $\mathcal{I}'$ be an institution structured over an inclusive institution $\mathcal{I}$ through Φ. In addition to the hypotheses of Dfn. 6.8 let us also assume that*

1. Φ *lifts pushouts and*
2. *each intersection-union square of $\mathcal{I}$-signatures is pushout.*

Given a parameterized $\mathcal{I}'$-signature $\iota\colon P \to \Sigma'$ and an $\mathcal{I}'$-signature morphism $v\colon P \to \Sigma'_1$ such that $\Phi(P)$ and $\Phi(\Sigma'_1)$ are disjoint, then $\Sigma'(\iota \Leftarrow v)$ may be defined as a pushout of $\mathcal{I}'$-signature morphisms as follows:

$$\begin{array}{ccc} P + \Sigma'_1 & \xrightarrow{(\iota;i')+i'_1} & \Sigma' \Cup \Sigma'_1 \\ {\scriptstyle v+1_{\Sigma'_1}}\downarrow & & \downarrow{\scriptstyle v'} \\ \Sigma'_1 & \xrightarrow[\iota']{} & \Sigma'(\iota \Leftarrow v) \end{array}$$

where

- *$P + \Sigma'_1$ is a co-product that lifts the disjoint union $\Phi(P) \cup \Phi(\Sigma'_1)$, and*
- *$(\iota, i') + i'_1$ and $v + 1_{\Sigma'_1}$, resp., are the unique $\mathcal{I}'$-signature morphism 'extending' $(\iota; i')$, i'_1 and v, $1_{\Sigma'_1}$, resp., according to the universal property of the co-product $P + \Sigma'_1$.*

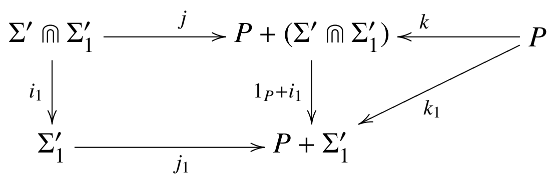
Moreover, if in addition

3. *the inclusion system for the $\mathcal{I}$-signatures is epic, and*
4. *each idempotent-by-extension $\mathcal{I}$-signature morphism admits free extensions along any $\mathcal{I}$-signature inclusion*

then we may choose the instance $\Sigma'(\iota \Leftarrow v)$ such that

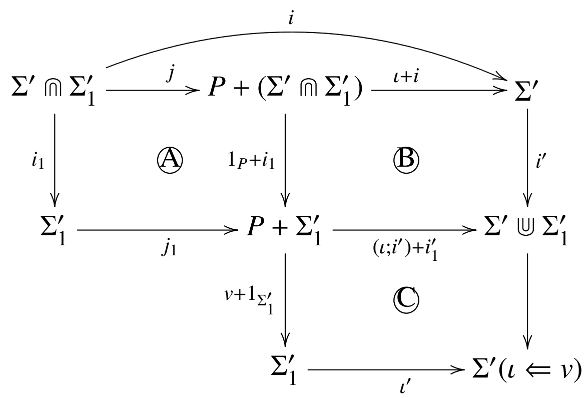
$$\Phi(\Sigma'_1) \subseteq \Phi(\Sigma'(\iota \Leftarrow v)).$$

Proof. For the proof of the first part of the proposition we let in the diagram below j, k and j_1, k_1, resp., be the co-cone morphisms of the co-products $P + (\Sigma' \Cap \Sigma'_1)$ and $P + \Sigma'_1$, resp.



The $\mathcal{I}'$-signature morphism $1_P + i_1$ is defined to be the unique morphism making the diagram commute; this is given by the universal property of the the co-product $P + (\Sigma' \Cap \Sigma'_1)$.

In the diagram below



by a general categorical argument we may establish that Ⓐ is a pushout square and since Ⓐ + Ⓑ is the pushout square defining $\Sigma' \Cup \Sigma'_1$ by a well know general categorical property about gluing pushout squares it follows that Ⓑ is a pushout square. By the same general categorical property it now follows that Ⓑ + Ⓒ is pushout square if and only if Ⓒ is pushout square, which proves the first part of the proposition.

For the second part of the proposition let us first establish that the $\mathcal{I}$-signature morphism below is idempotent-by-extension.

$$\Phi(P) \cup \Phi(\Sigma'_1) \xrightarrow{\Phi(v)+1_{\Sigma'_1}} \Phi(\Sigma'_1) \xrightarrow{\subseteq} \Phi(P) \cup \Phi(\Sigma'_1)$$

Let us denote this morphism by f. The idempotence of f is immediate. We also have that $f(\Phi(P) \cup \Phi(\Sigma'_1)) = \Phi(\Sigma'_1)$ since $e_f = \Phi(v) + 1_{\Sigma'_1}$ (because this is retract and from [17] we know that any retract is abstract surjection). Hence $\Phi(P) \cup \Phi(\Sigma'_1)$ is the disjoint union of $f(\Phi(P) \cup \Phi(\Sigma'_1))$ and $\Phi(P)$ which shows f idempotent-by-extension.

From Prop. 6.2 (with $\Phi(P) \cup \Phi(\Sigma_1')$ in the role of A, $\Phi(\Sigma') \cup \Phi(\Sigma_1')$ in the role of B, $\Phi(\Sigma_1')$ in the role of $f(A)$, and Σ in the role of $g(B)$) we obtain the following pushout squares:

$$
\begin{array}{ccccc}
\Phi(P) \cup \Phi(\Sigma_1') & \xrightarrow{\Phi(v)+1_{\Sigma_1'}} & \Phi(\Sigma_1') & \xrightarrow{\subseteq} & \Phi(P) \cup \Phi(\Sigma_1') \\
\downarrow{\scriptstyle\subseteq} & & \downarrow{\scriptstyle\subseteq} & & \downarrow{\scriptstyle\subseteq} \\
\Phi(\Sigma') \cup \Phi(\Sigma_1') & \xrightarrow[e_g]{} & \Sigma & \xrightarrow[\subseteq]{} & \Phi(\Sigma') \cup \Phi(\Sigma_1')
\end{array}
$$

The conclusion of the proposition now follows by lifting the left hand pushout square above to a pushout square of $\mathcal{I}'$-signature morphisms. □

Example 6.7. The instance `SG^(S⇐ pnat-as-sig)` of Ex. 6.5 may be obtained by applying Prop. 6.4 as follows. First let us note that the additional conditions of Prop. 6.4 may be checked rather easily. For example the existence of free extensions of *MSA* idempotent-by-extension signature morphisms is given by Prop. 6.3; note however that Prop. 6.3 is more general since it requires a condition much weaker than idempotency-by-extension.

Then according to the first part of Prop. 6.4 the instance `SG^(S⇐ pnat-as-sig)` may be obtained by a pushout of specifications as shown below:

$$
\begin{array}{ccc}
\texttt{SG} & & \\
\downarrow{\scriptstyle \texttt{pnat-as-sg}} & & \\
\texttt{PNAT+} & &
\end{array}
\qquad
\begin{array}{ccc}
(\texttt{SG} \star \texttt{S}) + (\texttt{PNAT+}) & \xrightarrow{\subseteq} & (\texttt{SG}) \cup (\texttt{PNAT+}) \\
\downarrow{\scriptstyle (\texttt{S}^{-1};\texttt{pnat-as-sg})+1_{PNAT+}} & & \downarrow \\
\texttt{PNAT+} & \longrightarrow & \texttt{SG}(\texttt{S} \Leftarrow \texttt{pnat-as-sg})
\end{array}
$$

Moreover, the second part of Prop. 6.4 allows us to choose the above pushout such that

$$\Phi(\texttt{PNAT+}) \subseteq \Phi(\texttt{SG}(\texttt{S} \Leftarrow \texttt{pnat-as-sg})).$$

7. Conclusion and Further Research

In this paper we have developed a theory of ‘abstract structured specifications’ involving two levels of institution independence and which includes among its instances the so-called theory or property oriented and the model oriented approaches to structuring specifications. Moreover, in the case of the latter approach our upper level of institution independence means that we may deal with the structuring of specifications without any reference to particular sets of specification building operators. Within such framework we have developed concepts and results about co-limits of abstract specifications, model amalgamation, normal forms, interpolation, compactness, and pushout-style parameterization with sharing.

We think that our proposed axiomatic approach involving two levels of institution independence constitutes a proper framework for the study and understanding of structuring and modularization, hence we expect the development of other modularization concepts and results within our framework. For example, it seems quite straightforward to lift the theory of multiple parameters recently developed in [15] to our framework of ‘abstract structured specifications’, including the equation

$$\Sigma'(\iota_1 \Leftarrow v_1)(\iota_2 \Leftarrow v_2) \cong \Sigma'(\iota_1 + \iota_2 \Leftarrow v_1 + v_2)$$

representing the isomorphism between the sequential and the parallel instantiation of multiple parameters.

Particular open research questions are the development of a proof theory that supports reusability of verifications in the style of [5] and to explore the relationship between our approach and the so-called 'development graphs' of [31], including proof theoretic aspects.

Acknowldgement

Thanks to the first (anonymous) referee for studying carefully this work and coming up with a series of comments and suggestions that have led to several corrections and an improvement in the citations. This work 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

References

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

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

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

[4] Tomasz Borzyszkowski. Generalized interpolation in CASL. *Information Processing Letters*, 76:19–24, 2001.

[5] Tomasz Borzyszkowski. Logical systems for structured specifications. *Theoretical Computer Science*, 286(2):197–245, 2002.

[6] Tomasz Borzyszkowski. Generalized interpolation in first-order logic. *Fundamenta Informaticæ*, 66(3):199–219, 2005.

[7] Rod Burstall and Joseph Goguen. Putting theories together to make specifications. In Raj Reddy, editor, *Proceedings, Fifth International Joint Conference on Artificial Intelligence*, pages 1045–1058. Department of Computer Science, Carnegie-Mellon University, 1977.

[8] 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.

[9] Maria-Victoria Cengarle. *Formal specifications with higher-order parameterization*. PhD thesis, Ludwig-Maximilians-Universitat, Munchen, 1994.

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

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

[12] Răzvan Diaconescu. An institution-independent proof of Craig Interpolation Theorem. *Studia Logica*, 77(1):59–79, 2004.

[13] Răzvan Diaconescu. Interpolation in Grothendieck institutions. *Theoretical Computer Science*, 311:439–461, 2004.

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

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

[16] 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.

[17] 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.

[18] Theodosis Dimitrakos and Tom Maibaum. On a generalized modularization theorem. *Information Processing Letters*, 74:65–71, 2000.

[19] Werner Fey. Pragmatics, concepts, syntax, semantics and correctness notions of ACT TWO: An algebraic module specification and interconnection language. Technical Report 88–26, Technical University of Berlin, Fachbereich Informatik, 1988.

[20] José L. Fiadeiro and Amilcar Sernadas. Structuring theories on consequence. In Donald Sannella and Andrzej Tarlecki, editors, *Recent Trends in Data Type Specification*, volume 332 of *Lecture Notes in Computer Science*, pages 44–72. Springer, 1988.

[21] Joseph Goguen. Higher-order functions considered unnecessary for higher-order programming. In David Turner, editor, *Research Topics in Functional Programming*, pages 309–352. Addison Wesley, 1990. University of Texas at Austin Year of Programming Series; preliminary version in SRI Technical Report SRI-CSL-88-1, January 1988.

[22] 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.

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

[24] Joseph Goguen, Timothy Winkler, José Meseguer, Kokichi Futatsugi, and Jean-Pierre Jouannaud. Introducing OBJ. In Joseph Goguen and Grant Malcolm, editors, *Software Engineering with OBJ: algebraic specification in action*. Kluwer, 2000.

[25] Daniel Găină and Andrei Popescu. An institution-independent proof of Robinson consistency theorem. *Studia Logica*, 85(1):41–73, 2007.

[26] Robert Harper, Donald Sannella, and Andrzej Tarlecki. Logic representation in LF. In David Pitt, David Rydeheard, Peter Dybjer, Andrew Pitts, and Axel Poigné, editors, *Proceedings, Conference on Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 250–272. Springer, 1989.

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

[28] S. Maehara. On the interpolation theorem of Craig. *Sugaku*, pages 235–237, 1961-1962. In Japanese.

[29] Thomas Maibaum, José Fiadeiro, and Martin Sadler. Stepwise program development in Π-institutions. Technical report, Imperial College, 1990.

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

[31] T. Mossakowski, S. Autexier, and D. Hutter. Development graphs - proof management for structured specification. *Journal of Logic and Algebraic Programming*, 67(1-2):114–145, 2006.

[32] Till Mossakowski, Răzvan Diaconescu, and Andrzej Tarlecki. What is a logic translation? *Logica Universalis*, 3(1):59–94, 2009.

[33] Till Mossakowski, Joseph Goguen, Răzvan Diaconescu, and Andrzej Tarlecki. What is a logic? In Jean-Yves Béziau, editor, *Logica Universalis*, pages 113–133. Birkhäuser, 2005.

[34] Marius Petria and Răzvan Diaconescu. Abstract Beth definability in institutions. *Journal of Symbolic Logic*, 71(3):1002–1028, 2006.

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

[36] Pieter-Hendrik Rodenburg. A simple algebraic proof of the equational interpolation theorem. *Algebra Universalis*, 28:48–51, 1991.

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

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

[39] Joseph Shoenfield. *Mathematical Logic*. Addison-Wesley, 1967.

[40] Andrzej Tarlecki. Bits and pieces of the theory of institutions. In David Pitt, Samson Abramsky, Axel Poigné, and David Rydeheard, editors, *Proceedings, Summer Workshop on Category Theory and Computer Programming*, volume 240 of *Lecture Notes in Computer Science*, pages 334–360. Springer, 1986.

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

[42] 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.

[43] Alfred Tarski. Über einige fundamentale Begriffe der Metamathematik. *C. R. Soc. Sci. Lettr. Varsovie*, Cl. III(23):22–29, 1930.

[44] 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.

[45] Paulo Veloso. On pushout consistency, modularity and interpolation for logical specifications. *Information Processing Letters*, 60(2):59–66, 1996.