

# On the Existence of Translations of Structured Specifications

Răzvan Diaconescu

*Simion Stoilow Institute of Mathematics of the Romanian Academy*

---

## Abstract

We provide a set of sufficient conditions for the existence of translations of structured specifications across specification formalisms. The most basic condition is the existence of a translation between the logical systems underlying the specification formalisms, which corresponds to the unstructured situation. Our approach is based upon institution theory and especially upon a recent abstract approach to structured specifications in which both the underlying logics and the structuring systems are treated fully abstractly. Hence our result is applicable to a wide range of actual specification formalisms that may employ different logics as well as different structuring systems, and is very relevant within the context of the fastly developing heterogeneous specification paradigm.

---

## 1. Introduction

Formal specification is an important paradigm that assist the development and maintenance of complex software systems; some argue that it is indispensable in the case of critical systems. Complex software systems involve very large specifications that cannot be developed and maintained in the absence of adequate structuring or modularisation. On the other hand, with the recent advance of the heterogeneous specification paradigm, there is a growing interest in the theory of translations between different specification formalisms. While this theory is quite developed at the level of translations between the underlying logics, which corresponds to the unstructured case, very little has been done for the structured specifications level. In my opinion, one reason for that situation lies in the fact until [5] structuring has been always treated concretely by making a choice of a particular set of structuring operators. With that kind of commitment the only variation between specification formalisms that is possible to consider is at the level of the underlying logic. This is a limitation since actual specification systems may involve modularisation constructs that cannot always be traced back to an apriori fixed set of core structuring operators.

The theory of abstractly structured specifications (ASS) of [5] provides a flexible approach to structuring systems and within that framework in [2] the author defines an adequate concept of translation of ASSs that is based upon the concept of comorphism from institution theory [10]. In my opinion, apart from the definition of the concept, the most important contribution of [2] consists of a minimalistic axiomatisation of the concept. In this work we build on the concept of translation of [2] and take a step forward by giving a set of general conditions, widely applicable, for the existence of translations of ASSs.

In the first part of the paper we recall very briefly general concepts of institution theory, and then we recall the main concepts from the theory of ASSs of [5] and also introduce a couple of new concepts required by our work. An important argument presented in the form of an example shows that the semantic normal

---

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

forms of [5], which play an important role in our main result, is significantly more general than its syntactic counterpart from the literature (e.g. [15]). The final technical section starts by recalling the concept of translation of ASSs of [2] and continues with the development of the main result of this paper, namely the existence of translations of ASSs. The general theory is illustrated by a relevant example displaying both concrete logics and concrete sets of structuring operators.

## 2. Institutions and Comorphisms

Institutions [1, 9] formalize the intuitive notion of logical system, including the syntax, semantics and the satisfaction between them and have been used intensively in computer science (e.g. [15]) and logic (e.g. [3]).

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

1. a category  $\text{Sign}^{\mathcal{I}}$ , whose objects are called *signatures*,
2. a functor  $\text{Sen}^{\mathcal{I}} : \text{Sign}^{\mathcal{I}} \rightarrow \text{Set}$  (to the category of sets), giving for each signature a set whose elements are called *sentences* over that signature,
3. a functor  $\text{Mod}^{\mathcal{I}} : (\text{Sign}^{\mathcal{I}})^{\text{op}} \rightarrow \mathbb{CAT}$  (from the opposite of  $\text{Sign}^{\mathcal{I}}$  to the category of categories) giving for each signature  $\Sigma$  a category whose objects are called  $\Sigma$ -*models*, and whose arrows are called  $\Sigma$ -*(model) homomorphisms*, and
4. a relation  $\models_{\Sigma}^{\mathcal{I}} \subseteq |\text{Mod}^{\mathcal{I}}(\Sigma)| \times \text{Sen}^{\mathcal{I}}(\Sigma)$  for each  $\Sigma \in |\text{Sign}^{\mathcal{I}}|$ , called  $\Sigma$ -*satisfaction*,

such that for each morphism  $\varphi : \Sigma \rightarrow \Sigma'$  in  $\text{Sign}^{\mathcal{I}}$ , the *satisfaction condition*

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

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

We denote the *reduct* functors  $\text{Mod}^{\mathcal{I}}(\varphi)$  by  $-\downarrow_{\varphi}$  and the sentence translations  $\text{Sen}^{\mathcal{I}}(\varphi)$  by  $\varphi(\_)$ . When there is no danger of ambiguity, we may skip the superscripts from the notations of the entities of the institution; for example,  $\text{Sign}^{\mathcal{I}}$  may be denoted simply by *Sign*. A  $\mathcal{I}$ -*theory* is any pair  $(\Sigma, E)$  such that  $\Sigma \in |\text{Sign}|$  and  $E \subseteq \text{Sen}(\Sigma)$ . For theory  $(\Sigma, E)$  we let  $\text{Mod}(\Sigma, E)$  denote the full subcategory of  $\text{Mod}(\Sigma)$  whose objects are the models satisfying all sentences of  $E$  and let  $E^{\bullet}$  denote the set of sentences satisfied by all models of  $\text{Mod}(\Sigma, E)$ .

The literature shows myriads of logical systems from computing or from mathematical logic captured as institutions. In fact, an informal thesis underlying institution theory is that any logic based on satisfaction between sentences and models of any kind may be captured by the above definition. Below we recall very briefly a couple of them that we will use in our examples.

**Example 2.1** (Many-sorted First-Order Logic). In the institution  $\mathcal{FOL}$  of many-sorted first-order logic with equality the signatures consist of sorts and typed functions and predicate symbols. The arities of functions are finite strings of sorts. Signature morphisms map symbols such that arities are preserved. Models are first-order structures interpreting sorts as sets, function symbols as functions, and predicate symbols as relations. The sentences are first-order formulas formed from atomic predicate sentences and equations by iteration of logical connectives ( $\wedge, \vee, \neg$  etc.) and (first-order) quantifiers  $\forall, \exists$ . Sentence translation means replacement of the translated symbols. Model reduct means reassembling the models components according to the signature morphism. Satisfaction is the usual Tarskian satisfaction of a first-order sentence in a first-order structure that is defined by induction on the structure of the sentences. Detailed definitions of variants of this rather common institution, that differ only slightly, can be found in very many places in the literature, e.g. [9, 15, 3].

**Example 2.2** (Partial Algebra). The institution  $\mathcal{PA}$  of partial algebra is similar to  $\mathcal{FOL}$  but functions can also be partial and there are no predicates. Equations evaluate to false if some component term involves some undefinedness or if they evaluate to different values.

**Example 2.3** (Institutions of Theories). For any institution  $\mathcal{I}$  the institution of its theories  $\mathcal{I}^{\text{th}}$  has finite theories  $(\Sigma, E)$  (with  $\Sigma$  any  $\mathcal{I}$ -signature and  $E \subseteq \text{Sen}^{\mathcal{I}}(\Sigma)$ ) as signatures. In  $\mathcal{I}^{\text{th}}$  the  $(\Sigma, E)$ -sentences are just the  $\Sigma$ -sentences in  $\mathcal{I}$  and the  $(\Sigma, E)$ -models are the  $\Sigma$ -models that satisfy all sentences in  $E$ . The satisfaction relation of  $\mathcal{I}^{\text{th}}$  is inherited from  $\mathcal{I}$ .

The notion of comorphism [11, 16, 17, 10] represents one of the most important kind of structure preserving mappings between institutions and provides an adequate formalisation of the informal concept of logic translation.

**Definition 2.2** (Comorphisms). An *institution comorphism*  $(\Phi, \alpha, \beta): \mathcal{I} \rightarrow \mathcal{I}'$  consists of

1. a functor  $\Phi: \text{Sign} \rightarrow \text{Sign}'$ ,
2. a natural transformation  $\alpha: \text{Sen} \Rightarrow \Phi; \text{Sen}'$ , and
3. a natural transformation  $\beta: \Phi^{\text{op}}; \text{Mod}' \Rightarrow \text{Mod}$

such that the following *satisfaction condition* holds

$$M' \models_{\Phi(\Sigma)} \alpha_{\Sigma}(e) \text{ iff } \beta_{\Sigma}(M') \models_{\Sigma} e$$

for each signature  $\Sigma \in |\text{Sign}|$ , for each  $\Phi(\Sigma)$ -model  $M'$ , and each  $\Sigma$ -sentence  $e$ .

**Example 2.4** (Encoding  $\mathcal{PA}$  into  $\mathcal{FOL}$ ). Let us briefly recall the emblematic case of the encodings of  $\mathcal{PA}$  into  $\mathcal{FOL}$ . There are several such encodings (details may be found in the literature, e.g. [12, 13, 3, 4]). All of them of them are *theoroidal*, i.e. they appear as comorphisms  $\mathcal{PA} \rightarrow \mathcal{FOL}^{\text{th}}$ . Perhaps the best known one encodes partial functions as total functions by adding for each sort an unary relation symbol standing for the defined values, the target presentations consisting of Horn sentences. In general one may think about logic encodings as being formalised as theoroidal comorphisms.

**Definition 2.3** (Amalgamation for comorphisms [3]). For any comorphism  $(\Phi, \alpha, \beta): \mathcal{I} \rightarrow \mathcal{I}'$  we say that a signature morphism  $\varphi_1: \Sigma \rightarrow \Sigma_1$  in  $\mathcal{I}$  has *weak model  $(\Phi, \beta)$ -amalgamation* when for each  $\Sigma_1$ -model  $M_1$  and each  $\Phi(\Sigma)$ -model  $M'$  with  $M_1 \upharpoonright_{\varphi_1} = \beta_{\Sigma}(M')$  there exists a  $\Phi(\Sigma_1)$ -model  $M'_1$  such that  $\beta_{\Sigma_1}(M'_1) = M_1$  and  $M'_1 \upharpoonright_{\Phi(\varphi_1)} = M'$ .

**Example 2.5.** It is easy to check that in the case of the theoroidal comorphism  $\mathcal{PA} \rightarrow \mathcal{FOL}^{\text{th}}$  that encodes partial functions as total functions mentioned at Ex. 2.4, each injective signature morphism has weak model  $(\Phi, \beta)$ -amalgamation.

### 3. Abstractly structured specifications

**Definition 3.1** (Structured institutions [5]). Given two institutions  $\mathcal{I} = (\text{Sign}, \text{Sen}, \text{Mod}, \models)$  and  $\mathcal{I}' = (\text{Sign}', \text{Sen}', \text{Mod}', \models')$ , we say that  $\mathcal{I}'$  is  $(\mathcal{I}, \text{sig})$ -structured when

- $\text{sig}: \text{Sign}' \rightarrow \text{Sign}$  is a functor,
- for each  $\mathcal{I}'$ -signature  $\Sigma'$  we have  $\text{Sen}(\text{sig}(\Sigma')) = \text{Sen}'(\Sigma')$ , and for each  $\mathcal{I}'$ -signature morphism  $\varphi'$  we have  $\text{Sen}(\text{sig}(\varphi')) = \text{Sen}'(\varphi')$ ,

- for each  $\mathcal{I}'$ -signature  $\Sigma'$  we have that  $Mod'(\Sigma')$  is a full subcategory of  $Mod(sig(\Sigma'))$  such that for each  $\mathcal{I}'$ -signature morphism  $\varphi' : \Sigma'_1 \rightarrow \Sigma'_2$  the diagram below commutes,

$$(1) \quad \begin{array}{ccc} Mod'(\Sigma'_1) & \xrightarrow{\subseteq} & Mod(sig(\Sigma'_1)) \\ \uparrow Mod'(\varphi') & & \uparrow Mod(sig(\varphi')) \\ Mod'(\Sigma'_2) & \xrightarrow{\subseteq} & Mod(sig(\Sigma'_2)) \end{array}$$

- for each  $\mathcal{I}'$ -signature  $\Sigma'$ , each  $\Sigma'$ -model  $M'$  and each  $\Sigma'$ -sentence  $\rho$  we have that

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

From a specification theoretic perspective, the  $\mathcal{I}'$ -signatures may be referred to as  $(\mathcal{I}, sig)$ -specifications.

In [5] several examples are presented in some detail; here let us just mention them rather briefly.

1. The Sannella-Tarlecki approach [14, 15] is covered by considering the structured specifications formed from finite  $\mathcal{I}$ -theories by iteration of some building operators. These structured specifications play the role of the  $\mathcal{I}'$ -signatures, whilst their denotations give the structuring functor  $sig$  and the model functor of  $\mathcal{I}'$ . Then the most common such set of building operators consists of unions, translations, and derivations as the signatures of  $\mathcal{I}'$ .

**BASIC.** Each finite theory  $(\Sigma, E)$  is a specification such that  $sig(\Sigma, E) = \Sigma$  and  $Mod'(\Sigma, E) = Mod(\Sigma, E)$ .

**UNION.** For any specifications  $SP_1$  and  $SP_2$  such that  $sig(SP_1) = sig(SP_2)$  we can take their *union*  $SP_1 \cup SP_2$  with  $sig(SP_1 \cup SP_2) = sig(SP_1) = sig(SP_2)$  and  $Mod'(SP_1 \cup SP_2) = Mod'(SP_1) \cap Mod'(SP_2)$ .

**TRANS.** For any specification  $SP$  and signature morphism  $\varphi : sig(SP) \rightarrow \Sigma'$  we can take its *translation along  $\varphi$*  denoted by  $SP \star \varphi$  and such that  $sig(SP \star \varphi) = \Sigma'$  and  $Mod'(SP \star \varphi) = \{M' \in Mod(\Sigma') \mid M' \upharpoonright_{\varphi} \in Mod'(SP)\}$ .

**DERIV.** For any specification  $SP'$  and any signature morphism  $\varphi : \Sigma \rightarrow sig(SP')$  we can take its *derivation along  $\varphi$*  denoted by  $\varphi \square SP'$  and such that  $sig(\varphi \square SP') = \Sigma$  and  $Mod'(\varphi \square SP') = \{M' \upharpoonright_{\varphi} \mid M' \in Mod'(SP')\}$ .

However the literature discusses many other possible sets of specifications building operators (see [15, 6], etc.) required by various modularisation constructs. All of them can be given semantics in the style of BASIC, UNION, TRANS, DERIV above.

2. In the case of the Goguen-Burstall approach [9, 8] in the role of the  $\mathcal{I}'$ -signatures one considers the theories  $(\Sigma, E)$  closed under semantic consequence (i.e.  $E = E^\bullet$ ). Then  $sig(\Sigma, E) = \Sigma$ .
3. One may also consider quotients of specifications by algebraic rules, such as commutativity and/or associativity of the union. A general theory of such quotienting is given in [5].
4. Other formalisms not necessarily rooted within specification theory may also be covered, such as the module systems for model expansion problems [18].

The following is a common property of structured specifications that holds in all examples mentioned above.

**Definition 3.2.** When the canonical mapping determined by  $sig$

$$Sig'(\Sigma'_1, \Sigma'_2) \rightarrow \{\varphi \in Sig(sig(\Sigma'_1), sig(\Sigma'_2)) \mid Mod'(\Sigma'_2) \upharpoonright_{\varphi} \subseteq Mod'(\Sigma'_1)\}.$$

are bijections then we say that  $\mathcal{I}'$  inherits the signature morphisms.

**Definition 3.3.** Let  $\mathcal{I}'$  be an institution that is  $(\mathcal{I}, sig)$ -structured.

- We say that  $\mathcal{I}'$  has *semantic basic specifications* when for each finite  $\mathcal{I}$ -theory  $(\Sigma, E)$  there exists a  $\mathcal{I}'$ -signature  $SP(\Sigma, E)$  such that  $sig(SP(\Sigma, E)) = \Sigma$  and  $Mod'(SP(\Sigma, E)) = Mod(\Sigma, E)$ .
- For any  $\mathcal{I}$ -signature morphism  $\varphi: \Omega \rightarrow \Sigma$ , we say that  $\mathcal{I}'$  has *semantic  $\varphi$ -derivations* [7] when for any  $\mathcal{I}'$ -signature  $\Sigma'$  such that  $sig(\Sigma') = \Sigma$  there exists a designated  $\mathcal{I}'$ -signature, denoted  $\varphi \square \Sigma'$ , such that  $sig(\varphi \square \Sigma') = \Omega$  and  $Mod'(\varphi \square \Sigma') = \{M' \upharpoonright_{\varphi} \mid M' \in Mod'(\Sigma')\}$ .

**Example 3.1.** In all examples of structured institutions mentioned above there is a straightforward way for semantic basic specifications given by  $SP(\Sigma, E) = (\Sigma, E)$  but also other ways in which  $SP(\Sigma, E)$  may be also defined as a properly structured specification (i.e. involving structuring operators such as UNION, TRANS).

Any structuring of specifications in the style of [14, 15] that has a DERIV operator has adequate semantic derivations. Quotients of such structurings also have semantic derivations. However note that the ‘hiding information’ operator of [8] in the Goguen-Burstall approach (defined on the closed theories by  $\varphi \square (\Sigma, E) = (\Omega, \varphi^{-1}(E))$ ) does *not* imply the existence of semantic derivations.

**Definition 3.4** (Semantic normal forms [5]). Given a  $(\mathcal{I}, sig)$ -structured institution  $\mathcal{I}'$ , a pair  $(\varphi, E)$  consisting of a signature morphism  $\varphi: sig(\Sigma') \rightarrow \Sigma$  and a set of sentences  $E \subseteq Sen(\Sigma)$  is a *semantic normal form* for an  $(\mathcal{I}, sig)$ -specification  $\Sigma'$  when  $Mod'(\Sigma') = Mod(\Sigma, E) \upharpoonright_{\varphi}$ . When  $E$  is finite we say that the normal form is *finitary*. We say that  $\mathcal{I}'$  admits (finitary) *semantic normal forms* when each  $(\mathcal{I}, sig)$ -specification has at least a (finitary) normal form.

**Example 3.2.** A classic result in specification literature (eg. [15, 3]) is that, under some very mild technical conditions, each specification structured with BASIC, UNION, TRANS, and DERIV has the same signature and class of models with a *normal form* specification, ie. a specification of the form  $\varphi \square (\Sigma, E)$  where  $E$  is a finite set of sentences for a signature  $\Sigma$ . These normal forms are *syntactic* in the sense that they are actual specifications. However syntactic normal forms obviously imply semantic normal forms.

The following example shows that the concept of semantic normal form is significantly wider than its syntactic counterpart from [15, 3]. Proper semantic normal forms may occur in the absence of syntactic normal forms and of a DERIV building operator.

**Example 3.3.** Let us consider the specifications built over  $\mathcal{FOL}$  with BASIC, UNION, TRANS and INTERS where the latter is defined as follows:

**INTERs.** For any specifications  $SP_1$  and  $SP_2$  we can take their *intersection*  $SP_1 \cap SP_2$  with

$$sig(SP_1 \cap SP_2) = sig(SP_1) \cap sig(SP_2),^1 \text{ and}$$

$$Mod'(SP_1 \cap SP_2) = Mod'(SP_1) \upharpoonright_{sig(SP_1) \cap sig(SP_2)} \cup Mod'(SP_2) \upharpoonright_{sig(SP_1) \cap sig(SP_2)}.$$

<sup>1</sup>The intersection of  $\mathcal{FOL}$ -signatures are set theoretic and can be defined component-wise on the sets of sorts first and then on the sets of operation or relations symbols corresponding to different ranks.

Intersection as specification building operator is very relevant in works involving module sharing in a significant way, such as in [6, 19] where parametrised specifications with sharing are studied. It has also been mentioned in the respective contexts in [8, 15].

Let us show that these structured specifications have semantic finitary normal forms  $(\varphi, E)$  with  $\varphi$  injective  $\mathcal{FOL}$ -signature morphism. This can be shown by induction on the structure of specifications. For the induction steps corresponding to BASIC, UNION and TRANS we may refer to the general results in the literature (eg. [15, 3]), therefore we focus here on the induction step corresponding to INTERS. Suppose that  $SP_1, SP_2$  have the semantic normal forms  $(\varphi_1: sig(SP_1) \rightarrow \Sigma_1, E_1)$  and  $(\varphi_2: sig(SP_2) \rightarrow \Sigma_2, E_2)$ , respectively. We consider the following pushout square of  $\mathcal{FOL}$ -signature morphisms.

$$\begin{array}{ccccc} sig(SP_1) & \xleftarrow[\cong]{i_1} & sig(SP_1) \cap sig(SP_2) & \xrightarrow[\cong]{i_2} & sig(SP_2) \\ \varphi_1 \downarrow & & \downarrow \varphi & & \downarrow \varphi_2 \\ \Sigma_1 & \xrightarrow{\theta_1} & \Sigma' & \xleftarrow{\theta_2} & \Sigma_2 \end{array}$$

Let us show that  $(\varphi, \theta_1(E_1) \vee \theta_2(E_2))$  is an appropriate semantic normal form for  $SP_1 \cap SP_2$  where  $\theta_1(E_1) \vee \theta_2(E_2) = \{\rho_1 \vee \rho_2 \mid \rho_1 \in \theta_1(E_1), \rho_2 \in \theta_2(E_2)\}$ . Here ‘appropriate’ means first that  $\varphi$  is injective and second that  $\theta_1(E_1) \vee \theta_2(E_2)$  is finite. While the latter follows immediately from the finiteness of  $E_1$  and  $E_2$  (cf. induction hypothesis) the former requires more explanation. Since by the induction hypothesis  $\varphi_k$  are injective, it follows that  $i_k; \varphi_k$  are injective too. By the property of pushouts of  $\mathcal{FOL}$ -signature morphisms we get that  $\theta_k$  are injective too. Hence  $\varphi = i_k; \varphi_k; \theta_k$  is injective.

In this situation the semantic normal form property means that

$$Mod(\Sigma_1, E_1) \downarrow_{\varphi_1} \downarrow_{i_1} \cup Mod(\Sigma_2, E_2) \downarrow_{\varphi_2} \downarrow_{i_2} = Mod(\theta_1(E_1) \vee \theta_2(E_2)) \downarrow_{\varphi}.$$

This can be established easily from the Satisfaction Condition and from the fact that  $\theta_k$ , being injective, are also *conservative*, ie. for each  $\Sigma_k$ -model  $M_k$  there exists a  $\Sigma'$ -model  $M'$  such that  $M_k = M' \downarrow_{\theta_k}$ .

This example can be easily generalised by replacing  $\mathcal{FOL}$  with an abstract institution with disjunctions and with an inclusion system [8, 3] for the category of the signatures that enjoys adequate properties and by replacing the class of the injective signature morphisms with an abstract class of  $\mathcal{D}$  of signature morphisms which together with an abstract class for the morphisms involved in TRANS satisfies some appropriate technical conditions.

#### 4. Translations of abstractly structured specifications

The following concept has been introduced in [2] as a formalisation of the informal notion of translation of ASSs.

**Definition 4.1** (Structured comorphism). We say that a comorphism  $(\Phi', \alpha', \beta'): I'_1 \rightarrow I'_2$  is  $((\Phi, \alpha, \beta), sig_1, sig_2)$ -structured, where  $(\Phi, \alpha, \beta): I_1 \rightarrow I_2$  is a comorphism, when

- $sig_1: Sign'_1 \rightarrow Sign_1$  and  $sig_2: Sign'_2 \rightarrow Sign_2$  are functors such that the following diagram commutes,

$$(2) \quad \begin{array}{ccc} Sign'_1 & \xrightarrow{\Phi'} & Sign'_2 \\ sig_1 \downarrow & & \downarrow sig_2 \\ Sign_1 & \xrightarrow{\Phi} & Sign_2 \end{array}$$

- $\mathcal{I}'_1$  is  $(\mathcal{I}_1, sig_1)$ -structured and  $\mathcal{I}'_2$  is  $(\mathcal{I}_2, sig_2)$ -structured, and
- for each  $\mathcal{I}'_1$ -signature  $\Sigma'_1$ ,  $\alpha'_{\Sigma'_1} = \alpha_{sig_1(\Sigma'_1)}$  and  $\beta'_{\Sigma'_1}$  is the restriction of  $\beta_{sig_1(\Sigma'_1)}$ :

$$(3) \quad \begin{array}{ccc} Mod'_2(\Phi'(\Sigma'_1)) & \xrightarrow{\beta'_{\Sigma'_1}} & Mod'_1(\Sigma'_1) \\ \subseteq \downarrow & & \downarrow \subseteq \\ Mod_2(\Phi(sig_1(\Sigma'_1))) & \xrightarrow{\beta_{sig_1(\Sigma'_1)}} & Mod_1(sig_1(\Sigma'_1)) \end{array}$$

The following result of [2] provides a minimal set of axioms for the concept of structured comorphism. This gives an economical way for proving structured comorphisms.

**Proposition 4.1.** *Given institutions  $\mathcal{I}'_1$  that is  $(\mathcal{I}_1, sig_1)$ -structured and  $\mathcal{I}'_2$  that is  $(\mathcal{I}_2, sig_2)$ -structured, functors  $\Phi, \Phi'$  such that the square (2) commutes, the following are equivalent:*

1. *there exists a comorphism  $(\Phi', \alpha', \beta'): \mathcal{I}'_1 \rightarrow \mathcal{I}'_2$  that is  $((\Phi, \alpha, \beta), sig_1, sig_2)$ -structured, and*
2. *for each  $\mathcal{I}'_1$ -signature  $\Sigma'_1$ , we have that*

$$(4) \quad \beta_{sig_1(\Sigma'_1)}(Mod'_2(\Phi'(\Sigma'_1))) \subseteq Mod'_1(\Sigma'_1).$$

The following is the main result of the paper and gives a set of sufficient conditions for the lifting of a comorphism between base institutions to a structured comorphism.

**Proposition 4.2.** *Let  $\mathcal{I}'_1$  and  $\mathcal{I}'_2$  be  $(\mathcal{I}_1, sig_1)$ , resp.  $(\mathcal{I}_2, sig_2)$ -structured institutions such that*

1.  *$\mathcal{I}'_1$  has finitary semantic normal forms  $(\varphi, E)$  with  $\varphi \in \mathcal{D}$  for  $\mathcal{D}$  a class of  $\mathcal{I}_1$ -signature morphisms,*
2.  *$\mathcal{I}'_2$  inherits the signature morphisms,*
3.  *$\mathcal{I}'_2$  has semantic basic specifications.*

*Any comorphism  $(\Phi, \alpha, \beta): \mathcal{I}_1 \rightarrow \mathcal{I}_2$  such that*

4.  *$\mathcal{I}'_2$  has  $\Phi(\varphi)$ -derivations for any  $\varphi \in \mathcal{D}$ , and*
5. *any morphism in  $\mathcal{D}$  has weak  $(\Phi, \beta)$ -amalgamation*

*determines a  $((\Phi, \alpha, \beta), sig_1, sig_2)$ -structured comorphism  $(\Phi', \alpha', \beta'): \mathcal{I}'_1 \rightarrow \mathcal{I}'_2$ .*

*Proof.* Let  $\Sigma'_1$  be a  $\mathcal{I}'_1$ -signature and let  $(\sigma_1: sig_1(\Sigma'_1) \rightarrow \Sigma_1, E_1)$  be a finitary normal form for  $\Sigma'_1$ . We define

$$\Phi'(\Sigma'_1) = \Phi(\sigma_1) \sqcap SP(\Phi(\Sigma_1), \alpha(E_1)).$$

For any  $\mathcal{I}'_1$ -signature morphism  $\theta'_1: \Sigma'_1 \rightarrow \Omega'_1$  such that  $\Phi'(\Omega'_1)$  is defined on the basis of selecting  $(\omega_1: sig_1(\Omega'_1) \rightarrow \Omega_1, T_1)$  as a normal form for  $\Omega'_1$ , since  $\mathcal{I}'_2$  inherits the signature morphisms, in order to define  $\Phi'(\theta'_1)$  it suffices to show that  $\Phi(sig_1(\theta'_1))$  satisfies the condition

$$Mod'_2(\Phi'(\Omega'_1)) \upharpoonright_{\Phi(sig_1(\theta'_1))} \subseteq Mod'_2(\Phi'(\Sigma'_1)).$$

By the definition of derivations and by the selected normal forms, this means that for any  $M_2 \in Mod_2(\Phi(\Omega_1), \alpha(T_1))$  we have to show that

$$(5) \quad M_2 \upharpoonright_{\Phi(\omega_1)} \upharpoonright_{\Phi(sig_1(\theta'_1))} \in Mod_2(\Phi(\Sigma_1), \alpha(E_1)) \upharpoonright_{\Phi(\sigma_1)}.$$

We have that

$$\begin{aligned}
& \beta_{sig_1(\Sigma'_1)}(M_2 \upharpoonright_{\Phi(\omega_1)} \upharpoonright_{\Phi(sig_1(\theta'_1))}) \\
&= \beta_{\Omega_1}(M_2) \upharpoonright_{\omega_1} \upharpoonright_{sig_1(\theta'_1)} && \text{(by the naturality of } \beta) \\
&\in Mod_1(\Omega_1, T_1) \upharpoonright_{\omega_1} \upharpoonright_{sig_1(\theta'_1)} && \text{(by the Satisfaction Condition of } (\Phi, \alpha, \beta)) \\
&= Mod'_1(\Omega'_1) \upharpoonright_{sig_1(\theta'_1)} && \text{(since } (\omega_1, T_1) \text{ is a normal form for } \Omega'_1) \\
&\subseteq Mod'_1(\Sigma'_1) && \text{(because } \theta'_1: \Sigma'_1 \rightarrow \Omega'_1, \text{ cf. (1))} \\
&= Mod_1(\Sigma_1, E_1) \upharpoonright_{\sigma_1} && \text{(since } (\sigma_1, E_1) \text{ is a normal form for } \Sigma'_1).
\end{aligned}$$

Hence there exists  $M_1 \in Mod_1(\Sigma_1, E_1)$  such that  $M_1 \upharpoonright_{\sigma_1} = \beta_{sig_1(\Sigma'_1)}(M_2 \upharpoonright_{\Phi(\omega_1)} \upharpoonright_{\Phi(sig_1(\theta'_1))})$ . By the amalgamation property of the square  $\mathbb{A}$  there exists  $N_2 \in Mod_2(\Phi(\Sigma_1))$  such that  $N_2 \upharpoonright_{\Phi(\sigma_1)} = M_2 \upharpoonright_{\Phi(\omega_1)} \upharpoonright_{\Phi(sig_1(\theta'_1))}$  and  $\beta_{\Sigma_1}(N_2) = M_1$ . By the Satisfaction Condition of  $(\Phi, \alpha, \beta)$  it follows that  $N_2 \models \alpha(E_1)$ . This completes the proof of (5).

$$\begin{array}{ccccccc}
Mod_2(\Phi(\Omega_1)) & \xrightarrow{-\upharpoonright_{\Phi(\omega_1)}} & Mod_2(\Phi(sig_1(\Omega'_1))) & \xrightarrow{-\upharpoonright_{\Phi(sig_1(\theta'_1))}} & Mod_2(\Phi(sig_1(\Sigma'_1))) & \xleftarrow{-\upharpoonright_{\Phi(\sigma_1)}} & Mod_2(\Phi(\Sigma_1)) \\
\beta_{\Omega_1} \downarrow & & \downarrow \beta_{sig_1(\Omega'_1)} & & \beta_{sig_1(\Sigma'_1)} \downarrow & \mathbb{A} & \downarrow \beta_{\Sigma_1} \\
Mod_1(\Omega_1) & \xrightarrow{-\upharpoonright_{\omega_1}} & Mod_1(sig_1(\Omega'_1)) & \xrightarrow{-\upharpoonright_{sig_1(\theta'_1)}} & Mod_1(sig_1(\Sigma'_1)) & \xleftarrow{-\upharpoonright_{\sigma_1}} & Mod_1(\Sigma_1)
\end{array}$$

Now, by virtue of Prop. 4.1 it remains only to show the relation (4). This holds by the following calculations.

$$\begin{aligned}
& \beta_{sig_1(\Sigma'_1)}(Mod'_2(\Phi'(\Sigma'_1))) \\
&= \beta_{sig_1(\Sigma'_1)}(Mod_2(\Phi(\Sigma_1), \alpha(E_1)) \upharpoonright_{\Phi(\sigma_1)}) && \text{(since } (\sigma_1, E_1) \text{ is a normal form for } \Sigma'_1) \\
&\subseteq \beta_{\Sigma_1}(Mod_2(\Phi(\Sigma_1), \alpha(E_1))) \upharpoonright_{\sigma_1} && \text{(by the naturality of } \beta) \\
&\subseteq Mod_1(\Sigma_1, E_1) \upharpoonright_{\sigma_1} && \text{(by the Satisfaction Condition of } (\Phi, \alpha, \beta)) \\
&= Mod'_1(\Sigma'_1) && \text{(since } (\sigma_1, E_1) \text{ is a normal form for } \Sigma'_1).
\end{aligned}$$

□

From the conditions of Prop. 4.2 only 1. and 4. are substantial, the rest are just technical conditions. According to the discussion in Ex. 3.1 the third condition underlying Prop. 4.2 provides space for considering different ways for structuring the result of the translation according to the structuring of  $I'_2$ , thus leading in fact to various different translations.

The following example is meant to illustrate how Prop. 4.2 may be applied in actual situations.

**Example 4.1.** Let us consider the specifications over  $\mathcal{PA}$  (partial algebra) as underlying logic and that are structured by BASIC, UNION, TRANS (by any signature morphism) and INTERS (cf. Ex. 3.3); let this be  $I'_1$  of Prop. 4.2. We consider the theoroidal comorphism  $\mathcal{PA} \rightarrow \mathcal{FOL}^{\text{th}}$  of Ex. 2.4 in the role of  $(\Phi, \alpha, \beta)$  of Prop. 4.2. We take now  $I'_2$  to be the institution of the specifications over  $\mathcal{FOL}^{\text{th}}$  that are structured by BASIC, UNION, TRANS (by any signature morphism) and DERIV (by injective signature morphisms).  $I'_1$  has finitary semantic normal forms  $(\varphi, E)$  with  $\varphi$  injective, like in Ex. 3.3. That  $I'_2$  inherits the signature morphism is obvious as, according to [15, 3] morphisms between structured specifications  $SP_1 \rightarrow SP_2$  are defined as the signature morphisms  $\chi: sig(SP_1) \rightarrow sig(SP_2)$  such that  $Mod(SP_2) \upharpoonright_{\chi} \subseteq Mod(SP_1)$ . The third condition underlying Prop. 4.2 can be fulfilled in various ways, leading to various different translations. The most straightforward way is  $SP(\Sigma, E) = (\Sigma, E)$ , however this corresponds to having the result of the translation always in normal form. Alternatively  $SP(\Sigma, E)$  can be chosen to be a specification structured by BASIC, UNION, and TRANS. The next condition of Prop. 4.2 is obvious and the last one holds according to Ex. 2.5.

We may obtain a translation to specifications over  $FOL$  instead of  $FOL^{\text{th}}$  by composing the obtained comorphism with the canonical comorphism that maps any  $FOL^{\text{th}}$ -based specification  $SP$  to the  $FOL$ -based specification  $SP \cup (\Sigma, E)$  where  $(\Sigma, E)$  is the  $FOL^{\text{th}}$ -signature of  $SP$ .

*Acknowledgement.* 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.

## 5. Conclusions

We have shown that in the presence of a translation between the underlying logics, in essence if the source structuring has semantic normal forms and the target structuring has derivations then we can have adequate translations at the level of abstractly structured specifications. We have argued that semantic normal forms are a significantly weaker condition than the existence of syntactic normal forms, for example derivations not being necessary at the source of the translation. Due to the concepts of structured institution and structured comorphism, our result is highly general as is independent both of the underlying logic and of concrete choices of structuring operators. The latter means also that the source and the target of the translation do not have to share their respective structuring systems. These give our main result a rather wide range of applications.

## Bibliography

- [1] 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.
- [2] Ionuț Țuțu. Comorphisms for structured institutions. *Information Processing Letters*, 113(894–900), 2013.
- [3] Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.
- [4] Răzvan Diaconescu. An encoding of partial algebras as total algebras. *Information Processing Letters*, 109(23–24):1245–1251, 2009.
- [5] Răzvan Diaconescu. An axiomatic approach to structuring specifications. *Theoretical Computer Science*, 433:20–42, 2012.
- [6] Răzvan Diaconescu and Ionuț Țuțu. On the algebra of structured specifications. *Theoretical Computer Science*, 412(28):3145–3174, 2011.
- [7] Răzvan Diaconescu and Ionuț Țuțu. Foundations for structuring behavioural specifications. *Journal of Logical and Algebraic Methods in Programming*, 83(3–4):319–338, 2014.
- [8] 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.
- [9] 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.
- [10] Joseph Goguen and Grigore Roșu. Institution morphisms. *Formal Aspects of Computing*, 13:274–307, 2002.
- [11] José Meseguer. General logics. In H.-D. Ebbinghaus et al., editors, *Proceedings, Logic Colloquium, 1987*, pages 275–329. North-Holland, 1989.
- [12] Till Mossakowski. Different types of arrow between logical frameworks. In F. Meyer auf der Heide and B. Monien, editors, *Proc. ICALP 96*, volume 1099 of *Lecture Notes in Computer Science*, pages 158–169. Springer Verlag, 1996.
- [13] Marius Petria and Răzvan Diaconescu. Abstract Beth definability in institutions. *Journal of Symbolic Logic*, 71(3):1002–1028, 2006.
- [14] Donald Sannella and Andrzej Tarlecki. Specifications in an arbitrary institution. *Information and Control*, 76:165–210, 1988.
- [15] Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specifications and Formal Software Development*. Springer, 2012.
- [16] Andrzej Tarlecki. Moving between logical systems. In Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Recent Trends in Data Type Specification*, volume 1130 of *Lecture Notes in Computer Science*, pages 478–502. Springer, 1996.
- [17] Andrzej Tarlecki. Towards heterogeneous specifications. In D. Gabbay and M. van Rijke, editors, *Proceedings, International Conference on Frontiers of Combining Systems (FroCoS'98)*, pages 337–360. Research Studies Press, 2000.

- [18] Shahab Tasharofi 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.
- [19] Ionuț Țuțu. Parameterisation for abstract structured specifications. *Theoretical Computer Science*, 517:102–142, 2014.