Parameterisation for Abstract Structured Specifications

Ionuţ Ţuţu[a,b]

[a] *Department of Computer Science, Royal Holloway University of London*
[b] *Institute of Mathematics of the Romanian Academy, Research group of the project ID-3-0439*

Abstract

We investigate multiple-parameterised specifications and their instantiation within the institution-independent framework of abstract structured specifications. Our work identifies a set of distinctive features of specifications languages that have a fundamental role in defining and instantiating parameterised specifications. We consider both simultaneous and sequential instantiation of parameters, and allow not only sharing between the body of the parameterised specification and the instances of the parameters, but also between the parameters of a generic specification. The developments conclude with the examination of the relation between the results of simultaneous and sequential instantiation of parameters, which are shown to be isomorphic under a given set of sufficient abstract conditions.

Keywords: Algebraic specification, Institution theory, Structured institutions, Parameterisation, Free extensions

1. Introduction

Modularisation plays a paramount role in managing the inherent complexity of large software development projects. This paper is dedicated to the study of parameterisation as one of the most important techniques used in structuring formal specifications. Parameterisation, or generalization, allows abstracting away those elements of a formal specification whose details are not part of the essence of the specified system, and can be obtained at a later time through particularisation.

Parameterised constructions arise naturally in both mathematics and computing science. Immediate examples can be found in the theory of algorithms and data types where most of the entities are parameterised by a combination of structures, operations on structures or properties of them. For instance, one can easily see that the list structure is generic with respect to the type of its elements. Most programming or specification languages allow the definition of such data types in a manner that expresses clearly their variable components. In the case of the list structure we identify a single parameter, namely the type of its elements. The particularisations of a generic entity are obtained by instantiating its parameters. This requires fitting argument mappings for each of the parameters meant to be instantiated. For the actual situation of generic lists, we can choose to instantiate the type of list's elements, which makes no assumptions on their structure, simply by replacing it with the type of natural numbers, thus obtaining lists of natural numbers. In conclusion, parameterisation as a technical device has a double function; it allows the writer to make explicit the parameters of a generic entity and at the same time it provides a suitable mechanism for instantiating the parameters.

From the point of view of formal specifications, parameterisation has an essential function in increasing the expressive power of the specification languages that support this mechanism. When certain conditions are met by the base specification language, the systematic use of parameterisation allows the development of complex module expressions in which new parameterised specifications can be obtained by partially instantiating existing generic specifications, or instantiating their parameters with other generic specifications. As it was pointed out in [16] we can gain in this way both the specification power of richer languages and the desired properties for specification and verification of the simpler ones.

Email address: ittutu@gmail.com (Ionuţ Ţuţu)

The structure and the contents of the paper

Our paper presents an overview of the theory of parameterised specifications in the abstract framework of institution theory [18]. Its goal is to establish those properties of specification languages with a primary function in building parameterised specifications and in supporting the instantiation of the parameters such that the resulting specifications satisfy expected algebraic properties. To this effect, our work upgrades the theory of parameterised specifications introduced in [13] in two fundamental ways:

1. We consider the general case of multiple-parameterised specifications in which the parameters can interact between them and also with any additional specifications involved in the instantiation process – as opposed to the original approach, which required parameters to have distinct signatures, disjoint from the signatures of other parameters or of the possible instances of the parameters.

2. The investigations follow the recent developments in the field of abstract structured specifications [10], and in this way are independent not only of the underlying logical system but also of the actual structuring operators. Consequently, the constructions and the results discussed here can be applied uniformly to specification frameworks that are based on either model-oriented [28, 13] or property-oriented [12] studies of modularisation.

These upgrades bring a significant increase in the flexibility of parameterised specifications, which is required by the way that parameterisation is used in practice. For instance, given the following CASL [26] specification of weighted lists, in order to obtain, by instantiation, the specification WEIGHTED_LIST [NAT] of weighted lists of natural numbers, one has to accept the sharing of the specification of natural numbers by the parameter of the generic specification WEIGHTED_LIST and its instance NAT. This is achieved in our example by treating the specification NAT as an import of the parameterised specification, i.e. as an auxiliary specification that is used by the parameter and is not meant to be instantiated.

spec NAT =
  free type $Nat ::= 0 \mid s\_\_(Nat)$
end

spec NAT_PLUS =
  NAT
then op $\_\_ + \_\_ : Nat \times Nat \to Nat$
  $\forall M, N : Nat$
  • $0 + N = M$
  • $s\ M + N = s\ (M + N)$
end

spec WEIGHTED_LIST [**sort** *Elt* **op** $weight : Elt \to Nat$] **given** NAT =
  NAT_PLUS
then free type $List ::= nil \mid \_\_\_\_(Elt; List)$
  op $weight : List \to Nat$
  $\forall E : Elt; L : List$
  • $weight(nil) = 0$
  • $weight(E\ L) = weight(E) + weight(L)$
end

Note that in this situation the sharing between the parameters and their instances is explicitly stated at design time through imports. Moreover, the imports of any generic CASL specification are fixed and common to all the parameters, to their instances and also to the body of the parameterised specification. From this perspective, what is distinctive about the approach discussed in the present paper is that the sharing between the parameters and their instances, or between the body of the parameterised specification and the instances of the parameters, is not fixed, but is decided at the time of the instantiation.

Another relevant example is given by the CafeOBJ [11] specification of generic pairs listed below. The parameters of PAIR are isomorphic renamings ELT.*FIRST* and ELT.*SECOND* of the specification ELT and have the signatures

given by the sorts *Elt.FIRST* and *Elt.SECOND*, respectively. For technical reasons, unless explicitly stated otherwise, the CafeOBJ specifications implicitly protect the predefined specification BOOL of boolean values. Therefore, the symbols defined by BOOL such as *true* and *false* are shared between the parameters ELT.*FIRST* and ELT.*SECOND*, and thus it is natural to consider that sharing may also occur between the parameters of a generic specification. Although this particular situation is supported by the current implementation of the system through module sharing, the semantics of CafeOBJ requires disjoint signatures for any two different parameters of a given parameterised module. The subsequent technical sections of the paper will present a number of more elaborated examples.

module$*$ ELT {
  protecting(BOOL)
  [*Elt*]
}

module! PAIR (*FIRST* :: ELT, *SECOND* :: ELT) {
  protecting(BOOL)
  [*Pair*]
  op $\langle \_, \_ \rangle$: *Elt.FIRST Elt.SECOND* $\rightarrow$ *Pair*
}

After a brief review of some basic notions of institution theory necessary for our work, § 3 introduces the main formal device supporting the study of abstract parameterised specifications – the concept of quasi-inclusion system. It is a slight generalization of the well known notion of inclusion system [12] that allows the existence of distinct quasi-inclusions opposite one to the other. The main advantage of these more general structures over inclusion systems is the possibility to lift them from the signatures of the underlying logic to structured specifications. We also introduce here operations and properties that appear naturally in the context of quasi-inclusion systems such as the operations of union and intersection of objects, the distributivity property of a quasi-inclusion system, the compatibility of two morphisms, the preservation of an object by a morphism and the join of two compatible morphisms.

Free extensions of morphisms along quasi-inclusions, discussed in § 4, represent the universal construction which allows the instantiation of parameterised specifications to produce results that reflect the intuition about instantiation, and moreover, that fulfil certain algebraic properties of interest in the actual practice of parameterised specifications. We investigate a number of properties of free extensions that highlight their behaviour with respect to the composition and the join of morphisms. These properties will constitute the necessary foundation for the later developments.

§ 5 examines parameterised objects and their instantiation in categories endowed with an auxiliary structure that is rich enough for the study of parameterisation – a distributive quasi-inclusion system. Furthermore, we require that the considered category satisfies a number of properties that are essential to parameterisation such as the existence of free extensions for a specific class of morphisms. We begin with the analysis of the simpler case of objects that have only one parameter, and then generalize the constructions to the more elaborated case of multiple-parameterised objects. For the later we discuss two distinct instantiation procedures and identify a set of conditions that are sufficient for guaranteeing that the possible instantiation scenarios produce isomorphic results.

Since the properties considered in § 5 are too restrictive for numerous categories of signatures belonging to the foundations of various specification languages (for example the OBJ family of languages [22, 11]), in § 6 we concentrate on the basic properties of functors that would assist the study of parameterisation in the framework of abstract structured specifications. In this sense, we introduce the property of strongly lifting cocones which, together with faithfulness, allows functors to lift both quasi-inclusion systems and colimits.

Parameterisation for abstract structured specifications, at the level of structured institutions, is discussed in § 7 by referring to the concepts and results developed in § 5 for the signatures of the base institution, and to those obtained in § 6 for the structuring functor. We focus on the analysis of multiple-parameterised specifications and the considered options for instantiating them – instantiate all the parameters at once by simultaneous instantiation, or one at a time by sequential instantiation. The examination of the isomorphism relation between the results of the two instantiation procedures concludes our work.

2. Preliminaries

This section gives an account of the elementary concepts and structures necessary for our work.

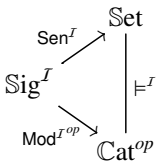
Categories

We assume some familiarity with basic notions of category theory such as functor or colimit. We use the terminology and the notations from [24], with a few exceptions. For a given category $\mathbb{C}$, we denote by $|\mathbb{C}|$ its class of objects and by $\mathbb{C}(A, B)$ the set of arrows with domain A and codomain B; the composition of arrows is considered in diagrammatic order and is denoted by ';'. A subcategory $\mathbb{C}'$ of $\mathbb{C}$ is *broad* when it has all the objects of $\mathbb{C}$, and is *full* when for every two objects A and B in $\mathbb{C}'$ we have $\mathbb{C}'(A, B) = \mathbb{C}(A, B)$. $\mathbb{S}$et designates the category having sets in the role of objects and functions in the role of arrows. The quasi-category of all categories and functors is denoted by $\mathbb{C}$at. A functor $F\colon \mathbb{C} \to \mathbb{C}'$ is *faithful* when for every two objects A and B in $\mathbb{C}$, the arrow map $F\colon \mathbb{C}(A, B) \to \mathbb{C}'(F(A), F(B))$ is injective.

Institutions

The notion of institution emerged from the general concept of language, advanced in [5]. It was introduced in [18] with the aim of formalizing the intuitive notion of logical system and thus providing a rigorous device for dealing with the increasing number of logics used as foundation for specification and programming languages.

Definition 2.1 (Institution)**.** An *institution* $\mathcal{I} = (\mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$ consists of



- a category $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ of *signatures* and *signature morphisms*,
- a *sentence functor* $\mathsf{Sen}^{\mathcal{I}} : \mathbb{S}\mathrm{ig}^{\mathcal{I}} \to \mathbb{S}\mathrm{et}$ providing, for each signature Σ, the set $\mathsf{Sen}^{\mathcal{I}}(\Sigma)$ of Σ-*sentences*, and for each signature morphism $\varphi\colon \Sigma \to \Sigma'$, the *sentence translation map* $\mathsf{Sen}^{\mathcal{I}}(\varphi)\colon \mathsf{Sen}^{\mathcal{I}}(\Sigma) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma')$,
- a *model functor* $\mathsf{Mod}^{\mathcal{I}} : \mathbb{S}\mathrm{ig}^{\mathcal{I}\,op} \to \mathbb{C}\mathrm{at}$ providing, for each signature Σ, the category $\mathsf{Mod}^{\mathcal{I}}(\Sigma)$ of Σ-*models* and Σ-*homomorphisms*, and for each morphism $\varphi\colon \Sigma \to \Sigma'$, the *reduct functor* $\mathsf{Mod}^{\mathcal{I}}(\varphi)\colon \mathsf{Mod}^{\mathcal{I}}(\Sigma') \to \mathsf{Mod}^{\mathcal{I}}(\Sigma)$,
- a family of *satisfaction* relations $\models^{\mathcal{I}}_{\Sigma} \subseteq \left|\mathsf{Mod}^{\mathcal{I}}(\Sigma)\right| \times \mathsf{Sen}^{\mathcal{I}}(\Sigma)$, indexed by signatures,

such that the following *satisfaction condition* holds:

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

for each signature morphism $\varphi\colon \Sigma \to \Sigma'$, Σ'-model M' and Σ-sentence ρ.

The reduct functor $\mathsf{Mod}^{\mathcal{I}}(\varphi)$ is often denoted by $\_{\restriction_\varphi}$, while the sentence translation $\mathsf{Sen}^{\mathcal{I}}(\varphi)$ is designated simply by φ. The satisfaction condition can also be presented diagrammatically as depicted below.

$$\begin{array}{ccc} \Sigma & & \left|\mathsf{Mod}^{\mathcal{I}}(\Sigma)\right| \xrightarrow{\models^{\mathcal{I}}_{\Sigma}} \mathsf{Sen}^{\mathcal{I}}(\Sigma) \\ {\scriptstyle\varphi}\downarrow & & \uparrow{\scriptstyle \_{\restriction_\varphi}} \qquad\qquad \downarrow{\scriptstyle\varphi} \\ \Sigma' & & \left|\mathsf{Mod}^{\mathcal{I}}(\Sigma')\right| \xrightarrow[\models^{\mathcal{I}}_{\Sigma'}]{} \mathsf{Sen}^{\mathcal{I}}(\Sigma') \end{array}$$

When the context is clear, we may omit the subscripts or superscripts from the notations introduced above. For instance, when the institution and the signature can be easily inferred, we will often denote the satisfaction relation simply by $\models$.

Example 2.1 (Many-Sorted Algebra – MSA)**.** A *many-sorted algebraic signature* is a pair (S, F) consisting of a set S of sort symbols and a family $\{F_{w\to s} \mid w \in S^*, s \in S\}$ of operation symbols, indexed by arities and sorts corresponding to the arguments and the results of the operations, respectively. A *morphism of algebraic signatures* $\varphi\colon (S, F) \to (S', F')$ consists of a translation map $\varphi^{op}\colon S \to S'$ for the sort symbols, and a family of translation maps $\{\varphi^{op}_{w\to s}\colon F_{w\to s} \to F'_{\varphi^{st}(w)\to\varphi^{st}(s)} \mid w \in S^*, s \in S\}$ for the operation symbols.

Given a signature (S, F), a *model* or *algebra* for (S, F) interprets each sort symbol $s \in S$ as a set M_s and each operation symbol $\sigma \in F_{w\to s}$ as a function $M_\sigma\colon M_w \to M_s$, where $M_{s_1\cdots s_n}$ denotes the Cartesian product $M_{s_1} \times \cdots \times M_{s_n}$. An (S, F)-*homomorphism* $h\colon M \to N$ is a family of functions $\{h_s\colon M_s \to N_s \mid s \in S\}$ indexed by the sorts such that $h_s(M_\sigma(m)) = N_\sigma(h_w(m))$ for each $\sigma \in F_{w\to s}$ and $m \in M_w$; the map $h_w\colon M_w \to N_w$ is defined as the component-wise extension of h given by $h_{s_1\cdots s_n}(m_1, \ldots, m_n) = (h_{s_1}(m_1), \ldots, h_{s_n}(m_n))$.
For a signature morphism $\varphi\colon (S, F) \to (S', F')$, the *reduct* $M'|_\varphi$ of an (S', F')-model M' is defined by $(M'|_\varphi)_x = M'_{\varphi(x)}$, for each sort or operation symbol x from (S, F).

The set of *(S, F)-sentences* is defined as the least set containing the quantifier-free atoms $t = t'$, with t and t' terms of the same sort, which is closed under the usual first order Boolean connectives and first order quantifiers. The translation of sentences along signature morphisms is defined by renaming the operation symbols, and implicitly the sort symbols as well, according to the considered signature morphism.

Finally, the satisfaction relation between models and sentences is the Tarskian satisfaction defined inductively on the structure of the sentences.

The next two examples refine the institution of many-sorted algebra primarily by considering signatures with a richer structure, obtained either by taking into account an order relation on sorts or by distinguishing between the operation symbols that are required to be interpreted as total functions and the other operation symbols, which are interpreted as partial functions.

Example 2.2 (Order-Sorted Algebra – OSA)**.** *Order-sorted signatures* [25, 19] are tuples $(S, \leq, F)$ defined by a many-sorted signature (S, F) together with a partial order $\leq$ on S that satisfies the following monotonicity condition: for any arities w_1, w_2, and sorts s_1, s_2, whenever $w_1 \leq w_2$ and the intersection $F_{w_1\to s_1} \cap F_{w_2\to s_2}$ is not empty, we also have $s_1 \leq s_2$. An *order-sorted signature morphism* $\varphi\colon (S, \leq, F) \to (S', \leq', F')$ is just a many-sorted signature morphism $\varphi\colon (S, F) \to (S', F')$ whose component on sorts is a monotonic function $\varphi^{st}\colon (S, \leq) \to (S', \leq')$.

At the semantics level, the order structure on the sorts is reflected as set-theoretic inclusions between the corresponding carriers. More precisely, an $(S, \leq, F)$-*model* is an (S, F)-algebra M that satisfies the following monotonicity conditions:

- $M_{s_1} \subseteq M_{s_2}$ whenever s_1 and s_2 are two sorts such that $s_1 \leq s_2$,
- $M_{\sigma\colon w_1\to s_1}$ and $M_{\sigma\colon w_2\to s_2}$ agree on M_{w_1}, in the sense that $M_{\sigma\colon w_1\to s_1}(m) = M_{\sigma\colon w_2\to s_2}(m)$ for all tuples $m \in M_{w_1}$, whenever $w_1 \leq w_2$ and $\sigma \in F_{w_1\to s_1} \cap F_{w_2\to s_2}$.

Similarly, an $(S, \leq, F)$-*homomorphism* $h\colon M \to N$ is an (S, F)-homomorphism such that h_{s_1} and h_{s_2} agree on M_{s_1}, for any two sorts s_1 and s_2 such that $s_1 \leq s_2$.

The sentences and their satisfaction by models are defined as in the case of many-sorted algebra (note that for any two sorts s_1 and s_2 such that $s_1 \leq s_2$, any term of sort s_1 is also a term of sort s_2).

Example 2.3 (Partial Algebra – PA)**.** A *partial algebraic signature* is a tuple (S, F, TF) defined by an algebraic signature (S, F) and a family of sets of operation symbols TF such that $TF_{w\to s} \subseteq F_{w\to s}$, for all $w \in S^*$ and $s \in S$. The symbols in TF are called *total operation symbols* while the ones in $F \setminus TF$ are called *partial operation symbols*. In the literature, the component-wise difference $F \setminus TF$ is usually denoted by PF, and the partial algebraic signatures are equivalently presented as tuples (S, TF, PF) [26, 9]. For notational convenience, we choose to emphasize only the subset of total operation symbols; we will obtain in this way, in the subsequent sections of the paper, simpler descriptions for a number of operations with signatures. A *morphism of partial algebraic signatures* $\varphi\colon (S, F, TF) \to (S', F', TF')$ is a many-sorted signature morphism $\varphi\colon (S, F) \to (S', F')$ that preserves the total operation symbols, i.e. $\varphi^{op}_{w\to s}(TF_{w\to s}) \subseteq TF'_{\varphi^{st}(w)\to\varphi^{st}(s)}$, for all $w \in S^*$ and $s \in S$.

Given a partial algebraic signature (S, F, TF), a model or *partial algebra* for (S, F, TF) interprets each sort symbol $s \in S$ as a set M_s, and each operation symbol $\sigma \in F_{w\to s}$ as a partial function $M_\sigma\colon M_w \rightharpoonup M_s$ that is total whenever σ is

total. The *partial algebra homomorphisms* $h\colon M \to N$ are defined just as in the case of many-sorted algebra, with the observation that $h_s(M_\sigma(m)) = N_\sigma(h_w(m))$ for all operation symbols $\sigma \in F_{w\to s}$ and $m \in M_w$ such that $M_\sigma(m)$ is defined.

The sentences and the satisfaction relation are also defined like in the case of may sorted algebra (with quantification only over total first order variables), but are based on three kinds of atomic sentences: def $\_$ (*definedness*), $\_ \overset{s}{=} \_$ (*strong equality*) and $\_ \overset{e}{=} \_$ (*existence equality*). The definedness def t of a term t holds in a model M when its interpretation in M, M_t, is defined. The strong equality $t \overset{s}{=} t'$ holds in M when both terms are undefined or when they are defined and have equal interpretations. The existence equality $t \overset{e}{=} t'$ holds when both terms are defined and have equal interpretations.

Structured specifications

Let us now recall the main institution-independent approaches to structuring formal specifications, namely the property-oriented one, based on the notion of theory [12], and the model-oriented one, based on the notion of structured specification [28].

In any institution $\mathcal{I} = (\mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$, for every signature $\Sigma \in \left|\mathbb{S}\mathrm{ig}^{\mathcal{I}}\right|$,

- for each Σ-model M and each set E of Σ-sentences, $M \models E$ if and only if $M \models e$, for all $e \in E$,
- for each set E of Σ-sentences, E^* denotes the class $\left\{M \in \left|\mathsf{Mod}^{\mathcal{I}}(\Sigma)\right| \;\middle|\; M \models E\right\}$,
- for each class $\mathbb{M}$ of Σ-models, $\mathbb{M}^*$ denotes the set $\{e \in \mathsf{Sen}^{\mathcal{I}}(\Sigma) \mid M \models e, \text{ for all } M \in \mathbb{M}\}$.

Definition 2.2 (Presentations and theories)**.** For every institution $\mathcal{I} = (\mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$, a *presentation* is a pair (Σ, E) consisting of a signature $\Sigma \in \left|\mathbb{S}\mathrm{ig}^{\mathcal{I}}\right|$ and a set of sentences $E \subseteq \mathsf{Sen}^{\mathcal{I}}(\Sigma)$. The presentations of $\mathcal{I}$ form a category $\mathbb{P}\mathrm{res}^{\mathcal{I}}$ having as arrows $\varphi\colon (\Sigma, E) \to (\Sigma', E')$ those signature morphisms $\varphi\colon \Sigma \to \Sigma'$ such that $E' \models^{\mathcal{I}}_{\Sigma'} \varphi(E)$; the arrows can be composed by following the definition of composition from the category of signatures $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$.

Theories are presentations (Σ, E) whose set of sentences is closed under semantic deduction, i.e. for every sentence $e \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$, $e \in E$ whenever $E \models^{\mathcal{I}}_{\Sigma} e$. They form a full subcategory of $\mathbb{P}\mathrm{res}^{\mathcal{I}}$, denoted $\mathbb{T}\mathrm{h}^{\mathcal{I}}$.

For any institution $\mathcal{I}$, the category of its presentations is equipped with a functor

$$\mathsf{Sig}\colon \mathbb{P}\mathrm{res}^{\mathcal{I}} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$$

defined by $\mathsf{Sig}(\Sigma, E) = \Sigma$ on presentations and $\mathsf{Sig}(\varphi) = \varphi$ on morphisms of presentations.

The following definition of structured specifications was first presented in [13]. It extends the set of specification building operators considered in [3] with new ones for non-protecting extensions and initial semantics. Unlike the description considered in [13], the union of specifications is defined here as partial rather than total, as in [28], and thus it no longer relies on an inclusive base institution.

Definition 2.3 (Structured specifications)**.** Let us consider an institution $\mathcal{I} = (\mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$ and fix two classes of signature morphisms $\mathcal{T}$ and $\mathcal{D}$. The $(\mathcal{T}, \mathcal{D})$-structured specifications of $\mathcal{I}$ are obtained from finite presentations by applying the specification building operators listed below. For each structured specification SP we consider its signature $\mathrm{Sig}[SP]$, its set of axioms $\mathrm{Ax}[SP]$ and its class of models $\mathrm{Mod}[SP]$. Together, the signature and the class of models of a specification describe its semantics.

PRES. Any *finite presentation* (Σ, E) is a structured specification such that

- $\mathrm{Sig}[(\Sigma, E)] = \Sigma$,
- $\mathrm{Ax}[(\Sigma, E)] = E$,
- $\mathrm{Mod}[(\Sigma, E)] = E^*$.

UNION. For any specifications SP_1 and SP_2 with the same signature Σ, their *union* $SP_1 \cup SP_2$ is also a structured specification, with

- $\mathrm{Sig}[SP_1 \cup SP_2] = \Sigma$,

– $\mathrm{Ax}[SP_1 \cup SP_2] = \mathrm{Ax}[SP_1] \cup \mathrm{Ax}[SP_2]$,

– $\mathrm{Mod}[SP_1 \cup SP_2] = \mathrm{Mod}[SP_1] \cap \mathrm{Mod}[SP_2]$.

TRANS. For any specification SP and signature morphism $\varphi\colon \mathrm{Sig}[SP] \to \Sigma'$ in $\mathcal{T}$, the *translation* of SP along φ, denoted $SP \star \varphi$, is a structured specification having

– $\mathrm{Sig}[SP \star \varphi] = \Sigma'$,

– $\mathrm{Ax}[SP \star \varphi] = \varphi(\mathrm{Ax}[SP])$,

– $\mathrm{Mod}[SP \star \varphi] = \left\{ M' \in \left| \mathsf{Mod}^{\mathcal{I}}(\Sigma') \right| \;\middle|\; M'|_{\varphi} \in \mathrm{Mod}[SP] \right\}$.

DERIV. For any specification SP' and signature morphism $\varphi\colon \Sigma \to \mathrm{Sig}[SP']$ in $\mathcal{D}$, the *derivation* of SP' along φ, denoted $\varphi \mid SP'$, is a structured specification with

– $\mathrm{Sig}[\varphi \mid SP'] = \Sigma$,

– $\mathrm{Ax}[\varphi \mid SP'] = \varphi^{-1}(\mathrm{Ax}[SP']^{**})$,

– $\mathrm{Mod}[\varphi \mid SP'] = \{M'|_{\varphi} \mid M' \in \mathrm{Mod}[SP']\}$.

$\mathcal{H}$-EXT. Given a family $\mathcal{H} = \{\mathcal{H}_\Sigma\}_{\Sigma \in |\mathbb{S}\mathrm{ig}^{\mathcal{I}}|}$ of classes of homomorphisms, the *$\mathcal{H}$-extension* of a specification SP, denoted $\mathcal{H}(SP)$, is a structured specification such that

– $\mathrm{Sig}[\mathcal{H}(SP)] = \mathrm{Sig}[SP]$,

– $\mathrm{Ax}[\mathcal{H}(SP)] = \mathrm{Ax}[SP]$,

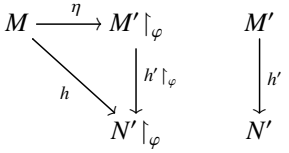
– $\mathrm{Mod}[\mathcal{H}(SP)] = \left\{ M' \in \left| \mathsf{Mod}^{\mathcal{I}}(\mathrm{Sig}[SP]) \right| \;\middle|\; M' \models \mathrm{Ax}[SP] \text{ and there exists } h\colon M \to M' \text{ in } \mathcal{H}_{\mathrm{Sig}[SP]} \text{ with } M \in \mathrm{Mod}[SP] \right\}$.

$\mathcal{H}$-FREE. Given a family $\mathcal{H} = \{\mathcal{H}_\Sigma\}_{\Sigma \in |\mathbb{S}\mathrm{ig}^{\mathcal{I}}|}$ of classes of homomorphisms, for any two specifications SP and SP', and any signature morphism $\varphi\colon \mathrm{Sig}[SP] \to \mathrm{Sig}[SP']$, the *$\mathcal{H}$-free restriction* of SP' to SP through φ, denoted $SP' \,!_{\mathcal{H}}\, (\varphi, SP)$, is a structured specification such that

– $\mathrm{Sig}[SP' \,!_{\mathcal{H}}\, (\varphi, SP)] = \mathrm{Sig}[SP']$,

– $\mathrm{Ax}[SP' \,!_{\mathcal{H}}\, (\varphi, SP)] = \mathrm{Ax}[SP']$,

– $\mathrm{Mod}[SP' \,!_{\mathcal{H}}\, (\varphi, SP)] = \{M' \in \mathrm{Mod}[SP'] \mid$ there exists $M \in \mathrm{Mod}[SP]$ and $\eta\colon M \to M'|_{\varphi}$ in $\mathcal{H}_{\mathrm{Sig}[SP]}$ such that for all $h\colon M \to N'|_{\varphi}$ with $N' \in \mathrm{Mod}[SP']$ there exists a unique arrow $h'\colon M' \to N'$ that satisfies $h = \eta; h'|_{\varphi}\}$.



For any institution $\mathcal{I}$ and classes of signature morphisms $\mathcal{T}$ and $\mathcal{D}$, the $(\mathcal{T}, \mathcal{D})$-structured specifications form a category $\mathbb{S}\mathrm{pec}^{\mathcal{I}}$ whose arrows $\varphi\colon SP \to SP'$ are signature morphisms $\varphi\colon \mathrm{Sig}[SP] \to \mathrm{Sig}[SP']$ such that for all models $M' \in \mathrm{Mod}[SP']$ it holds that $M'|_{\varphi} \in \mathrm{Mod}[SP]$; as in the case of presentation morphisms, the composition of specification morphisms is inherited from the category of signatures $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$. The signature map $\mathrm{Sig}[\_]$ can be straightforwardly extended to a functor

$$\mathsf{Sig}\colon \mathbb{S}\mathrm{pec}^{\mathcal{I}} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$$

defined by $\mathsf{Sig}(SP) = \mathrm{Sig}[SP]$ on structured specifications and $\mathsf{Sig}(\varphi) = \varphi$ on morphisms of specifications.

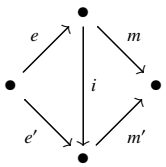
3. Quasi-inclusions

The idea of axiomatising inclusions in an abstract categorical framework and using them as a fundamental device for modularisation of formal specifications, in a way that is independent of any underlying logic, was first suggested in [18]. The first answer was given in [12] with the introduction of the abstract notion of inclusion system, an instrument whose purpose was to provide a categorical description for the extension of theories. Inclusion systems have been further developed in [6, 7], and used throughout a series of works in the general study of module algebra [9, 21] and within the abstract framework of institution-independent model theory [9], with focus on definability [2] and axiomatisability [8, 27, 20]. This paper continues the categorical study of inclusions as a fundamental concept in formalizing operations such as importing, hiding and parameterisation for abstract structured specifications. We consider the slightly more general notion of quasi-inclusion system which has more similarities with the concept of image factorization system [23]. Quasi-inclusion systems can be regarded as the categorical counterpart of entailment relations, in the same way as inclusion systems correspond categorically to set-theoretic inclusions, and image factorization systems to set-theoretic injections.

Throughout the literature there are many equivalent definitions, or different only by a small extent, of both inclusion systems [12, 6, 9] and image factorization systems [23, 17, 18]. Let us first recall some basic definitions and properties about these notions from a point of view we consider closest to our approach.

Definition 3.1 (Image factorization system)**.** An *image factorization system* for a category $\mathbb{C}$ consists of a pair $\langle \mathbb{M}, \mathbb{E} \rangle$ of broad subcategories of $\mathbb{C}$ such that

- all morphisms of $\mathbb{E}$ are epimorphisms in $\mathbb{C}$,
- all morphisms of $\mathbb{M}$ are monomorphisms in $\mathbb{C}$,
- all isomorphisms of $\mathbb{C}$ are in both $\mathbb{M}$ and $\mathbb{E}$, and
- any morphism f of $\mathbb{C}$ can be factored as $f = e; m$, with $e \in \mathbb{E}$ and $m \in \mathbb{M}$, uniquely up to isomorphism, i.e. for any other factorization $e'; m'$ of f such that $e' \in \mathbb{E}$ and $m' \in \mathbb{M}$, there exists a unique isomorphism i of $\mathbb{C}$ such that $e; i = e'$ and $i; m' = m$.



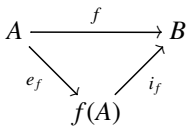
Inclusion systems are more convenient than image factorization systems from the point of view of modularisation. One of their main advantages is that subobjects are uniquely presented through specific morphisms instead of equivalence classes of morphisms.

Definition 3.2 (Inclusion system)**.** An *inclusion system* for a category $\mathbb{C}$ consists of a pair $\langle \mathbb{I}, \mathbb{E} \rangle$ of broad subcategories of $\mathbb{C}$ such that

- $\mathbb{I}$ is a partial order, i.e. a category such that (a) between any two objects there can be at most one arrow and (b) any isomorphism is an identity, and
- any morphism f in $\mathbb{C}$ can be factored uniquely as $f = e_f; i_f$, with $e_f \in \mathbb{E}$ and $i_f \in \mathbb{I}$.

The morphisms of $\mathbb{I}$ and $\mathbb{E}$ are called *abstract inclusions* and *abstract surjections*, respectively. The domain of the inclusion i_f involved in the factorization of a morphism $f\colon A \to B$ is called the *image* of f and is denoted by $f(A)$. Since $\mathbb{I}$ is a partial order, and thus, for any two objects A and B, the set $\mathbb{I}(A, B)$ can have at most one element, we may denote the inclusion from A to B, when it exists, with $A \subseteq B$.

Example 3.1. In the category $\mathbb{S}$et of sets and functions, the set-theoretic inclusions in the role of abstract inclusions together with the surjective functions in the role of abstract surjections form an inclusion system.



Example 3.2. In the category $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ of many-sorted signatures we can consider the following two non-trivial inclusion systems:

the strong $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ *inclusion system,* obtained by defining: (*a*) the abstract inclusions $(S,F) \xrightarrow{\subseteq} (S',F')$ as those signature morphisms having all the components $S \xrightarrow{\subseteq} S'$ and $F_{w\to s} \xrightarrow{\subseteq} F'_{w\to s}$, for all $w \in S^*$ and $s \in S$, as set-theoretic inclusions; (*b*) the abstract surjections as those signature morphisms $\varphi\colon (S,F) \to (S',F')$ such that $S' = \varphi^{st}(S)$ and $F'_{w'\to s'} = \bigcup_{\varphi^{st}(ws)=w's'} \varphi^{op}_{w\to s}(F_{w\to s})$, for all $w' \in S'^*$ and $s' \in S'$;

the closed $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ *inclusion system,* obtained by defining: (*a*) the abstract inclusions $(S,F) \xrightarrow{\subseteq} (S',F')$ as those signature morphisms whose components are all set-theoretic inclusions, with $F_{w\to s} = F'_{w\to s}$, for all $w \in S^*$ and $s \in S$; (*b*) the abstract surjections as those signature morphisms $\varphi\colon (S,F) \to (S',F')$ that are surjective on sorts.

Similar strong and closed inclusion systems can be defined for various categories of signatures related to the many-sorted ones described above. When the context is clear, we may also use the terms 'strong abstract surjection' or 'closed abstract inclusion' to refer to a particular strong or closed inclusion system.

Example 3.3. For the category $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ of order-sorted signatures, we take into account the following two inclusion systems:

the strong $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ *inclusion system,* in which: (*a*) the abstract inclusions $(S,\leq,F) \xrightarrow{\subseteq} (S',\leq',F')$ are signature morphisms with the $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reduct $(S,F) \xrightarrow{\subseteq} (S',F')$ an abstract inclusion in the strong sense; (*b*) the abstract surjections are signature morphisms $\varphi\colon (S,\leq,F) \to (S',\leq',F')$ such that the $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reduct $\varphi\colon (S,F) \to (S',F')$ is a strong abstract surjection and $\leq'$ is the least monotonic partial order on S' that includes the image of $\leq$ through φ^{st};

the closed $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ *inclusion system,* in which: (*a*) the abstract inclusions $(S,\leq,F) \xrightarrow{\subseteq} (S',\leq',F')$ are signature morphisms such that the $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reduct $(S,F) \xrightarrow{\subseteq} (S',F')$ is a closed abstract inclusion, and $\leq$ is the restriction of $\leq'$ to S; (*b*) the abstract surjections are just morphisms $\varphi\colon (S,\leq,F) \to (S',\leq',F')$ whose $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reducts $\varphi\colon (S,F) \to (S',F')$ are abstract surjections in the closed sense.

Example 3.4. For the category $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{PA}}}$ of partial algebraic signatures, we take into account the following two inclusion systems:

the strong $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{PA}}}$ *inclusion system,* in which: (*a*) the abstract inclusions $(S,F,TF) \xrightarrow{\subseteq} (S',F',TF')$ are signature morphisms whose $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reducts $(S,F) \xrightarrow{\subseteq} (S',F')$ are abstract inclusions in the strong sense; (*b*) the abstract surjections are morphisms $\varphi\colon (S,F,TF) \to (S',F',TF')$ such that the $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reduct $\varphi\colon (S,F) \to (S',F')$ is a strong abstract surjection and $TF'_{w'\to s'} = \bigcup_{\varphi^{st}(ws)=w's'} \varphi^{op}_{w\to s}(TF_{w\to s})$, for all $w' \in S'^*$ and $s' \in S'$;

the closed $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{PA}}}$ *inclusion system,* in which: (*a*) the abstract inclusions $(S,F,TF) \xrightarrow{\subseteq} (S',F',TF')$ are signature morphisms with the $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reduct $(S,F) \xrightarrow{\subseteq} (S',F')$ a closed abstract inclusion and $TF_{w\to s} = TF'_{w\to s}$, for all $w \in S^*$ and $s \in S$; (*b*) the abstract surjections are just morphisms $\varphi\colon (S,F,TF) \to (S',F',TF')$ whose $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$-reducts $\varphi\colon (S,F) \to (S',F')$ are abstract surjections in the closed sense.

In some cases it is possible to derive inclusion systems for more complex categories from given inclusion systems of simpler, base categories. A first example was reported in [12], with a more comprehensive analysis in [9].

Example 3.5. Whenever we have an institution whose category of signatures admits an inclusion system, the category of its theories can inherit the inclusion system from the category of signatures in the following two ways:

the strong theory inclusion system, having as abstract inclusions the theory morphisms $(\Sigma, E) \xrightarrow{\subseteq} (\Sigma', E')$ with the underlying signature morphism $\Sigma \xrightarrow{\subseteq} \Sigma'$ an abstract inclusion, and as abstract surjections the theory morphisms $\varphi\colon (\Sigma, E) \to (\Sigma', E')$ such that the underlying signature morphism $\varphi\colon \Sigma \to \Sigma'$ is an abstract surjection satisfying $E' = \varphi(E)^{**}$.

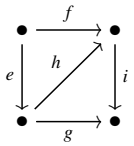
$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\varphi} & (\Sigma', E') \\ & {\scriptstyle e_\varphi}\searrow \quad \nearrow{\scriptstyle i_\varphi} & \\ & (\varphi(\Sigma), e_\varphi(E)^{**}) & \end{array}$$

the closed theory inclusion system, having as abstract inclusions the theory morphisms $\varphi\colon (\Sigma, E) \xrightarrow{\subseteq} (\Sigma', E')$ such that the underlying signature morphism $\varphi\colon \Sigma \xrightarrow{\subseteq} \Sigma'$ is an inclusion satisfying $E = \varphi^{-1}(E')$, and as abstract surjections the theory morphisms $(\Sigma, E) \to (\Sigma', E')$ with the underlying signature morphism $\Sigma \to \Sigma'$ an abstract surjection.

$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\varphi} & (\Sigma', E') \\ & {\scriptstyle e_\varphi}\searrow \quad \nearrow{\scriptstyle i_\varphi} & \\ & (\varphi(\Sigma), i_\varphi^{-1}(E')) & \end{array}$$

The following property first appeared in the study of factorization systems [23].

Definition 3.3. For any subcategory $\mathbb{I}$ of a category $\mathbb{C}$, let $\mathbb{E}_\mathbb{I}$ denote the class of all morphisms e of $\mathbb{C}$ for which the following *diagonal-fill property* holds: for any morphisms $f, g \in \mathbb{C}$ and $i \in \mathbb{I}$ such that $f; i = e; g$ there exists a morphism h satisfying $e; h = f$ and $h; i = g$.



Fact 3.1. For any subcategory $\mathbb{I}$ of a category $\mathbb{C}$, $\mathbb{E}_\mathbb{I}$ is a broad subcategory of $\mathbb{C}$.

The diagonal-fill property proved to be an essential technical device in showing that, for any inclusion system $\langle \mathbb{I}, \mathbb{E} \rangle$ of a category $\mathbb{C}$, the subcategory $\mathbb{E}$ of abstract surjections is determined by $\mathbb{I}$. An equivalent definition of inclusion systems, based only on the subcategory of abstract inclusions, was first proposed in [6]. To this effect, an *inclusion system* of a category $\mathbb{C}$ consists of a broad subcategory $\mathbb{I}$ of $\mathbb{C}$ such that

- $\mathbb{I}$ is a partial order,
- for any two composable morphisms $f \in \mathbb{C}$ and $i \in \mathbb{I}$, if $f; i \in \mathbb{I}$ then $f \in \mathbb{I}$, and
- any morphism $f \in \mathbb{C}$ can be factored as $e_f; i_f$, for some $e_f \in \mathbb{E}_\mathbb{I}$ and $i_f \in \mathbb{I}$.

Starting from this characterisation of inclusion systems, we obtain the concept of quasi-inclusion system by dropping the antisymmetry property of the subcategory of abstract inclusions, i.e. we no longer require that any two abstract inclusions dual one to the other are equal.

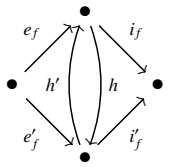
Definition 3.4 (Quasi-inclusion system)**.** A *quasi-inclusion system* of a category $\mathbb{C}$ is a broad preordered subcategory $\mathbb{I}$ of $\mathbb{C}$ such that

- for any two composable morphisms $f \in \mathbb{C}$ and $i \in \mathbb{I}$, if $f; i \in \mathbb{I}$ then $f \in \mathbb{I}$, and
- any morphism $f \in \mathbb{C}$ can be factored as $e_f; i_f$, where $e_f \in \mathbb{E}_\mathbb{I}$ and $i_f \in \mathbb{I}$.

The morphisms of $\mathbb{I}$ are called *abstract quasi-inclusions* or *quasi-inclusive morphisms* when the intent is to differentiate them from other morphisms of $\mathbb{C}$; the morphisms of $\mathbb{E}_\mathbb{I}$ are called *abstract surjections*. We often denote a quasi-inclusion $i\colon A \to B$ by $A \sqsubseteq B$.

Proposition 3.1 (Uniqueness of factorization)**.** *Given a quasi-inclusion system $\mathbb{I}$ of a category $\mathbb{C}$, every morphism $f \in \mathbb{C}$ can be factored as $e_f; i_f$, with $e_f \in \mathbb{E}_\mathbb{I}$ and $i_f \in \mathbb{I}$, uniquely up to a quasi-inclusive isomorphism, i.e. for any two factorizations $e_f; i_f = e'_f; i'_f$ with $e_f, e'_f \in \mathbb{E}_\mathbb{I}$ and $i_f, i'_f \in \mathbb{I}$ there exists a (unique) isomorphism $h \in \mathbb{I}$ such that $e_f; h = e'_f$ and $h; i'_f = i_f$.*

Proof. Let us consider two factorizations $e_f; i_f$ and $e'_f; i'_f$ of a morphism f. By the diagonal-fill property of the abstract surjections e_f and e'_f we obtain two morphisms h and h' such that $e_f; h = e'_f$, $h; i'_f = i_f$, $e'_f; h' = e_f$ and $h'; i_f = i'_f$.



Since $h; i'_f$ and i'_f are both quasi-inclusions we deduce that h is a quasi-inclusion as well. Analogously, h' is also a quasi-inclusion. Therefore, because the category of abstract inclusions is a preorder, we conclude that h and h' are isomorphisms inverse one to the other. □

Proposition 3.1 allows us to consider the *image* of a morphism $f\colon A \to B$, denoted $f(A)$, as the domain of the quasi-inclusion i_f. Notice that, because i_f is not uniquely determined by f, neither is $f(A)$, but they are uniquely determined up to a quasi-inclusive isomorphism. Whenever there is a risk of confusion we will explicitly state the abstract surjection e_f and the quasi-inclusion i_f involved in the factorization.

The converse of Proposition 3.1 holds trivially.

Fact 3.2. Let $\mathbb{I}$ be a quasi-inclusion system of a category $\mathbb{C}$. If $f\colon A \to B$ and $g\colon A \to B$ are two morphisms such that $f(A) \cong_\mathbb{I} g(A)$ and $e_f; (f(A) \cong_\mathbb{I} g(A)) = e_g$ then $f = g$.

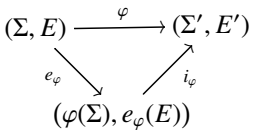
Notice that, for an arbitrary but fixed institution, in general we cannot lift the inclusion system of signatures to the category of presentations due to the uniqueness requirement of the factorization. More precisely, it is often the case that a particular closed set of sentences can be represented by a multitude of semantically equivalent subsets. For such subsets we can get presentation morphisms dual one to the other, and with identities as underlying signature morphisms, that are not necessarily equal.

$$(\Sigma, E_1) \underset{1_\Sigma}{\overset{1_\Sigma}{\rightleftarrows}} (\Sigma, E_2)$$

This is no longer an impediment if we are working with quasi-inclusion systems.

Proposition 3.2. *Let $\mathcal{I}$ be an institution such that the category of signatures $\mathbb{S}\mathrm{ig}^\mathcal{I}$ admits a quasi-inclusion system. Then we can consider the following two quasi-inclusion systems for the category $\mathbb{P}\mathrm{res}^\mathcal{I}$ of $\mathcal{I}$-presentations:*

the strong presentation quasi-inclusion system, having as abstract quasi-inclusions the morphisms of presentations $(\Sigma, E) \xrightarrow{\sqsubseteq} (\Sigma', E')$ with the underlying signature morphism $\Sigma \xrightarrow{\sqsubseteq} \Sigma'$ a quasi-inclusion; in this case, the abstract surjections are presentation morphisms $\varphi\colon (\Sigma, E) \to (\Sigma', E')$ such that the underlying signature morphism $\varphi\colon \Sigma \to \Sigma'$ is an abstract surjection satisfying $E' \models\!\models \varphi(E)$.



the closed presentation quasi-inclusion system, having as abstract quasi-inclusions the morphisms of presentations $\varphi\colon (\Sigma, E) \xrightarrow{\sqsubseteq} (\Sigma', E')$ *such that the underlying signature morphism* $\varphi\colon \Sigma \xrightarrow{\sqsubseteq} \Sigma'$ *is a quasi-inclusion satisfying* $E \between \varphi^{-1}(E'^{**})$*; the abstract surjections are presentation morphisms* $(\Sigma, E) \to (\Sigma', E')$ *with the underlying signature morphism* $\Sigma \to \Sigma'$ *an abstract surjection.*

$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\quad\varphi\quad} & (\Sigma', E') \\ & {\scriptstyle e_\varphi}\searrow \quad \nearrow{\scriptstyle i_\varphi} & \\ & (\varphi(\Sigma), i_\varphi^{-1}(E'^{**})) & \end{array}$$

Proof. Since the strong presentation quasi-inclusion system will prove to be more relevant for our study, we will focus here only on its proof. The case of the closed presentation quasi-inclusion system can be treated similarly.

According to the definition we can easily see that the quasi-inclusions of presentations form a broad preordered subcategory of $\mathbb{P}\mathrm{res}^{\mathcal{I}}$ and that a presentation morphism φ is inclusive whenever $\varphi;\iota$ is inclusive, for some quasi-inclusion ι. Thus, all we need to show is that the abstract surjections for $\mathbb{P}\mathrm{res}^{\mathcal{I}}$ are precisely the presentation morphisms $\varphi\colon (\Sigma, E) \to (\Sigma', E')$ that are abstract surjections for $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ and satisfy $E' \between \varphi(E)$.

For the direct implication we first need to prove the diagonal-fill property for φ in $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$. Let us consider two signature morphisms $\theta\colon \Sigma \to \Sigma_1$ and $\theta'\colon \Sigma' \to \Sigma'_1$ and a quasi-inclusion of signatures $\iota\colon \Sigma_1 \to \Sigma'_1$ such that $\theta;\iota = \varphi;\theta'$.

$$\begin{array}{ccc} \Sigma & \xrightarrow{\theta} & \Sigma_1 \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\iota} \\ \Sigma' & \xrightarrow[\theta']{} & \Sigma'_1 \end{array} \qquad\qquad \begin{array}{ccc} (\Sigma, E) & \xrightarrow{\theta} & (\Sigma_1, \theta(E)) \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\iota} \\ (\Sigma', E') & \xrightarrow[\theta']{} & (\Sigma'_1, \theta'(E')) \end{array}$$

Since φ is a morphism of presentations we have $E' \models \varphi(E)$, and thus $\theta'(E') \models \theta'(\varphi(E)) = \iota(\theta(E))$. Therefore, we can extend the morphisms of signatures θ, θ' and ι to morphisms of presentations as in the diagram depicted above. By the diagonal-fill property of φ in $\mathbb{P}\mathrm{res}^{\mathcal{I}}$ we deduce that there exists a mediating signature morphisms $\xi\colon \Sigma' \to \Sigma_1$ such that $\varphi;\xi = \theta$ and $\xi;\iota = \theta'$.

Let us now concentrate on the semantic equivalence $E' \between \varphi(E)$. Since $E' \models \varphi(E)$ is a direct consequence of the fact that φ is a morphisms of presentations, we only need to prove that $\varphi(E) \models E'$. This follows easily by considering the commutative square of presentation morphisms depicted below. By the diagonal-fill property of φ in $\mathbb{P}\mathrm{res}^{\mathcal{I}}$ we deduce that $1_{\Sigma'}$ is a presentation morphism $(\Sigma', E') \to (\Sigma', \varphi(E))$, and thus $\varphi(E) \models E'$.

$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\varphi} & (\Sigma', \varphi(E)) \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle 1_{\Sigma'}} \\ (\Sigma', E') & \xrightarrow[1_{\Sigma'}]{} & (\Sigma', E') \end{array}$$

For the opposite implication let $\theta\colon (\Sigma, E) \to (\Sigma_1, E_1)$ and $\theta'\colon (\Sigma', E') \to (\Sigma'_1, E'_1)$ be two morphisms of presentations and $\iota\colon (\Sigma_1, E_1) \to (\Sigma'_1, E'_1)$ a quasi-inclusion such that $\theta;\iota = \varphi;\theta'$. According to the diagonal-fill property of φ in $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ we know there exists a signature morphism $\xi\colon \Sigma' \to \Sigma_1$ such that $\varphi;\xi = \theta$ and $\xi;\iota = \theta'$. Moreover, due to the fact that θ is a morphism of presentations, it follows that $E_1 \models \xi(\varphi(E))$, and thus $E_1 \models \xi(E')$ because $E' \between \varphi(E)$. Since the morphisms θ, θ' and ι were considered arbitrary, we conclude that φ has the diagonal-fill property in $\mathbb{P}\mathrm{res}^{\mathcal{I}}$.

$$\begin{array}{ccc} (\Sigma, E) & \xrightarrow{\theta} & (\Sigma_1, E_1) \\ {\scriptstyle\varphi}\downarrow & & \downarrow{\scriptstyle\iota} \\ (\Sigma', E') & \xrightarrow[\theta']{} & (\Sigma'_1, E'_1) \end{array}$$

□

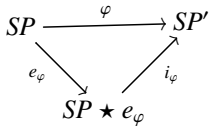
Proposition 3.2 illustrates how quasi-inclusion systems can capture categorically the entailment relations between the sets of sentences of an institution. More precisely, by considering a fixed signature, we reduce presentations to sets of sentences for which quasi-inclusions $E \xrightarrow{\sqsubseteq} E'$ exist if and only if $E' \models E$.

The construction of the quasi-inclusion systems for the category of presentations of an institution can be generalized without difficulty to the case of structured specifications. Although these are the only examples of quasi-inclusion systems in the paper that are not examples of inclusion systems as well, we prefer the concept of quasi-inclusion system because it is sufficient from a technical point of view for the development of parameterisation and at the same time it has weaker conditions that need to be checked in the actual situations.

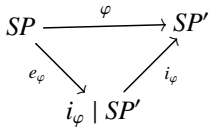
The next statement can be straightforwardly proved by following the same lines as in the proof of Proposition 3.2, with the translation operator as a natural upgrade of the translation of sentences along signature morphisms. For this reason we will omit the proof here.

Proposition 3.3. *If $\mathcal{I}$ is an institution whose category of signatures $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ admits a quasi-inclusion system, the category $\mathbb{S}\mathrm{pec}^{\mathcal{I}}$ of $\mathcal{I}$-structured specifications can inherit the quasi-inclusion system of the signatures in the following two ways:*

the strong specification quasi-inclusion system, having as quasi-inclusions the specification morphisms $SP \xrightarrow{\sqsubseteq} SP'$ with the underlying signature morphism $\mathsf{Sig}(SP) \xrightarrow{\sqsubseteq} \mathsf{Sig}(SP')$ a quasi-inclusion, and as abstract surjections the specification morphisms $\varphi\colon SP \to SP'$ such that the underlying signature morphism $\varphi\colon \mathsf{Sig}(SP) \to \mathsf{Sig}(SP')$ is an abstract surjection satisfying $SP' \between SP \star \varphi$, i.e. $\mathrm{Sig}[SP'] = \mathrm{Sig}[SP \star \varphi]$ and $\mathrm{Mod}[SP'] = \mathrm{Mod}[SP \star \varphi]$.



the closed specification quasi-inclusion system, having as quasi-inclusions the specification morphisms $SP \xrightarrow{\sqsubseteq} SP'$ such that the underlying signature morphism $\mathsf{Sig}(SP) \xrightarrow{\sqsubseteq} \mathsf{Sig}(SP')$ is a quasi-inclusion satisfying $SP \between \iota \mid SP'$, where ι denotes the quasi-inclusion $\mathrm{Sig}[SP] \sqsubseteq \mathrm{Sig}[SP']$, and as abstract surjections the specification morphisms $\varphi\colon SP \to SP'$ with the underlying signature morphism $\varphi\colon \mathsf{Sig}(SP) \to \mathsf{Sig}(SP')$ an abstract surjection.



As we might expect, quasi-inclusion systems share many elementary properties of inclusion systems.

Proposition 3.4. *If $\mathbb{C}$ is a category endowed with a quasi-inclusion system $\mathbb{I}$,*

- *all quasi-inclusions are monomorphisms,*
- *all isomorphisms of $\mathbb{C}$ are abstract surjections,*
- *any retract of $\mathbb{C}$ is an abstract surjection.*

Proof. The first property is an immediate consequence of the Proposition 3.1 and Fact 3.2 discussed above, while the second one can be easily proved by checking the diagonal-fill property holds for isomorphisms.

For the third statement, let $f\colon A \to B$ be a retract in $\mathbb{C}$, i.e. there exists $g\colon B \to A$ such that $g; f = 1_B$. By factoring f as $e_f; i_f$ we obtain $g; e_f; i_f = 1_B$ which implies that $g; e_f$ is a quasi-inclusion. Since i_f is the quasi-inclusion $f(A) \sqsubseteq B$ and $g; e_f$ is the quasi-inclusion $B \sqsubseteq f(A)$, it follows that i_f is a quasi-inclusive isomorphism, and hence an abstract surjection. From this observation we conclude that f is an abstract surjection. □

Definition 3.5. A quasi-inclusion system $\mathbb{I}$ of a category $\mathbb{C}$

– has *unions* when it has least upper bounds, i.e. coproducts in $\mathbb{I}$, denoted by $\sqcup$,

– has *intersections* when it has greatest lower bounds, i.e. products in $\mathbb{I}$, denoted by $\sqcap$,

– is *distributive* when it has unions and intersections, and for any three objects A, B and C the following two properties hold:

$$A \sqcap (B \sqcup C) \cong_{\mathbb{I}} (A \sqcap B) \sqcup (A \sqcap C) \quad \text{and} \quad A \sqcup (B \sqcap C) \cong_{\mathbb{I}} (A \sqcup B) \sqcap (A \sqcup C).$$

Note that the universal constructions in the category of quasi-inclusions are not necessarily unique, but unique only up to a quasi-inclusive isomorphism. When there is no danger of ambiguity, the union $A \sqcup B$ (or intersection $A \sqcap B$) of two objects A and B denotes any representative of the considered class of isomorphic objects.

Example 3.6. The inclusion system of $\mathbb{S}\mathrm{et}$ is distributive, with the set-theoretic unions and intersections in the role of their respective categorical counterparts.

Example 3.7. For the category $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ of many-sorted signatures, the strong inclusion system is distributive. The union $(S_\cup, F_\cup)$ of two many-sorted algebraic signatures (S_1, F_1) and (S_2, F_2) is defined by the cospan of inclusions $(S_1, F_1) \xrightarrow{\subseteq} (S_\cup, F_\cup) \xleftarrow{\supseteq} (S_2, F_2)$, where

– $S_\cup = S_1 \cup S_2$,

– $(F_\cup)_{w \to s} = \bigcup_{\substack{i \in \{1,2\} \\ w \in S_i^*,\, s \in S_i}} (F_i)_{w \to s}$, for all $w \in S_\cup^*$ and $s \in S_\cup$.

The intersection $(S_\cap, F_\cap)$ of two many-sorted signatures (S_1, F_1) and (S_2, F_2) is given by the pair of inclusions $(S_1, F_1) \xleftarrow{\supseteq} (S_\cap, F_\cap) \xrightarrow{\subseteq} (S_2, F_2)$, where

– $S_\cap = S_1 \cap S_2$,

– $(F_\cap)_{w \to s} = (F_1)_{w \to s} \cap (F_2)_{w \to s}$, for all $w \in S_\cap^*$ and $s \in S_\cap$.

Let us note that the closed $\underline{\mathrm{MSA}}$ inclusion system does not have unions because the abstract inclusions forbid the introduction of new operation symbols with already known arities and sorts. For example, we may choose the signatures $(\{\mathtt{s}\}, \{\mathtt{c} \colon [] \to \mathtt{s}\})$ and $(\{\mathtt{s}\}, \{\mathtt{c}' \colon [] \to \mathtt{s}\})$ which do not admit a least upper bound. This entails that for the algebraic signatures considered above none of the closed inclusion systems has unions, and hence none of them is distributive.

A similar observation can be made about the strong inclusion system of $\underline{\mathrm{OSA}}$ signatures. As a result of the antisymmetry of the order relation on sorts, some pairs of signatures may fail to admit upper bounds. One of the simplest examples of pairs with this property is given by the signatures $(\{\mathtt{s}, \mathtt{s}'\}, \{\mathtt{s} \leq \mathtt{s}'\}, \emptyset)$ and $(\{\mathtt{s}, \mathtt{s}'\}, \{\mathtt{s}' \leq \mathtt{s}\}, \emptyset)$.

We regard the existence of unions of signatures as a necessary condition even for the most elementary developments on the modularisation of formal specifications that take into account the possible sharing between entities. For this reason, although this may imply more meticulous reasoning about syntactic elements such as terms, we follow the lines of [26] and redefine the notion of *order-sorted signature* to correspond to tuples $(S, \leq, F)$ such that (S, F) is a many-sorted algebraic signature and $\leq$ is a *preorder* relation on S.

Example 3.8. In $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$, the strong inclusion system of (preorder-based) order-sorted signatures has unions and intersections. The union $(S_\cup, \leq_\cup, F_\cup)$ of two $\underline{\mathrm{OSA}}$ signatures $(S_1, \leq_1, F_1)$ and $(S_2, \leq_2, F_2)$ is defined by the cospan of inclusions $(S_1, \leq_1, F_1) \xrightarrow{\subseteq} (S_\cup, \leq_\cup, F_\cup) \xleftarrow{\supseteq} (S_2, \leq_2, F_2)$, where

– $(S_\cup, F_\cup) = (S_1, F_1) \cup (S_2, F_2)$, and

– $\leq_\cup = (\leq_1 \cup \leq_2)^{m*}$, i.e. the monotonic, reflexive and transitive closure of the union $\leq_1 \cup \leq_2$.

The intersection $(S_\cap, \leq_\cap, F_\cap)$ of two $\underline{\mathrm{OSA}}$ signatures $(S_1, \leq_1, F_1)$ and $(S_2, \leq_2, F_2)$ is given by the pair of inclusions $(S_1, \leq_1, F_1) \xleftarrow{\supseteq} (S_\cap, \leq_\cap, F_\cap) \xrightarrow{\subseteq} (S_2, \leq_2, F_2)$, where

– $(S_\cap, F_\cap) = (S_1, F_1) \cap (S_2, F_2)$, and

– $\leq_\cap = \leq_1 \cap \leq_2$.

Note that even with a more relaxed definition for signatures, the strong inclusion system of the category $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ is not distributive. This can be seen by analysing the possible unions and intersections of the signatures $(S_1, \leq_1, F_1)$, $(S_2, \leq_2, F_2)$ and $(S_3, \leq_3, F_3)$, with $S_1 = S_2 = S_3 = \{\mathtt{s}, \mathtt{s}', \mathtt{s}''\}$, $F_1 = F_2 = F_3 = \emptyset$, and the preorders $\leq_1$, $\leq_2$ and $\leq_3$ generated by $\{\mathtt{s} \leq_1 \mathtt{s}''\}$, $\{\mathtt{s} \leq_2 \mathtt{s}'\}$ and $\{\mathtt{s}' \leq_3 \mathtt{s}''\}$, respectively. For these signatures we obtain

$$(S_1, \leq_1, F_1) \cap ((S_2, \leq_2, F_2) \cup (S_3, \leq_3, F_3)) \supsetneq ((S_1, \leq_1, F_1) \cap (S_2, \leq_2, F_2)) \cup ((S_1, \leq_1, F_1) \cap (S_3, \leq_3, F_3)).$$

Without distributivity, many results local to $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ that are related to the instantiation of multiple-parameterised order-sorted signatures may not hold for particular examples of signatures and signature morphisms. However, as we will see in the later sections, we can obtain the desired results through the forgetful functor $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}} \to \mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ that discards the order relation on sorts. Furthermore, by similar considerations, these properties can be lifted to the more general categories of presentations and structured specifications of $\underline{\mathrm{OSA}}$.

Example 3.9. For the category $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{PA}}}$ of partial algebraic signatures, the strong inclusion system is distributive. The union $(S_\cup, F_\cup, TF_\cup)$ of two $\underline{\mathrm{PA}}$ signatures (S_1, F_1, TF_1) and (S_2, F_2, TF_2) is defined by

– $(S_\cup, F_\cup) = (S_1, F_1) \cup (S_2, F_2)$, and

– $TF_\cup = TF_1 \cup TF_2$.

The intersection $(S_\cap, F_\cap, TF_\cap)$ of two $\underline{\mathrm{PA}}$ signatures (S_1, F_1, TF_1) and (S_2, F_2, TF_2) is given by

– $(S_\cap, F_\cap) = (S_1, F_1) \cap (S_2, F_2)$, and

– $TF_\cap = TF_1 \cap TF_2$.

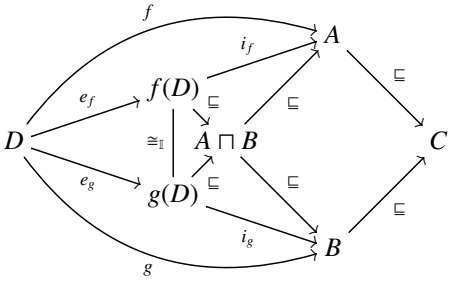
Since the distributivity property is essential in our study on generic specifications, throughout the subsequent dedicated sections of this paper we will only refer to these examples of distributive quasi-inclusion systems.

The following result, discussed in a restricted form in [12], shows that whenever the considered category has pullbacks, the existence of intersections is guaranteed by the existence of unions.

Proposition 3.5. *Let $\mathbb{I}$ be a quasi-inclusion system with unions for a category $\mathbb{C}$. Then $\mathbb{I}$ has intersections if and only if every cospan of quasi-inclusions $A \overset{\sqsubseteq}{\to} C \overset{\sqsupseteq}{\leftarrow} B$ admits a pullback in $\mathbb{C}$.*

Proof. We first assume that $\mathbb{I}$ has intersections and that $A \sqsubseteq C$, $B \sqsubseteq C$ are abstract quasi-inclusions. Our aim is to prove that $A \overset{\sqsubseteq}{\to} C \overset{\sqsupseteq}{\leftarrow} B$ has a pullback in $\mathbb{C}$ given by $A \sqcap B \sqsubseteq A$ and $A \sqcap B \sqsubseteq B$. Since this pair is obviously a pullback cone, we investigate the universal mapping property.

Let $f\colon D \to A$ and $g\colon D \to B$ be two morphisms such that $f; (A \sqsubseteq C) = g; (B \sqsubseteq C)$. By decomposing the arrows f and g as $e_f; i_f$ and $e_g; i_g$, respectively, we obtain $e_f; i_f; (A \sqsubseteq C) = e_g; i_g; (B \sqsubseteq C)$. It follows that $f(D) \cong_{\mathbb{I}} g(D)$ by the uniqueness of factorization. Since $f(D)$ and $g(D)$ are subobjects of both A and B we obtain, by the universal mapping property of the intersection, the quasi-inclusions $f(D) \sqsubseteq A \sqcap B$ and $g(D) \sqsubseteq A \sqcap B$. Using Fact 3.2 we can conclude that $e_f; (f(D) \sqsubseteq A \sqcap B) = e_g; (g(D) \sqsubseteq A \sqcap B)$, and denote this morphism by h. It can be easily checked that $h; (A \sqcap B \sqsubseteq A) = f$ and $h; (A \sqcap B \sqsubseteq B) = g$. Moreover, h is unique with these properties since all quasi-inclusions are monos.



For the opposite implication, we show how intersections can be built by means of unions and pushouts. Let A and B be two objects and $f\colon D \to A$ and $g\colon D \to B$ be the structural morphisms of a pullback cone of $A \overset{\sqsubseteq}{\to} A \sqcup B \overset{\sqsupseteq}{\leftarrow} B$. By factoring f and g as above we deduce the existence of the quasi-inclusive isomorphism $f(D) \cong_{\mathbb{I}} g(D)$. We proceed by proving that $f(D)$ (or equivalently, $g(D)$) is the intersection of A and B.

According to the definition of the intersection, we need to show that $f(D)$ is the vertex of a product of A and B in the category of quasi-inclusions. Let us thus consider two quasi-inclusions $E \sqsubseteq A$ and $E \sqsubseteq B$. Since $(E \sqsubseteq A); (A \sqsubseteq A \sqcup B) = (E \sqsubseteq B); (B \sqsubseteq A \sqcup B)$ we deduce by the universal mapping property of the pullback given by f and g that there exists a unique morphism $h\colon E \to D$ such that $h; f = (E \sqsubseteq A)$ and $h; g = (E \sqsubseteq B)$. Furthermore, by taking into account the fact that $h; f = h; e_f; i_f$ and i_f are both quasi-inclusions, we deduce that $h; e_f$ is a quasi-inclusion as well; therefore, $E \sqsubseteq f(D)$. Since $E \sqsubseteq A$ and $E \sqsubseteq B$ were considered to be arbitrary, we conclude that $f(D)$ is the intersection of A and B. □

Compatible arrows

The concept of compatible signature morphisms, as defined in [26], has been generalized to abstract inclusion systems in [13] for morphisms with the same codomain. Here we consider an even more general notion of compatibility for arbitrary morphisms.

Definition 3.6 (Compatible arrows)**.** Let us consider a category equipped with a quasi-inclusion system having unions and intersections. Two arrows $f\colon A \to A'$ and $g\colon B \to B'$ are *compatible* when

$$(A \sqcap B \sqsubseteq A); f; (A' \sqsubseteq A' \sqcup B') = (A \sqcap B \sqsubseteq B); g; (B' \sqsubseteq A' \sqcup B').$$

An arrow $f\colon A \to A'$ *preserves* an object B when f and 1_B are compatible. Furthermore, it *strongly preserves* an object B when, in addition to preserving B, it satisfies $A' \sqcap B \sqsubseteq A \sqcap B$.

Example 3.10. In the category $\mathbb{S}$et of sets, two functions $f\colon A \to A'$ and $g\colon B \to B'$ are compatible if and only if they agree on their common domain, i.e. $f(x) = g(x)$ for all $x \in A \cap B$.

Example 3.11. Two morphisms of $\underline{\text{MSA}}$ signatures $\varphi_1\colon (S_1, F_1) \to (S_1', F_1')$ and $\varphi_2\colon (S_2, F_2) \to (S_2', F_2')$ are compatible if and only if their components on sorts, φ_1^{st} and φ_2^{st}, are compatible functions, just as the translations of operation symbols $(\varphi_1^{op})_{w \to s}$ and $(\varphi_2^{op})_{w \to s}$, for all $w \in (S_1 \cap S_2)^*$ and $s \in S_1 \cap S_2$.

The details of the above example can be naturally extended to morphisms of order-sorted signatures or partial algebraic signatures by noticing they are essentially morphisms of many-sorted signatures which satisfy additional properties.

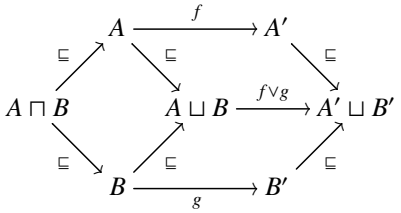
Assumption 3.1. *In the rest of this paper, whenever we will refer to compatibility we will implicitly assume that the considered category is endowed with a quasi-inclusion system having unions and intersections.*

The following elementary properties about the preservation of objects will prove to be useful in the later developments of the theory.

Fact 3.3. An arrow $f\colon A \to A'$ *preserves* an object B if and only if $(A \sqcap B \sqsubseteq A); f$ is a quasi-inclusion.

Fact 3.4. For any arrow $f\colon A \to A'$ such that $A' \sqsubseteq A$ the notions of preserved object and strongly preserved object concur.

Definition 3.7 (Join of compatible arrows)**.** For every two compatible arrows $f\colon A \to A'$ and $g\colon B \to B'$ in a category $\mathbb{C}$, the *join* $f \vee g$ denotes the unique arrow $A \sqcup B \to A' \sqcup B'$ such that $(A \sqsubseteq A \sqcup B); (f \vee g) = f; (A' \sqsubseteq A' \sqcup B')$ and $(B \sqsubseteq A \sqcup B); (f \vee g) = g; (B' \sqsubseteq A' \sqcup B')$, provided that the intersection-union square $[A \sqcap B, A, B, A \sqcup B]$ describes a pushout in $\mathbb{C}$.



Let us observe that, since unions are only unique up to a quasi-inclusive isomorphism, the join $f \vee g$ of two compatible arrows as above is not uniquely determined by f and g but rather by f, g and the unions $A \sqcup B$ and $A' \sqcup B'$. For this reason, whenever confusion can arise we will explicitly state the domain and the codomain of the join $f \vee g$.

Assumption 3.2. *In our benchmark quasi-inclusion systems, every intersection-union square is also a pushout square. However, in an arbitrary setting, the intersection-union squares might not necessarily describe pushouts. This property of the quasi-inclusion system will be implicitly assumed whenever we will use the join operator.*

Example 3.12. For any two compatible functions $f\colon A \to A'$ and $g\colon B \to B'$ the join $f \vee g\colon A \cup B \to A' \cup B'$ is defined by

$$(f \vee g)(x) = \begin{cases} f(x) & \text{for all } x \in A, \\ g(x) & \text{for all } x \in B. \end{cases}$$

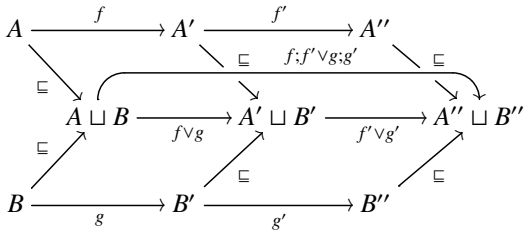
Example 3.13. In the category of many-sorted signatures, the join $\varphi_1 \vee \varphi_2$ of two compatible signature morphisms $\varphi_1\colon (S_1, F_1) \to (S_1', F_1')$ and $\varphi_2\colon (S_2, F_2) \to (S_2', F_2')$ is given by

$$(\varphi_1 \vee \varphi_2)^{st}(s) = \begin{cases} \varphi_1^{st}(s) & \text{for all } s \in S_1, \\ \varphi_2^{st}(s) & \text{for all } s \in S_2, \end{cases}$$

$$(\varphi_1 \vee \varphi_2)^{op}_{w\to s}(\sigma) = \begin{cases} (\varphi_1^{op})_{w\to s}(\sigma) & \text{for all } \sigma \in (F_1)_{w\to s}, \\ (\varphi_2^{op})_{w\to s}(\sigma) & \text{for all } \sigma \in (F_2)_{w\to s}. \end{cases}$$

Proposition 3.6. *If $f\colon A \to A'$, $g\colon B \to B'$ and $f'\colon A' \to A''$, $g'\colon B' \to B''$ are two pairs of compatible morphisms, $f; f'$ and $g; g'$ are also compatible and their join is given by $f; f' \vee g; g' = (f \vee g); (f' \vee g')$.*

Proof. Let us consider the following diagram.



The compatibility of $f; f'$ and $g; g'$ can be easily checked through a straightforward calculation.

$$\begin{aligned}(A \sqcap B \sqsubseteq A); f; f'; (A'' \sqsubseteq A'' \sqcup B'') &= (A \sqcap B \sqsubseteq A); f; (A' \sqsubseteq A' \sqcup B'); (f' \vee g') \\ &= (A \sqcap B \sqsubseteq B); g; (B' \sqsubseteq A' \sqcup B'); (f' \vee g') \\ &= (A \sqcap B \sqsubseteq B); g; g'; (B'' \sqsubseteq A'' \sqcup B'')\end{aligned}$$

For the equality of $f; f' \vee g; g'$ and $(f \vee g); (f' \vee g')$, since by hypothesis the union of A and B is a pushout of their intersection, it suffices to show that $(f \vee g); (f' \vee g')$ satisfies the defining properties of the join $f; f' \vee g; g'$.

$$\begin{aligned}(A \sqsubseteq A \sqcup B); (f \vee g); (f' \vee g') &= f; (A' \sqsubseteq A' \sqcup B'); (f' \vee g') \\ &= f; f'; (A'' \sqsubseteq A'' \sqcup B'') \\ (B \sqsubseteq A \sqcup B); (f \vee g); (f' \vee g') &= g; (B' \sqsubseteq A' \sqcup B'); (f' \vee g') \\ &= g; g'; (B'' \sqsubseteq A'' \sqcup B'')\end{aligned}$$

□

Corollary 3.1. *If $f\colon A \to C$ and $g\colon B \to C$ are two compatible arrows then for any arrow $h\colon C \to D$, $f; h$ and $g; h$ are also compatible and $(f \vee g); h = f; h \vee g; h$.*

Proposition 3.7. *Any two quasi-inclusions are compatible. Moreover, the join of any two quasi-inclusions is a quasi-inclusion as well.*

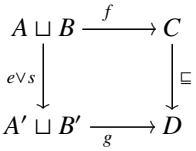
Proof. The first part of the statement follows trivially from the definition of compatibility and the compositionality of quasi-inclusions. For the second part, let $A \sqsubseteq A'$ and $B \sqsubseteq B'$ be two quasi-inclusive morphisms such that the intersection-union square $[A \sqcap B, A, B, A \sqcup B]$ describes a pushout. By the universal property of the union $A \sqcup B$ there exists a quasi-inclusion $A \sqcup B \sqsubseteq A' \sqcup B'$. Since this arrow verifies

$$(A \sqsubseteq A \sqcup B); (A \sqcup B \sqsubseteq A' \sqcup B') = (A \sqsubseteq A'); (A' \sqsubseteq A' \sqcup B') \text{ and}$$
$$(B \sqsubseteq A \sqcup B); (A \sqcup B \sqsubseteq A' \sqcup B') = (B \sqsubseteq B'); (B' \sqsubseteq A' \sqcup B')$$

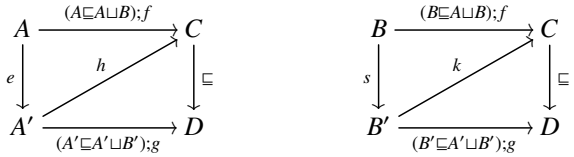
by the uniqueness of the join arrow of $A \sqsubseteq A'$ and $B \sqsubseteq B'$, we conclude that $(A \sqsubseteq A') \vee (B \sqsubseteq B') = (A \sqcup B \sqsubseteq A' \sqcup B')$. □

Proposition 3.8. *The join of any two compatible abstract surjections is an abstract surjection.*

Proof. Let $e\colon A \to A'$ and $s\colon B \to B'$ be two compatible abstract surjections. In order to show that $e \vee s$ is an abstract surjection, by the diagonal-fill property, we consider two arbitrary morphisms $f\colon A \sqcup B \to C$ and $g\colon A' \sqcup B' \to D$ such that C is a subobject of D and the equality $f; (C \sqsubseteq D) = (e \vee s); g$ holds.



By composing the above equality to the left with the quasi-inclusions $A \sqsubseteq A \sqcup B$ and $B \sqsubseteq A \sqcup B$ we obtain $(A \sqsubseteq A \sqcup B); f = e; (A' \sqsubseteq A' \sqcup B'); g$ and $(B \sqsubseteq A \sqcup B); f = s; (B' \sqsubseteq A' \sqcup B'); g$, respectively. The diagonal-fill properties of e and s provide us two morphisms h and k, respectively, that make the following two diagrams commutative.



Note that $(A' \sqcap B' \sqsubseteq A'); h; (C \sqsubseteq D) = (A' \sqcap B' \sqsubseteq A' \sqcup B'); g = (A' \sqcap B' \sqsubseteq B'); k; (C \sqsubseteq D)$. Since $C \sqsubseteq D$ is mono, by Proposition 3.4, we conclude that the morphisms h and k are compatible, and thus there exists the join $h \vee k\colon A' \sqcup B' \to C$. Moreover, by Proposition 3.6 and Corollary 3.1, using the fact that the join of the structural quasi-inclusions of any union yields the identity of the union, we infer that $(e \vee s); (h \vee k) = f$ by the equalities

$$(e \vee s); (h \vee k) = e; h \vee s; k = (A \sqsubseteq A \sqcup B); f \vee (B \sqsubseteq A \sqcup B); f = ((A \sqsubseteq A \sqcup B) \vee (B \sqsubseteq A \sqcup B)); f = f,$$

and that $(h \vee k); (C \sqsubseteq D) = g$ by the equalities

$$(h \vee k); (C \sqsubseteq D) = h; (C \sqsubseteq D) \vee k; (C \sqsubseteq D) = (A' \sqsubseteq A' \sqcup B'); g \vee (B' \sqsubseteq A' \sqcup B'); g = ((A' \sqsubseteq A' \sqcup B') \vee (B' \sqsubseteq A' \sqcup B')); g = g.$$

Given the above explanations, we conclude that $e \vee s$ is an abstract surjection. □

Corollary 3.2. *Let $f\colon A \to A'$ and $g\colon B \to B'$ be two compatible morphisms and $e_f; i_f$, $e_g; i_g$ their corresponding factorizations. Then*

- *e_f and e_g are compatible,*
- *$f \vee g$ can be factored as $(e_f \vee e_g); (i_f \vee i_g)$, and*
- *$(f \vee g)(A \sqcup B) \cong_{\mathbb{I}} f(A) \sqcup g(B)$.*

For the remaining part of this section we focus on properties related to the preservation of objects by arrows. More precisely, we are interested in properties of classes of objects that are preserved by certain arrows and in properties of classes of arrows that preserve certain objects.

Fact 3.5. An arrow $f\colon A \to A'$ preserves an object B if and only if it preserves all the subobjects of B.

Proposition 3.9. *If the quasi-inclusion system is distributive, any arrow $f\colon A \to A'$ preserving two objects B and C preserves their union $B \sqcup C$ as well.*

Proof. Let us consider an arrow $f\colon A \to A'$ that preserves the objects B and C. According to Fact 3.3 it is enough to check that $(A \sqcap (B \sqcup C) \sqsubseteq A); f$ is quasi-inclusive. By the distributivity of the quasi-inclusion system, our goal is further reduced to showing that $((A \sqcap B) \sqcup (A \sqcap C) \sqsubseteq A); f$ is quasi-inclusive. The conclusion of the statement now follows from Proposition 3.7, based on the fact that $(A \sqcap B \sqsubseteq A); f$ and $(A \sqcap C \sqsubseteq A); f$ are both quasi-inclusions. □

Fact 3.6. If $f\colon A \to A'$ and $f'\colon A' \to A''$ are two composable morphisms then any object preserved by both f and f' is preserved by $f; f'$ as well.

Proposition 3.10. *If $f\colon A \to A'$ and $g\colon B \to B'$ are two compatible morphisms then any object preserved by $f \vee g$ is preserved by both f and g. Moreover, if the quasi-inclusion system is distributive then any object preserved by f and g is also preserved by $f \vee g$.*

Proof. Assuming $f \vee g$ preserves an arbitrary but fixed object C, or equivalently, that $((A \sqcup B) \sqcap C \sqsubseteq A \sqcup B); (f \vee g)$ is a quasi-inclusion, we infer that both arrows $(A \sqcap C \sqsubseteq A); f; (A' \sqsubseteq A' \sqcup B')$ and $(B \sqcap C \sqsubseteq B); g; (B' \sqsubseteq A' \sqcup B')$ are quasi-inclusions, as explained by the two sequences of equalities below. If follows that $(A \sqcap C \sqsubseteq A); f$ and $(B \sqcap C \sqsubseteq B); g$ are quasi-inclusions, thus concluding that both f and g preserve C.

$$\begin{aligned}
(A \sqcap C \sqsubseteq A); f; (A' \sqsubseteq A' \sqcup B') &= (A \sqcap C \sqsubseteq A); (A \sqsubseteq A \sqcup B); (f \vee g) \\
&= (A \sqcap C \sqsubseteq (A \sqcup B) \sqcap C); ((A \sqcup B) \sqcap C \sqsubseteq A \sqcup B); (f \vee g) \\
&= (A \sqcap C \sqsubseteq (A \sqcup B) \sqcap C); ((A \sqcup B) \sqcap C \sqsubseteq A' \sqcup B') \\
&= (A \sqcap C \sqsubseteq A' \sqcup B') \\
(B \sqcap C \sqsubseteq B); g; (B' \sqsubseteq A' \sqcup B') &= (B \sqcap C \sqsubseteq B); (B \sqsubseteq A \sqcup B); (f \vee g) \\
&= (B \sqcap C \sqsubseteq (A \sqcup B) \sqcap C); ((A \sqcup B) \sqcap C \sqsubseteq A \sqcup B); (f \vee g) \\
&= (B \sqcap C \sqsubseteq (A \sqcup B) \sqcap C); ((A \sqcup B) \sqcap C \sqsubseteq A' \sqcup B') \\
&= (B \sqcap C \sqsubseteq A' \sqcup B')
\end{aligned}$$

For the reverse implication, because the quasi-inclusion system is distributive, we know there exists a quasi-inclusive isomorphism between $(A \sqcup B) \sqcap C$ and $(A \sqcap C) \sqcup (B \sqcap C)$. By Corollary 3.1, based on the fact that the join of the structural quasi-inclusions of any union yields the identity of the union, it follows that the morphism $((A \sqcap C) \sqcup (B \sqcap C) \sqsubseteq A \sqcup B); (f \vee g)$ can be written as the join of its restrictions to $A \sqcap C$ and $B \sqcap C$.

$$\begin{aligned}
((A \sqcap C) \sqcup (B \sqcap C) \sqsubseteq A \sqcup B); (f \vee g) = (A \sqcup C \sqsubseteq (A \sqcap C) \sqcup (B \sqcap C)); ((A \sqcap C) \sqcup (B \sqcap C) \sqsubseteq A \sqcup B); (f \vee g) \vee \\
(B \sqcup C \sqsubseteq (A \sqcap C) \sqcup (B \sqcap C)); ((A \sqcap C) \sqcup (B \sqcap C) \sqsubseteq A \sqcup B); (f \vee g)
\end{aligned}$$

Therefore, according to Proposition 3.7, in order to prove that $((A \sqcup B) \sqcap C); (f \vee g)$ is quasi-inclusive it suffices to show that the two restrictions considered above are quasi-inclusions.

$$\begin{aligned}
&(A \sqcap C \sqsubseteq (A \sqcap C) \sqcup (B \sqcap C)); ((A \sqcap C) \sqcup (B \sqcap C) \sqsubseteq A \sqcup B); (f \vee g) \\
&\quad = (A \sqcap C \sqsubseteq A \sqcup B); (f \vee g) \\
&\quad = (A \sqcap C \sqsubseteq A); (A \sqsubseteq A \sqcup B); (f \vee g) \\
&\quad = (A \sqcap C \sqsubseteq A); f; (A' \sqsubseteq A' \sqcup B') \\
&\quad = (A \sqcap C \sqsubseteq A' \sqcup B')
\end{aligned}$$

$$
\begin{aligned}
&(B \sqcap C \sqsubseteq (A \sqcap C) \sqcup (B \sqcap C)); ((A \sqcap C) \sqcup (B \sqcap C) \sqsubseteq A \sqcup B); (f \vee g) \\
&\quad = (B \sqcap C \sqsubseteq A \sqcup B); (f \vee g) \\
&\quad = (B \sqcap C \sqsubseteq B); (B \sqsubseteq A \sqcup B); (f \vee g) \\
&\quad = (B \sqcap C \sqsubseteq B); g; (B' \sqsubseteq A' \sqcup B') \\
&\quad = (B \sqcap C \sqsubseteq A' \sqcup B')
\end{aligned}
$$

□

4. Free Extensions

The concept of free extension of a morphism along an inclusion, introduced in [13], was developed as a refined form of a specific pushout, aimed at providing a more fine-grained categorical representation of the intuition behind the substitution of generic components in a given system. In this section we explore some of the properties of free extensions defined along quasi-inclusions, with a greater emphasis on their behaviour relative to the composition and the join of arrows. These properties will later emerge as essential statements used to support desired attributes of generic systems.

Definition 4.1 (Free extension)**.** Let $\mathbb{C}$ be a category endowed with a quasi-inclusion system $\mathbb{I}$ having unions and intersections, and let $f\colon A \to A'$ be a morphism. A *free extension* of f along a quasi-inclusion $A \sqsubseteq B$ is a morphism $f^B\colon B \to B'$, with $A' \sqsubseteq B'$, such that the square depicted below is a pushout square, and every object preserved by f is preserved by f^B too.

$$
\begin{array}{ccc}
A & \xrightarrow{\sqsubseteq} & B \\
{\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f^B} \\
A' & \xrightarrow[\sqsubseteq]{} & B'
\end{array}
$$

Example 4.1. In $\mathbb{S}$et, a function $f\colon A \to A'$ admits a free extension along an inclusion $A \subseteq B$ if and only if A' and $B \setminus A$ are disjoint. When this condition is met, the free extension $f^B\colon B \to B'$ is defined by $B' = (B \setminus A) \cup A'$ and

$$
f^B(a) = \begin{cases} f(a) & \text{for all } a \in A, \\ a & \text{for all } a \in B \setminus A. \end{cases}
$$

As a result of this characterisation, in $\mathbb{S}$et, the free extension $f^B\colon B \to B'$ of a function $f\colon A \to A'$ exists and satisfies $B' \subseteq B$ whenever $A' \subseteq A$.

The following result was first proved in [13] for signature endomorphisms. Here we extend it to signature morphisms whose codomain is a subobject of the domain. Since the proof can be done in an analogous manner to the original case of endomorphisms, we will omit it and focus solely on the construction of the free extensions of interest.

Proposition 4.1. *Any* MSA *signature morphism* $\varphi_1\colon (S_1, F_1) \to (S_1', F_1')$ *such that* (S_1', F_1') *is a subsignature of* (S_1, F_1) *admits free extensions along any inclusion* $(S_1, F_1) \subseteq (S_2, F_2)$*. Moreover, if* φ_1 *preserves a fixed signature* (S_0, F_0)*, there exists a free extension* $\varphi_2\colon (S_2, F_2) \to (S_2', F_2')$ *that strongly preserves* (S_0, F_0)*, defined as follows:*

- *the component on sorts,* $\varphi_2^{st}\colon S_2 \to S_2'$*, is the free extension of the function* φ_1^{st} *along* $S_1 \subseteq S_2$*; it is defined by* $S_2' = (S_2 \setminus S_1) \cup S_1'$ *and*
$$
\varphi_2^{st}(s) = \begin{cases} \varphi_1^{st}(s) & \text{for all } s \in S_1, \\ s & \text{for all } s \in S_2 \setminus S_1 \end{cases}
$$
- *for each arity* $w_2' \in S_2'^*$ *and each sort* $s_2' \in S_2'$*, the set* $(F_2')_{w_2' \to s_2'}$ *is given by the codomain of the free extension* $\theta'_{w_2' \to s_2'}$ *depicted below, where for any signature morphism* $\varphi\colon (S, F) \to (S', F')$*, the disjoint union*

$\biguplus_{\varphi^{st}(ws)=w's'} F_{w\to s}$ *is defined by*

$$\biguplus_{\varphi^{st}(ws)=w's'} F_{w\to s} = \{(\sigma, w, s, F_0) \mid \sigma \in F_{w\to s}, \varphi^{st}(ws) = w's', ws \neq w's'\} \cup \{\sigma \mid w' \in S^*, s' \in S, \sigma \in F_{w'\to s'}, \varphi^{st}(w's') = w's'\}$$

and $\theta_{w'_2\to s'_2}$ *is the function mapping each operation symbol* $(\sigma_1, w_1, s_1, F_0)$ *or* σ_1 *to* $(\varphi_1^{op})_{w_1\to s_1}(\sigma_1)$*;*

$$\begin{array}{ccc} \biguplus_{\varphi_1^{st}(w_1s_1)=w'_2s'_2} (F_1)_{w_1\to s_1} & \xrightarrow{\subseteq} & \biguplus_{\varphi_2^{st}(w_2s_2)=w'_2s'_2} (F_2)_{w_2\to s_2} \\ \downarrow{\scriptstyle \theta_{w'_2\to s'_2}} & & \downarrow{\scriptstyle \theta'_{w'_2\to s'_2}} \\ \biguplus_{w'_1s'_1=w'_2s'_2} (F'_1)_{w'_1\to s'_1} & \xrightarrow[\subseteq]{} & (F'_2)_{w'_2\to s'_2} \end{array}$$

– *for each arity* $w_2 \in S_2^*$ *and each sort* $s_2 \in S_2$*, the function* $(\varphi_2^{op})_{w_2\to s_2}$ *is given by the composition of the canonical injection* $(F_2)_{w_2\to s_2} \to \biguplus_{\varphi_2^{st}(ws)=\varphi_2^{st}(w_2s_2)}(F_2)_{w\to s}$ *with the free extension* $\theta'_{\varphi_2^{st}(w_2)\to\varphi_2^{st}(s_2)}$.

The existence of free extensions can be lifted from many-sorted signatures to order-sorted signatures or partial algebraic signatures through the forgetful functors that eliminate the details related to the preorder on sorts or the totality implied by specific operation symbols. These functors will be studied in a more general setting in the later sections of the paper.

Fact 4.1. Let $\varphi_1 \colon (S_1, \leq_1, F_1) \to (S'_1, \leq'_1, F'_1)$ be an order-sorted signature morphism such that $(S'_1, \leq'_1, F'_1)$ is a subsignature of $(S_1, \leq_1, F_1)$. For any inclusion $(S_1, \leq_1, F_1) \subseteq (S_2, \leq_2, F_2)$, the morphism φ_1 admits a free extension $\varphi_2 \colon (S_2, \leq_2, F_2) \to (S'_2, \leq'_2, F'_2)$ that is obtained by lifting from the category of many-sorted signatures the free extension $\varphi_2 \colon (S_2, F_2) \to (S'_2, F'_2)$ of $\varphi_1 \colon (S_1, F_1) \to (S'_1, F'_1)$, and by defining $\leq'_2$ as the monotonic, reflexive and transitive closure of $\varphi_2^{st}(\leq_2) \cup \leq'_1$.

$$\begin{array}{ccc} (S_1, \leq_1, F_1) & \xrightarrow{\subseteq} & (S_2, \leq_2, F_2) \\ \downarrow{\scriptstyle \varphi_1} & & \downarrow{\scriptstyle \varphi_2} \\ (S'_1, \leq'_1, F'_1) & \xrightarrow[\subseteq]{} & (S'_2, (\varphi_2^{st}(\leq_2) \cup \leq'_1)^{m*}, F'_2) \end{array}$$

Let us note that, contrary to the case of many-sorted signatures, in $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ it is not always possible to choose the free extension of a morphism given as above such that it strongly preserves a fixed object preserved by the first morphism. To clarify this, let Σ_0 be the order-sorted signature $(\{\mathtt{s}, \mathtt{s}'\}, \{\mathtt{s} \leq_0 \mathtt{s}'\}, \emptyset)$ and φ_1 the signature morphism from $\Sigma_1 = (\{\mathtt{t}, \mathtt{s}'\}, \emptyset, \emptyset)$ to $\Sigma'_1 = (\{\mathtt{s}'\}, \emptyset, \emptyset)$, defined by $\varphi_1^{st}(\mathtt{t}) = \mathtt{s}'$ and $\varphi_1^{st}(\mathtt{s}') = \mathtt{s}'$. Observe that φ_1 trivially preserves Σ_0, and that it has a unique free extension along the inclusion $\Sigma_1 \subseteq \Sigma_2$, where $\Sigma_2 = (\{\mathtt{s}, \mathtt{t}, \mathtt{s}'\}, \{\mathtt{s} \leq_2 \mathtt{t}\}, \emptyset)$, namely the signature morphism φ_2 from Σ_2 to $\Sigma'_2 = (\{\mathtt{s}, \mathtt{s}'\}, \{\mathtt{s} \leq'_2 \mathtt{s}'\}, \emptyset)$, given by $\varphi_2^{st}(\mathtt{s}) = \mathtt{s}$, $\varphi_2^{st}(\mathtt{t}) = \mathtt{s}'$ and $\varphi_2^{st}(\mathtt{s}') = \mathtt{s}'$. As expected, the signature Σ_0 is preserved by φ_2 and its $\underline{\mathrm{MSA}}$ reduct is strongly preserved by φ_2. However, the preorder relation $\leq_0$ of Σ_0 does not satisfy $\leq'_2 \cap \leq_0 \subseteq \leq_2 \cap \leq_0$, and thus Σ_0 is not strongly preserved by φ_2.

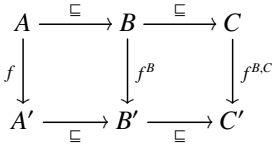
Fact 4.2. Let $\varphi_1 \colon (S_1, F_1, TF_1) \to (S'_1, F'_1, TF'_1)$ be a morphism of $\underline{\mathrm{PA}}$ signatures such that (S'_1, F'_1, TF'_1) is a subsignature of (S_1, F_1, TF_1). For any inclusion $(S_1, F_1, TF_1) \subseteq (S_2, F_2, TF_2)$, the morphism φ_1 admits a free extension $\varphi_2 \colon (S_2, F_2, TF_2) \to (S'_2, F'_2, TF'_2)$ that is obtained by lifting from the category of many-sorted signatures the free extension $\varphi_2 \colon (S_2, F_2) \to (S'_2, F'_2)$ of $\varphi_1 \colon (S_1, F_1) \to (S'_1, F'_1)$, and by defining TF'_2 as the union $\varphi_2^{op}(TF_2) \cup TF'_1$.

$$\begin{array}{ccc} (S_1, F_1, TF_1) & \xrightarrow{\subseteq} & (S_2, F_2, TF_2) \\ \downarrow{\scriptstyle \varphi_1} & & \downarrow{\scriptstyle \varphi_2} \\ (S'_1, F'_1, TF'_1) & \xrightarrow[\subseteq]{} & (S'_2, F'_2, \varphi_2^{op}(TF_2) \cup TF'_1) \end{array}$$

Similarly to the case of order-sorted signature morphisms, it is not always possible to choose the free extension φ_2 such that it strongly preserves a given PA signature Σ_0 preserved by φ_1. For example, let us consider the partial algebraic signature $\Sigma_0 = (\{\mathtt{s}\}, \{\mathtt{b}\colon [] \to \mathtt{s}\}, \{\mathtt{b}\colon [] \to \mathtt{s}\})$ and the signature morphism φ_1 from $\Sigma_1 = (\{\mathtt{s}\}, \{\mathtt{a}\colon [] \to \mathtt{s}, \mathtt{b}\colon [] \to \mathtt{s}\}, \emptyset)$ to $\Sigma_1' = (\{\mathtt{s}\}, \{\mathtt{b}\colon [] \to \mathtt{s}\}, \emptyset)$ given by $\varphi_1^{st}(\mathtt{s}) = \mathtt{s}$ and $\varphi_1^{op}(\mathtt{a}) = \varphi_1^{op}(\mathtt{b}) = \mathtt{b}$. In addition, let Σ_2 be the signature $(\{\mathtt{s}\}, \{\mathtt{a}\colon [] \to \mathtt{s}, \mathtt{b}\colon [] \to \mathtt{s}\}, \{\mathtt{a}\colon [] \to \mathtt{s}\})$. It is easy to see that φ_1 admits a unique free extension along the inclusion $\Sigma_1 \subseteq \Sigma_2$, namely the signature morphism $\varphi_2\colon \Sigma_2 \to \Sigma_2'$ defined by $\Sigma_2' = (\{\mathtt{s}\}, \{\mathtt{b}\colon [] \to \mathtt{s}\}, \{\mathtt{b}\colon [] \to \mathtt{s}\})$, $\varphi_2^{st}(\mathtt{s}) = \mathtt{s}$ and $\varphi_2^{op}(\mathtt{a}) = \varphi_2^{op}(\mathtt{b}) = \mathtt{b}$. Notice that in this situation the operation symbol $\mathtt{b}\colon [] \to \mathtt{s}$ appears as total only in the codomain of φ_2, and thus the inclusion $\Sigma_2' \cap \Sigma_0 \subseteq \Sigma_2 \cap \Sigma_0$ does not hold.

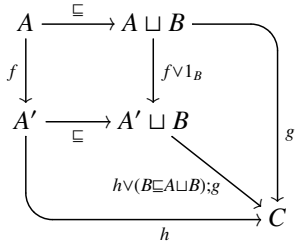
Fact 4.3. If $f^B\colon B \to B'$ is a free extension of $f\colon A \to A'$ along the quasi-inclusion $A \sqsubseteq B$ then every object preserved by f^B is also preserved by f.

Fact 4.4. Consider a commutative diagram like below where f^B is a free extension of f along the quasi-inclusion $A \sqsubseteq B$. Then $f^{B,C}$ is a free extension of f^B along the quasi-inclusion $B \sqsubseteq C$ if and only if $f^{B,C}$ is a free extension of f along $A \sqsubseteq C$.



Proposition 4.2. *Let $\mathbb{C}$ be a category endowed with a distributive quasi-inclusion system $\mathbb{I}$. If $f\colon A \to A'$ is a morphism strongly preserving an object B then $f \vee 1_B$ is a free extension of f along the quasi-inclusion $A \sqsubseteq A \sqcup B$.*

Proof. Let $f\colon A \to A'$ be a morphism and B an object such that f and 1_B are compatible. By the definition of $f \vee 1_B$ we know that the square $[A, A \sqcup B, A', A' \sqcup B]$ depicted below is commutative. In order to show that it is a pushout square we consider two morphisms $g\colon A \sqcup B \to C$ and $h\colon A' \to C$ such that $(A \sqsubseteq A \sqcup B); g = f; h$.



Since f and 1_B are compatible we conclude the morphism $(A \sqcap B \sqsubseteq A); f$ is quasi-inclusive. Moreover, because f strongly preserves B, the quasi-inclusion $A' \sqcap B \sqsubseteq A'$ can be factored as $(A' \sqcap B \sqsubseteq A \sqcap B); (A \sqcap B \sqsubseteq A')$. This allows us to draw the following sequence of equalities.

$$\begin{aligned}(A' \sqcap B \sqsubseteq A'); h &= (A' \sqcap B \sqsubseteq A \sqcap B); (A \sqcap B \sqsubseteq A'); h \\ &= (A' \sqcap B \sqsubseteq A \sqcap B); (A \sqcap B \sqsubseteq A); f; h \\ &= (A' \sqcap B \sqsubseteq A \sqcap B); (A \sqcap B \sqsubseteq A); (A \sqsubseteq A \sqcup B); g \\ &= (A' \sqcap B \sqsubseteq A \sqcap B); (A \sqcap B \sqsubseteq B); (B \sqsubseteq A \sqcup B); g \\ &= (A' \sqcap B \sqsubseteq B); (B \sqsubseteq A \sqcup B); g\end{aligned}$$

It follows that h and $(B \sqsubseteq A \sqcup B); g$ are compatible; therefore, we can consider the morphism $h \vee (B \sqsubseteq A \sqcup B); g$ which verifies the next two equalities.

$$(f \vee 1_B); (h \vee (B \sqsubseteq A \sqcup B); g) = g \text{ and}$$

$$(A' \sqsubseteq A' \sqcup B); (h \vee (B \sqsubseteq A \sqcup B); g) = h,$$

where the former holds since

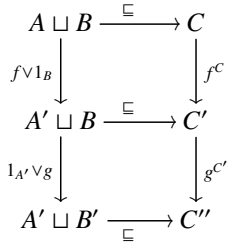
$$\begin{aligned}(A \sqsubseteq A \sqcup B); (f \vee 1_B); (h \vee (B \sqsubseteq A \sqcup B); g) &= (A \sqsubseteq A \sqcup B); (f; h \vee (B \sqsubseteq A \sqcup B); g) \\ &= f; h \\ &= (A \sqsubseteq A \sqcup B); g\end{aligned}$$

and

$$\begin{aligned}(B \sqsubseteq A \sqcup B); (f \vee 1_B); (h \vee (B \sqsubseteq A \sqcup B); g) &= (B \sqsubseteq A \sqcup B); (f; h \vee (B \sqsubseteq A \sqcup B); g) \\ &= (B \sqsubseteq A \sqcup B); g.\end{aligned}$$

Furthermore, for any other morphism $k \colon A' \sqcup B \to C$ such that $(f \vee 1_B); k = g$ and $(A' \sqsubseteq A' \sqcup B); k = h$ we obtain the equalities $(B \sqsubseteq A' \sqcup B); k = (B \sqsubseteq A \sqcup B); (f \vee 1_B); k = (B \sqsubseteq A \sqcup B); g$, implying that $k = h \vee (B \sqsubseteq A \sqcup B); g$. To conclude the argument, notice that by Proposition 3.10 every object preserved by f, and also trivially preserved by 1_B, is preserved by $f \vee 1_B$ as well. □

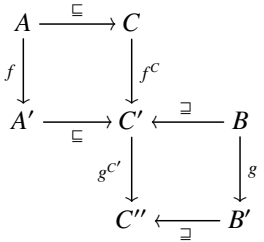
Proposition 4.3. *Let $f \colon A \to A'$ and $g \colon B \to B'$ be two compatible morphisms such that f preserves B and g preserves A'. If $f^C \colon C \to C'$ and $g^{C'} \colon C' \to C''$ are free extensions of $f \vee 1_B$ and $1_{A'} \vee g$, respectively, then $f^C; g^{C'}$ is a free extension of $f \vee g$.*



Proof. Let us first notice that $(f \vee 1_B); (1_{A'} \vee g) = f \vee g$, by Proposition 3.6. Based on the general properties of gluing together pushout squares [1], because the two inner squares depicted above describe pushouts, we conclude the outer square $[A \sqcup B, C, A' \sqcup B', C'']$ describes a pushout too. Then $f^C; g^{C'}$ is a free extension of $f \vee g$ if every object preserved by $f \vee g$ is also preserved by $f^C; g^{C'}$.

By Proposition 3.10 we know that any object preserved by the join $f \vee g$ is preserved by both f and g; therefore, any object preserved by $f \vee g$ must be preserved by f^C and g^C, and thus, by Fact 3.6, is preserved by $f^C; g^{C'}$ as well. □

Proposition 4.4. *Let $f \colon A \to A'$ and $g \colon B \to B'$ be two compatible morphisms such that f strongly preserves B and g strongly preserves A'. If $f^C \colon C \to C'$ and $g^{C'} \colon C' \to C''$ are free extensions of f and g, respectively, and f^C strongly preserves B, then $f^C; g^{C'}$ is a free extension of $f \vee g$.*

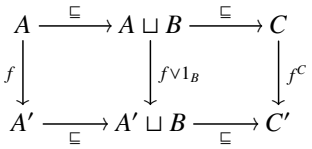


Proof. Observe that B is a subobject of C, because $B \cong_{\mathbb{I}} B \sqcap C' \sqsubseteq B \sqcap C$. It follows that $A \sqcup B$ is a subobject of C as well. Furthermore, since $[A \sqcap B, A, B, A \sqcup B]$ is an intersection-union square, and thus a pushout square, the equalities below entail $(A \sqcup B \sqsubseteq C); f^C = (f \vee 1_B); (A' \sqcup B \sqsubseteq C')$. Notice that by Fact 3.3 we know $(B \sqsubseteq C); f^C$ is a quasi-inclusion.

$$(A \sqsubseteq A \sqcup B); (A \sqcup B \sqsubseteq C); f^C = f; (A' \sqsubseteq A' \sqcup B); (A' \sqcup B \sqsubseteq C') = (A \sqsubseteq A \sqcup B); (f \vee 1_B); (A' \sqcup B \sqsubseteq C')$$

$$(B \sqsubseteq A \sqcup B); (A \sqcup B \sqsubseteq C); f^C = (B \sqsubseteq C); f^C = (B \sqsubseteq C') = (B \sqsubseteq A \sqcup B); (f \vee 1_B); (A' \sqcup B \sqsubseteq C')$$

By these explanations we can draw the following commutative diagram.

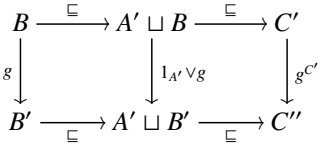


We know that $f \vee 1_B$ is a free extension of f, from Proposition 4.2, and that f^C is a free extension of f, from hypothesis; therefore, by Fact 4.4, f^C is a free extension of $f \vee 1_B$.

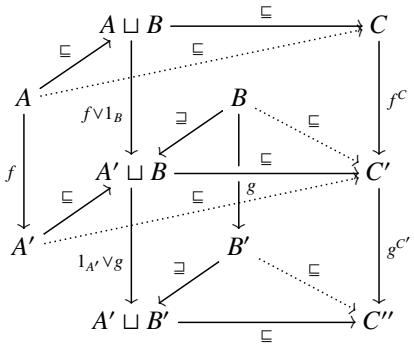
Through a similar argument to the one given above, we can draw the commutative diagram below, where the commutativity of the right inner square is given by the next two sequences of equalities.

$$(A' \sqsubseteq A' \sqcup B); (A' \sqcup B \sqsubseteq C'); g^{C'} = (A' \sqsubseteq C'); g^{C'} = (A' \sqsubseteq C'') = (A' \sqsubseteq A' \sqcup B); (1_{A'} \vee g); (A' \sqcup B' \sqsubseteq C'')$$

$$(B \sqsubseteq A' \sqcup B); (A' \sqcup B \sqsubseteq C'); g^{C'} = g; (B' \sqsubseteq A' \sqcup B'); (A' \sqcup B' \sqsubseteq C'') = (B \sqsubseteq A' \sqcup B); (1_{A'} \vee g); (A' \sqcup B' \sqsubseteq C'')$$



Because $1_{A'} \vee g$ is a free extension of g, from Proposition 4.2, and $g^{C'}$ is also a free extension of g, from hypothesis, we reach the conclusion that $g^{C'}$ is a free extension of $1_{A'} \vee g$. All these arguments allow us to draw the following concise diagrammatic presentation of the proof.



By the above explanations f^C and $g^{C'}$ are free extensions of $f \vee 1_B$ and $1_{A'} \vee g$, respectively. For this reason, by Proposition 4.3, $f^C; g^{C'}$ is a free extension of $f \vee g$, thus concluding the argument. □

5. Parameterised Objects

Parameterisation constitutes one of the most important techniques used in constructing complex structured entities. It provides a bidirectional mechanism that allows the development of new parameterised descriptions of a given generic design by abstracting away certain elements from particular existing specifications. This assumes a dedicated apparatus for parameter instantiation allowing the reuse or composition of the parameterised specifications whenever it is needed.

Our approach to parameterisation follows the lines of [13]. It refines the pushout-style parameterisation introduced in [4] and [5] by considering a more restrictive definition of parameter passing, based on specific inclusion preserving pushouts, namely the ones that correspond to free extensions.

Assumption 5.1. *Here we discuss parameterisation from a local, syntactic point of view, in the sense that we define notions such as parameterised object and instantiation of parameters within a given category with appropriate additional structure. More precisely, throughout this section we assume an arbitrary but fixed category endowed with a quasi-inclusion system having unions and intersections.*

Parameterised objects, usually considered in a category of formal specifications, are often defined in the literature as arbitrary specification morphisms [9, 29] or abstract inclusions [13]. In this paper we regard quasi-inclusions as foundations for parameterisation. This allows us to give a formal, categorical representation of the intuition about generic entities, with a wide range of examples, while providing a convenient theory for reasoning about sharing and multiple parameters.

Definition 5.1 (Parameterised object)**.** A *parameterised object*, denoted $\Sigma(P)$, consists of a quasi-inclusion $P \sqsubseteq \Sigma$. The objects P and Σ are called the *parameter* and the *body* of the parameterised object, respectively.

Example 5.1. It is often the case that data types are inherently generic and characterised as initial models of suitable many-sorted signatures. For example, a parameterised many-sorted signature of generic lists can be defined through the inclusion of signatures $(S_{\mathtt{ELT}}, F_{\mathtt{ELT}}) \xrightarrow{\subseteq} (S_{\mathtt{LIST}}, F_{\mathtt{LIST}})$ detailed bellow.

$$S_{\mathtt{ELT}} = \{\mathtt{Elt}\} \qquad S_{\mathtt{LIST}} = \{\mathtt{Elt}, \mathtt{List}\}$$
$$F_{\mathtt{ELT}} = \emptyset \qquad F_{\mathtt{LIST}} = \{\mathtt{nil} \colon [] \to \mathtt{List}, \_\_ \colon \mathtt{Elt}\ \mathtt{List} \to \mathtt{List}\}$$

The instantiation of parameters takes into account the possible sharing between the body of the considered parameterised object and the instance of the parameter.

Definition 5.2 (Parameter instantiation)**.** Given any parameterised object $\Sigma(P)$ and morphism $v \colon P \to P'$ that preserves P', the *instance of the parameterised object* $\Sigma(P)$ *by* v, denoted $\Sigma(P \Leftarrow v)$, is defined by the pushout square depicted below.

$$\begin{array}{ccc} P \sqcup (\Sigma \sqcap P') & \xrightarrow{\sqsubseteq} & \Sigma \\ {\scriptstyle v \vee (\Sigma \sqcap P' \sqsubseteq P')}\downarrow & & \downarrow{\scriptstyle v'} \\ P' & \xrightarrow[i]{} & \Sigma(P \Leftarrow v) \end{array}$$

Example 5.2. The many-sorted signature $(S^{\mathtt{NAT}}_{\mathtt{LIST}}, F^{\mathtt{NAT}}_{\mathtt{LIST}})$ of lists of natural numbers can be obtained through the instantiation of the parameterised signature of generic lists, $(S_{\mathtt{LIST}}, F_{\mathtt{LIST}})(S_{\mathtt{ELT}}, F_{\mathtt{ELT}})$, by the fitting argument morphism $v \colon (S_{\mathtt{ELT}}, F_{\mathtt{ELT}}) \to (S_{\mathtt{NAT}}, F_{\mathtt{NAT}})$ that maps the sort `Elt` to `Nat`. The following diagram gives a simplified representation of the relevant pushout square.

$$\begin{array}{ccc} \Big(\{\mathtt{Elt}\}, \emptyset\Big) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Elt}, \mathtt{List}\},\\ \{\mathtt{nil} \colon [] \to \mathtt{List}, \_\_ \colon \mathtt{Elt}\ \mathtt{List} \to \mathtt{List}\}\end{array}\right) \\ {\scriptstyle \mathtt{Elt} \mapsto \mathtt{Nat}}\downarrow & & \downarrow{\scriptstyle \mathtt{Elt} \mapsto \mathtt{Nat}} \\ \left(\begin{array}{l}\{\mathtt{Nat}\},\\ \{\mathtt{0} \colon [] \to \mathtt{Nat}, \mathtt{s}\_ \colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right) & \xrightarrow[\subseteq]{} & \left(\begin{array}{l}\{\mathtt{Nat}, \mathtt{List}\},\\ \{\mathtt{0} \colon [] \to \mathtt{Nat}, \mathtt{s}\_ \colon \mathtt{Nat} \to \mathtt{Nat},\\ \ \mathtt{nil} \colon [] \to \mathtt{List}, \_\_ \colon \mathtt{Nat}\ \mathtt{List} \to \mathtt{List}\}\end{array}\right) \end{array}$$

Under certain additional constraints satisfied by the quasi-inclusion system, the instantiation of parameters can be equivalently defined through a simpler pushout square, whose construction requires only the use of unions. This approach to instantiation of parameters was first discussed in [13].

Proposition 5.1. *If the quasi-inclusion system is distributive and every intersection-union square is a pushout square then the instantiation of parameters presented in Definition 5.2 can be equivalently obtained through a pushout square as depicted below.*

$$\begin{array}{ccc} P \sqcup P' & \xrightarrow{\sqsubseteq} & \Sigma \sqcup P' \\ {\scriptstyle v \vee 1_{P'}}\downarrow & & \downarrow{\scriptstyle \vartheta} \\ P' & \xrightarrow[i]{} & \Sigma(P \Leftarrow v) \end{array}$$

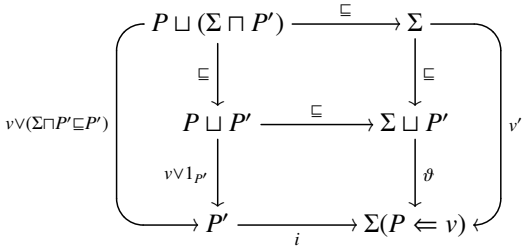
Proof. We will show that any instance $\Sigma(P \Leftarrow v)$ of a parameterised object $\Sigma(P)$ that can be obtained through the construction of a pushout square as in Definition 5.2 can also be obtained through a pushout square as depicted above, and vice versa. To this end, the first step in our proof is to show that the existence of any of the two commutative squares follows from the existence of the other.

Let us begin by noticing that the join $v \vee (\Sigma \sqcap P' \sqsubseteq P')$ can be factored as $(P \sqcup (\Sigma \sqcap P') \sqsubseteq P \sqcap P'); (v \vee 1_{P'})$. This follows from the universal property of $P \sqcup (\Sigma \sqcap P')$, i.e. from the fact that the intersection-union square $[P \sqcap (\Sigma \sqcap P'), P, \Sigma \sqcap P', P \sqcup (\Sigma \sqcap P')]$ is also a pushout square, by composing the two morphisms to the left with $P \sqsubseteq P \sqcup (\Sigma \sqcap P')$ and $\Sigma \sqcap P' \sqsubseteq P \sqcup (\Sigma \sqcap P')$.

$$\begin{aligned}(P \sqsubseteq P \sqcup (\Sigma \sqcap P')); &(P \sqcup (\Sigma \sqcap P') \sqsubseteq P \sqcup P'); (v \vee 1_{P'}) \\ &= (P \sqsubseteq P \sqcup P'); (v \vee 1_{P'}) \\ &= v \\ &= (P \sqsubseteq P \sqcup (\Sigma \sqcap P')); (v \vee (\Sigma \sqcap P' \sqsubseteq P'))\end{aligned}$$

$$\begin{aligned}(\Sigma \sqcap P' \sqsubseteq P \sqcup (\Sigma \sqcap P')); &(P \sqcup (\Sigma \sqcap P') \sqsubseteq P \sqcup P'); (v \vee 1_{P'}) \\ &= (\Sigma \sqcap P' \sqsubseteq P'); (P' \sqsubseteq P \sqcup P'); (v \vee 1_{P'}) \\ &= (\Sigma \sqcap P' \sqsubseteq P') \\ &= (\Sigma \sqcap P' \sqsubseteq P \sqcup (\Sigma \sqcap P')); (v \vee (\Sigma \sqcap P' \sqsubseteq P'))\end{aligned}$$

This observation allows us to draw the following commutative diagram by choosing v' as $(\Sigma \sqsubseteq \Sigma \sqcap P'); \vartheta$, for any morphisms ϑ and i such that the lower square commutes. Hence, whenever the pushout square $[P \sqcup P', \Sigma \sqcup P', P', \Sigma(P \Leftarrow v)]$ exists, we can build a commutative square $[P \sqcup (\Sigma \sqcap P'), \Sigma, P', \Sigma(P \Leftarrow v)]$ as in Definition 5.2.



Conversely, given any two morphisms v' and i such that the outer square is commutative, we have that

$$\begin{aligned}(\Sigma \sqcap P' \sqsubseteq \Sigma); v' &= (\Sigma \sqcap P' \sqsubseteq P \sqcup (\Sigma \sqcap P')); (P \sqcup (\Sigma \sqcap P') \sqsubseteq \Sigma); v' \\ &= (\Sigma \sqcap P' \sqsubseteq P \sqcup (\Sigma \sqcap P')); (v \vee (\Sigma \sqcap P' \sqsubseteq P')); i \\ &= (\Sigma \sqcap P' \sqsubseteq P'); i,\end{aligned}$$

which means that v' and i are compatible. For this reason we can choose the morphism $\vartheta = v' \vee i$ such that v' can also be written as $(\Sigma \sqsubseteq \Sigma \sqcap P'); \vartheta$. Note that the choice of ϑ also verifies the equality $(P \sqcup P' \sqsubseteq \Sigma \sqcup P'); \vartheta = (v \vee 1_{P'}); i$; this follows from the universal property of the union $P \sqcup P'$, as presented below.

$$\begin{aligned}(P \sqsubseteq P \sqcup P'); (P \sqcup P' \sqsubseteq \Sigma \sqcup P'); \vartheta &= (P \sqsubseteq P \sqcup (\Sigma \sqcap P')); (P \sqcup (\Sigma \sqcap P') \sqsubseteq \Sigma); (\Sigma \sqsubseteq \Sigma \sqcup P'); (v' \vee i) \\ &= (P \sqsubseteq P \sqcup (\Sigma \sqcap P')); (P \sqcup (\Sigma \sqcap P') \sqsubseteq P \sqcup P'); (v \vee 1_{P'}); i \\ &= (P \sqsubseteq P \sqcup P'); (v \vee 1_{P'}); i\end{aligned}$$

$$\begin{aligned}(P' \sqsubseteq P \sqcup P'); (P \sqcup P' \sqsubseteq \Sigma \sqcup P'); \vartheta &= (P' \sqsubseteq \Sigma \sqcup P'); (v' \vee i) \\ &= (P' \sqsubseteq P \sqcup P'); (v \vee 1_{P'}); i\end{aligned}$$

The next step is to show that the outer square $[P \sqcup (\Sigma \sqcap P'), \Sigma, P', \Sigma(P \Leftarrow v)]$ describes a pushout if and only if the lower square $[P \sqcup P', \Sigma \sqcup P', P', \Sigma(P \Leftarrow v)]$ describes a pushout. According to a general result about gluing together pushout squares [1], it is enough to prove that the upper square $[P \sqcup (\Sigma \sqcap P'), \Sigma, P \sqcup P', \Sigma \sqcup P']$ is a pushout square. For

this we show that the upper square is an intersection-union square and thus a pushout square by hypothesis. We need to check that $(P \sqcup P') \sqcup \Sigma \cong_{\mathbb{I}} \Sigma \sqcup P'$ and $(P \sqcup P') \sqcap \Sigma \cong_{\mathbb{I}} P \sqcup (\Sigma \sqcap P')$. The first relation is trivial since the union is associative and commutative (modulo a quasi-inclusive isomorphism) and P is a subobject of Σ. For the second relation we use, in addition, the distributivity property to obtain $(P \sqcup P') \sqcap \Sigma \cong_{\mathbb{I}} (P \sqcap \Sigma) \sqcup (P' \sqcap \Sigma) \cong_{\mathbb{I}} P \sqcup (\Sigma \sqcap P')$. □

Assumption 5.2. *For the remaining part of this section we will assume the considered quasi-inclusion system is distributive and that any intersection-union square is a pushout square.*

The characterisation provided by Proposition 5.1 allows us to define a stronger, more restricted form of parameter instantiation that proves to be more convenient for dealing with objects that have multiple parameters.

Definition 5.3 (Parameter instantiation based on free extensions)**.** Let $\Sigma(P)$ be a parameterised object and $v\colon P \to P'$ a fitting argument morphism preserving P'. The *instance of* $\Sigma(P)$ *by* v *based on free extensions* is defined as the codomain $\Sigma(P \Leftarrow v)$ of the free extension ϑ of $v \vee 1_{P'}$ along the quasi-inclusion $P \sqcup P' \sqsubseteq \Sigma \sqcup P'$.

$$\begin{array}{ccc} P \sqcup P' & \xrightarrow{\sqsubseteq} & \Sigma \sqcup P' \\ {\scriptstyle v \vee 1_{P'}}\downarrow & & \downarrow{\scriptstyle \vartheta} \\ P' & \xrightarrow[\sqsubseteq]{} & \Sigma(P \Leftarrow v) \end{array}$$

Note that for any morphism $v\colon P \to P'$ that preserves its codomain P', such as the ones used in the instantiation of parameters, the join $v \vee 1_{P'}$ trivially verifies $(P' \sqsubseteq P \sqcup P'); (v \vee 1_{P'}) = 1_{P'}$. It follows that $v \vee 1_{P'}$ is a retract having as section a quasi-inclusion. Moreover, by Proposition 3.4, $v \vee 1_{P'}$ is an abstract surjection.

Multiple parameters

Although highly useful, simple parameterisation is quite limited from a structural point of view due to its monadic nature. We can obtain more general parameterisation, with an arbitrarily large number of parameters, by iterating simple parameterisation for a given list of parameters. The theory presented in this paper makes no additional assumptions on the parameters; it supports sharing not only between the body of a parameterised object and an instance of one of its parameters, but also between the parameters.

Definition 5.4 (Multiple parameters)**.** A *multiple-parameterised object* is an object with a finite number of parameters $\{P_i \sqsubseteq \Sigma \mid i \in [n]\}$, denoted $\Sigma(P_i \mid i \in [n])$.

Example 5.3. One of the simplest and most natural illustrations of the notion of multiple-parameterised object emerges from the concept of Cartesian product of sets. A multiple-parameterised many-sorted signature of generic pairs may be presented by the cospan $(S_{\texttt{ELT}_1}, F_{\texttt{ELT}_1}) \xrightarrow{\subseteq} (S_{\texttt{PAIR}}, F_{\texttt{PAIR}}) \xleftarrow{\supseteq} (S_{\texttt{ELT}_2}, F_{\texttt{ELT}_2})$ detailed below.

$$\begin{array}{lll} S_{\texttt{ELT}_1} = \{\texttt{Elt}_1\} & S_{\texttt{PAIR}} = \{\texttt{Elt}_1, \texttt{Elt}_2, \texttt{Pair}\} & S_{\texttt{ELT}_2} = \{\texttt{Elt}_2\} \\ F_{\texttt{ELT}_1} = \emptyset & F_{\texttt{PAIR}} = \{\langle \_, \_ \rangle\colon \texttt{Elt}_1\ \texttt{Elt}_2 \to \texttt{Pair}\} & F_{\texttt{ELT}_2} = \emptyset \end{array}$$

Note that every multiple-parameterised object can be seen as a single-parameterised one by taking the union of its parameters; thus, the contribution of the concept of multiple-parameterised object is only notational.

Fact 5.1. For any multiple-parameterised object $\Sigma(P_i \mid i \in [n])$, $\Sigma(\bigsqcup_{i\in[n]} P_i)$ is a single-parameterised object.

The following result allows us to define the simultaneous instantiation of multiple parameters as a particular case of single-parameter instantiation.

Proposition 5.2. *Let* $\{v_i\colon P_i \to P'_i \mid i \in [n]\}$ *be a (non-empty) set of pairwise compatible morphisms. Then there exists a unique morphism* $\bigvee_{i\in[n]} v_i\colon \bigsqcup_{i\in[n]} P_i \to \bigsqcup_{i\in[n]} P'_i$ *such that*

$$\left(P_j \sqsubseteq \bigsqcup_{i\in[n]} P_i\right); \bigvee_{i\in[n]} v_i = v_j; \left(P'_j \sqsubseteq \bigsqcup_{i\in[n]} P'_i\right), \quad \textit{for any } j \in [n].$$

Moreover, for any set of objects $\{Q_j \mid j \in [m]\}$*, if* v_i *preserves* Q_j*, for all* $i \in [n]$ *and* $j \in [m]$*, then the join* $\bigvee_{i\in[n]} v_i$ *preserves the union* $\bigsqcup_{j\in[m]} Q_j$*.*

Proof. Without loss of generality we may assume that $\bigsqcup_{i\in[n]} P_i$ and $\bigsqcup_{i\in[n]} P'_i$ are the iterated unions given by

$$\bigsqcup_{i\in[n+1]} P_i = \bigsqcup_{i\in[n]} P_i \sqcup P_n \quad \text{and} \quad \bigsqcup_{i\in[n+1]} P'_i = \bigsqcup_{i\in[n]} P'_i \sqcup P'_n$$

for any $n > 0$, and by $\bigsqcup_{i\in[1]} P_i = P_0$ and $\bigsqcup_{i\in[1]} P'_i = P'_0$.

We prove the conclusion of the first part of the proposition by induction on the number of morphisms. In the base case, the statement is trivial. For the induction step, by the distributivity of the quasi-inclusion system, notice that $\bigvee_{i\in[n]} v_i$ and v_n are compatible, since the morphisms $\{v_i \colon P_i \to P'_i \mid i \in [n+1]\}$ are pairwise compatible. Consequently, we can define

$$\bigvee_{i\in[n+1]} v_i = \bigvee_{i\in[n]} v_i \vee v_n.$$

Following a direct calculation one can easily see that $\bigvee_{i\in[n+1]} v_i$ is unique with the property stated above.

For the second part of the statement let us consider a set of objects $\{Q_j \mid j \in [m]\}$, and assume that v_i preserves Q_j, for any $i \in [n]$ and $j \in [m]$, or equivalently, by Fact 3.3, that $(P_i \sqcap Q_j \sqsubseteq P_i); v_i$ is quasi-inclusive, for any $i \in [n]$ and $j \in [m]$. By the distributivity of the quasi-inclusion system, the conclusion will follow once we prove that $(P_i \sqcap Q_j \sqsubseteq \bigsqcup_{k\in[n]} P_k); \bigvee_{k\in[n]} v_k$ is quasi-inclusive, for any $i \in [n]$ and $j \in [m]$. This is achieved through the next calculation.

$$\begin{aligned}
\Big(P_i \sqcap Q_j \sqsubseteq \bigsqcup_{k\in[n]} P_k\Big); \bigvee_{k\in[n]} v_k &= (P_i \sqcap Q_j \sqsubseteq P_i); \Big(P_i \sqsubseteq \bigsqcup_{k\in[n]} P_k\Big); \bigvee_{k\in[n]} v_k \\
&= (P_i \sqcap Q_j \sqsubseteq P_i); v_i; \Big(P'_i \sqsubseteq \bigsqcup_{k\in[n]} P'_k\Big) \\
&= \Big(P_i \sqcap Q_j \sqsubseteq \bigsqcup_{k\in[n]} P'_k\Big)
\end{aligned}$$

□

Definition 5.5 (Simultaneous instantiation of parameters)**.** Let us consider a multiple-parameterised object $\Sigma(P_i \mid i \in [n])$ and a set of pairwise compatible morphisms $\{v_i \colon P_i \to P'_i \mid i \in [n]\}$ such that any morphism v_i preserves any object P'_j, for $i, j \in [n]$. The *simultaneous instantiation* of $\Sigma(P_i \mid i \in [n])$ by $\{v_i \colon P_i \to P'_i \mid i \in [n]\}$, denoted $\Sigma(P_i \Leftarrow v_i \mid i \in [n])$, is defined as the single-parameter instantiation $\Sigma(\bigsqcup_{i\in[n]} P_i \Leftarrow \bigvee_{i\in[n]} v_i)$. In addition, the simultaneous instantiation $\Sigma(P_i \Leftarrow v_i \mid i \in [n])$ is said to be *based on free extensions* when $\Sigma(\bigsqcup_{i\in[n]} P_i \Leftarrow \bigvee_{i\in[n]} v_i)$ is based on free extensions.

Example 5.4. In $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$, the many-sorted signature $\left(S^{\mathtt{NAT},\mathtt{BOOL}}_{\mathtt{PAIR}}, F^{\mathtt{NAT},\mathtt{BOOL}}_{\mathtt{PAIR}}\right)$ of pairs of natural numbers and Boolean values can be obtained by simultaneously instantiating the two parameters of the generic specification of pairs, $(S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})((S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}), (S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}))$. The fitting argument morphisms we consider are

$$v_1 \colon (S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \to (S_{\mathtt{NAT}}, F_{\mathtt{NAT}}) \quad \text{and} \quad v_2 \colon (S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}) \to (S_{\mathtt{BOOL}}, F_{\mathtt{BOOL}}),$$

mapping the sorts $\mathtt{Elt}_1$ and $\mathtt{Elt}_2$ to $\mathtt{Nat}$ and $\mathtt{Bool}$, respectively. They are compatible and preserve both $(S_{\mathtt{Nat}}, F_{\mathtt{NAT}})$ and $(S_{\mathtt{BOOL}}, F_{\mathtt{BOOL}})$, thus allowing us to draw the following instantiation diagram.

$$\begin{array}{ccc}
\Big(\{\mathtt{Elt}_1, \mathtt{Elt}_2\}, \emptyset\Big) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Elt}_1, \mathtt{Elt}_2, \mathtt{Pair}\},\\ \{\langle _, _\rangle \colon \mathtt{Elt}_1\ \mathtt{Elt}_2 \to \mathtt{Pair}\}\end{array}\right) \\
\Big\downarrow{\scriptstyle \mathtt{Elt}_1\mapsto\mathtt{Nat},\ \mathtt{Elt}_2\mapsto\mathtt{Bool}} & & \Big\downarrow{\scriptstyle \mathtt{Elt}_1\mapsto\mathtt{Nat},\ \mathtt{Elt}_2\mapsto\mathtt{Bool}} \\
\left(\begin{array}{l}\{\mathtt{Nat}, \mathtt{Bool}\},\\ \{\mathtt{0} \colon [] \to \mathtt{Nat}, \mathtt{s_} \colon \mathtt{Nat} \to \mathtt{Nat},\\ \ \mathtt{true} \colon [] \to \mathtt{Bool}, \mathtt{false} \colon [] \to \mathtt{Bool}\}\end{array}\right) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Nat}, \mathtt{Bool}, \mathtt{List}\},\\ \{\mathtt{0} \colon [] \to \mathtt{Nat}, \mathtt{s_} \colon \mathtt{Nat} \to \mathtt{Nat},\\ \ \mathtt{true} \colon [] \to \mathtt{Bool}, \mathtt{false} \colon [] \to \mathtt{Bool},\\ \ \langle _, _\rangle \colon \mathtt{Nat}\ \mathtt{Bool} \to \mathtt{Pair}\}\end{array}\right)
\end{array}$$

What is distinctive about multiple-parameterised objects is the possibility to instantiate them partially and further obtain, in this way, new parameterised objects with more or less parameters than the original one, depending on the instances of the parameters. For example, we can instantiate the $(S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1})$ parameter of the generic signature $(S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})((S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}), (S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}))$ to obtain a new generic signature that corresponds to pairs of natural numbers and arbitrary elements. In order to achieve this, we temporarily forget the other parameter, and regard the generic signature of pairs as a single-parameterised object $(S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})(S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1})$. This object can then be instantiated by v_1 to obtain the parameterised signature $\left(S^{\mathtt{NAT}}_{\mathtt{PAIR}}, F^{\mathtt{NAT}}_{\mathtt{PAIR}}\right)(S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2})$, with the sorts $\{\mathtt{Nat}, \mathtt{Elt}_2, \mathtt{Pair}\}$ and the operations $\{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s}\_\colon \mathtt{Nat} \to \mathtt{Nat}, \langle \_, \_ \rangle\colon \mathtt{Nat}\ \mathtt{Elt}_2 \to \mathtt{Pair}\}$. Note that the signature morphisms involved in this process have been chosen such that all the symbols of $(S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2})$ are preserved, and hence we can continue with the instantiation of the remaining parameter through the morphism v_2 such that we obtain the same signature $\left(S^{\mathtt{NAT,BOOL}}_{\mathtt{PAIR}}, F^{\mathtt{NAT,BOOL}}_{\mathtt{PAIR}}\right)$ as in the simultaneous instantiation $(S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})((S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \Leftarrow v_1, (S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}) \Leftarrow v_2)$.

In a general setting, the result of the instantiation of a given parameterised object by some fixed fitting argument morphism is unique only up to isomorphism. In many concrete situations this closure of the class of instances under arbitrary renaming may lead to unintended identification of some syntactic elements. For instance, in the case of the parameterised signature of pairs, after the first instantiation step we could have obtained a signature with an isomorphic but different set of sorts than $\{\mathtt{Nat}, \mathtt{Elt}_2, \mathtt{Pair}\}$; consequently, the desired sequence of instantiations meant to provide the signature $(S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})((S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \Leftarrow v_1, (S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}) \Leftarrow v_2)$ would have been interrupted. Such situations can be avoided by restricting the class of possible isomorphic results of the instantiations.

Proposition 5.3. *Let $\Sigma(P_i \mid i \in [n])$ be a multiple-parameterised object and $v_i\colon P_i \to P'_i$ a morphism that preserves P'_i and all P_j, for $j \in [n] \setminus \{i\}$. If the instantiation of parameters is based on free extensions then $\Sigma(P_i \Leftarrow v_i)$ is a parameterised object, with the parameters $\{P_j \mid j \in [n] \setminus \{i\}\}$.*

Proof. Let us consider the following pushout square used in the instantiation of $\Sigma(P_i \mid i \in [n])$ by v_i.

$$\begin{array}{ccc} P_i \sqcup P'_i & \xrightarrow{\sqsubseteq} & \Sigma \sqcup P'_i \\ {\scriptstyle v_i \vee 1_{P'_i}}\downarrow & & \downarrow{\scriptstyle \vartheta_i} \\ P'_i & \xrightarrow[\sqsubseteq]{} & \Sigma(P_i \Leftarrow v_i) \end{array}$$

Our aim is to prove that, for any $j \in [n] \setminus \{i\}$, P_j is a subobject of $\Sigma(P_i \Leftarrow v_i)$. For this it is enough to show that ϑ_i preserves all the parameters P_j, with $j \in [n] \setminus \{i\}$.

Let P_j be a parameter of Σ such that $j \in [n] \setminus \{i\}$. From hypothesis we know that P_j is preserved by v_i. One can easily see that it is also preserved by $1_{P'_i}$, and thus, by Proposition 3.10, deduce that $v_i \vee 1_{P'_i}$ preserves the object P_j. Then ϑ_i also preserves P_j because it is a free extension of $v_i \vee 1_{P'_i}$. □

Proposition 5.3 allows us to consider partial instances of multiple-parameterised objects. In this sense, a *partial instance* of a multiple-parameterised object $\Sigma(P_i \mid i \in [n])$ is just an instance of $\Sigma(P_i \mid i \in I)$, for some subset I of $[n]$, by fitting argument morphisms that preserve the parameters $\{P_i \mid i \in [n] \setminus I\}$. The use of free extensions guarantees that the uninstantiated parameters of $\Sigma(P_i \mid i \in [n])$, i.e. the objects P_i such that $i \in [n] \setminus I$, remain parameters of the partial instances. Therefore, we can continue the instantiation process until we obtain a (complete) instance.

Definition 5.6 (Sequential instantiation of parameters)**.** Let us consider a multiple-parameterised object $\Sigma(P_i \mid i \in [n])$ and a set of pairwise compatible morphisms $\{v_i\colon P_i \to P'_i \mid i \in [n]\}$ such that for every $i \in [n]$, v_i preserves P'_i and all P_j, where $j \in [n] \setminus [i+1]$. The *sequential instantiation* of $\Sigma(P_i \mid i \in [n])$ by $\{v_i\colon P_i \to P'_i \mid i \in [n]\}$, denoted $\Sigma(P_i \Leftarrow v_i)_{i \in [n]}$, is defined as the iterated single-parameter instantiation (based on free extensions) $\Sigma(P_0 \Leftarrow v_0) \cdots (P_{n-1} \Leftarrow v_{n-1})$.

Example 5.5. The many-sorted signature $\left(S^{\mathtt{NAT,BOOL}}_{\mathtt{PAIR}}, F^{\mathtt{NAT,BOOL}}_{\mathtt{PAIR}}\right)$ of pairs of natural numbers and Boolean values can be obtained by sequentially instantiating the parameters $(S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1})$ and $(S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2})$ of the signature $(S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})$

through the fitting argument morphisms v_1 and v_2, respectively. This process follows the diagram depicted below.

$$\begin{array}{ccc}
(S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \cup (S_{\mathtt{NAT}}, F_{\mathtt{NAT}}) & \xrightarrow{\subseteq} & (S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}}) \cup (S_{\mathtt{NAT}}, F_{\mathtt{NAT}}) \\
\downarrow{\scriptstyle v_1 \vee 1_{(S_{\mathtt{NAT}}, F_{\mathtt{NAT}})}} & & \downarrow{\scriptstyle \vartheta_1} \\
(S_{\mathtt{NAT}}, F_{\mathtt{NAT}}) & \xrightarrow[\subseteq]{} & (S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})((S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \Leftarrow v_1) \\
 & & \downarrow{\scriptstyle \subseteq} \\
(S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}) \cup (S_{\mathtt{BOOL}}, F_{\mathtt{BOOL}}) & \xrightarrow{\subseteq} & (S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})((S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \Leftarrow v_1) \cup (S_{\mathtt{BOOL}}, F_{\mathtt{BOOL}}) \\
\downarrow{\scriptstyle v_2 \vee 1_{(S_{\mathtt{BOOL}}, F_{\mathtt{BOOL}})}} & & \downarrow{\scriptstyle \vartheta_2} \\
(S_{\mathtt{BOOL}}, F_{\mathtt{BOOL}}) & \xrightarrow[\subseteq]{} & (S_{\mathtt{PAIR}}, F_{\mathtt{PAIR}})((S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \Leftarrow v_1)((S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}) \Leftarrow v_2)
\end{array}$$

The relevant details of the free extensions ϑ_1 and ϑ_2 considered above can be seen in the next two diagrams.

$$\begin{array}{ccc}
\left(\begin{array}{l}\{\mathtt{Elt}_1, \mathtt{Nat}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s_}\colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Elt}_1, \mathtt{Elt}_2, \mathtt{Pair}, \mathtt{Nat}\},\\ \{\langle _, _\rangle\colon \mathtt{Elt}_1\ \mathtt{Elt}_2 \to \mathtt{Pair},\\ \ \mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s_}\colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right) \\
\downarrow{\scriptstyle \mathtt{Elt}_1 \mapsto \mathtt{Nat}} & & \downarrow{\scriptstyle \mathtt{Elt}_1 \mapsto \mathtt{Nat}} \\
\left(\begin{array}{l}\{\mathtt{Nat}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s_}\colon \mathtt{Nat} \to \mathtt{Nat}\}\end{array}\right) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Nat}, \mathtt{Elt}_2, \mathtt{Pair}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s_}\colon \mathtt{Nat} \to \mathtt{Nat},\\ \ \langle _, _\rangle\colon \mathtt{Nat}\ \mathtt{Elt}_2 \to \mathtt{Pair}\}\end{array}\right)
\end{array}$$

$$\begin{array}{ccc}
\left(\begin{array}{l}\{\mathtt{Elt}_2, \mathtt{Bool}\},\\ \{\mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\}\end{array}\right) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Nat}, \mathtt{Elt}_2, \mathtt{Pair}, \mathtt{Bool}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s_}\colon \mathtt{Nat} \to \mathtt{Nat},\\ \ \langle _, _\rangle\colon \mathtt{Nat}\ \mathtt{Elt}_2 \to \mathtt{Pair},\\ \ \mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\}\end{array}\right) \\
\downarrow{\scriptstyle \mathtt{Elt}_2 \mapsto \mathtt{Bool}} & & \downarrow{\scriptstyle \mathtt{Elt}_2 \mapsto \mathtt{Bool}} \\
\left(\begin{array}{l}\{\mathtt{Bool}\},\\ \{\mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool}\}\end{array}\right) & \xrightarrow{\subseteq} & \left(\begin{array}{l}\{\mathtt{Nat}, \mathtt{Bool}, \mathtt{Pair}\},\\ \{\mathtt{0}\colon [] \to \mathtt{Nat}, \mathtt{s_}\colon \mathtt{Nat} \to \mathtt{Nat},\\ \ \mathtt{true}\colon [] \to \mathtt{Bool}, \mathtt{false}\colon [] \to \mathtt{Bool},\\ \ \langle _, _\rangle\colon \mathtt{Nat}\ \mathtt{Bool} \to \mathtt{Pair}\}\end{array}\right)
\end{array}$$

We now have two main instantiation procedures for multiple-parameterised objects. It feels natural to ask whether or not they produce isomorphic results. Reflecting on their corresponding premises, let us first observe that the possible situations in which they can be applied form intersecting classes, in the sense that in some cases only simultaneous instantiation can be used, while in others we can discuss solely about sequential instantiation. In our study we will consider those situations in which the premises of both instantiation procedures are satisfied.

In general, intricate sharing between the instances of the parameters and the body of the parameterised object may lead to non-isomorphic results of the two instantiation procedures, even when the simultaneous instantiation is also based on free extensions. A first example was discussed in [13] with respect to more specific notions of simultaneous and sequential instantiation. We consider here a slightly more involved construction, based on the many-sorted signature

$(S_{\mathtt{PAIR}^{\mathtt{obs}}}, F_{\mathtt{PAIR}^{\mathtt{obs}}})$ of generic pairs that support an observation $\mathtt{obs}$ as described below.

$$\begin{aligned} S_{\mathtt{PAIR}^{\mathtt{obs}}} &= \{\mathtt{Elt}_1, \mathtt{Elt}_2, \mathtt{Pair}\} \\ F_{\mathtt{PAIR}^{\mathtt{obs}}} &= \{\langle \_, \_ \rangle \colon \mathtt{Elt}_1\ \mathtt{Elt}_2 \to \mathtt{Pair}, \mathtt{obs} \colon \mathtt{Pair} \to \mathtt{Elt}_1\} \end{aligned}$$

For defining the context of the instantiation procedures that we intend to analyse, let us denote by $(S_{\mathtt{NAT}^{\mathtt{obs}}}, F_{\mathtt{NAT}^{\mathtt{obs}}})$ the extension of $(S_{\mathtt{NAT}}, F_{\mathtt{NAT}})$ obtained by adding the sort $\mathtt{Pair}$ and the operation symbol $\mathtt{obs} \colon \mathtt{Pair} \to \mathtt{Nat}$. The fitting argument morphisms $v_1 \colon (S_{\mathtt{ELT}_1}, F_{\mathtt{ELT}_1}) \to (S_{\mathtt{NAT}}, F_{\mathtt{NAT}})$ and $v_2 \colon (S_{\mathtt{ELT}_2}, F_{\mathtt{ELT}_2}) \to (S_{\mathtt{NAT}^{\mathtt{obs}}}, F_{\mathtt{NAT}^{\mathtt{obs}}})$ that we take into account are both mapping the corresponding sorts of their domains to $\mathtt{Nat}$.

By simultaneously instantiating the two parameters of $(S_{\mathtt{PAIR}^{\mathtt{obs}}}, F_{\mathtt{PAIR}^{\mathtt{obs}}})$ we can obtain the many-sorted signature $\left(S^{\mathtt{NAT},\mathtt{NAT}^{\mathtt{obs}}}_{\mathtt{PAIR}^{\mathtt{obs}}}, F^{\mathtt{NAT},\mathtt{NAT}^{\mathtt{obs}}}_{\mathtt{PAIR}^{\mathtt{obs}}}\right)$ that is defined by the following sets of sorts and operation symbols.

$$\begin{aligned} S^{\mathtt{NAT},\mathtt{NAT}^{\mathtt{obs}}}_{\mathtt{PAIR}^{\mathtt{obs}}} &= \{\mathtt{Nat}, \mathtt{Pair}\} \\ F^{\mathtt{NAT},\mathtt{NAT}^{\mathtt{obs}}}_{\mathtt{PAIR}^{\mathtt{obs}}} &= \{\mathtt{0} \colon [] \to \mathtt{Nat}, \mathtt{s}\_ \colon \mathtt{Nat} \to \mathtt{Nat}, \\ &\qquad \langle \_, \_ \rangle \colon \mathtt{Nat}\ \mathtt{Nat} \to \mathtt{Pair}, \\ &\qquad \mathtt{obs} \colon \mathtt{Pair} \to \mathtt{Nat}, \\ &\qquad \mathtt{obs}' \colon \mathtt{Pair} \to \mathtt{Nat}\} \end{aligned}$$

If we instantiate $(S_{\mathtt{PAIR}^{\mathtt{obs}}}, F_{\mathtt{PAIR}^{\mathtt{obs}}})$ sequentially we may conclude the partial instantiation of the first parameter by v_1 with a result $\left(S^{\mathtt{NAT}}_{\mathtt{PAIR}^{\mathtt{obs}}}, F^{\mathtt{NAT}}_{\mathtt{PAIR}^{\mathtt{obs}}}\right)$ having the sorts $S^{\mathtt{NAT}}_{\mathtt{PAIR}^{\mathtt{obs}}} = \{\mathtt{Nat}, \mathtt{Elt}_2, \mathtt{Pair}\}$ and the operation symbols

$$\begin{aligned} F^{\mathtt{NAT}}_{\mathtt{PAIR}^{\mathtt{obs}}} &= \{\mathtt{0} \colon [] \to \mathtt{Nat}, \mathtt{s}\_ \colon \mathtt{Nat} \to \mathtt{Nat}, \\ &\qquad \langle \_, \_ \rangle \colon \mathtt{Nat}\ \mathtt{Elt}_2 \to \mathtt{Pair}, \\ &\qquad \mathtt{obs} \colon \mathtt{Pair} \to \mathtt{Nat}\}. \end{aligned}$$

This choice has a negative impact on the development of the sequential instantiation of $(S_{\mathtt{PAIR}^{\mathtt{obs}}}, F_{\mathtt{PAIR}^{\mathtt{obs}}})$. It allows the instantiation process to advance towards a signature that is not isomorphic with the one obtained through the simultaneous instantiation; for example, we may get the signature $\left(S^{\mathtt{NAT},\mathtt{NAT}^{\mathtt{obs}}}_{\mathtt{PAIR}^{\mathtt{obs}}}, \left(F^{\mathtt{NAT},\mathtt{NAT}^{\mathtt{obs}}}_{\mathtt{PAIR}^{\mathtt{obs}}}\right)'\right)$ that has the same set of sorts $\{\mathtt{Nat}, \mathtt{Pair}\}$ and a more restricted set of operation symbols

$$\begin{aligned} \left(F^{\mathtt{NAT},\mathtt{NAT}^{\mathtt{obs}}}_{\mathtt{PAIR}^{\mathtt{obs}}}\right)' &= \{\mathtt{0} \colon [] \to \mathtt{Nat}, \mathtt{s}\_ \colon \mathtt{Nat} \to \mathtt{Nat}, \\ &\qquad \langle \_, \_ \rangle \colon \mathtt{Nat}\ \mathtt{Nat} \to \mathtt{Pair}, \\ &\qquad \mathtt{obs} \colon \mathtt{Pair} \to \mathtt{Nat}\}. \end{aligned}$$

One can easily describe many other similar examples by allowing the partial instantiations to produce signatures that share with the instances of the remaining parameters symbols such as $\mathtt{obs} \colon \mathtt{Pair} \to \mathtt{Nat}$, which do not belong to the body of the parameterised signature or to the instance of the current parameter. In order to prevent such scenarios and to obtain results that are isomorphic to those provided by the simultaneous instantiation of parameters we will consider a restricted form of sequential instantiation.

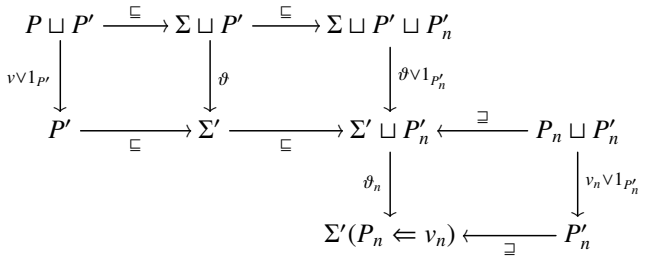
Definition 5.7 (Strong sequential instantiation of parameters)**.** For any multiple-parameterised object $\Sigma(P_i \mid i \in [n])$ and any set of pairwise compatible morphisms $\{v_i \colon P_i \to P'_i \mid i \in [n]\}$ such that v_i preserves P'_i and all P_j, for every $i \in [n]$ and $j \in [n] \setminus [i+1]$, a *sequential instantiation* $\Sigma(P_i \Leftarrow v_i)_{i \in [n]}$ is said to be *strong* if and only if any partial instance of $\Sigma(P_i \mid i \in [n])$ is obtained through a free extension that strongly preserves the considered instances of the remaining parameters.

Let us note that even though most multiple-parameterised objects of interest admit strong sequential instantiations, this property is not guaranteed in general. For instance, according to Example 4.1, Fact 3.4 and Proposition 4.1, in categories such as $\mathbb{S}\mathrm{et}$ and $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ the strong sequential instantiation is always defined, while in $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ and $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{PA}}}$ the existence of strong sequential instantiations has to be verified for each multiple-parameterised signature that is considered (see Facts 4.1 and 4.2). The following rather technical proposition advances our first main result.

Theorem 5.1. *Let $\Sigma(P_i \mid i \in [n])$ be a multiple-parameterised object and $\{v_i \colon P_i \to P'_i \mid i \in [n]\}$ a set of pairwise compatible morphisms such that for every $i \in [n]$, v_i preserves P_j and P'_k, for all $j \in [n] \setminus \{i\}$ and $k \in [n]$. If the simultaneous instantiation of parameters is based on free extensions and the sequential instantiation of parameters is strong then their results are vertices of isomorphic cocones that correspond to free extensions of the same morphism and along the same quasi-inclusion.*

Proof. We prove this result by induction on the number of parameters. For parameterised objects with only one parameter the simultaneous and the sequential instantiations coincide, and hence the conclusion is immediate.

Let us consider that the statement holds for an arbitrary but fixed number n of parameters, and denote by v the join $\bigvee_{i\in[n]} v_i$ and by P and P' the unions $\bigsqcup_{i\in[n]} P_i$ and $\bigsqcup_{i\in[n]} P'_i$, respectively. Through the induction hypothesis we can safely assume the result Σ' of the sequential instantiation is the vertex of a free extension ϑ of $v \vee 1_{P'}$ along $P \sqcup P' \sqsubseteq \Sigma \sqcup P'$. Since v_i preserves P'_n, for all $i \in [n]$, it follows by Proposition 5.2 that v preserves P'_n as well. Therefore, we can also assume, by hypothesis, that ϑ strongly preserves P'_n. These allow us to draw the following commutative diagram providing an alternative representation of the sequential instantiation $\Sigma(P_i \Leftarrow v_i)_{i\in[n+1]}$.



Notice that by Proposition 4.2 we have that $\vartheta \vee 1_{P'_n}$ is a free extension of ϑ; therefore, through the properties of the horizontal gluing of squares describing free extensions, we deduce $\vartheta \vee 1_{P'_n}$ is a free extension of $v \vee 1_{P'}$ as well.

By Proposition 5.2, the object $P_n \sqcup P'_n$ is preserved by both $v \vee 1_{P'}$ and $\vartheta \vee 1_{P'_n}$, while P' is preserved by $v_n \vee 1_{P'_n}$. Moreover, according to Fact 3.4, $P_n \sqcup P'_n$ is strongly preserved by $v \vee 1_{P'}$; it is also strongly preserved by $\vartheta \vee 1_{P'_n}$ because it is a subobject of the domain $\Sigma \sqcup P' \sqcup P'_n$ of $\vartheta \vee 1_{P'_n}$. At the same time, by Fact 3.4, the object P' is strongly preserved by $v_n \vee 1_{P'_n}$. Based on these observations, we can deduce through Proposition 4.4 that $(\vartheta \vee 1_{P'_n}); \vartheta_n$ is a free extension of the join $(v \vee 1_{P'}) \vee (v_n \vee 1_{P'_n}) \colon (P \sqcup P_n) \sqcup (P' \sqcup P'_n) \to (P' \sqcup P'_n)$, which can also be written as $\bigvee_{i\in[n+1]} v_i \vee 1_{\bigsqcup_{i\in[n+1]} P'_i}$. We have thus proved that the result $\Sigma(P_i \Leftarrow v_i)_{i\in[n+1]}$ of the sequential instantiation is the vertex of a free extension of $\bigvee_{i\in[n+1]} v_i \vee 1_{\bigsqcup_{i\in[n+1]} P'_i}$ along $\Sigma \sqcup \bigsqcup_{i\in[n+1]} P'_i$; therefore, it is isomorphic with $\Sigma(\bigsqcup_{i\in[n+1]} P_i \Leftarrow \bigvee_{i\in[n+1]} v_i)$. □

An immediate corollary of the above theorem, which concludes the present section, is that under certain hypotheses successfully fulfilled in most actual situations, for every strong sequential parameter instantiation we can choose to instantiate some of the parameters or all of them simultaneously.

Corollary 5.1. *Let $\Sigma(P_i \mid i \in [n])$ be a multiple-parameterised object and $\{v_i \colon P_i \to P'_i \mid i \in [n]\}$ a set of pairwise compatible morphisms such that for every $i \in [n]$, v_i preserves P_j and P'_k, for all $j \in [n] \setminus \{i\}$ and $k \in [n]$. If the simultaneous instantiation of parameters is based on free extensions and the sequential instantiation of parameters is strong then the two instantiation procedures produce isomorphic results.*

$$\Sigma\Big(\bigsqcup_{i\in[n]} P_i \Leftarrow \bigvee_{i\in[n]} v_i\Big) \quad \cong \quad \Sigma(P_i \Leftarrow v_i)_{i\in[n]}$$

6. Lifting Properties

The previous sections have been committed to the study of parameterisation at the level of a fixed category that is endowed with a sufficient additional structure. To be more precise, our approach to parameterisation applies to categories equipped with a distributive quasi-inclusion system that satisfies the following two properties:

– any intersection-union square is a pushout square,

– any retract whose section is a quasi-inclusion admits free extensions (along any reasonable quasi-inclusion) that strongly preserve any given object preserved by the retract.

Although very general, the theory developed so far is not flexible enough to be applied in the study of some important categories of signatures, such as the category of presentations or the category of structured specifications of a given institution. Surprisingly, in most cases, one of the first properties that fail to be satisfied is the distributivity of quasi-inclusion system, in spite of the fact that the more complex hypotheses hold, such as the one referring to the existence of free extensions.

In the study of structured specifications [29, 13] this impediment is overcome by defining the necessary concepts for parameterisation in terms of simpler underlying structures, which are given by basic signatures. Results about signatures that are relevant in the examination of parameterised specifications can then be lifted to the level of structured specifications because parameter instantiation involves a special kind of finite colimit and there exists a forgetful functor from the category of structured specifications to the category of signatures that lifts finite colimits. This fundamental result was first proved in [18] for the theories of an institution, and then extended in [29] for the more general case of structured specifications.

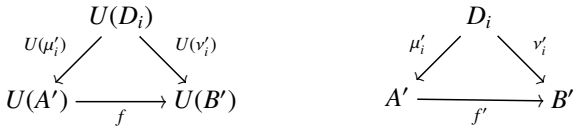
We generalize the analysis of parameterised specifications by considering distinguished functors about which we prove they lift both quasi-inclusion systems and colimits. It is easy to check that the new assumptions are naturally fulfilled by most of the concrete forgetful mappings between categories of signatures that are relevant for formal specification. We regard the class of these functors as an essential technical device for reasoning about parameterisation at the level of structured institutions.

Strong liftings

The intuition of our approach follows from the study of the quasi-inclusion systems for presentations and structured specifications of a given institution. The theory we develop here shares many similarities with the work presented in [6] that concentrates on the inheritance of inclusion systems for the theories of an institution from the category of its signatures, and also with the study of discrete and indiscrete structures from [14].

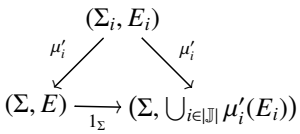
As argued in the section dedicated to quasi-inclusion systems, we are interested in investigating quasi-inclusions and strong abstract surjections. To this end, one can easily see that every quasi-inclusion of presentations of structured signatures lies in the inverse image (with respect to the forgetful signature functor) of the subcategory of quasi-inclusions of the category of underlying signatures. The case of strong abstract surjections requires a supplementary property captured by the following concept of strong cocone.

Definition 6.1 (Strong cocone)**.** Consider a functor $U\colon \mathbb{C}' \to \mathbb{C}$ and a diagram $D\colon \mathbb{J} \to \mathbb{C}'$. A cocone $\mu'\colon D \Rightarrow A'$ is *U-strong* when for all cocones $\nu'\colon D \Rightarrow B'$, every cocone homomorphism $U(\mu') \to U(\nu')$ of $D; U$ can be lifted to a cocone homomorphism $\mu' \to \nu'$ of D. More precisely, the property requires that for every arrow $f\colon U(A') \to U(B')$ of $\mathbb{C}$ such that $U(\mu'_i); f = U(\nu'_i)$, for all $i \in |\mathbb{J}|$, there exists an arrow $f'\colon A' \to B'$ of $\mathbb{C}'$ such that $U(f') = f$ and $\mu'_i; f' = \nu'_i$, for all $i \in |\mathbb{J}|$.

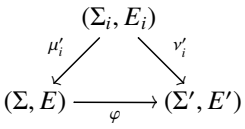


Lemma 6.1. *Let $D\colon \mathbb{J} \to \mathbb{P}\mathrm{res}^{\mathcal{I}}$ be a diagram in the category of presentations of an institution $\mathcal{I}$, and let us denote by (Σ_i, E_i) the presentation D_i, for any $i \in |\mathbb{J}|$. A cocone $\mu'\colon D \Rightarrow (\Sigma, E)$ is* Sig*-strong if and only if $E \models\!\models \bigcup_{i\in|\mathbb{J}|} \mu'_i(E_i)$.*

Proof. For the direct implication let us first notice that, given a Sig-strong cocone $\mu'\colon D \Rightarrow (\Sigma, E)$, every sentence in $\mu'_i(E_i)$ is a semantic consequence of E, for every $i \in |\mathbb{J}|$ (because μ'_i is a morphism of presentations). We deduce that $E \models \bigcup_{i\in|\mathbb{J}|} \mu'_i(E_i)$. Furthermore, since μ' is Sig-strong, it follows that the identity 1_Σ can be lifted to a morphism of presentations $(\Sigma, E) \to (\Sigma, \bigcup_{i\in|\mathbb{J}|} \mu'_i(E_i))$, and thus $\bigcup_{i\in|\mathbb{J}|} \mu'_i(E_i) \models E$.



Let us now focus on the opposite implication and consider a cocone $\nu' : D \Rightarrow (\Sigma', E')$ and a signature morphism $\varphi : \Sigma \to \Sigma'$ such that $\mu'_i ; \varphi = \nu'_i$ for all $i \in |\mathbb{J}|$. According to the definition, in order to prove that μ' is Sig-strong it suffices to show that φ is a morphism of presentations $(\Sigma, E) \to (\Sigma', E')$, i.e. $E' \models \varphi(E)$. For every $i \in |\mathbb{J}|$, since ν'_i is a morphism of presentations and $\mu'_i ; \varphi = \nu'_i$, it holds that $E' \models \varphi(\mu'_i(E_i))$; therefore, $E' \models \varphi(\bigcup_{i \in |\mathbb{J}|} \mu'_i(E_i))$. Since $E \between \bigcup_{i \in |\mathbb{J}|} \mu'_i(E_i)$ we conclude that $E' \models \varphi(E)$.



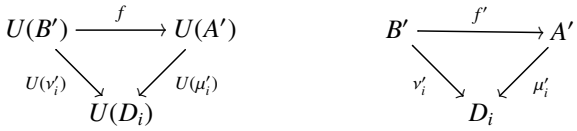
□

The proof of Lemma 6.1 can be extended without any difficulty to structured specifications.

Lemma 6.2. *Let $D : \mathbb{J} \to \mathbb{S}\mathrm{pec}^{\mathcal{I}}$ be a finite diagram in the category of structured specifications of an institution $\mathcal{I}$, and let us denote by SP_i the specification D_i, for any $i \in |\mathbb{J}|$. A cocone $\mu' : D \Rightarrow SP$ is* Sig*-strong if and only if $SP \between \bigcup_{i \in |\mathbb{J}|} SP_i \star \mu'_i$.*

The notion of closed cone is dual to the concept of strong cocone.

Definition 6.2 (Closed cone)**.** Consider a functor $U : \mathbb{C}' \to \mathbb{C}$ and a diagram $D : \mathbb{J} \to \mathbb{C}'$. A cone $\mu' : A' \Rightarrow D$ is *U-closed* when for all cones $\nu' : B' \Rightarrow D$, every cone homomorphism $f : U(\nu') \to U(\mu')$ of D; U can be lifted to a cone homomorphism $f' : \nu' \to \mu'$ of D.



The following result can be easily proved in a similar manner to Lemma 6.1.

Proposition 6.1. *Let $D : \mathbb{J} \to \mathbb{P}\mathrm{res}^{\mathcal{I}}$ be a diagram in the category of presentations of an institution $\mathcal{I}$, and let us denote by (Σ_i, E_i) the presentation D_i, for any $i \in |\mathbb{J}|$. A cone $\mu' : (\Sigma, E) \Rightarrow D$ is* Sig*-closed if and only if $E \between \bigcap_{i \in |\mathbb{J}|} \mu'^{-1}_i(E_i^{**})$.*

Note that in the case of structured specifications we cannot discuss about closed cones for arbitrary finite diagrams in the absence of a structuring operator that promotes the generalization of the intersection of presentations. Since our language allows derivation, the new dedicated operator, denoted here by $\cap$, may be defined only for structured specification with the same signature. Considering two arbitrary specifications SP_1 and SP_2 with the signature Σ, the semantics of $SP_1 \cap SP_2$ may be determined by

- $\mathrm{Sig}[SP_1 \cap SP_2] = \Sigma$,
- $\mathrm{Ax}[SP_1 \cap SP_2] = \mathrm{Ax}[SP_1]^{**} \cap \mathrm{Ax}[SP_2]^{**}$,
- $\mathrm{Mod}[SP_1 \cap SP_2] = \mathrm{Mod}[SP_1] \cup \mathrm{Mod}[SP_2]$.

Nevertheless, the closed quasi-inclusion system of the category of structured specifications only requires the concept of closed morphism, i.e. a closed cone of a diagram $D : \mathbb{1} \to \mathbb{S}\mathrm{pec}^{\mathcal{I}}$ with respect to the functor Sig. In this sense, a morphism of specifications $\varphi : SP \to SP'$ is Sig-closed if and only if $SP \between \varphi \mid S\, P'$.

The following definition introduces the main property that we require to be fulfilled by functors in order to lift quasi-inclusion systems as well as colimits. Note that by dualization we automatically obtain a property of functors that ensures the lifting of limits. Additionally, this dual property can also be proved to play a primary role in lifting quasi-inclusion systems.

Definition 6.3. A functor $U : \mathbb{C}' \to \mathbb{C}$ *strongly lifts (finite) cocones* if for any (finite) diagram $D : \mathbb{J} \to \mathbb{C}'$ and every cocone $\mu : D; U \Rightarrow A$, there exists a U-strong cocone $\mu' : D \Rightarrow A'$ such that $U(A') = A$ and $U(\mu') = \mu$. Dually, a functor $U : \mathbb{C}' \to \mathbb{C}$ *closely lifts (finite) cones* if for any (finite) diagram $D : \mathbb{J} \to \mathbb{C}'$ and every cone $\mu : A \Rightarrow D; U$, there exists a U-closed cone $\mu' : A' \Rightarrow D$ such that $U(A') = A$ and $U(\mu') = \mu$.

The idea that the signature functors of the presentations or the structured specifications of an institution have the property of strongly lifting finite cocones has been already suggested in Lemmas 6.1 and 6.2. In addition, these functors share another important property for our study on parameterisation – they are both faithful.

Proposition 6.2. *For any institution $\mathcal{I}$, the forgetful functor* $\mathsf{Sig} : \mathbb{P}\mathrm{res}^{\mathcal{I}} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$ *is faithful and it strongly lifts cocones. Let us consider a diagram $D : \mathbb{J} \to \mathbb{P}\mathrm{res}^{\mathcal{I}}$ for which we denote D_i by (Σ_i, E_i), where $i \in |\mathbb{J}|$. Then any cocone $\mu : D; \mathsf{Sig} \Rightarrow \Sigma$ admits the* Sig*-strong lifting $\mu : D \Rightarrow (\Sigma, \bigcup_{i \in |\mathbb{J}|} \mu_i(E_i))$.*

Proposition 6.3. *For any institution $\mathcal{I}$, the forgetful functor* $\mathsf{Sig} : \mathbb{S}\mathrm{pec}^{\mathcal{I}} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$ *is faithful and it strongly lifts finite cocones. Considering a finite diagram $D : \mathbb{J} \to \mathbb{S}\mathrm{pec}^{\mathcal{I}}$ for which we denote D_i by SP_i, for all $i \in |\mathbb{J}|$, any cocone $\mu : D; \mathsf{Sig} \Rightarrow \Sigma$ admits the* Sig*-strong lifting $\mu : D \Rightarrow \bigcup_{i \in |\mathbb{J}|} SP_i \star \mu_i$.*

Instances of faithful functors that strongly lift cocones can also be easily discovered by analysing the connection between order-sorted or partial algebraic signatures and their underlying many-sorted components.

Proposition 6.4. *The functor $U : \mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}} \to \mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ defined on objects by $U(S, \leq, F) = (S, F)$ and on morphisms by $U(\varphi) = \varphi$ is faithful and it strongly lifts cocones. If $D : \mathbb{J} \to \mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ is a diagram for which we denote D_i by $(S_i, \leq_i, F_i)$, for all $i \in |\mathbb{J}|$, any cocone $\mu : D; U \Rightarrow (S, F)$ admits the U-strong lifting $\mu : D \Rightarrow (S, (\bigcup_{i \in |\mathbb{J}|} \mu_i^{st}(\leq_i))^{m*}, F)$.*

Proof. Let us consider a diagram $D : \mathbb{J} \to \mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ and a cocone $\mu : D; U \Rightarrow (S, F)$ as above, and let $\leq$ denote the relation $(\bigcup_{i \in |\mathbb{J}|} \mu_i^{st}(\leq_i))^{m*}$. Note that $\leq$ is the monotonic, reflexive and transitive closure of the union $\bigcup_{i \in |\mathbb{J}|} \mu_i^{st}(\leq_i)$ and that it can be equivalently described as the least monotonic preorder on S that includes $\bigcup_{i \in |\mathbb{J}|} \mu_i^{st}(\leq_i)$.

The fact that μ is a cocone $D \Rightarrow (S, \leq, F)$ in $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ is an immediate consequence of the definition of $\leq$. Hence, all we need to prove is that μ is a U-strong cocone. Let us thus consider a cocone $\nu : D \Rightarrow (S', \leq', F')$ in $\mathbb{S}\mathrm{ig}^{\underline{\mathrm{OSA}}}$ and a many-sorted signature morphism $\varphi : (S, F) \to (S', F')$ such that $\mu_i; \varphi = \nu_i$ for all $i \in |\mathbb{J}|$.

$$\begin{array}{ccc} & (S_i, \leq_i, F_i) & \\ {\scriptstyle \mu_i}\swarrow & & \searrow{\scriptstyle \nu_i} \\ (S, \leq, F) & \xrightarrow[\varphi]{} & (S', \leq', F') \end{array}$$

According to the definition of U-strong cocones it suffices to show that φ is a morphism of order-sorted signatures from $(S, \leq, F)$ to $(S', \leq', F')$. This follows easily by first noticing that the preimage $(\varphi^{st})^{-1}(\leq')$ of $\leq'$ is a monotonic preorder on S. Furthermore, since ν_i^{st} is monotonic and $\mu_i^{st}; \varphi^{st} = \nu_i^{st}$ we deduce that $\mu_i^{st}(\leq_i) \subseteq (\varphi^{st})^{-1}(\leq')$, for all $i \in |\mathbb{J}|$. By the characterisation of $\leq$ as the least monotonic preorder on S that includes the relations $\mu_i(\leq_i)$, for $i \in |\mathbb{J}|$, we deduce that $\leq \subseteq (\varphi^{st})^{-1}(\leq')$. It follows that $\varphi^{st}(\leq) \subseteq \leq'$, thus concluding our proof. □

The strong lifting of cocones from the category of many-sorted signatures to the category of partial algebraic signatures can be proved in a similar manner.

Proposition 6.5. *The functor $U : \mathbb{S}\mathrm{ig}^{\underline{\mathrm{PA}}} \to \mathbb{S}\mathrm{ig}^{\underline{\mathrm{MSA}}}$ defined on objects by $U(S, F, TF) = (S, F)$ and on morphisms by $U(\varphi) = \varphi$ is faithful and it strongly lifts cocones. If $D : \mathbb{J} \to \mathbb{S}\mathrm{ig}^{\underline{\mathrm{PA}}}$ is a diagram for which we denote D_i by (S_i, F_i, TF_i), for all $i \in |\mathbb{J}|$, any cocone $\mu : D; U \Rightarrow (S, F)$ admits the U-strong lifting $\mu : D \Rightarrow (S, F, \bigcup_{i \in |\mathbb{J}|} \mu_i^{op}(TF_i))$.*

The following compositionality result is rather straightforward.

Fact 6.1. If $U' : \mathbb{C}'' \to \mathbb{C}'$ and $U : \mathbb{C}' \to \mathbb{C}$ are faithful functors that strongly lift cocones, then $U'; U$ is faithful and it strongly lifts cocones as well.

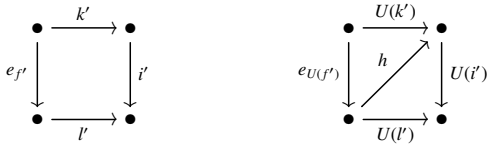
Given the high resemblance between inclusion systems and quasi-inclusion systems, the next result can be regarded as an adaptation of the construction of inclusion systems, as investigated in [6].

Theorem 6.1. *If $U\colon \mathbb{C}' \to \mathbb{C}$ is a faithful functor that strongly lifts (finite) cocones, every quasi-inclusion system $\mathbb{I}$ of $\mathbb{C}$ can be lifted to the quasi-inclusion system $U^{-1}(\mathbb{I})$ of $\mathbb{C}'$.*

Proof. Consider a functor $U\colon \mathbb{C}' \to \mathbb{C}$ satisfying the above premises, and a quasi-inclusion system $\mathbb{I}$ of the category $\mathbb{C}$. Let us first notice that $U^{-1}(\mathbb{I})$ is indeed a broad preordered subcategory of $\mathbb{C}'$, which follows immediately from the definitions of U and $\mathbb{I}$, and further denote it by $\mathbb{I}'$.

It is easy to see that for any two composable morphisms f' and i' in $\mathbb{C}'$ and $\mathbb{I}'$, respectively, f' is a quasi-inclusion whenever $f';i'$ is a quasi-inclusion. Assuming that $f';i'$ is a quasi-inclusion, or equivalently, that $U(f';i') \in \mathbb{I}$, it follows that $U(f');U(i') \in \mathbb{I}$; therefore, by the definition of the quasi-inclusion system, $U(f')$ is an arrow in $\mathbb{I}$, and thus f' is a quasi-inclusion.

The more involved part of the argument is to show that every morphism $f' \in \mathbb{C}'$ can be factored as the composition of an abstract surjection and a quasi-inclusion, i.e. it can be written as $e_{f'};i_{f'}$, with $e_{f'} \in \mathbb{E}_{\mathbb{I}'}$ and $i_{f'} \in \mathbb{I}'$. For this let us consider an arbitrary but fixed morphism $f'\colon A' \to B'$ in $\mathbb{C}'$. Then $U(f')\colon U(A') \to U(B')$ is a morphism in $\mathbb{C}$, and thus it can be factored as $U(A') \xrightarrow{e_{U(f')}} C \xrightarrow{i_{U(f')}} U(B')$. By hypothesis, the morphism $e_{U(f')}$ can be lifted to a U-strong morphism $e_{f'}\colon A' \to C'$ whose membership to $\mathbb{E}_{\mathbb{I}'}$ is discussed next. More precisely, we argue that $e_{f'}$ is an abstract surjection by checking the corresponding diagonal-fill property. Let us consider a commutative square in $\mathbb{C}'$ as below, in which i' a quasi-inclusion. By applying the functor U we obtain a commutative square in $\mathbb{C}$, with the abstract surjection $U(e_{f'}) = e_{U(f')}$ and the quasi-inclusion $U(i')$ on opposite sides. Thus we deduce the existence of a morphism h, depicted in the right square below, such that $e_{U(f')};h = U(k')$ and $h;U(i') = U(l')$.



Since $e_{f'}$ is U-strong and h satisfies $U(e_{f'});h = U(k')$ we conclude there exists a lifting h' of h such that $e'_{f};h' = k'$. Moreover, because U is faithful and $U(h');U(i') = U(l')$, we also obtain the equality $h';i' = l'$. Thus we conclude that $e_{f'}$ is an abstract surjection.

Following a similar argument, based on the U-strongness of $e_{f'}$ and the equality $U(e_{f'});i_{U(f')} = U(f')$, we deduce that $i_{U(f')}$ can be lifted to a quasi-inclusion $i_{f'}$ satisfying $e_{f'};i_{f'} = f'$. This concludes the proof. □

A trivial corollary of Theorem 6.1 is that a cocone or cocone homomorphism in $\mathbb{C}'$ consists solely of quasi-inclusions if and only if its image through U has this property. As a result, any U-strong lifting of a cocone that consists of quasi-inclusions is also strong with respect to the restriction of U to quasi-inclusions.

Corollary 6.1. *If $U\colon \mathbb{C}' \to \mathbb{C}$ is a faithful functor that strongly lifts (finite) cocones and $\mathbb{I}$ is a quasi-inclusion system of $\mathbb{C}$ then the restriction of U to a functor $U^{-1}(\mathbb{I}) \to \mathbb{I}$ is also faithful and it strongly lifts (finite) cocones.*

Based on the facts that every identity is trivially a closed morphism and that the composition of any two closed morphisms is closed as well, the following statement can be proved through an analogous technique as the one employed in the argument of Theorem 6.1.

Theorem 6.2. *If $U\colon \mathbb{C}' \to \mathbb{C}$ is a faithful functor that closely lifts (finite) cones, every quasi-inclusion system $\mathbb{I}$ of $\mathbb{C}$ can be lifted to a quasi-inclusion system of $\mathbb{C}'$ given by the U-closed morphisms of $U^{-1}(\mathbb{I})$.*

Observe that, intuitively, the functors that strongly lift cocones are used in constructing quasi-inclusion systems such that the image of any morphism is defined minimally with respect to the strong abstract surjections; the functors that closely lift cones generate quasi-inclusion systems in which the image of any morphism is defined maximally with respect to the closed abstract inclusions.

We now turn our attention on the study of constructions and properties related to the lifting of colimits.

Proposition 6.6. *Let $U\colon \mathbb{C}' \to \mathbb{C}$ be a faithful functor and $D\colon \mathbb{J} \to \mathbb{C}'$ a diagram in $\mathbb{C}'$. Then any U-strong lifting of a colimiting cocone of D; U is a colimiting cocone of D.*

Proof. Let us consider a U-strong cocone $\mu' : D \Rightarrow A'$ of D whose image through U is a colimiting cocone $\mu : D; U \Rightarrow A$ of $D; U$. It is easy to see that for any other cocone $\nu' : D \Rightarrow B'$ of D, $U(\nu') : D; U \Rightarrow U(B')$ is a cocone of $D; U$, and thus, by the universal mapping property of μ, there exists a cocone homomorphism $f : U(\mu') \to U(\nu')$. Furthermore, because μ' is U-strong, f can be lifted to a cocone homomorphism $f' : \mu' \to \nu'$. To conclude, notice that by the faithfulness of U and the uniqueness of f, there can be no other homomorphisms of cocones from μ' to ν'. □

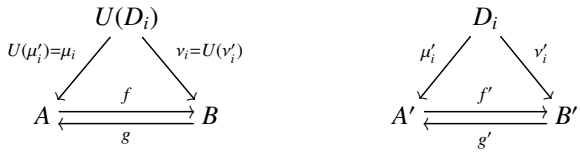
The next result follows immediately by combining the essential properties of the functors that strongly lift cocones and Proposition 6.6.

Theorem 6.3. *Every faithful functor that strongly lifts (finite) cocones lifts (finite) colimits as well.*

Notice that in our setting, the construction of colimits by strongly lifting colimiting cocones does not require all the details of the considered diagram, but only the image of the diagram in the base category and information about a particular subclass of its vertices. For example, computing the pushout of a cospan $SP_1 \to SP \leftarrow SP_2$ in the category of structured specifications depends upon its image $\mathsf{Sig}(SP_1) \to \mathsf{Sig}(SP) \leftarrow \mathsf{Sig}(SP_2)$ in the category of signatures and details about SP_1 and SP_2 only. A similar situation has been examined in [15, 14] as part of the development of algebraic semantics for coordination.

Proposition 6.7. *If $U : \mathbb{C}' \to \mathbb{C}$ is a faithful functor, the U-strong liftings of any two isomorphic cocones are also isomorphic.*

Proof. Let $D : \mathbb{J} \to \mathbb{C}'$ be a diagram in $\mathbb{C}'$ and $\mu : D; U \to A$ together with $\nu : D; U \to B$ two isomorphic cocones of $D; U$ admitting the U-strong liftings $\mu' : D \to A'$ and $\nu' : D \to B'$, respectively. Since μ and ν are isomorphic, we deduce the existence of two morphisms $f : A \to B$ and $g : B \to A$, inverse one to the other, making the diagram displayed below commutative.



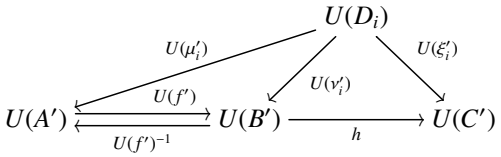
By the U-strongness of μ' and ν', the two morphisms f and g of $\mathbb{C}$ can be lifted to the morphisms $f' : A' \to B'$ and $g' : B' \to A'$, respectively, in the category $\mathbb{C}'$. Because U is faithful, by the next calculation, we deduce that f' and g' are also inverse one to the other, thus concluding the proof.

$$U(f'; g') = U(f'); U(g') = f; g = 1_A = 1_{U(A')} = U(1_{A'})$$
$$U(g'; f') = U(g'); U(f') = g; f = 1_B = 1_{U(B')} = U(1_{B'})$$

□

Proposition 6.8. *Given a functor $U : \mathbb{C}' \to \mathbb{C}$, any cocone isomorphic with a U-strong cocone is U-strong as well.*

Proof. Let $D : \mathbb{J} \to \mathbb{C}'$ be a diagram and $\mu' : D \Rightarrow A'$ together with $\nu' : D \Rightarrow B'$ two isomorphic cocones of D such that the former is U-strong. In order to show that ν' is U-strong too, let us consider an arbitrary but fixed cocone $\xi' : D \Rightarrow C'$ and a cocone homomorphism $h : U(\nu') \to U(\xi')$. We obtain the following commutative diagram, where f' denotes the isomorphism between μ' and ν'.



It is easy to check that $U(f'); h$ is a cocone homomorphism from $U(\mu')$ to $U(\xi')$. Since μ' is U-strong, we conclude there exists a cocone homomorphism $g' : \mu' \to \xi'$ such that $U(g') = U(f'); h$. We can now define the morphism $h' : B' \to C'$

as the composition $f'^{-1}; g'$. Notice that h' is a cocone homomorphism since both arrows f'^{-1} and g' are cocone homomorphisms. Moreover, by a straightforward calculation we have $U(h') = U(f'^{-1}); U(g') = U(f'^{-1}); U(f'); h = h$. We conclude that ν' is U-strong. □

Corollary 6.2. *If $U\colon \mathbb{C}' \to \mathbb{C}$ is a faithful functor, the U-strong liftings of isomorphic cocones are unique up to isomorphism.*

By now we know that whenever we have a faithful functor $U\colon \mathbb{C}' \to \mathbb{C}$ that strongly lifts cocones, we can lift any quasi-inclusion system of $\mathbb{C}$ to a quasi-inclusion system of $\mathbb{C}'$, and thus we can define parameterised objects in $\mathbb{C}'$ provided that we can define them in $\mathbb{C}$. Moreover, assuming that U strongly lifts cocones and that $\mathbb{C}$ is equipped with a quasi-inclusion system, it follows by Corollary 6.1 that the restriction of U to quasi-inclusions is a functor that strongly lifts cocones as well. As a result, by Theorem 6.3, the functor U lifts unions: for any two objects A' and B' of $\mathbb{C}'$ such that $U(A') \sqcup U(B')$ exists, the union $A' \sqcup B'$, i.e. the coproduct of A' and B' in the category of the quasi-inclusions of $\mathbb{C}'$, is given by the vertex of the U-strong lifting of the union of $U(A')$ and $U(B')$ in $\mathbb{C}$.

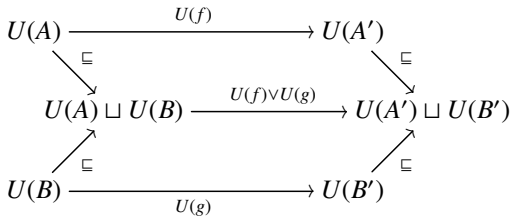
Even in the situations when U strongly lifts cocones and $\mathbb{C}$ admits a quasi-inclusion system with unions, thus ensuring the existence of unions and pushouts in $\mathbb{C}'$, we still do not have all the necessary constructions for instantiating parameters. More precisely, based on the functor U, we need to define a notion of relative compatibility between arrows that can be further used in the description of a join operator.

Definition 6.4 (Compatible arrows)**.** Let $U\colon \mathbb{C}' \to \mathbb{C}$ be a functor whose codomain is a category equipped with a quasi-inclusion system that has unions and intersections. Two arrows $f\colon A \to A'$ and $g\colon B \to B'$ of $\mathbb{C}'$ are *compatible with respect to U* or *U-compatible* when $U(f)$ and $U(g)$ are compatible in $\mathbb{C}$. An arrow $f\colon A \to A'$ *preserves* an object B *with respect to U* or more concisely, *U-preserves B*, when f and 1_B are compatible with respect to U.

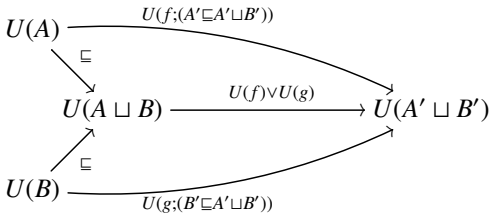
Proposition 6.9. *Let $U\colon \mathbb{C}' \to \mathbb{C}$ be a faithful functor that strongly lifts finite cocones. In addition, let us assume its codomain, the category $\mathbb{C}$, is endowed with a quasi-inclusion system having unions and intersections such that every intersection-union square is a pushout square. Then for any two U-compatible arrows $f\colon A \to A'$ and $g\colon B \to B'$, there exists a unique arrow $f \vee_U g\colon A \sqcup B \to A' \sqcup B'$ such that*

$$(A \sqsubseteq A \sqcup B); (f \vee_U g) = f; (A' \sqsubseteq A' \sqcup B') \quad \textit{and} \quad (B \sqsubseteq A \sqcup B); (f \vee_U g) = g; (B' \sqsubseteq A' \sqcup B').$$

Proof. Let us consider two U-compatible arrows $f\colon A \to A'$ and $g\colon B \to B'$. It immediately follows that $U(f)$ and $U(g)$ are compatible, and thus there exists a unique arrow $U(f) \vee U(g)$ from $U(A) \sqcup U(B)$ to $U(A') \sqcup U(B')$ making the diagram below commutative.



By Theorem 6.1, the inverse image of the quasi-inclusion system of $\mathbb{C}$ through the functor U is a quasi-inclusion system of $\mathbb{C}'$. Moreover, since U is faithful and it strongly lifts cocones it follows by Corollary 6.1 that its restriction to quasi-inclusions has these properties as well. Hence, by Theorem 6.3, the quasi-inclusion system of $\mathbb{C}'$ has unions. Furthermore, by Proposition 6.6 and Corollary 6.2, the colimiting cocones of the unions are U-strong. These observations allow us to draw the next commutative diagram, and further deduce, by the U-strongness of the corresponding cocone of the union $A \sqcup B$, the existence of the arrow $f \vee_U g\colon A \sqcup B \to A' \sqcup B'$ such that $U(f \vee_U g) = U(f) \vee U(g)$.



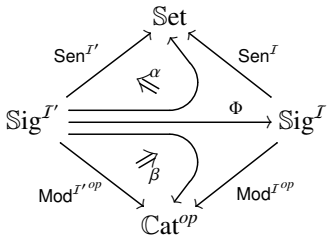
Notice that, because U is faithful, the arrow $f \vee_U g$ easily satisfies the desired equalities. Moreover, by the uniqueness of $U(f) \vee U(g)$ and the faithfulness of U, it is the only arrow of $\mathbb{C}'$ with this property. □

7. Parameterisation for Structured Institutions

Our most general approach to parameterisation lies at the highly conceptual level of structured institutions. This axiomatic resolution to structured specifications was introduced in [10] for supporting a consistent analysis of structured specifications that is independent of the choice of specification building operators. The theory reasons about structured specifications abstractly, as signatures of an upper level institution considered on top of a base institution that abstracts the underlying logical system.

We recall the fundamental concepts and examples necessary for developing and illustrating new results, and conclude with an extension of Corollary 5.1 about the isomorphic relation between the results of the sequential and the simultaneous instantiation of multiple parameters. The concept of institution morphism, introduced in [18], formalizes the structure preserving mappings from more complex institutions to simpler ones.

Definition 7.1 (Institution morphism)**.** An *institution morphism* is a triple $(\Phi, \alpha, \beta)\colon \mathcal{I}' \to \mathcal{I}$ consisting of



- a functor $\Phi\colon \mathbb{S}\mathrm{ig}^{\mathcal{I}'} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$ called *signature functor*,
- a natural transformation $\alpha\colon \Phi; \mathsf{Sen}^{\mathcal{I}} \Rightarrow \mathsf{Sen}^{\mathcal{I}'}$ called *sentence transformation*,
- a natural transformation $\beta\colon \mathsf{Mod}^{\mathcal{I}'} \Rightarrow \Phi^{op}; \mathsf{Mod}^{\mathcal{I}}$ called *model transformation*

such that the following *satisfaction condition* holds:

$$M' \models^{\mathcal{I}'}_{\Sigma'} \alpha_{\Sigma'}(\rho) \quad \text{if and only if} \quad \beta_{\Sigma'}(M') \models^{\mathcal{I}}_{\Phi(\Sigma')} \rho$$

for all $\mathcal{I}'$-signatures Σ', Σ'-model M' and $\Phi(\Sigma')$-sentence ρ.

Definition 7.2 (Structured institution)**.** An institution $\mathcal{I}' = (\mathbb{S}\mathrm{ig}^{\mathcal{I}'}, \mathsf{Sen}^{\mathcal{I}'}, \mathsf{Mod}^{\mathcal{I}'}, \models^{\mathcal{I}'})$ is *structured* over a base institution $\mathcal{I} = (\mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$ through a functor $\Phi\colon \mathbb{S}\mathrm{ig}^{\mathcal{I}'} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$, called the *structuring functor*, if Φ can be extended to an institution morphism from $\mathcal{I}'$ to $\mathcal{I}$ whose sentence transformations are all identities and model transformations are all full subcategory inclusions. More precisely, $\mathcal{I}'$ is structured over $\mathcal{I}$ through Φ if there exists an institution morphism $(\Phi, \alpha, \beta)\colon \mathcal{I}' \to \mathcal{I}$ such that the following properties hold:

- for every signature $\Sigma' \in \left|\mathbb{S}\mathrm{ig}^{\mathcal{I}'}\right|$, the arrow $\alpha_{\Sigma'}\colon \mathsf{Sen}^{\mathcal{I}}(\Phi(\Sigma')) \to \mathsf{Sen}^{\mathcal{I}'}(\Sigma')$ is the identity map of $\mathsf{Sen}^{\mathcal{I}'}(\Sigma')$; moreover, for any signature morphism $\varphi'\colon \Sigma'_1 \to \Sigma'_2$ of $\mathbb{S}\mathrm{ig}^{\mathcal{I}'}$ we obtain a commutative square as depicted below; in conclusion, the functors $\Phi; \mathsf{Sen}^{\mathcal{I}}$ and $\mathsf{Sen}^{\mathcal{I}'}$ are equal on both objects and arrows;

$$\begin{array}{ccc} \Sigma'_1 & \mathsf{Sen}^{\mathcal{I}}(\Phi(\Sigma'_1)) & \xrightarrow[1_{\mathsf{Sen}^{\mathcal{I}'}(\Sigma'_1)}]{\alpha_{\Sigma'_1}} \mathsf{Sen}^{\mathcal{I}'}(\Sigma'_1) \\ {\scriptstyle\varphi'}\downarrow & {\scriptstyle\mathsf{Sen}^{\mathcal{I}}(\Phi(\varphi'))}\downarrow & \downarrow{\scriptstyle\mathsf{Sen}^{\mathcal{I}'}(\varphi')} \\ \Sigma'_2 & \mathsf{Sen}^{\mathcal{I}}(\Phi(\Sigma'_2)) & \xrightarrow[\alpha_{\Sigma'_2}]{1_{\mathsf{Sen}^{\mathcal{I}'}(\Sigma'_2)}} \mathsf{Sen}^{\mathcal{I}'}(\Sigma'_2) \end{array}$$

– for every signature $\Sigma' \in \left|\mathbb{S}\mathrm{ig}^{\mathcal{I}'}\right|$, the arrow $\beta_{\Sigma'} : \mathsf{Mod}^{\mathcal{I}'}(\Sigma') \to \mathsf{Mod}^{\mathcal{I}}(\Phi(\Sigma'))$ is the corresponding functor of the full subcategory inclusion $\mathsf{Mod}^{\mathcal{I}'}(\Sigma') \subseteq \mathsf{Mod}^{\mathcal{I}}(\Phi(\Sigma'))$; moreover, for any signature morphism $\varphi' : \Sigma'_1 \to \Sigma'_2$ of $\mathbb{S}\mathrm{ig}^{\mathcal{I}'}$ we obtain a commutative square as presented below;

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

– for every signature $\Sigma' \in \left|\mathbb{S}\mathrm{ig}^{\mathcal{I}'}\right|$, every model $M' \in \left|\mathsf{Mod}^{\mathcal{I}'}(\Sigma')\right|$ and every sentence $e' \in \mathsf{Sen}^{\mathcal{I}'}(\Sigma')$,

$$M' \models^{\mathcal{I}'}_{\Sigma'} e' \quad \text{if and only if} \quad M' \models^{\mathcal{I}}_{\Phi(\Sigma')} e'.$$

Example 7.1. Discussed in more detail in [9], the institution $\mathcal{I}^{pres} = (\mathbb{S}\mathrm{ig}^{pres}, \mathsf{Sen}^{pres}, \mathsf{Mod}^{pres}, \models^{pres})$ of presentations over a base institution $\mathcal{I} = (\mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$ is obtained by extending the sentence functor, the model functor and the satisfaction relation from base signatures to presentations. Its structure is defined as follows:

– $\mathbb{S}\mathrm{ig}^{pres}$ is the category $\mathbb{P}\mathrm{res}^{\mathcal{I}}$ of presentations of $\mathcal{I}$,

– Sen^{pres} maps every presentation (Σ, E) to the set $\mathsf{Sen}^{\mathcal{I}}(\Sigma)$ of Σ-sentences,

– Mod^{pres} maps every presentation (Σ, E) to the full subcategory of $\mathsf{Mod}^{\mathcal{I}}(\Sigma)$ determined by the Σ-models that satisfy E,

– $\models^{pres}$ is the satisfaction relation induced by $\models^{\mathcal{I}}$, in the sense that for all (Σ, E)-models M and (Σ, E)-sentences e, $M \models^{pres}_{(\Sigma,E)} e$ if and only if $M \models^{\mathcal{I}}_{\Sigma} e$.

The institution $\mathcal{I}^{pres}$ is structured over $\mathcal{I}$ through the signature functor $\mathsf{Sig} : \mathbb{S}\mathrm{ig}^{pres} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$ that maps every presentation (Σ, E) to its underlying signature Σ.

Example 7.2. The institution $\mathcal{I}^{spec} = (\mathbb{S}\mathrm{ig}^{spec}, \mathsf{Sen}^{spec}, \mathsf{Mod}^{spec}, \models^{spec})$ of structured specifications over a base institution $\mathcal{I} = (\mathbb{S}\mathrm{ig}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$, with respect to designated classes of signature morphisms $\mathcal{T}$ and $\mathcal{D}$, is organized according to [10] as follows:

– $\mathbb{S}\mathrm{ig}^{spec}$ is the category $\mathbb{S}\mathrm{pec}^{\mathcal{I}}$ of $(\mathcal{T}, \mathcal{D})$-structured specifications of $\mathcal{I}$,

– Sen^{spec} maps every structured specification SP to the set $\mathsf{Sen}^{\mathcal{I}}(\mathsf{Sig}(SP))$ of $\mathbb{S}\mathrm{ig}(SP)$-sentences,

– Mod^{spec} maps every structured specification SP to the full subcategory of $\mathsf{Mod}^{\mathcal{I}}(\mathsf{Sig}(SP))$ determined by the models of SP,

– $\models^{spec}$ is the satisfaction relation induced by $\models^{\mathcal{I}}$.

Similarly to the case of $\mathcal{I}^{pres}$, the institution $\mathcal{I}^{spec}$ of $(\mathcal{T}, \mathcal{D})$-structured specifications is structured over its base institution $\mathcal{I}$ through the signature functor $\mathsf{Sig} : \mathbb{S}\mathrm{ig}^{spec} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$.

Our study of abstract parameterised specifications in the context of structured institutions is independent of the considered sentence and model transformations, thus allowing its presentation in a more general setting given by arbitrary institution morphisms. For this reason we employ the following terminology.

Definition 7.3 (Pre-structured institution)**.** We say that an institution $\mathcal{I}'$ is *pre-structured* over a base institution $\mathcal{I}$ through a functor $\Phi : \mathbb{S}\mathrm{ig}^{\mathcal{I}'} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$ if there exists an institution morphism from $\mathcal{I}'$ to $\mathcal{I}$ whose signature functor is Φ.

Example 7.3. The institution OSA is pre-structured over MSA through the forgetful functor that discards the order relation on sorts. This functor together with

– the sentence transformations given by the set-theoretic inclusions $\mathsf{Sen}^{\underline{\mathrm{MSA}}}(S,F) \subseteq \mathsf{Sen}^{\underline{\mathrm{OSA}}}(S,F,\leq)$, indexed by order-sorted signatures $(S,\leq,F)$, and

– the model transformations given by the subcategory inclusions $\mathsf{Mod}^{\underline{\mathrm{OSA}}}(S,F,\leq) \subseteq \mathsf{Mod}^{\underline{\mathrm{MSA}}}(S,F)$, indexed by order-sorted signatures $(S,\leq,F)$,

forms a morphism of institutions from $\underline{\mathrm{OSA}}$ to $\underline{\mathrm{MSA}}$.

Based on the compositionality of institution morphisms [30], the following result in combination with Fact 6.1 will allow us to analyse the connections between the possible objects obtained by instantiating multiple parameters even for structured institutions such as $\underline{\mathrm{OSA}}^{spec}$, whose base institution does not have all the required properties for the analysis, but in turn it is pre-structured over another institution that is sufficiently simple, in this case $\underline{\mathrm{MSA}}$.

Fact 7.1. If an institution $\mathcal{I}''$ is pre-structured over $\mathcal{I}'$ through a functor Φ' and $\mathcal{I}'$ is pre-structured over $\mathcal{I}$ through Φ, then $\mathcal{I}''$ is pre-structured over $\mathcal{I}$ through $\Phi';\Phi$.

Assumption 7.1. *For the rest of this section let us assume that $\mathcal{I}$ is an institution whose category of signatures satisfies the properties required for the local study of parameterisation developed in § 5: (a) it is equipped with a distributive quasi-inclusion system $\mathbb{I}$, (b) all intersection-union squares are pushout squares, and (c) all retracts having quasi-inclusions as sections admit free extensions that strongly preserve given signatures preserved by the original morphisms. In addition, we also assume that $\mathcal{I}'$ is a pre-structured institution over $\mathcal{I}$ through a signature functor $\Phi\colon \mathbb{S}\mathrm{ig}^{\mathcal{I}'} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$ that is faithful and strongly lifts finite cocones.*

We recall from [10] the notion of parameterised signature, and adapt it to the more general context of quasi-inclusions and pre-structured institutions.

Definition 7.4 (Parameterised signature)**.** A *parameterised signature with respect to* Φ or *$\mathcal{I}'$-parameterised signature*, denoted $\Sigma'(\iota)$, consists of a signature morphism $\iota\colon P \to \Sigma'$ of $\mathbb{S}\mathrm{ig}^{\mathcal{I}'}$ such that $\Phi(\iota)$ is the quasi-inclusion $\Phi(P) \sqsubseteq \Phi(\Sigma')$. Similarly to the case of multiple-parameterised objects, we can consider *multiple-parameterised $\mathcal{I}'$-signatures*, denoted $\Sigma'(\iota_i\colon P_i \to \Sigma' \mid i \in [n])$.

The following elementary properties of parameterised signatures are immediate.

Fact 7.2. An $\mathcal{I}'$-signature $\Sigma'(\iota\colon P \to \Sigma')$ is parameterised with respect to Φ if and only if $\Phi(\Sigma')(\Phi(P))$ is a parameterised object in $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$.

Fact 7.3. If the signature functor $\Phi\colon \mathbb{S}\mathrm{ig}^{\mathcal{I}'} \to \mathbb{S}\mathrm{ig}^{\mathcal{I}}$ is faithful and it strongly lifts finite cocones then the parameterised signatures with respect to Φ are precisely the parameterised objects of $\mathbb{S}\mathrm{ig}^{\mathcal{I}'}$ relative to the quasi-inclusion system $\Phi^{-1}(\mathbb{I})$. For this reason, in such situations we also denote the single-parameterised $\mathcal{I}'$-signatures $\Sigma'(\iota\colon P \to \Sigma')$ simply by $\Sigma'(P)$, and the multiple-parameterised $\mathcal{I}'$-signatures $\Sigma'(\iota_i\colon P_i \to \Sigma' \mid i \in [n])$ by $\Sigma'(P_i \mid i \in [n])$.

The definition of the instantiation of parameters that we consider here can be regarded as a reflection through the signature functor of the corresponding notion for the case of parameterised objects discussed in § 5. Let us note this approach differs significantly from the one employed in [10], which assumed different properties of the structuring functor, such as lifting of coproducts and the existence of a left adjoint.

Definition 7.5 (Parameter instantiation)**.** Given any parameterised $\mathcal{I}'$-signature $\Sigma'(P)$ and morphism $\nu\colon P \to P'$ that preserves P' with respect to Φ, the *instance of* $\Sigma'(P)$ *by* ν, denoted $\Sigma'(P \Leftarrow_\Phi \nu)$, is defined by the following pushout square.

$$\begin{array}{ccc} P \sqcup P' & \xrightarrow{\sqsubseteq} & \Sigma' \sqcup P' \\ {\scriptstyle \nu \vee_\Phi 1_{P'}}\downarrow & & \downarrow{\scriptstyle \vartheta} \\ P' & \xrightarrow[i]{} & \Sigma'(P \Leftarrow_\Phi \nu) \end{array}$$

The *instance of the parameterised $\mathcal{I}'$-signature $\Sigma'(P)$ by ν based on free extensions* can be obtained by restricting the pushout squares considered above to those squares whose images through Φ describe free extensions, i.e. by taking into account only the pushout squares for which $\Phi(\vartheta)$ is a free extension of $\Phi(\nu) \vee 1_{\Phi(P')}$ and i is a quasi-inclusion.

Fact 7.4. For any parameterised $\mathcal{I}'$-signature $\Sigma'(P)$, and morphism $\nu\colon P \to P'$ that preserves P' with respect to Φ, the instance $\Sigma'(P \Leftarrow_\Phi \nu)$ of the parameterised $\mathcal{I}'$-signature $\Sigma'(P)$ by ν can be obtained through a Φ-strong lifting of the pushout used in the instantiation $\Phi(\Sigma')(\Phi(P) \Leftarrow \Phi(\nu))$ of the parameterised object $\Phi(\Sigma')(\Phi(P))$ by the fitting argument morphism $\Phi(\nu)$.

$$\begin{array}{ccc} \Phi(P) \sqcup \Phi(P') & \xrightarrow{\sqsubseteq} & \Phi(\Sigma') \sqcup \Phi(P') \\ \downarrow{\scriptstyle \Phi(\nu) \vee 1_{\Phi(P')}} & & \downarrow{\scriptstyle \Phi(\vartheta)} \\ \Phi(P') & \xrightarrow[\Phi(i)]{} & \Phi(\Sigma')(\Phi(P) \Leftarrow \Phi(\nu)) \end{array}$$

Example 7.4. By making use of parameterisation techniques, one can develop constructions with a high degree of modularity in specification languages that feature simple yet sufficiently expressive structuring operators. This intuition will become more clear after the introduction of instantiation procedures for multiple-parameterised signatures. We can still see that even in its monadic form, parameterisation contributes significantly to structuring specifications. To illustrate this, let us further extend the examination of generic lists.

Based on the parameterised many-sorted signature of generic lists presented in Example 5.1 we can easily construct a parameterised MSA-specification of lists, LIST(ELT), by following the definitions below.

$$\begin{aligned} \mathtt{ELT} &= ((S_{\mathtt{ELT}}, F_{\mathtt{ELT}}), \emptyset) \\ \mathtt{LIST(ELT)} &= ((S_{\mathtt{LIST}}, F_{\mathtt{LIST}}), \emptyset)\ !\ \mathtt{ELT} \end{aligned}$$

The construction employs a simplified representation of the free specification $((S_{\mathtt{LIST}}, F_{\mathtt{LIST}}), \emptyset)\ !_{\mathcal{H}}\ (\iota, \mathtt{ELT})$, adapted to the case in which $\mathcal{H}$ is the family of classes of identities and ι is the inclusion $(S_{\mathtt{ELT}}, F_{\mathtt{ELT}}) \subseteq (S_{\mathtt{LIST}}, F_{\mathtt{LIST}})$.

Let us observe that LIST(ELT) can be regarded as the object map of a functor *List*: $\mathbb{S}\mathrm{et} \to \mathbb{S}\mathrm{et}$ that associates to each set Elt the set List(Elt) of all finite sequences of elements from Elt, and to each function op: Elt → Elt′ the function map(op): List(Elt) → List(Elt′) corresponding to the element-wise application of op. In order to capture the behaviour of *List* on arrows we first need to consider the signatures $(S_{\mathtt{MAP}}, F_{\mathtt{MAP}})$ and $(S_{\mathtt{LIST\text{–}MAP}}, F_{\mathtt{LIST\text{–}MAP}})$, together with the set of $(S_{\mathtt{LIST\text{–}MAP}}, F_{\mathtt{LIST\text{–}MAP}})$-equations $E_{\mathtt{LIST\text{–}MAP}}$.

$$\begin{aligned} S_{\mathtt{ELT}'} &= \{\mathtt{Elt}'\} & S_{\mathtt{LIST\text{–}MAP}} &= \{\mathtt{Elt}, \mathtt{List}, \mathtt{Elt}', \mathtt{List}'\} \\ F_{\mathtt{ELT}'} &= \emptyset & F_{\mathtt{LIST\text{–}MAP}} &= \{\mathtt{nil}\colon [] \to \mathtt{List}, \_\_\colon \mathtt{Elt}\ \mathtt{List} \to \mathtt{List}, \\ & & & \quad \mathtt{nil}\colon [] \to \mathtt{List}', \_\_\colon \mathtt{Elt}'\ \mathtt{List}' \to \mathtt{List}', \\ & & & \quad \mathtt{op}\colon \mathtt{Elt} \to \mathtt{Elt}', \mathtt{map}\colon \mathtt{List} \to \mathtt{List}'\} \\ S_{\mathtt{MAP}} &= \{\mathtt{Elt}, \mathtt{Elt}'\} & E_{\mathtt{LIST\text{–}MAP}} &= \{(\forall \emptyset)\ \mathtt{map(nil)} = \mathtt{nil}, \\ F_{\mathtt{MAP}} &= \{\mathtt{op}\colon \mathtt{Elt} \to \mathtt{Elt}'\} & & \quad (\forall \{E\colon \mathtt{Elt}, L\colon \mathtt{List}\})\ \mathtt{map}(E\ L) = \mathtt{op}(E)\ \mathtt{map}(L)\} \end{aligned}$$

The arrow component of *List* can be specified through the parameterised specification LIST–MAP(MAP) that is explained in the next few lines (with explicit fitting argument morphisms and translations of specifications). Let us note that the union of specifications over different signatures is allowed by implicitly translating them to union signature.

$$\begin{aligned} \mathtt{ELT}' &= ((S_{\mathtt{ELT}'}, F_{\mathtt{ELT}'}), \emptyset) \\ \mathtt{MAP} &= ((S_{\mathtt{MAP}}, F_{\mathtt{MAP}}), \emptyset) \\ \mathtt{LIST\text{–}MAP(MAP)} &= \mathtt{LIST}(\mathtt{ELT} \Leftarrow_{\mathsf{Sig}} \{\mathtt{Elt} \mapsto \mathtt{Elt}\}\colon \mathtt{ELT} \to \mathtt{MAP}) \star \{\mathtt{List} \mapsto \mathtt{List}\} \cup \\ & \quad \mathtt{LIST}(\mathtt{ELT} \Leftarrow_{\mathsf{Sig}} \{\mathtt{Elt} \mapsto \mathtt{Elt}'\}\colon \mathtt{ELT} \to \mathtt{MAP}) \star \{\mathtt{List} \mapsto \mathtt{List}'\} \cup \\ & \quad ((S_{\mathtt{LIST\text{–}MAP}}, F_{\mathtt{LIST\text{–}MAP}}), E_{\mathtt{LIST\text{–}MAP}}) \end{aligned}$$

We can now easily obtain specifications of various transformations on lists that are compatible with the list structure just by specifying their action on elements. For example, the operation that filters a list of natural numbers by marking with true all its elements that satisfy a given predicate P, may be specified as follows. We first consider a suitable signature for P, namely $(S_{\mathtt{P}}, F_{\mathtt{P}}) = (\{\mathtt{Nat}, \mathtt{Bool}\}, \{\mathtt{P}\colon \mathtt{Nat} \to \mathtt{Bool}\})$. Then we can introduce a parameterised structured

specification for $\mathtt{P}$.

$$\begin{aligned}
\mathtt{NAT} &= ((S_{\mathtt{NAT}}, F_{\mathtt{NAT}}), \emptyset) \,!\, \emptyset \\
\mathtt{BOOL} &= ((S_{\mathtt{BOOL}}, F_{\mathtt{BOOL}}), \emptyset) \,!\, \emptyset \\
\mathtt{P(ELT)} &= \mathtt{ELT} \cup \mathtt{BOOL} \cup ((S_{\mathtt{MAP}}, F_{\mathtt{MAP}}), \emptyset) \star \{\mathtt{Elt}' \mapsto \mathtt{Bool}, \mathtt{op} \mapsto \mathtt{P}\} \\
\mathtt{NAT\text{–}P}((S_{\mathtt{P}}, F_{\mathtt{P}}), \emptyset) &= \mathtt{P(ELT} \Leftarrow_{\mathsf{Sig}} \{\mathtt{Elt} \mapsto \mathtt{Nat}\} \colon \mathtt{ELT} \to \mathtt{NAT})
\end{aligned}$$

Finally, we can proceed with the instantiation of $\mathtt{LIST\text{–}MAP}$ and define $\mathtt{LIST\text{–}MAP}_{\mathtt{NAT\text{–}P}}$ as $\mathtt{LIST\text{–}MAP(MAP} \Leftarrow_{\mathsf{Sig}} v)$, where $v \colon \mathtt{MAP} \to \mathtt{NAT\text{–}P}((S_{\mathtt{P}}, F_{\mathtt{P}}), \emptyset)$ is the specification morphism given by $v^{st}(\mathtt{Elt}) = \mathtt{Nat}$, $v^{st}(\mathtt{Elt}') = \mathtt{Bool}$ and $v^{op}(\mathtt{op}) = \mathtt{P}$. Note that by construction, the resulting specification $\mathtt{LIST\text{–}MAP}_{\mathtt{NAT\text{–}P}}$ inherits the parameter $((S_{\mathtt{P}}, F_{\mathtt{P}}), \emptyset)$ of $\mathtt{NAT\text{–}P}$, and thus we can indeed regard it as a specification of generic filters on finite lists of natural numbers. The diagram below depicts the instantiation process.

$$\begin{array}{c}
\overset{v}{\frown} \\
\mathtt{LIST\text{–}MAP(MAP)} \qquad\qquad \mathtt{NAT\text{–}P}((S_{\mathtt{P}}, F_{\mathtt{P}}), \emptyset) \\
\hline
\mathtt{LIST\text{–}MAP(MAP} \Leftarrow_{\mathsf{Sig}} v)((S_{\mathtt{P}}, F_{\mathtt{P}}), \emptyset)
\end{array}$$

The above example illustrates how parameterisation can be used in building libraries of formal specifications with a high degree of reusability. At the same time, it shows that special care is needed when designing such libraries, as one may easily obtain (by instantiation) inconsistent specifications. For example, given the specification $\mathtt{LIST\text{–}MAP(MAP} \Leftarrow_{\mathsf{Sig}} v)((S_{\mathtt{P}}, F_{\mathtt{P}}), \emptyset)$, any instance of the parameter $((S_{\mathtt{P}}, F_{\mathtt{P}}), \emptyset)$ such that the (translation of the) sort $\mathtt{Nat}$ cannot be interpreted as the set of natural numbers will yield an inconsistent instance of the parameterised specification. This is a natural consequence of the fact that the body of the parameterised specification defines new constraints on the sort $\mathtt{Nat}$ introduced by the parameter, and it could be prevented, for instance, by considering only specifications with conservative parameters, i.e. with parameters whose models can always be expanded to models of the parameterised specification.

The following result plays the role of Proposition 5.2 in supporting the definition of simultaneous instantiation of parameters.

Proposition 7.1. *If $\{v_i \colon P_i \to P'_i \mid i \in [n]\}$ is a set of pairwise compatible $\mathcal{I}'$-signature morphisms with respect to Φ, there exists a unique $\mathcal{I}'$-signature morphism $\bigvee_{\Phi i \in [n]} v_i \colon \bigsqcup_{i\in[n]} P_i \to \bigsqcup_{i\in[n]} P'_i$ satisfying*

$$\Big(P_j \sqsubseteq \bigsqcup_{i\in[n]} P_i\Big); \bigvee_{i\in[n]}{}_{\Phi} v_i = v_j; \Big(P'_j \sqsubseteq \bigsqcup_{i\in[n]} P'_i\Big), \quad \textit{for any } j \in [n].$$

In addition, for any set of $\mathcal{I}'$-signatures $\{Q_j \mid j \in [m]\}$, if v_i preserves Q_j with respect to Φ, for any $i \in [n]$ and $j \in [m]$, then $\bigvee_{\Phi i\in[n]} v_i$ preserves $\bigsqcup_{j\in[m]} Q_j$ with respect to Φ.

Proof. Let $\{v_i \colon P_i \to P'_i \mid i \in [n]\}$ be a (non-empty) set of pairwise compatible $\mathcal{I}'$-signature morphisms with respect to Φ. It follows that $\{\Phi(v_i) \colon \Phi(P_i) \to \Phi(P'_i) \mid i \in [n]\}$ is a set of pairwise compatible $\mathcal{I}$-signature morphisms.

The existence of $\bigvee_{\Phi i\in[n]} v_i$ is immediate. Based on the fact that the union $\bigsqcup_{i\in[n]} P_i$ is given by the Φ-strong lifting of $\bigsqcup_{i\in[n]} \Phi(P_i)$, the morphism $\bigvee_{\Phi i\in[n]} v_i$ can be defined as the lifting of the join $\bigvee_{i\in[n]} \Phi(v_i)$, about which we know from Proposition 5.2 that it satisfies

$$\Phi\Big(\Big(P_j \sqsubseteq \bigsqcup_{i\in[n]} P_i\Big); \bigvee_{i\in[n]}{}_{\Phi} v_i\Big) = \Big(\Phi(P_j) \sqsubseteq \bigsqcup_{i\in[n]} \Phi(P_i)\Big); \bigvee_{i\in[n]} \Phi(v_i) = \Phi(v_j); \Big(\Phi(P'_j) \sqsubseteq \bigsqcup_{i\in[n]} \Phi(P'_i)\Big) = \Phi\Big(v_j; \Big(P'_j \sqsubseteq \bigsqcup_{i\in[n]} P'_i\Big)\Big)$$

for all $j \in [n]$. Therefore, since Φ is faithful, it follows that $\bigvee_{\Phi i\in[n]} v_i$ verifies the equality $(P_j \sqsubseteq \bigsqcup_{i\in[n]} P_i); \bigvee_{\Phi i\in[n]} v_i = v_j; (P'_j \sqsubseteq \bigsqcup_{i\in[n]} P'_i)$, for any $j \in [n]$. In addition, if we assumed $u \colon \bigsqcup_{i\in[n]} P_i \to \bigsqcup_{i\in[n]} P'_i$ to be another morphism such that $(P_j \sqsubseteq \bigsqcup_{i\in[n]} P_i); u = v_j; (P'_j \sqsubseteq \bigsqcup_{i\in[n]} P'_i)$ for all $j \in [n]$, then by applying Φ we would obtain the equalities $(\Phi(P_j) \sqsubseteq \bigsqcup_{i\in[n]} \Phi(P_i)); \Phi(u) = \Phi(v_j); (\Phi(P'_j) \sqsubseteq \bigsqcup_{i\in[n]} \Phi(P'_i))$, for $j \in [n]$. Since $\bigvee_{i\in[n]} \Phi(v_i)$ is the unique morphism satisfying these equalities, we deduce that $\Phi(u) = \bigvee_{i\in[n]} \Phi(v_i)$, which implies $u = \bigvee_{\Phi i\in[n]} v_i$ because Φ is faithful.

For the second part of the proposition, let $\{Q_j \mid j \in [m]\}$ be a set of $\mathcal{I}'$-signatures such that, for all $i \in [n]$ and $j \in [m]$, v_i preserves Q_j with respect to Φ. We deduce that for all $i \in [n]$ and $j \in [m]$, $\Phi(v_i)$ preserves $\Phi(Q_j)$. Therefore, by Proposition 5.2, the join $\bigvee_{i\in[n]} \Phi(v_i)$ preserves the union $\bigsqcup_{j\in[m]} \Phi(Q_j)$. Since $\Phi(\bigvee_{\Phi i\in[n]} v_i) = \bigvee_{i\in[n]} \Phi(v_i)$ and $\Phi(\bigsqcup_{j\in[m]} Q_j) = \bigsqcup_{j\in[m]} \Phi(Q_j)$, we reach the conclusion that $\bigvee_{\Phi i\in[n]} v_i$ preserves $\bigsqcup_{j\in[m]} Q_j$ with respect to Φ. □

Definition 7.6 (Simultaneous instantiation of parameters)**.** Let $\Sigma'(P_i \mid i \in [n])$ be a multiple-parameterised $\mathcal{I}'$-signature and $\{v_i\colon P_i \to P'_i \mid i \in [n]\}$ a set of pairwise compatible morphisms with respect to Φ such that any morphism v_i preserves any object P'_j with respect to Φ, for $i, j \in [n]$. The *simultaneous instantiation* of $\Sigma'(P_i \mid i \in [n])$ by $\{v_i\colon P_i \to P'_i \mid i \in [n]\}$, denoted $\Sigma'(P_i \Leftarrow_\Phi v_i \mid i \in [n])$, is defined as the (single) $\mathcal{I}'$-parameter instantiation $\Sigma'(\bigsqcup_{i\in[n]} P_i \Leftarrow_\Phi \bigvee_{\Phi i\in[n]} v_i)$, and is said to be *based on free extensions* if $\Sigma'(\bigsqcup_{i\in[n]} P_i \Leftarrow_\Phi \bigvee_{\Phi i\in[n]} v_i)$ has this property.

The sequential instantiation of parameters can be extended from the signatures of the base institution in a similar way. Its definition relies on the following straightforward property.

Fact 7.5. Let $\Sigma'(P_i \mid i \in [n])$ be a multiple-parameterised $\mathcal{I}'$-signature and $v_i\colon P_i \to P'_i$ a morphism that preserves P'_i and all P_j with respect to Φ, for $j \in [n] \setminus \{i\}$. If the $\mathcal{I}'$-instantiation of parameters is based on free extensions then $\Sigma'(P_i \Leftarrow_\Phi v_i)$ is a parameterised $\mathcal{I}'$-signature, with the parameters $\{P_j \mid j \in [n] \setminus \{i\}\}$.

Definition 7.7 (Sequential instantiation of parameters)**.** Let us consider a parameterised $\mathcal{I}'$-signature $\Sigma'(P_i \mid i \in [n])$ and a set of pairwise Φ-compatible $\mathcal{I}'$-signature morphisms $\{v_i\colon P_i \to P'_i \mid i \in [n]\}$ such that for every $i \in [n]$, v_i preserves P'_i and all P_j with respect to Φ, where $j \in [n] \setminus [i+1]$. The *sequential instantiation* of $\Sigma'(P_i \mid i \in [n])$ by $\{v_i\colon P_i \to P'_i \mid i \in [n]\}$, denoted $\Sigma'(P_i \Leftarrow_\Phi v_i)_{i\in[n]}$, is defined as the iterated (single) parameter $\mathcal{I}'$-instantiation $\Sigma(P_0 \Leftarrow_\Phi v_0)\cdots(P_{n-1} \Leftarrow_\Phi v_{n-1})$. In addition, the sequential instantiation $\Sigma'(P_i \Leftarrow_\Phi v_i)_{i\in[n]}$ is *strong* when any partial instance of $\Sigma'(P_i \mid i \in [n])$ is obtained through the Φ-strong lifting of a free extension that strongly preserves the underlying $\mathcal{I}$-signatures of the instances of the remaining parameters.

The possibility to instantiate only some of the parameters of a multiple-parameterised specification, and obtain in this way new parameterised specifications that preserve the original uninstantiated parameters, promotes even more the advantages of modularisation.

Example 7.5. We can reorganize the parameters of the specification LIST–MAP discussed in Example 7.4 to allow more interactions with other specifications, and hence further increase its expressive power.

$$\begin{aligned}\texttt{LIST–MAP}(\texttt{ELT}, \texttt{MAP}, \texttt{ELT}') = {} & \texttt{LIST}(\texttt{ELT} \Leftarrow_{\mathsf{Sig}} \{\texttt{Elt} \mapsto \texttt{Elt}\}\colon \texttt{ELT} \to \texttt{ELT}) \star \{\texttt{List} \mapsto \texttt{List}\} \cup \\ & \texttt{LIST}(\texttt{ELT} \Leftarrow_{\mathsf{Sig}} \{\texttt{Elt} \mapsto \texttt{Elt}'\}\colon \texttt{ELT} \to \texttt{ELT}') \star \{\texttt{List} \mapsto \texttt{List}'\} \cup \\ & ((S_{\texttt{LIST–MAP}}, F_{\texttt{LIST–MAP}}), E_{\texttt{LIST–MAP}})\end{aligned}$$

Let us notice that sharing occurs between the parameters ELT and MAP, as well as between MAP and ELT′. For this reason, the three parameters may be instantiated in three possible ways: all three simultaneously, the first two in parallel without changing the sort Elt′, followed by the third one, or the last two in parallel without changing the sort Elt, followed by the first one.

The next lines elaborate on the last instantiation scheme and illustrate an alternative course for obtaining the specification of filters on finite lists of natural numbers. We consider the specification morphisms

$$v_{\texttt{P}}\colon \texttt{MAP} \to \texttt{P}(\texttt{ELT}) \quad \text{and} \quad v_{\texttt{BOOL}}\colon \texttt{ELT}' \to \texttt{BOOL}$$

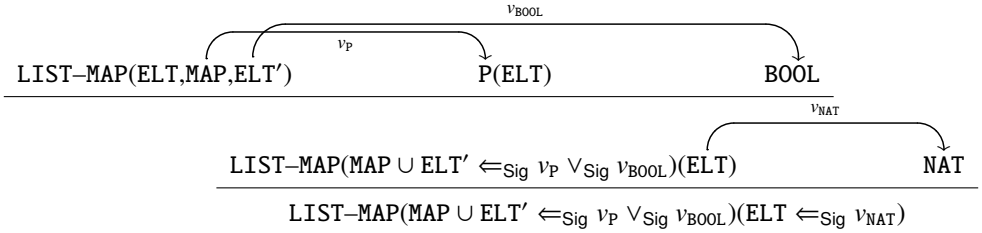
given by $v^{st}_{\texttt{P}}(\texttt{Elt}) = \texttt{Elt}$, $v^{st}_{\texttt{P}}(\texttt{Elt}') = \texttt{Bool}$, $v^{op}_{\texttt{P}}(\texttt{op}) = \texttt{P}$, and $v^{st}_{\texttt{BOOL}}(\texttt{Elt}') = \texttt{Bool}$. The parameters MAP and ELT′ can be instantiated simultaneously through $v_{\texttt{P}}$ and $v_{\texttt{BOOL}}$, thus yielding the parameterised specification

$$\texttt{LIST–MAP}(\texttt{MAP} \cup \texttt{ELT}' \Leftarrow_{\mathsf{Sig}} v_{\texttt{P}} \vee_{\mathsf{Sig}} v_{\texttt{BOOL}})(\texttt{ELT}).$$

We may now continue with the instantiation of ELT through the fitting argument morphism $v_{\texttt{NAT}}\colon \texttt{ELT} \to \texttt{NAT}$ that maps the sort Elt to Nat. In this manner we obtain the final result of the instantiation.

$$\texttt{LIST–MAP}(\texttt{MAP} \cup \texttt{ELT}' \Leftarrow_{\mathsf{Sig}} v_{\texttt{P}} \vee_{\mathsf{Sig}} v_{\texttt{BOOL}})(\texttt{ELT} \Leftarrow_{\mathsf{Sig}} v_{\texttt{NAT}})$$

The complete instantiation process is outlined in the diagram below.



All the notions and results advanced thus far can be seen as a systematic development of a methodology aimed at increasing the expressive power of specification languages through parameterisation. The subsequent fundamental corollary of our work clarifies a set of sufficient conditions the underlying logical system should satisfy and also acceptable choices that can be made in instantiating the parameters such that all the possible instantiation schemes produce the same class of isomorphic results.

Theorem 7.1. *Let $\Sigma'(P_i \mid i \in [n])$ be a multiple-parameterised $\mathcal{I}'$-signature and $\{\nu_i \colon P_i \to P'_i \mid i \in [n]\}$ a set of pairwise Φ-compatible $\mathcal{I}'$-signature morphisms such that for every $i \in [n]$, ν_i preserves P_j and P'_k with respect to Φ, for all $j \in [n] \setminus \{i\}$ and $k \in [n]$. If the simultaneous $\mathcal{I}'$-instantiation of parameters is based on free extensions and the sequential $\mathcal{I}'$-instantiation of parameters is strong then the two instantiation procedures produce isomorphic results.*

$$\Sigma'\Big(\bigsqcup_{i\in[n]} P_i \Leftarrow_\Phi \bigvee_{i\in[n]}{}_\Phi\, \nu_i\Big) \quad \cong \quad \Sigma'(P_i \Leftarrow_\Phi \nu_i)_{i\in[n]}$$

Proof. By translating the above prerequisites along Φ we obtain the parameterised object $\Phi(\Sigma')(\Phi(P_i) \mid i \in [n])$ in $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$ and the set of morphisms $\{\Phi(\nu_i) \colon \Phi(P_i) \to \Phi(P'_i) \mid i \in [n]\}$ with the property that $\Phi(\nu_i)$ preserves $\Phi(P_j)$ and $\Phi(P'_k)$, for all $j \in [n] \setminus \{i\}$ and $k \in [n]$. Given that the simultaneous $\mathcal{I}'$-instantiation is based on free extensions and that the sequential $\mathcal{I}'$-instantiation is strong, we deduce that the simultaneous and the sequential $\mathcal{I}$-instantiations of $\Phi(\Sigma')(\Phi(P_i) \mid i \in [n])$ by $\{\Phi(\nu_i) \colon \Phi(P_i) \to \Phi(P'_i) \mid i \in [n]\}$ are also based on free extensions and strong, respectively. Therefore, by Corollary 5.1, there exists an isomorphism between the results $\Phi(\Sigma')(\bigsqcup_{i\in[n]} \Phi(P_i) \Leftarrow \bigvee_{i\in[n]} \Phi(\nu_i))$ and $\Phi(\Sigma')(\Phi(P_i) \Leftarrow \Phi(\nu_i))_{i\in[n]}$ of the simultaneous and the sequential instantiation of $\Phi(\Sigma')(\Phi(P_i) \mid i \in [n])$ by $\{\Phi(\nu_i) \colon \Phi(P_i) \to \Phi(P'_i) \mid i \in [n]\}$. Moreover, by Theorem 5.1, we know that $\Phi(\Sigma')(\bigsqcup_{i\in[n]} \Phi(P_i) \Leftarrow \bigvee_{i\in[n]} \Phi(\nu_i))$ and $\Phi(\Sigma')(\Phi(P_i) \Leftarrow \Phi(\nu_i))_{i\in[n]}$ are vertices of two isomorphic cocones over the same finite diagram. Since by hypothesis Φ is a faithful functor that strongly lifts cocones, we also know by Fact 7.4 that $\Sigma'(\bigsqcup_{i\in[n]} P_i \Leftarrow_\Phi \bigvee_{\Phi i\in[n]} \nu_i)$ and $\Sigma'(P_i \Leftarrow_\Phi \nu_i)_{i\in[n]}$ are obtained through Φ-strong liftings of the cocones corresponding to the two parameter instantiations formed in $\mathbb{S}\mathrm{ig}^{\mathcal{I}}$. This allows us to conclude by Corollary 6.2 that the results of the simultaneous and the sequential $\mathcal{I}'$-instantiations presented above are isomorphic. □

8. Conclusions

In the present paper we extended the theory of pushout-style parameterisation in two main directions. First, by imposing minimum restrictions on the instantiation of parameters, thus allowing both sharing between various parameters, and between the body of the parameterised specification and the instances of the parameters. Second, by developing all the concepts and results within the high-level framework of abstract structured specifications, such that they are independent of both the underlying logical system and the concrete structuring operators.

Our efforts concentrated on the examination of multiple-parameterised specifications and their possible instantiation scenarios. Given a base institution $\mathcal{I}$, we showed that a pre-structured institution $\mathcal{I}'$ over $\mathcal{I}$ through a structuring functor Φ enjoys good properties with respect to parameterisation when

1. the category of $\mathcal{I}$-signatures is equipped with a distributive quasi-inclusion system such that:
   - any intersection-union square is also a pushout square, and

– any retract whose section is a quasi-inclusion admits free extensions that strongly preserve any fixed signature preserved by the retract;

2. the structuring functor Φ is faithful and it strongly lifts cocones.

We discussed two main instantiation procedures, distinct not only by their inner workings but also by the situations in which they can be employed. To this regard, simultaneous instantiation only requires that the shared part of two or more parameters is instantiated in the same way, while sequential instantiation requires that the instantiation of a parameter does not change any of the remaining parameters. When both simultaneous and sequential instantiation procedures can be applied, the results of all possible instantiations of a parameterised specification prove to be isomorphic, assuming that a specified set of sufficient conditions hold. We investigated these conditions for a number of base logical systems and structuring formalisms and showed that they are smoothly satisfied.

Acknowledgements

The author is grateful to Răzvan Diaconescu and José Fiadeiro for their continuous encouragement, support and helpful discussions throughout the course of this work. Acknowledgements are also due to the two anonymous referees for their careful study of the original manuscript and for their detailed technical comments, which have helped to significantly improve the paper. This research 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.

Bibliography

[1] Jiri Adámek, Horst Herrlich, and George Strecker. *Abstract and Concrete Categories: The Joy of Cats*. Dover books on mathematics. Dover Publications, 2009. reprint.

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

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

[4] Rod M. Burstall and Joseph A. Goguen. Putting theories together to make specifications. In Raj Reddy, editor, *International Joint Conference on Artificial Intelligence*, pages 1045–1058. William Kaufmann, 1977.

[5] Rod M. Burstall and Joseph A. Goguen. The semantics of clear, a specification language. In Dines Bjørner, editor, *Abstract Software Specifications*, volume 86 of *Lecture Notes in Computer Science*, pages 292–332. Springer, 1979.

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

[7] Virgil E. Căzănescu and Grigore Roşu. Weak inclusion systems: Part two. *Journal of Universal Computer Science*, 6(1):5–21, 2000.

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

[9] Răzvan Diaconescu. *Institution-independent model theory*. Studies in Universal Logic. Birkhäuser, 2008.

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

[11] Răzvan Diaconescu and Kokichi Futatsugi. *CafeOBJ report: the language, proof techniques, and methodologies for object-oriented algebraic specification*. AMAST series in computing. World Scientific, 1998.

[12] Răzvan Diaconescu, Joseph A. Goguen, and Petros Stefaneas. Logical support for modularisation. In Gérard Huet and Gordon Plotkin, editors, *Logical Environments*, pages 83–130. Cambridge University Press, 1993.

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

[14] José L. Fiadeiro. *Categories for Software Engineering*. Springer, 2004.

[15] José L. Fiadeiro and Antónia Lopes. Algebraic semantics of coordination or what is in a signature. In Armando Martin Haeberer, editor, *Algebraic Methodology and Software Technology*, volume 1548 of *Lecture Notes in Computer Science*, pages 293–307. Springer, 1998.

[16] Joseph A. Goguen. Higher-order functions considered unnecessary for higher-order programming. In David A. Turner, editor, *Research topics in functional programming*, pages 309–351. Addison-Wesley, 1990.

[17] Joseph A. Goguen and Rod M. Burstall. Some fundamental algebraic tools for the semantics of computation. Part 2: Signed and abstract theories. *Theoretical Computer Science*, 31:263–295, 1984.

[18] Joseph A. Goguen and Rod M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.

[19] Joseph A. Goguen and Răzvan Diaconescu. An Oxford survey of order sorted algebra. *Mathematical Structures in Computer Science*, 4(3):363–392, 1994.

[20] Joseph A. Goguen and Grigore Roşu. Institution morphisms. *Formal Aspects of Computing*, 13(3–5):274–307, 2002.

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

[22] Joseph A. Goguen, Timothy Winkler, José Meseguer, Kokichi Futatsugi, and Jean-Pierre Jouannaud. Introducing OBJ. In Joseph A. Goguen and Grant Malcolm, editors, *Software engineering with OBJ: algebraic specification in action*, Advances in formal methods. Kluwer Academic, 2000.

[23] Horst Herrlich and George Strecker. *Category theory: an introduction*. Allyn and Bacon series in advanced mathematics. Allyn and Bacon, 1973.
[24] Saunders Mac Lane. *Categories for the working mathematician*. Graduate texts in mathematics. Springer, 1998.
[25] José Meseguer and Joseph A. Goguen. Order-sorted algebra solves the constructor-selector, multiple representation, and coercion problems. *Information and Computation*, 103(1):114–158, 1993.
[26] Peter Mosses. *CASL Reference Manual: The Complete Documentation Of The Common Algebraic Specification Language*. Lecture Notes in Computer Science. Springer, 2004.
[27] Grigore Roşu. Axiomatizability in inclusive equational logics. *Mathematical Structures in Computer Science*, 12(5):541–563, 2002.
[28] Donald Sannella and Andrzej Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76(2/3):165–210, 1988.
[29] Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2011.
[30] Andrzej Tarlecki. Moving between logical systems. In Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Specification of Abstract Data Types*, volume 1130 of *Lecture Notes in Computer Science*, pages 478–502. Springer, 1995.