

Parchments for CafeOBJ logics^{*}

Till Mossakowski¹, Wiesław Pawłowski², Donald Sannella³, and
Andrzej Tarlecki⁴

¹ Faculty of Computer Science, University of Magdeburg

² Institute of Informatics, University of Gdańsk

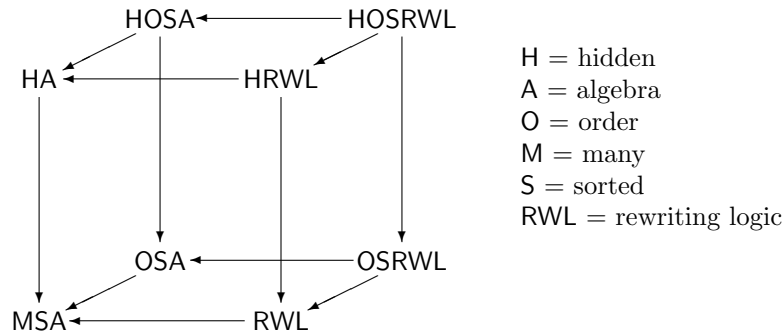
³ Laboratory for Foundations of Computer Science, University of Edinburgh

⁴ Institute of Informatics, University of Warsaw

Abstract. This paper addresses issues arising in the systematic construction of large logical systems. We rely on a model-theoretic view of logical systems, captured by *institutions* that are in turn presented by *parchments*. We define their categories, and study constructions that may be carried out in these categories. In particular we show how limits of parchments may be used to combine features involved in various logical systems, sometimes necessarily augmenting the universal construction by additional systematic adjustments. We illustrate these developments by sketching how the logical systems that form the logical foundations of CafeOBJ may be built in this manner.

1 Introduction

This paper is written as a tribute to Professor Kokichi Futatsugi, the leader of the algebraic specification community in Japan, whom we have had a chance to meet many times over the years. One of his major undertakings was the very successful CafeOBJ project [DF98], which led to the development of a system that implements and executes algebraic specifications, in the tradition of the OBJ family [GWM⁺00]. The system is based on solid logical foundations given by a family of logical systems linked by a number of logic morphisms, referred to as the *CafeOBJ cube* [DF02]:



^{*} This work has been partially supported by the Polish Ministry of Science and Higher Education, grant N206 493138 (AT).

The eight logical systems listed above and the twelve arrows that link them are formalised as, respectively, *institutions* and *institution morphisms* [GB92]. The institution diagram above may be viewed as an *indexed institution*; the actual logical system that underlies CafeOBJ is given as the *Grothendieck institution* [Dia02] built out of it. Even if we prefer to think of the cube above as a heterogeneous logical environment [MT09] and work with heterogeneous specifications, technically the differences are negligible and the main point is to understand properly the CafeOBJ cube of institutions and their morphisms.

As far as we are aware, while the CafeOBJ literature presents the institutions involved in a manner that is sufficient to understand and work with them well, there is no document that presents the institutions involved formally in complete detail; this applies even more to the institution morphisms that link them. In a way, this is rather expected, as the details appear to be quite obvious, largely routine and repetitive from one institution to another, and from one morphism to another. So, the CafeOBJ authors present the interesting aspects of the institutions, leaving out the details.¹

The main point of the present paper concerns the methodology of logic definitions. [DF02] defines the CafeOBJ cube in a top-down manner. Although in the literature, the concepts of order-sorting, rewriting logic and hidden algebra have been defined and studied (and integrated) separately, the technical presentation in [DF02] starts with a large combined institution, from which suitable subinstitutions are obtained subsequently. A drawback of this approach is the difficulty of changing individual feature components in a simple way. For example, [Dia07] claims that the base institution of equational logic could be replaced by membership equational logic, but to our knowledge, this has never been worked out. Indeed, working this out would imply a lot of tedious repetition of the original CafeOBJ definitions. However, even if the details seem routine and repetitive, one cannot just leave them out without a risk of unforeseen interactions between the modifications and the other features.

We therefore propose a bottom-up approach to the CafeOBJ cube. We present each of the features separately, and obtain the combined institution via a general universal construction. At each step of the combination, the details may be fine-tuned, if needed. This approach has the benefit of increased modularity: we can change certain feature components and then automatically re-generate the whole picture by repeating the universal constructions involved. In this paper, we concentrate on the methodology of this approach and therefore take the liberty of deviating from some details of the CafeOBJ institutions as defined in [DF02].

Like CafeOBJ, we follow Goguen and Burstall [GB92] and work within the theory of institutions as a formal framework to study and use logical systems. We will, however, look more closely at the structure of logical sentences and their semantics, and consider institutions to be presented by *parchments* [GB86]. We employ the version of parchments introduced in [MTP98] to avoid the technically

¹ We should stress though that this point may be due to our lack of complete knowledge of the CafeOBJ literature.

unnecessary and methodologically dubious blending of models into the syntactic aspects of logical systems.

We study various ways to extend, combine and modify these *model-theoretic parchments*, thus obtaining new logical systems and morphisms between them. We sketch how the logical systems in the CafeOBJ cube and morphisms between them may be obtained in such a way.

We start by recalling some standard algebraic notions (Sect. 2) and the basic concepts of the theory of institutions (Sect. 3). Then the less standard notions of model-theoretic parchment and parchment morphism are recalled in Sect. 4. The crucial property here is that when such parchments and their morphisms are *institutional*, they present institutions and institution morphisms, respectively. In Sect. 5 we discuss some simple ways to extend, combine and modify model-theoretic parchments and their morphisms, and in particular the use of limits in various parchment categories to combine institutions presented by parchments. We show how this works on some simple examples, sketching how the institutions and morphisms in the CafeOBJ cube may arise.

2 Algebraic preliminaries

We briefly recall the key concepts and notations used throughout this paper; we refer to [ST12] for details omitted here.

First-order signatures are triples $\Theta = \langle S, \Omega, \Pi \rangle$, consisting of a set S of sorts, set Ω of operation names classified by their profiles (we write $f: s_1 \times \dots \times s_n \rightarrow s$, $n \geq 0$, to indicate that f has the arity $s_1 \dots s_n \in S^*$ and result sort $s \in S$) and set Π of predicate names classified by their profiles (we write $p: s_1 \times \dots \times s_n$, $n \geq 0$, to indicate that the predicate p has arity $s_1 \dots s_n \in S^*$). *First-order signature morphisms* map sorts, operation and predicate names to sorts, operation and predicate names, respectively, preserving their arities and result sorts. This yields the category **FOSig**.

Given a first-order signature $\Theta = \langle S, \Omega, \Pi \rangle$, a Θ -*structure* A consists of an S -sorted carrier set $|A| = \langle |A|_s \rangle_{s \in S}$, for each operation name $f: s_1 \times \dots \times s_n \rightarrow s$, a function $f_A: |A|_{s_1} \times \dots \times |A|_{s_n} \rightarrow |A|_s$, and for each predicate name $p: s_1 \times \dots \times s_n$, a relation $p_A \subseteq |A|_{s_1} \times \dots \times |A|_{s_n}$. A Θ -*homomorphism* $h: A \rightarrow B$ between two such Θ -structures is a family of maps $h = \langle h_s: |A|_s \rightarrow |B|_s \rangle_{s \in S}$ that preserves results of operations and predicate relations; h is *closed* if it also reflects predicate relations. **Str**(Θ) is the category of Θ -structures and their (not necessarily closed) homomorphisms. For any first-order morphism $\theta: \Theta \rightarrow \Theta'$, we have the usual *reduct* functor **Str**(θ): **Str**(Θ') \rightarrow **Str**(Θ), often written as $-|_{\theta}$. This yields a functor **Str**: **FOSig**^{op} \rightarrow **Cat**.²

² **Cat** denotes the (quasi-)category of all categories. We will gloss over fine foundational distinctions between categories at various levels of the hierarchy of universes [Mac71], and use the same term *category* to refer to (quasi-)categories of all categories, of all institutions, etc.

For any signature morphism $\theta: \Theta \rightarrow \Theta'$, the θ -reduct has a left adjoint $F_\theta: \mathbf{Str}(\Theta) \rightarrow \mathbf{Str}(\Theta')$; for any $A \in |\mathbf{Str}(\Theta)|$, $F_\theta(A) \in |\mathbf{Str}(\Theta')|$ is its *free extension* with unit $\eta_\theta: A \rightarrow F_\theta(A)|_\theta$.

Logic denotes a special signature with $*$ as the only sort, no operations and a unique predicate $D: *$. \mathbf{FOSig}_* is the subcategory of \mathbf{FOSig} that has signatures that extend *Logic* and signature morphisms that are identities on *Logic*.

The category \mathbf{FOSig} is cocomplete, with the standard colimit construction. The functor \mathbf{Str} is continuous, which in particular implies that the amalgamation property holds. This carries over to \mathbf{FOSig}_* and the restriction of \mathbf{Str} to \mathbf{FOSig}_* .

For any signature Θ , Θ -terms and their evaluation in Θ -structures are defined as usual. In particular, the algebra T_Θ of terms with predicates interpreted as empty relations is initial in $\mathbf{Str}(\Theta)$. For any (ground) term $t \in |T_\Theta|$ and structure $A \in |\mathbf{Str}(\Theta)|$, we write $t_A \in |A|$ for the value of t in A (which is the value of the unique homomorphism $!_A: T_\Theta \rightarrow A$ on t).

Θ -equations and predicate applications, as well as their satisfaction in Θ -structures, are defined as usual.

Any signature morphism $\theta: \Theta \rightarrow \Theta'$ determines the obvious translation of Θ -terms to Θ' -terms, given by $\theta: T_\Theta \rightarrow T_{\Theta'}|_\theta$. This translation further extends to Θ -equations and predicate applications. Then for any term $t \in |T_\Theta|$, and Θ' -structure A' , the crucial property is that $\theta(t)_{A'} = t_{A'}|_\theta$. This yields the famous *satisfaction condition* for equations and predicate applications: given any Θ -equation or predicate application φ and structure $A' \in |\mathbf{Str}(\Theta')|$, $A' \models_{\Theta'} \theta(\varphi)$ iff $A'|_\theta \models_\Theta \varphi$.

3 Institutions

Goguen and Burstall [GB92] formalised the notion of a logical system as an *institution*, thus starting a line of important developments of adequately abstract and general approaches to the foundations of software specifications and formal system development (as envisaged by the work on CLEAR [BG80], and carried forward by [ST88], see [ST12]), as well as a modern and elegant version of very abstract model theory (as proposed in [Tar86], see [Dia08]). Another important line of work which exploits institutions and their various morphisms [GR02] aims at moving between logical systems within a heterogeneous logical environment, comparing logical systems, and building complex logical systems in a systematic manner. In our view, in spite of work on various aspects of this area [Tar96, MTP97, MTP98, Tar00, CMRS01, CGR03, Mos03, Mos05, MT09], there is much to add here. The current paper is a contribution to this field.

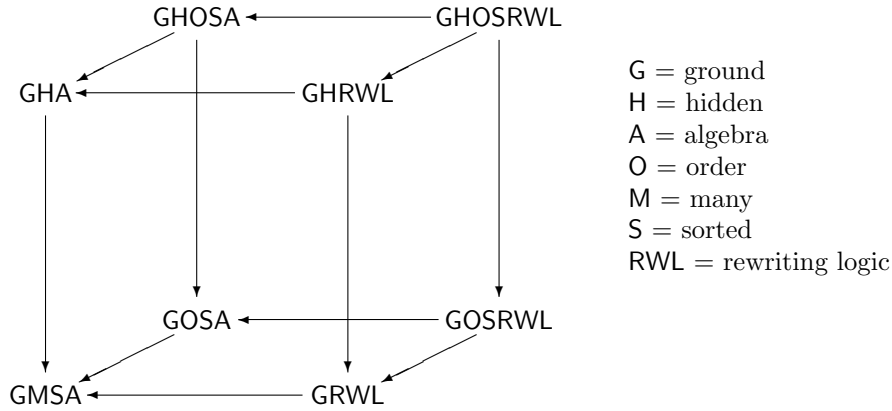
An *institution* $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ consists of:

- a category \mathbf{Sign} of *signatures*;
- a functor $\mathbf{Sen}: \mathbf{Sign} \rightarrow \mathbf{Set}$ which for any signature $\Sigma \in |\mathbf{Sign}|$ yields a set $\mathbf{Sen}(\Sigma)$ of *sentences*, and for any signature morphism $\sigma: \Sigma \rightarrow \Sigma'$, a σ -*translation of sentences*, often written as $\sigma: \mathbf{Sen}(\Sigma) \rightarrow \mathbf{Sen}(\Sigma')$;

- a functor $\mathbf{Mod} : \mathbf{Sign}^{op} \rightarrow \mathbf{Class}^3$ which for any signature Σ yields a class $\mathbf{Mod}(\Sigma)$ of *models*, and for any morphism $\sigma : \Sigma \rightarrow \Sigma'$, a σ -*reduct of models* often written as $_{|\sigma} : \mathbf{Mod}(\Sigma') \rightarrow \mathbf{Mod}(\Sigma)$; and
- a *satisfaction relation* $\models_{\Sigma} \subseteq \mathbf{Mod}(\Sigma) \times \mathbf{Sen}(\Sigma)$ for any signature $\Sigma \in |\mathbf{Sign}|$

such that the following *satisfaction condition* holds: for any signature morphism $\sigma : \Sigma \rightarrow \Sigma'$, sentence $\varphi \in \mathbf{Sen}(\Sigma)$ and model $M' \in \mathbf{Mod}(\Sigma')$, $M' \models_{\Sigma'} \sigma(\varphi)$ iff $M'_{|\sigma} \models_{\Sigma} \varphi$.

For simplicity of presentation, we will look at examples of logical systems drawn from the *CafeOBJ* cube in their *ground* versions, without variables:



Variables and (universal) quantification may be introduced in a rather standard way, see Sect. 6 for some hints. Moreover, we will simplify all of the logical systems involved by disregarding the fact that all statements in *CafeOBJ* may be conditional [DF02] — hence there are no conditional statements in the logics below. Adding conditions to the sentences of each of the logics considered is straightforward. Thus, we consider ground atomic sentences of *CafeOBJ* logics.

Furthermore, we will only attempt to capture the essential features of the logics in the *CafeOBJ* cube, rather than follow their published definitions. Consequently, the exact details of the logics presented below may depart from their *CafeOBJ* inspirations.

Finally, the logics of *CafeOBJ* seem to be set up incrementally, so that for instance strict equations, behavioural (hidden) equations and rewriting statements coexist rather than replacing one another [DF02]. So, the version of rewriting logic we consider covers equations (inherited from many-sorted equational logic)

³ **Class** is the quasi-category of all classes (or discrete categories). The standard definition of institution puts **Cat** here. There would be no problem in doing the same, and we realise that this is important for the semantics of *CafeOBJ* specifications. However, since model morphisms are orthogonal to the issues discussed in this paper, we decided to leave them out to simplify the technicalities and notation somewhat.

as well as rewriting statements. For presentation purposes, we will introduce another logical system, GPRWL, capturing ground rewriting statements only.⁴

Example 3.1. A trivial algebraic institution: $\mathbf{A} = \langle \mathbf{AlgSig}, \mathbf{Sen}^\emptyset, \mathbf{Alg}, \models^\emptyset \rangle$, with *algebraic signatures* (i.e., first-order signatures with no predicates, so that \mathbf{AlgSig} is the full subcategory of \mathbf{FOSig}), with *algebras* (i.e., structures over algebraic signatures) as models (with reducts inherited from the definition of algebras as first-order structures, so that \mathbf{Alg} is a “subfunctor” of \mathbf{Str}), and with no sentences whatsoever (so that $\mathbf{Sen}^\emptyset(\Sigma) = \emptyset$).

Example 3.2. Ground equational institution: $\mathbf{GMSA} = \langle \mathbf{AlgSig}, \mathbf{GEQ}, \mathbf{Alg}, \models \rangle$, where for each algebraic signature $\Sigma \in |\mathbf{AlgSig}|$, $\mathbf{GEQ}(\Sigma)$ is the set of ground (no variables) Σ -equations, with the translations along signature morphisms and the satisfaction of equations in algebras defined in the standard way (as recalled in Sect. 2).

Example 3.3. The institution of ground order-sorted equational logic: $\mathbf{GOSA} = \langle \mathbf{OSSig}, \mathbf{GOSEQ}, \mathbf{OAlg}, \models \rangle$, where

- An order-sorted signature $\langle \Sigma, \leq \rangle$ is an algebraic signature Σ with a partial ordering \leq on its set of sorts. Order-sorted signature morphisms are like algebraic signature morphisms which in addition must preserve the ordering. This yields the category \mathbf{OSSig} of order-sorted signatures and their morphisms.
- For each order-sorted signature $\langle \Sigma, \leq \rangle$:
 - $\langle \Sigma, \leq \rangle$ -terms are built as usual, except that in addition to the operations in Σ , a subsort inclusion $\iota_{s \leq s'} : s \rightarrow s'$ and retract $r_{s \leq s'} : s' \rightarrow s$ is available when $s \leq s'$. Then $\mathbf{Sen}^{\mathbf{GOSA}}(\langle \Sigma, \leq \rangle)$ contains equations between such ground terms.
 - An order-sorted $\langle \Sigma, \leq \rangle$ -algebra A is a Σ -algebra where for any sorts $s \leq s'$, $|A|_s \subseteq |A|_{s'}$.
 - Evaluation of order-sorted $\langle \Sigma, \leq \rangle$ -terms is as usual, except that the inclusions $\iota_{s \leq s'}$ are interpreted as inclusions from $|A|_s$ to $|A|_{s'} \supseteq |A|_s$, and retracts $r_{s \leq s'}$ as maximal partial identities from $|A|_{s'}$ to $|A|_s \subseteq |A|_{s'}$. So, term evaluation is *partial*.⁵ A ground order-sorted equation $t = t'$ holds in an order-sorted algebra A , written as usual $A \models t = t'$, if the values in A of both t and t' are defined and equal.

Example 3.4. Ground rewriting institution $\mathbf{GPRWL} = \langle \mathbf{AlgSig}, \mathbf{GRW}, \mathbf{RAlg}, \models \rangle$ with algebraic signatures, and then for each signature $\Sigma \in |\mathbf{AlgSig}|$,

- sentences in $\mathbf{GRW}(\Sigma)$ are *rewritings* (or *transitions*) $t \Rightarrow t'$ between (ground) terms of a common sort,

⁴ \mathbf{P} stands for “pure”.

⁵ We depart here from $\mathbf{CafeOBJ}$, which to handle partiality either refers to the order-sorted [GM92] tradition, relying on the use of “error supersorts”, with retracts yielding “erroneous terms” rather than being undefined, or vaguely mentions membership equational logic [Mes98].

- models in $\mathbf{RAlg}(\Sigma)$ are *rewriting algebras*⁶, i.e., Σ -algebras $A \in \mathbf{Alg}(\Sigma)$ additionally equipped with a *rewriting relation* $\preceq_s \subseteq |A|_s \times |A|_s$ on the carrier of each sort s in Σ , where the family of the rewriting relations is required to be a *precongruence* on A , i.e., a preorder that is preserved (in the obvious sense) by all of the operations in A ,
- a Σ -rewriting $t \Rightarrow t'$ holds in a rewriting algebra $A \in \mathbf{RAlg}(\Sigma)$, written as usual $A \models t \Rightarrow t'$, if $t_A \preceq t'_A$.

Example 3.5. The institution of ground behavioural equational logic $\mathbf{GHA} = \langle \mathbf{BehSig}, \mathbf{GBEQ}, \mathbf{Alg}, \models \rangle$, where:⁷

- A *behavioural signature* $\langle \Sigma, OBS \rangle$ consists of an algebraic signature Σ together with the indicated set OBS of *observable* sorts in Σ . Behavioural signature morphisms are those algebraic signature morphisms that preserve the sets of observable and of non-observable sorts and, stating the extra condition somewhat informally, add no new terms leading from an “old” non-observable sort to an observable sort. This defines the category of behavioural signatures \mathbf{BehSig} .
- For each behavioural signature $\langle \Sigma, OBS \rangle$,
 - sentences are pairs of (ground) terms of a common sort, just like Σ -equations, but we write them here as $t \sim t'$,
 - models are just Σ -algebras,
 - for each Σ -algebra A , let \approx_A be the *indistinguishability* relation, i.e., the largest congruence on the subalgebra of A generated by the sorts in OBS that is the identity on the carriers of sorts in OBS (so that $a \approx_A b$ iff, relying on standard concepts and notation, for all contexts C of an observable sort, $C_A[a] = C_A[b]$). A ground Σ -equation $t \sim t'$ *behaviourally* holds in A , written $A \models t \sim t'$, if $t_A \approx_A t'_A$.

Given institutions $\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ and $\mathbf{INS}' = \langle \mathbf{Sign}', \mathbf{Sen}', \mathbf{Mod}', \langle \models_{\Sigma'} \rangle_{\Sigma' \in |\mathbf{Sign}'|} \rangle$, an *institution morphism* $\mu: \mathbf{INS} \rightarrow \mathbf{INS}'$ consists of

- a functor $\mu^{Sig}: \mathbf{Sign} \rightarrow \mathbf{Sign}'$,
- a natural transformation $\mu^{Sen}: \mu^{Sig}; \mathbf{Sen}' \rightarrow \mathbf{Sen}$, and
- a natural transformation $\mu^{Mod}: \mathbf{Mod} \rightarrow (\mu^{Sig})^{op}; \mathbf{Mod}'$

such that the following *satisfaction condition* holds: for any signature $\Sigma \in |\mathbf{Sign}|$, sentence $\varphi' \in \mathbf{Sen}'(\mu^{Sig}(\Sigma))$ and model $M \in \mathbf{Mod}(\Sigma)$, $M \models_{\Sigma} \mu^{Sen}(\varphi')$ iff $\mu_{\Sigma}^{Mod}(M) \models'_{\mu^{Sig}(\Sigma)} \varphi'$.

⁶ This terminology follows [DF02], even though recently some **CafeOBJ** authors go back to the more traditional term *(pre)ordered algebras*.

⁷ This is a crude version of the behavioural (hidden) equational logic of **CafeOBJ** presented in [DF02], which has a more subtle treatment of observability, specifying the set of operations that may be used as observers rather than indicating observable sorts, much in the style of constructor observational logic COL [BH06], going back perhaps to [SW83,ST87]. We omit here *coherence* statements, which are trivial in our behavioural institution.

Institution morphisms compose in the obvious, component-wise manner. We thus have a category \mathcal{JNS} of institutions and their morphisms.

Example 3.6. There are evident institution morphisms from the institutions GMSA , GOSA , GPRWL , GHA given in Examples 3.2, 3.3, 3.4 and 3.5, respectively, to the institution \mathbf{A} of Example 3.1; in each case signatures are mapped to their underlying algebraic signatures, and models to their underlying algebras (some of these mappings are identities, of course).

Example 3.7. The trivial morphism from GOSA to \mathbf{A} of Example 3.6 extends easily to a morphism from GOSA to GMSA , with the translation of (ground) equations in GMSA to order-sorted equations being the identity.

Example 3.8. The trivial morphism from GHA to \mathbf{A} of Example 3.6 does not extend to an institution morphism from GHA to GMSA — one may try to map equations $t = t'$ to behavioural equations $t \sim t'$ and check that one implication of the satisfaction condition would in general fail.

However, we may construct a different morphism, based on a signature functor that maps any behavioural signature $\langle\langle S, \Omega \rangle, OBS\rangle$ to the algebraic signature $\langle OBS, \Omega_{OBS} \rangle$ with observable sorts only and operations limited to *observable operations*, i.e., operations with observable arity and result sorts. Algebras are then mapped to their appropriate reducts, and (ground) equations over such limited signatures are mapped to their behavioural versions. It is easy to see that the satisfaction condition holds for such equations.

It is relatively easy to show completeness of the category \mathcal{JNS} of institutions and their morphisms:

Theorem 3.9 ([Tar86]). *\mathcal{JNS} is complete.*

In essence, the limit of a diagram of institutions is built by first defining the category of signatures as the limit of the categories of signatures of the institutions in the diagram. Signatures so defined in essence combine individual signatures in the institutions in the diagram linked by the signature functors of the institution morphisms involved. Then for each such “combined” signature, the set of sentences is defined as the colimit of the sets of sentences over the corresponding individual signatures with sentence translations between them given by the institution morphisms. Dually, the class of models is defined as the limit of the model classes over the corresponding individual signatures with model translations between them given by the institution morphisms. Finally, the satisfaction relation is defined uniquely so that the satisfaction condition holds for each of the resulting projection morphisms.

Example 3.10. The institution GRWL is defined as a pullback of GMSA and GPRWL over \mathbf{A} (via the trivial morphisms of Example 3.6). It has algebraic signatures (common to GMSA and GPRWL), rewriting algebras of GPRWL as models (mapped onto the class of algebras of GMSA) and sentences that are either equations (coming from GMSA) or rewritings (from GPRWL), with satisfaction inherited from the appropriate component institutions.

Example 3.11. Similarly, we may consider a pullback of GMSA and GHA over A (via the morphisms of Example 3.6). It has behavioural signatures as signatures, behavioural algebras as models, and sentences that are either (ground) equations of GMSA or behavioural equations of GHA. The morphism from GHA to GMSA of Example 3.8 is not involved here, and the two sets of sentences remain separate, even though one might want to identify equations between terms of observable sorts with their behavioural versions.

Example 3.12. We may also form a pullback of GOSA and GRWL over GMSA (via the morphism of Example 3.7 and the morphism given by the pullback construction of GRWL in Example 3.10). This would not be quite satisfactory though: in such a pullback institution, sentences would be either equations between order-sorted terms, as expected, or rewritings, but only between ordinary many-sorted terms. There would be no rewritings between order-sorted terms that involve subsort inclusions and retracts, which we would like to include in a combination of order-sorted algebra and rewriting logic as well. On the positive side: as expected, equations between the terms we have in GRWL would be glued together with their corresponding order-sorted equations.

Example 3.13. Another interesting pullback that is not adequate as a logic combination is the pullback of GHA and GRWL over GMSA (via the morphism of Example 3.8 and the morphism given by the pullback construction of GRWL in Example 3.10). The pullback institution has behavioural signatures as signatures (that map to the algebraic signatures in GRWL as in the morphism given in Example 3.8), and as models algebras with carriers equipped with a rewriting preorder on observable sorts only, preserved by observable operations. As sentences, we would get behavioural equations, here including standard equations between terms built using solely observable operations, and rewritings between such terms only. Clearly, what would be “missing” are rewritings between terms involving operations with non-observable result sorts.

4 Model-theoretic parchments

Examples 3.12 and 3.13 illustrate a major problem with using institutions and their limits as a tool for combining logical systems. Since logical sentences in institutions are regarded as unstructured entities, this works as expected only when we put together logical systems with sentences that capture distinct properties that do not interact with each other, as in Examples 3.10 and 3.11. Otherwise, we would prefer to combine the ways sentences are built, rather than sets of sentences as such. Consequently, we have to look more closely at sentence construction. To capture this, Goguen and Burstall [GB86] introduced *parchments*, an algebraic way to present institutions, where the syntax of sentences is given by the initial (term) algebra over a signature that lists the operations for constructing sentences and other auxiliary syntactic phrases. Parchments also presented models as signature morphisms into a special “large” signature, naming all potential denotations for signature components, with an indicated

Procrustean structure comprising all these denotations. Semantics of syntactic phrases is then captured by mapping the initial syntax to the corresponding reduct of the Procrustean algebra. The disadvantage is not only the need to use such “large” objects (with all the foundational worries they bring) but also that we inherently mix together model-theoretic and syntactic aspects of logical systems presented in such a way. To avoid this, in [MTP98] we proposed a version of parchments that keeps the models separate and splits the Procrustean semantic object into smaller objects appropriate for each model considered.

This means that model-theoretic parchments comprise signatures and models in the same way as institutions do. However, while in institutions sentences are given directly by the sentence functor, model-theoretic parchments feature a language functor that maps each signature of the model-theoretic parchment to a first-order signature with an algebraic part representing the abstract syntax of sentences⁸; sentences are then generated as terms of the distinguished sort $*$. Moreover, instead of a satisfaction relation, model-theoretic parchments, for each signature and model, feature an *evaluation structure* that gives interpretation for the syntactic constructs used to build sentences. The interpretation of terms in the evaluation structure determines the meaning of syntactic phrases used in sentences, and of sentences themselves. A sentence holds in a model when in the evaluation structure for this model the sentence as a term evaluates to a logical value designated by the special predicate D . Finally, the satisfaction condition is ensured by suitable coherence homomorphisms between evaluation structures.

A *model-theoretic parchment* (or briefly: *parchment*) $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ consists of:

- a category \mathbf{Sign} of signatures;
- a functor $\mathbf{L}: \mathbf{Sign} \rightarrow \mathbf{FOSig}_*$ that for any signature $\Sigma \in |\mathbf{Sign}|$ yields a first-order signature $\mathbf{L}(\Sigma)$ that gives the abstract syntax for sentences;
- a functor $\mathbf{Mod}: \mathbf{Sign}^{op} \rightarrow \mathbf{Class}$ (as for institutions); and
- a family \mathcal{G} that in turn consists of:⁹
 - for any signature $\Sigma \in |\mathbf{Sign}|$ and model $M \in \mathbf{Mod}(\Sigma)$, an $\mathbf{L}(\Sigma)$ -structure $\mathcal{G}_\Sigma(M) \in |\mathbf{Str}(\mathbf{L}(\Sigma))|$; and
 - for any signature morphism $\sigma: \Sigma \rightarrow \Sigma'$ and model $M' \in \mathbf{Mod}(\Sigma')$, an $\mathbf{L}(\Sigma)$ -homomorphism $\mathcal{G}_\sigma(M'): \mathcal{G}_\Sigma(M'|_\sigma) \rightarrow \mathcal{G}_{\Sigma'}(M')|_{\mathbf{L}(\sigma)}$ such that for any signature morphisms $\sigma_1: \Sigma_0 \rightarrow \Sigma_1$, $\sigma_2: \Sigma_1 \rightarrow \Sigma_2$ and model $M_2 \in \mathbf{Mod}(\Sigma_2)$, $\mathcal{G}_{\sigma_1; \sigma_2}(M_2) = \mathcal{G}_{\sigma_1}(M_2|_{\sigma_2}); \mathcal{G}_{\sigma_2}(M_2)|_{\mathbf{L}(\sigma_1)}$.

Informally, for any signature $\Sigma \in |\mathbf{Sign}|$ and model $M \in \mathbf{Mod}(\Sigma)$, the *evaluation structure* $\mathcal{G}_\Sigma(M)$ determines semantic evaluation of $\mathbf{L}(\Sigma)$ -phrases in the model M . Then the *mediating homomorphisms* $\mathcal{G}_\sigma(M'): \mathcal{G}_\Sigma(M'|_\sigma) \rightarrow \mathcal{G}_{\Sigma'}(M')|_{\mathbf{L}(\sigma)}$

⁸ This relies on the usual correspondence between context-free grammars and algebraic signatures.

⁹ \mathcal{G} may be viewed as a signature-preserving functor between Grothendieck categories built by “flattening” $\mathbf{Mod}: \mathbf{Sign}^{op} \rightarrow \mathbf{Class}$ and $\mathbf{L}^{op}; \mathbf{Str}: \mathbf{Sign}^{op} \rightarrow \mathbf{Cat}$, respectively, cf. [TBG91]. We prefer to indicate the components of \mathcal{G} explicitly, so we refrain from spelling out and using this alternative formulation.

ensure that this evaluation changes smoothly when we move from one signature to another, and so is in a sense uniform for the entire logical system presented by the parchment. However, the uniformity as captured by the mediating homomorphisms implies that semantic properties are preserved, but not necessarily reflected, by model reducts w.r.t. signature morphisms.

We think of the set $|\mathcal{G}_\Sigma(M)|_*$ as the set of logical values for evaluation of Σ -sentences in $M \in \mathbf{Mod}(\Sigma)$. By allowing arbitrary sets of values here we naturally accommodate various forms of many-valued logics, with non-standard logical values permitted. Then the predicate $D: *$ designates the logical values that indicate which sentences “hold” in the model, thus enabling a classical two-valued understanding of satisfaction on top of possibly many-valued sentence evaluation.

A parchment as above is *institutional* if for any signature morphism $\sigma: \Sigma \rightarrow \Sigma'$ and model $M' \in \mathbf{Mod}(\Sigma')$, $\mathcal{G}_\sigma(M'): \mathcal{G}_\Sigma(M'|_\sigma) \rightarrow \mathcal{G}_{\Sigma'}(M')|_{\mathbf{L}(\sigma)}$ is a closed $\mathbf{L}(\Sigma)$ -homomorphism on the subsignature *Logic*, that is $|\mathcal{G}_\sigma(M')|_*$ preserves and reflects the predicate $D: *$.¹⁰ Then, such an institutional parchment is *Boolean* if for any signature $\Sigma \in |\mathbf{Sign}|$ and model $M \in \mathbf{Mod}(\Sigma)$, $|\mathcal{G}_\Sigma(M)|_* = \mathit{Bool}$, where $\mathit{Bool} = \{tt, ff\}$, and $\mathit{D}_{\mathcal{G}_\Sigma(M)} = \{tt\}$ (it follows that the homomorphisms $\mathcal{G}_\sigma(M')$ are identities on the sort $*$). We say that a parchment is *strict* if for any signature morphism $\sigma: \Sigma \rightarrow \Sigma'$ and model $M' \in \mathbf{Mod}(\Sigma')$, $\mathcal{G}_\sigma(M'): \mathcal{G}_\Sigma(M'|_\sigma) \rightarrow \mathcal{G}_{\Sigma'}(M')|_{\mathbf{L}(\sigma)}$ is the identity; in particular, all strict parchments are institutional.

Any institutional parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ *presents* the institution $\mathcal{J}(\mathbf{P}) = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$, which inherits signatures and models directly from \mathbf{P} , and

- for $\Sigma \in |\mathbf{Sign}|$, $\mathbf{Sen}(\Sigma) = |T_{\mathbf{L}(\Sigma)}|_*$, where $T_{\mathbf{L}(\Sigma)}$ is the initial $\mathbf{L}(\Sigma)$ -structure (so that Σ -sentences are ground $\mathbf{L}(\Sigma)$ -terms of sort $*$),
- for $\sigma: \Sigma \rightarrow \Sigma'$, $\mathbf{Sen}(\sigma) = (!_\sigma)_*$, where $!_\sigma: T_{\mathbf{L}(\Sigma)} \rightarrow T_{\mathbf{L}(\Sigma')}|_{\mathbf{L}(\sigma)}$ is the unique $\mathbf{L}(\Sigma)$ -homomorphism given by the initiality of $T_{\mathbf{L}(\Sigma)}$, and
- for $\Sigma \in |\mathbf{Sign}|$, $\varphi \in |T_{\mathbf{L}(\Sigma)}|_*$, and $M \in \mathbf{Mod}(\Sigma)$, $M \models_\Sigma \varphi$ iff $\varphi_{\mathcal{G}_\Sigma(M)} \in \mathit{D}_{\mathcal{G}_\Sigma(M)}$ (i.e., in $\mathcal{G}_\Sigma(M)$ the predicate D holds on the value of φ viewed as a $\mathbf{L}(\Sigma)$ -term of sort $*$).

One can check now that $\mathcal{J}(\mathbf{P})$ so defined is indeed an institution, where for $\sigma: \Sigma \rightarrow \Sigma'$, $\varphi \in \mathbf{Sen}(\Sigma)$ and $M' \in \mathbf{Mod}(\Sigma')$, the satisfaction condition follows since the homomorphism $\mathcal{G}_\sigma(M')$ preserves and reflects the predicate $D: *$.

A parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ is *atomic* if for all $\Sigma \in |\mathbf{Sign}|$, the first-order signature $\mathbf{L}(\Sigma)$ has no operations with $*$ in their arity. In that case, no sentence is constructed out of other sentences, and so, informally, all the sentences are atomic.

Example 4.1. A Boolean parchment that presents the institution \mathbf{A} of Example 3.1 is $\mathbf{P}_\mathbf{A} = \langle \mathbf{AlgSig}, \mathbf{L}^\mathbf{A}, \mathbf{Alg}, \mathcal{G}^\mathbf{A} \rangle$ with algebraic signatures, with algebras as models, and where for any $\Sigma \in |\mathbf{AlgSig}|$, $\mathbf{L}^\mathbf{A}(\Sigma)$ extends *Logic* by

¹⁰ This requirement is deliberately weaker than that imposed on “logical” parchments in [MTP98].

adding (the sorts and operations of) Σ , and then for any algebra $A \in \mathbf{Alg}(\Sigma)$, $\mathcal{G}_{\Sigma}^A(A) \in \mathbf{Str}(\mathbf{L}^A(\Sigma))$ coincides with A on Σ , with identity mediating homomorphisms.

Example 4.2. A Boolean parchment that in essence presents the institution GMSA of Example 3.2 is $\mathbf{P}_{\text{GMSA}} = \langle \mathbf{AlgSig}, \mathbf{L}^{\text{GMSA}}, \mathbf{Alg}, \mathcal{G}^{\text{GMSA}} \rangle$, with algebraic signatures, with algebras as models, and where for any $\Sigma \in |\mathbf{AlgSig}|$:

- $\mathbf{L}^{\text{GMSA}}(\Sigma)$ extends *Logic* by adding Σ and for each sort s in Σ , a binary operation $eq: s \times s \rightarrow *$;
- for any algebra $A \in \mathbf{Alg}(\Sigma)$, $\mathcal{G}_{\Sigma}^{\text{GMSA}}(A) \in \mathbf{Str}(\mathbf{L}^{\text{GMSA}}(\Sigma))$ is A on Σ , and interprets eq as the diagonal function, yielding tt if its two arguments coincide, and ff if they are distinct; and
- mediating homomorphisms are identities.

Now, $\mathbf{L}^{\text{GMSA}}(\Sigma)$ -terms of sort $*$ are of the form $eq(t, t')$, for Σ -terms t and t' of a common sort. Such a term evaluates to tt in $\mathcal{G}_{\Sigma}^{\text{GMSA}}(A)$ if the terms t and t' evaluate in $\mathcal{G}_{\Sigma}^{\text{GMSA}}(A)$ (or equivalently, in A) to equal values. Consequently, the parchment \mathbf{P}_{GMSA} presents the institution GMSA, modulo the details of the actual notation used for sentences (we will disregard such differences from now on).

Example 4.3. A Boolean parchment that presents the institution GOSA of Example 3.3 is $\mathbf{P}_{\text{GOSA}} = \langle \mathbf{OSSig}, \mathbf{L}^{\text{GOSA}}, \mathbf{OSAlg}, \mathcal{G}^{\text{GOSA}} \rangle$, with order-sorted signatures, with order-sorted algebras as models, and then for any order-sorted signature $\langle \Sigma, \leq \rangle$:

- $\mathbf{L}^{\text{GOSA}}(\langle \Sigma, \leq \rangle)$ extends *Logic* by Σ and all the subsort inclusions and retracts, as well as the operation $eq: s \times s \rightarrow *$ for each sort s in Σ ;
- for each order-sorted algebra $A \in \mathbf{OSAlg}(\langle \Sigma, \leq \rangle)$, $\mathcal{G}_{\langle \Sigma, \leq \rangle}^{\text{GOSA}}(A)$ expands A on Σ by adding an “undefined” element \perp to the carrier of each sort s in Σ and extending the interpretation of all operations in A so that they are strict on \perp (yield \perp as the result on any tuple of arguments that contains \perp), and interprets subsort inclusions and retracts in the obvious way (retracts map to \perp the elements of the supersort that are not in the subsort) and the eq operations as the diagonal on the “defined” elements in $|A|$ and yielding ff when any of its arguments is \perp ; and
- mediating morphisms are identities again.

Now, an order-sorted $\langle \Sigma, \leq \rangle$ -term with a defined value in an order-sorted $\langle \Sigma, \leq \rangle$ -algebra A evaluates to the same value in $\mathcal{G}_{\langle \Sigma, \leq \rangle}^{\text{GOSA}}(A)$; if it is undefined in A then in $\mathcal{G}_{\langle \Sigma, \leq \rangle}^{\text{GOSA}}(A)$ it has the value \perp . Hence, $eq(t, t')$ evaluates to tt in $\mathcal{G}_{\langle \Sigma, \leq \rangle}^{\text{GOSA}}(A)$ iff the values of t and t' in A are defined and equal. Consequently, the parchment \mathbf{P}_{GOSA} indeed presents the institution GOSA.

Example 4.4. A Boolean parchment that presents the institution GPRWL of Example 3.4 is $\mathbf{P}_{\text{GPRWL}} = \langle \mathbf{AlgSig}, \mathbf{L}^{\text{GPRWL}}, \mathbf{RAlg}, \mathcal{G}^{\text{GPRWL}} \rangle$, with algebraic signatures, with rewriting algebras as models, and where for any $\Sigma \in |\mathbf{AlgSig}|$:

- $\mathbf{L}^{\text{GPRWL}}(\Sigma)$ extends *Logic* by adding Σ and for each sort s in Σ a binary operation $\text{rwrt}: s \times s \rightarrow *$,
- for any rewriting algebra $A \in \mathbf{RAlg}(\Sigma)$, $\mathcal{G}_{\Sigma}^{\text{GPRWL}}(A) \in \mathbf{Str}(\mathbf{L}^{\text{GPRWL}}(\Sigma))$ is the standard algebra part of A on Σ , and interprets each rwrt so that it yields tt if its two arguments are in the rewriting precongruence, and ff otherwise, and
- mediating homomorphisms are identities.

Example 4.5. A Boolean parchment that presents the institution GHA of Example 3.5 is $\mathbf{P}_{\text{GHA}} = \langle \mathbf{BehSig}, \mathbf{L}^{\text{GHA}}, \mathbf{Alg}, \mathcal{G}^{\text{GHA}} \rangle$, with behavioural signatures and algebras as models, and where for any $\Sigma \in |\mathbf{AlgSig}|$:

- $\mathbf{L}^{\text{GHA}}(\Sigma)$ extends *Logic* by adding Σ and for each sort s in Σ a binary operation $\text{beq}: s \times s \rightarrow *$, and
- for any algebra $A \in \mathbf{Alg}(\Sigma)$, $\mathcal{G}_{\Sigma}^{\text{GHA}}(A) \in \mathbf{Str}(\mathbf{L}^{\text{GHA}}(\Sigma))$ is A on Σ , and interprets beq to capture the indistinguishability relation, yielding tt if its two arguments are related by \approx_A , and ff otherwise, and
- mediating homomorphisms are identities.¹¹

Clearly, all parchments in Examples 4.1–4.5 are atomic and strict.

Given two parchments $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ and $\mathbf{P}' = \langle \mathbf{Sign}', \mathbf{L}', \mathbf{Mod}', \mathcal{G}' \rangle$, a *parchment morphism* $\gamma: \mathbf{P} \rightarrow \mathbf{P}'$ consists of:

- a functor $\gamma^{\text{Sig}}: \mathbf{Sign} \rightarrow \mathbf{Sign}'$,
- a natural transformation $\gamma^{\text{Lan}}: \gamma^{\text{Sig}}; \mathbf{L}' \rightarrow \mathbf{L}$,
- a natural transformation $\gamma^{\text{Mod}}: \mathbf{Mod} \rightarrow (\gamma^{\text{Sig}})^{\text{op}}; \mathbf{Mod}'$,
- a family of homomorphisms $\gamma_{\Sigma, M}^{\mathcal{G}}: \mathcal{G}'_{\gamma^{\text{Sig}}(\Sigma)}(\gamma_{\Sigma}^{\text{Mod}}(M)) \rightarrow \mathcal{G}_{\Sigma}(M)|_{\gamma_{\Sigma}^{\text{Lan}}}$, for $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, such that for any signature morphism $\sigma: \Sigma_1 \rightarrow \Sigma_2$ in \mathbf{Sign} and model $M_2 \in \mathbf{Mod}(\Sigma_2)$ we have

$$\gamma_{\Sigma_1, M_2|_{\sigma}}^{\mathcal{G}}; \mathcal{G}_{\sigma}(M_2)|_{\gamma_{\Sigma_1}^{\text{Lan}}} = \mathcal{G}'_{\sigma'}(\gamma_{\Sigma_2}^{\text{Mod}}(M_2)); \gamma_{\Sigma_2, M_2}^{\mathcal{G}}|_{\mathbf{L}'(\sigma')}$$

where $\sigma' = \gamma^{\text{Sig}}(\sigma): \Sigma'_1 \rightarrow \Sigma'_2$.

The naturality condition in the last item captures the identity of two composed $\mathbf{L}'(\Sigma'_1)$ -homomorphisms of type

$$\mathcal{G}'_{\Sigma'_1}(\gamma_{\Sigma_1}^{\text{Mod}}(M_2|_{\sigma})) = \mathcal{G}'_{\Sigma'_1}(\gamma_{\Sigma_2}^{\text{Mod}}(M_2)|_{\sigma'}) \rightarrow \mathcal{G}_{\Sigma_2}(M_2)|_{\gamma_{\Sigma_1}^{\text{Lan}}; \mathbf{L}(\sigma)} = \mathcal{G}_{\Sigma_2}(M_2)|_{\mathbf{L}'(\sigma'); \gamma_{\Sigma_2}^{\text{Lan}}}.$$

This may look scary, but we encourage the reader to “type” the morphisms in question and make sure that the condition is not only correctly stated, but is indeed natural.¹²

¹¹ The extra condition imposed in Example 3.5 on signature morphisms in \mathbf{BehSig} plays a crucial role here: for signature morphisms that add new contexts to “observe” old sorts, the identity map indicated here as the mediating homomorphism may fail to preserve the operation beq .

¹² In fact, $\gamma^{\mathcal{G}}$ is a natural transformation between suitably re-indexed functors \mathcal{G}' and \mathcal{G} , see footnote 9.

As with parchments, where only institutional parchments presented institutions, not every parchment morphism presents an institution morphism. We say that a parchment morphism as above is *institutional* if all homomorphisms $\gamma_{\Sigma, M}^{\mathcal{G}}$ are closed on the subsignature *Logic*.¹³ It follows that in institutional parchment morphisms¹⁴ between Boolean parchments, the homomorphisms $\gamma_{\Sigma, M}^{\mathcal{G}}$ are identities on the subsignature *Logic*. If all homomorphisms $\gamma_{\Sigma, M}^{\mathcal{G}}$ are identities, we say that the parchment morphism is *strict*.

For institutional parchments \mathbf{P} and \mathbf{P}' as above, each institutional parchment morphism $\gamma = \langle \gamma^{Sig}, \gamma^{Lan}, \gamma^{Mod}, \gamma^{\mathcal{G}} \rangle: \mathbf{P} \rightarrow \mathbf{P}'$ presents an institution morphism $\mathcal{J}(\gamma): \mathcal{J}(\mathbf{P}) \rightarrow \mathcal{J}(\mathbf{P}')$, defined as follows:

- $(\mathcal{J}(\gamma))^{Sig} = \gamma^{Sig}$,
- $(\mathcal{J}(\gamma))^{Mod} = \gamma^{Mod}$,
- for $\Sigma \in |\mathbf{Sign}|$, let $\Sigma' = \gamma^{Sig}(\Sigma)$; then $(\mathcal{J}(\gamma))_{\Sigma}^{Sen}: |T_{\mathbf{L}'(\Sigma')}|_* \rightarrow |T_{\mathbf{L}(\Sigma)}|_*$ is given as the $*$ component of the unique $\mathbf{L}'(\Sigma')$ -homomorphism $!_{\Sigma}: T_{\mathbf{L}'(\Sigma')} \rightarrow T_{\mathbf{L}(\Sigma)}|_{\gamma_{\Sigma}^{Lan}}$.

One can check now that $\mathcal{J}(\gamma)$ so defined is indeed an institution morphism $\mathcal{J}(\gamma): \mathcal{J}(\mathbf{P}) \rightarrow \mathcal{J}(\mathbf{P}')$. In particular, the satisfaction condition follows since for any signature $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, the homomorphism $\gamma_{\Sigma, M}^{\mathcal{G}}$ reflects and preserves the predicate $D: *$.

Example 4.6. There are evident parchment morphisms from the parchments \mathbf{P}_{GMSA} , \mathbf{P}_{GOSA} , $\mathbf{P}_{\text{GPRWL}}$, \mathbf{P}_{GHA} , given in Examples 4.2, 4.3, 4.4, and 4.5, respectively, to \mathbf{P}_{A} given in Example 4.1, presenting the corresponding institution morphisms from Example 3.6. In each case signatures are mapped to their underlying algebraic signatures, models are mapped to the underlying algebras, and the maps on abstract syntax signatures are simply inclusions. All these parchment morphisms are strict (i.e., all the $\gamma^{\mathcal{G}}$ homomorphisms are identities) except for the morphism from \mathbf{P}_{GOSA} to \mathbf{P}_{A} , where for any order-sorted signature $\langle \Sigma, \leq \rangle$ and $A \in \mathbf{OSAlg}(\langle \Sigma, \leq \rangle)$, $\gamma_{\langle \Sigma, \leq \rangle, A}^{\mathcal{G}}: \mathcal{G}_{\langle \Sigma, \leq \rangle}^A(A) \rightarrow \mathcal{G}_{\langle \Sigma, \leq \rangle}^{\text{GOSA}}(A)|_{\mathbf{L}^A(\Sigma)}$ is identity on $*$ and inclusion on sorts from Σ (“adding” undefined elements \perp).

Example 4.7. The parchment morphism from \mathbf{P}_{GOSA} to \mathbf{P}_{A} extends to the obvious strict parchment morphism from \mathbf{P}_{GOSA} to \mathbf{P}_{GMSA} , where the abstract syntax signatures are mapped by inclusions. This parchment morphism presents the institution morphism from GOSA to GMSA given in Example 3.7.

Example 4.8. The institution morphism from GHA to GMSA given in Example 3.8 is presented by a strict parchment morphism from \mathbf{P}_{GHA} to \mathbf{P}_{GMSA} : signatures and models are mapped as in the institution morphism (so, forgetting about non-observable parts of behavioural signatures and their algebras), and abstract syntax signatures are mapped essentially by inclusions, except that the *eq* operations are renamed to *beq*.

¹³ Again, this is weaker than the corresponding condition imposed in [MTP98].

¹⁴ To clarify: *institutional parchment morphism* refers to a parchment morphism that is institutional (rather than to an institutional-parchment morphism, i.e., a morphism between institutional parchments).

5 Constructions in parchment categories

The rather straightforward composition of parchment morphisms $\gamma_1: \mathbf{P}_0 \rightarrow \mathbf{P}_1$ and $\gamma_2: \mathbf{P}_1 \rightarrow \mathbf{P}_2$ is the parchment morphism $\gamma: \mathbf{P}_0 \rightarrow \mathbf{P}_2$ defined as follows:

- $\gamma^{Sig} = \gamma_1^{Sig}; \gamma_2^{Sig}$,
- $\gamma^{Lan} = (\gamma_1^{Sig}; \gamma_2^{Lan}); \gamma_1^{Lan}$,
- $\gamma^{Mod} = \gamma_1^{Mod}; ((\gamma_1^{Sig})^{op}; \gamma_2^{Mod})$,
- for any \mathbf{P}_0 -signature Σ_0 and any Σ_0 -model M_0 , let $\Sigma_1 = \gamma_1^{Sig}(\Sigma_0)$ and $M_1 = (\gamma_1^{Mod})_{\Sigma_0}(M_0)$; then $\gamma_{\Sigma_0, M_0}^{\mathcal{G}} = (\gamma_2^{\mathcal{G}})_{\Sigma_1, M_1}; (\gamma_1^{\mathcal{G}})_{\Sigma_0, M_0} |_{(\gamma_2^{Lan})_{\Sigma_1}}$.

This defines a category \mathcal{PAR} of parchments and their morphisms. \mathcal{JPAR} denotes the subcategory of institutional parchments with institutional parchment morphisms. The construction of institutions and institutions morphisms from institutional parchments and their institutional morphisms, respectively, as given in Sect. 4, yields a functor $\mathcal{J}: \mathcal{JPAR} \rightarrow \mathcal{INS}$.

We can combine parchments using limits:

Theorem 5.1 ([MTP98]). *\mathcal{PAR} is complete.*

Instead of a detailed proof (which may be found in the full version of [MTP98]), let us just mention that the construction of limits in \mathcal{PAR} essentially follows the same idea as for institutions, see Thm. 3.9, and illustrate how this works for pullbacks.

Given parchments $\mathbf{P}_0 = \langle \mathbf{Sign}_0, \mathbf{L}_0, \mathbf{Mod}_0, \mathcal{G}_0 \rangle$, $\mathbf{P}_1 = \langle \mathbf{Sign}_1, \mathbf{L}_1, \mathbf{Mod}_1, \mathcal{G}_1 \rangle$, $\mathbf{P}_2 = \langle \mathbf{Sign}_2, \mathbf{L}_2, \mathbf{Mod}_2, \mathcal{G}_2 \rangle$ and parchment morphisms $\gamma_1: \mathbf{P}_1 \rightarrow \mathbf{P}_0$ and $\gamma_2: \mathbf{P}_2 \rightarrow \mathbf{P}_0$, we sketch the construction of their pullback in \mathcal{PAR} as a parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ with morphisms $\gamma_3: \mathbf{P} \rightarrow \mathbf{P}_1$ and $\gamma_4: \mathbf{P} \rightarrow \mathbf{P}_2$:

- The category **Sign** of signatures with $\gamma_3^{Sig}: \mathbf{Sign} \rightarrow \mathbf{Sign}_1$ and $\gamma_4^{Sig}: \mathbf{Sign} \rightarrow \mathbf{Sign}_2$ is obtained as a pullback of $\gamma_1^{Sig}: \mathbf{Sign}_1 \rightarrow \mathbf{Sign}_0$ and $\gamma_2^{Sig}: \mathbf{Sign}_2 \rightarrow \mathbf{Sign}_0$ in **Cat**.
- For each signature $\Sigma \in |\mathbf{Sign}|$, with $\Sigma_1 = \gamma_3^{Sig}(\Sigma)$, $\Sigma_2 = \gamma_4^{Sig}(\Sigma)$ and $\Sigma_0 = \gamma_1^{Sig}(\Sigma_1) (= \gamma_2^{Sig}(\Sigma_2))$:
 - the abstract syntax signature $\mathbf{L}(\Sigma)$ with $(\gamma_3^{Lan})_{\Sigma}: \mathbf{L}_1(\Sigma_1) \rightarrow \mathbf{L}(\Sigma)$ and $(\gamma_4^{Lan})_{\Sigma}: \mathbf{L}_2(\Sigma_2) \rightarrow \mathbf{L}(\Sigma)$ is given as a pushout of $(\gamma_1^{Lan})_{\Sigma_1}: \mathbf{L}_0(\Sigma_0) \rightarrow \mathbf{L}_1(\Sigma_1)$ and $(\gamma_2^{Lan})_{\Sigma_2}: \mathbf{L}_0(\Sigma_0) \rightarrow \mathbf{L}_2(\Sigma_2)$ in **FOSig***,
 - the class of models $\mathbf{Mod}(\Sigma)$ with $(\gamma_3^{Mod})_{\Sigma}: \mathbf{Mod}(\Sigma) \rightarrow \mathbf{Mod}_1(\gamma_3^{Sig}(\Sigma))$ and $(\gamma_4^{Mod})_{\Sigma}: \mathbf{Mod}(\Sigma) \rightarrow \mathbf{Mod}_2(\gamma_4^{Sig}(\Sigma))$ is obtained as a pullback of $(\gamma_1^{Mod})_{\Sigma_1}: \mathbf{Mod}_1(\Sigma_1) \rightarrow \mathbf{Mod}_0(\Sigma_0)$ and $(\gamma_2^{Mod})_{\Sigma_2}: \mathbf{Mod}_2(\Sigma_2) \rightarrow \mathbf{Mod}_0(\Sigma_0)$ in **Class**, and
 - for each model $M \in \mathbf{Mod}(\Sigma)$, in an attempt to make the construction of $\mathcal{G}_{\Sigma}(M)$ readable, let us introduce a number of abbreviations:
 - * $M_1 = (\gamma_3^{Mod})_{\Sigma}(M)$, $M_2 = (\gamma_4^{Mod})_{\Sigma}(M)$ and $M_0 = (\gamma_1^{Mod})_{\Sigma_1}(M_1) (= (\gamma_2^{Mod})_{\Sigma_2}(M_2))$,
 - * $G_1 = (\mathcal{G}_1)_{\Sigma_1}(M_1)$, $G_2 = (\mathcal{G}_2)_{\Sigma_2}(M_2)$, and $G_0 = (\mathcal{G}_0)_{\Sigma_0}(M_0)$,

$$\begin{aligned}
* \quad & \theta_1 = (\gamma_1^{Lan})_{\Sigma_1} : \mathbf{L}_0(\Sigma_0) \rightarrow \mathbf{L}_1(\Sigma_1), \theta_2 = (\gamma_2^{Lan})_{\Sigma_2} : \mathbf{L}_0(\Sigma_0) \rightarrow \mathbf{L}_2(\Sigma_2), \\
& \theta_3 = (\gamma_3^{Lan})_{\Sigma} : \mathbf{L}_1(\Sigma_1) \rightarrow \mathbf{L}(\Sigma), \theta_4 = (\gamma_4^{Lan})_{\Sigma} : \mathbf{L}_2(\Sigma_2) \rightarrow \mathbf{L}(\Sigma), \\
& \text{and } \theta = \theta_1; \theta_3 (= \theta_2; \theta_4).
\end{aligned}$$

We have two $\mathbf{L}_0(\Sigma_0)$ -homomorphisms $(\gamma_1^{\mathcal{G}})_{\Sigma_1, M_1} : G_0 \rightarrow G_1|_{(\gamma_1^{Lan})_{\Sigma_1}}$ and $(\gamma_2^{\mathcal{G}})_{\Sigma_2, M_2} : G_0 \rightarrow G_2|_{(\gamma_2^{Lan})_{\Sigma_2}}$. Using freeness, we get $(\gamma_1^{\mathcal{G}})_{\Sigma_1, M_1}^{\#} : F_{\theta_1}(G_0) \rightarrow G_1$ in $\mathbf{Str}(\mathbf{L}_1(\Sigma_1))$ and $(\gamma_2^{\mathcal{G}})_{\Sigma_2, M_2}^{\#} : F_{\theta_2}(G_0) \rightarrow G_2$ in $\mathbf{Str}(\mathbf{L}_2(\Sigma_2))$. Then, since (up to natural isomorphism) $F_{\theta_3}(F_{\theta_1}(G_0))$ and $F_{\theta_4}(F_{\theta_2}(G_0))$ coincide with $F_{\theta}(G_0)$, we may assume that $F_{\theta_3}((\gamma_1^{\mathcal{G}})_{\Sigma_1, M_1}^{\#}) : F_{\theta}(G_0) \rightarrow F_{\theta_3}(G_1)$ and $F_{\theta_4}((\gamma_2^{\mathcal{G}})_{\Sigma_2, M_2}^{\#}) : F_{\theta}(G_0) \rightarrow F_{\theta_4}(G_2)$. Let now $\mathcal{G}_{\Sigma}(M)$ with $\mathbf{L}(\Sigma)$ -homomorphisms $g_1 : F_{\theta_3}(G_1) \rightarrow \mathcal{G}_{\Sigma}(M)$ and $g_2 : F_{\theta_4}(G_2) \rightarrow \mathcal{G}_{\Sigma}(M)$ be their pushout in $\mathbf{Str}(\mathbf{L}(\Sigma))$. Finally, put $(\gamma_3^{\mathcal{G}})_{\Sigma, M} = (\eta_{\theta_3})_{G_1; g_1}|_{\theta_3} : G_1 \rightarrow \mathcal{G}_{\Sigma}(M)|_{\theta_3}$ and $(\gamma_4^{\mathcal{G}})_{\Sigma, M} = (\eta_{\theta_4})_{G_2; g_2}|_{\theta_4} : G_2 \rightarrow \mathcal{G}_{\Sigma}(M)|_{\theta_4}$.

Then, for each signature morphism $\sigma : \Sigma \rightarrow \Sigma'$:

- $\mathbf{L}(\sigma) : \mathbf{L}(\Sigma) \rightarrow \mathbf{L}(\Sigma')$ is given by the pushout property of $\mathbf{L}(\Sigma)$,
- $\mathbf{Mod}(\sigma) : \mathbf{Mod}(\Sigma') \rightarrow \mathbf{Mod}(\Sigma)$ is given by the pullback property of $\mathbf{Mod}(\Sigma)$, and
- for any model $M' \in \mathbf{Mod}(\Sigma')$, $\mathcal{G}_{\sigma}(M') : \mathcal{G}_{\Sigma}(M'|_{\sigma}) \rightarrow \mathcal{G}_{\Sigma'}(M')|_{\mathbf{L}(\Sigma)}$ is given by the pushout property of $\mathcal{G}_{\Sigma}(M'|_{\sigma})$.

It is routine (but very tedious!) to check that the above indeed defines a parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ with parchment morphisms $\gamma_3 : \mathbf{P} \rightarrow \mathbf{P}_1$ and $\gamma_4 : \mathbf{P} \rightarrow \mathbf{P}_2$ which form a pullback of $\gamma_1 : \mathbf{P}_1 \rightarrow \mathbf{P}_0$ and $\gamma_2 : \mathbf{P}_2 \rightarrow \mathbf{P}_0$.

Proposition 5.2. *The limit in \mathcal{PAR} of a diagram of institutional parchments and institutional parchment morphisms is not necessarily an institutional parchment, but the limiting cone consists of institutional parchment morphisms.*

This is the first sign of worry that a programme to “just” use the standard limit construction to put together logical systems presented by institutional parchments linked by institutional parchment morphisms is doomed. Here is another negative result, perhaps expected after Prop. 5.2, to show that this idea cannot work in general:

Proposition 5.3 ([MTP98]). *The category \mathcal{JPAR} of institutional parchments and their institutional morphisms is not complete.*

The source of these negative results is that the free constructions involved in building the evaluation structures in the limit parchment in general add new values, possibly also new logical values (of sort $*$). The predicate $\mathbf{D} : *$ does not hold on these new values over a given signature (so that the limit projection morphisms are institutional). However, there may be extensions of the signature considered where the new logical values are glued together with “old” logical values (due to identification of some parts of syntax) and when $\mathbf{D} : *$ holds on them, the mediating homomorphism is not closed — which yields the negative part of Prop. 5.2. Then, even when this does not happen and the limit parchment is institutional, there may be common compatible extensions of the parchments in

the diagram (institutional cones over this diagram) that designate the predicate $D: * \text{ to hold for some of the new logical values. Consequently, the unique parchment morphism from such a cone to the limit in } \mathcal{PAR} \text{ need not be institutional. This shows that for a parchment diagram in } \mathcal{JPAR}, \text{ even if its limit given by Thm. 5.1 is an institutional parchment and so the limit cone fits entirely into } \mathcal{JPAR}, \text{ it still does not have to be a limit of this diagram in } \mathcal{JPAR}.$

In fact, this is as expected: there is nothing like a free lunch, we cannot get meanings for essentially new combinations of syntactic constructs involved for free. The upshot is that the new logical values added by the free constructions involved in the limits in \mathcal{PAR} indicate the need for some decision concerning the meaning of such new phrases. Technically, this may take the form of consistently choosing a family of congruences on the evaluation structures that glue together new and old logical values.

Example 5.4. One can easily construct the combination of \mathbf{P}_{GOSA} and $\mathbf{P}_{\text{GPRWL}}$ as a pullback of \mathbf{P}_{GOSA} and $\mathbf{P}_{\text{GPRWL}}$ over \mathbf{P}_A , via the parchment morphisms given in Example 4.6. The pullback parchment has order-sorted signatures as signatures, and order-sorted algebras as models. For any order-sorted signature $\langle \Sigma, \leq \rangle$, the abstract syntax signature extends *Logic* by Σ and subsort inclusions and retracts, as well as by $eq: s \times s \rightarrow *$ and $rurt: s \times s \rightarrow *$ for each sort s in Σ . So, in contrast to Example 3.12, the abstract syntax here covers rewritings between all order-sorted terms. Then, for any order-sorted algebra $A \in \mathbf{OSAlg}(\langle \Sigma, \leq \rangle)$, the evaluation structure will comprise the carriers of A extended with the undefined element \perp , operations from Σ and eq interpreted as in \mathbf{P}_{GOSA} , and operations $rurt$ interpreted as in $\mathbf{P}_{\text{GPRWL}}$ on arguments from $|A|$, but on pairs of arguments containing \perp interpreted as new “free” logical values. It is now our decision to define how to interpret rewritings between terms with undefined values. The obvious choice — though technically not the only one possible — is to identify the freely added logical values with *ff* (thus setting rewritings between undefined terms to never hold) which would complete an adequate combination of the logical systems given by \mathbf{P}_{GOSA} and $\mathbf{P}_{\text{GPRWL}}$.

The above example captures well a general situation; let’s have a closer look at the issue of when a parchment combination is “satisfactory”.

Consider a family $\mathcal{P} = \langle \mathbf{P}_i = \langle \mathbf{Sign}_i, \mathbf{L}_i, \mathbf{Mod}_i, \mathcal{G}_i \rangle \rangle_{i \in \mathcal{I}}$ of parchments. A parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ with parchment morphisms $\gamma_i: \mathbf{P} \rightarrow \mathbf{P}_i$, $i \in \mathcal{I}$, is a *complete joint extension of* \mathcal{P} if for all signatures $\Sigma \in |\mathbf{Sign}|$ and models $M \in \mathbf{Mod}(\Sigma)$, the homomorphisms $(\gamma_i^{\mathcal{G}})_{\Sigma, M}: (\mathcal{G}_i)_{\Sigma_i}(M_i) \rightarrow \mathcal{G}_{\Sigma}(M)|_{(\gamma_i^{Lan})_{\Sigma}}$, $i \in \mathcal{I}$ (where $\Sigma_i = \gamma_i^{Sig}(\Sigma)$, $M_i = (\gamma_i^{Mod})_{\Sigma}(M)$) are jointly surjective on the sort $*$. So, informally, a parchment \mathbf{P} gives a complete joint extension of a family of parchments if each logical value in \mathbf{P} corresponds to some logical value in at least one of the parchments jointly extended. If all of the parchments in \mathcal{P} are institutional, then the complete joint extension is *institutional* if \mathbf{P} is institutional and all morphisms γ_i are institutional as well.

Proposition 5.5. *If a limit in \mathcal{PAR} of a diagram of institutional parchments and institutional parchment morphisms is a complete joint extension of the parchments in the diagram, then it is a limit in \mathcal{JPAR} as well.*

Things work particularly easily when the parchment extensions involved in the diagram do not interfere with each other. To keep the presentation relatively simple, we look at pullbacks only.

We say that institutional parchment morphisms $\gamma_1: \mathbf{P}_1 \rightarrow \mathbf{P}_0$ and $\gamma_2: \mathbf{P}_2 \rightarrow \mathbf{P}_0$ in \mathcal{JPAR} do not interfere, if for any signatures $\Sigma_1 \in |\mathbf{Sign}_1|$ and $\Sigma_2 \in |\mathbf{Sign}_2|$ such that $\gamma_1^{Sig}(\Sigma_1) = \gamma_2^{Sig}(\Sigma_2) = \Sigma_0$, we have that the term algebra over the pushout (in \mathbf{FOSig}_*) signature of $(\gamma_1^{Lan})_{\Sigma_1}: \mathbf{L}_0(\Sigma_0) \rightarrow \mathbf{L}_1(\Sigma_1)$ and $(\gamma_2^{Lan})_{\Sigma_2}: \mathbf{L}_0(\Sigma_0) \rightarrow \mathbf{L}_2(\Sigma_2)$ has as the carrier of sort $*$ the pushout in \mathbf{Set} of $!_{\Sigma_1}: T_{\mathbf{L}_0(\Sigma_0)} \rightarrow T_{\mathbf{L}_1(\Sigma_1)}|_{(\gamma_1^{Lan})_{\Sigma_1}}$ and $!_{\Sigma_2}: T_{\mathbf{L}_0(\Sigma_0)} \rightarrow T_{\mathbf{L}_2(\Sigma_2)}|_{(\gamma_2^{Lan})_{\Sigma_2}}$ restricted to the functions on the carriers of sort $*$.

Informally, this condition captures the fact that the new syntactic constructs added in \mathbf{P}_1 and \mathbf{P}_2 , respectively, do not interact with each other to build new sentences that would not come from either \mathbf{P}_1 or \mathbf{P}_2 . In particular, it requires both parchments to be atomic (except for some degenerate cases). It is rather obvious that in such a situation we can put the two parchments together without further ado:

Proposition 5.6. *If two morphisms $\gamma_1: \mathbf{P}_1 \rightarrow \mathbf{P}_0$ and $\gamma_2: \mathbf{P}_2 \rightarrow \mathbf{P}_0$ in \mathcal{JPAR} do not interfere then their pullback in \mathcal{PAR} is also a pullback in \mathcal{JPAR} . Moreover the functor $\mathcal{J}: \mathcal{JPAR} \rightarrow \mathcal{JNS}$ maps this pullback to a pullback in \mathcal{JNS} .*

Example 5.7. Define a parchment \mathbf{P}_{GRWL} as the pullback of \mathbf{P}_{GMSA} and \mathbf{P}_{GPRWL} over \mathbf{P}_A (via the morphisms sketched in Example 4.6). It is easy to see that the two parchment morphisms do not interfere, and the pullback presents the pullback of the corresponding institutions given in Example 3.10; in particular, \mathbf{P}_{GRWL} presents GRWL.

Example 5.8. Similarly, \mathbf{P}_{GHA} and \mathbf{P}_{GMSA} over \mathbf{P}_A (via the morphisms of Example 4.6) do not interfere. Their pullback presents the institution sketched in Example 3.11, where standard ground equations and ground behavioural equations coexist.

Before we return to the general case of an arbitrary combination of institutional parchments, let's have a look at a simpler situation, when given a parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$, we want to add to it some new syntactic constructs, as captured by a natural transformation $\alpha: \mathbf{L} \rightarrow \mathbf{L}'$ between functors from \mathbf{Sign} to \mathbf{FOSig}_* . We may now build another parchment $\mathcal{F}_\alpha(\mathbf{P}) = \langle \mathbf{Sign}, \mathbf{L}', \mathbf{Mod}, \mathcal{G}' \rangle$, with the same signatures and models as \mathbf{P} , with the richer abstract syntax signatures given by \mathbf{L}' , with the evaluation structures that freely extend the evaluation structures of \mathbf{P} , i.e., for $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, $\mathcal{G}'_\Sigma(M) = F_{\alpha_\Sigma}(\mathcal{G}_\Sigma(M))$, and with the mediating homomorphisms defined as follows. For $\sigma: \Sigma_1 \rightarrow \Sigma_2$ and $M_2 \in \mathbf{Mod}(\Sigma_2)$, with $M_2|_\sigma = M_1$, we have morphisms $\mathcal{G}_\sigma(M_2): \mathcal{G}_{\Sigma_1}(M_1) \rightarrow \mathcal{G}_{\Sigma_2}(M_2)|_{\mathbf{L}(\sigma)}$ and $\eta_{\alpha_{\Sigma_2}}: \mathcal{G}_{\Sigma_2}(M_2) \rightarrow$

$F_{\alpha_{\Sigma_2}}(\mathcal{G}_{\Sigma_2}(M'))|_{\alpha_{\Sigma_2}} = \mathcal{G}'_{\Sigma_2}(M_2)|_{\alpha_{\Sigma_2}}$, which yield $\mathcal{G}_\sigma(M_2); \eta_{\alpha_{\Sigma_2}}|_{\mathbf{L}(\sigma)}: \mathcal{G}_{\Sigma_1}(M_1) \rightarrow (\mathcal{G}'_{\Sigma_2}(M_2)|_{\alpha_{\Sigma_2}})|_{\mathbf{L}(\sigma)} = (\mathcal{G}'_{\Sigma_2}(M_2)|_{\mathbf{L}'(\sigma)})|_{\alpha_{\Sigma_1}}$. Since $\mathcal{G}'_{\Sigma_1}(M_1) = F_{\alpha_{\Sigma_1}}(\mathcal{G}_{\Sigma_1}(M_1))$, we can now define $\mathcal{G}'_\sigma(M_2): \mathcal{G}'_{\Sigma_1}(M_1) \rightarrow \mathcal{G}'_{\Sigma_2}(M_2)|_{\mathbf{L}'(\sigma)}$ as the unique morphism such that $\eta_{\alpha_{\Sigma_1}}; \mathcal{G}'_\sigma(M_2)|_{\alpha_{\Sigma_1}} = \mathcal{G}_\sigma(M_2); \eta_{\alpha_{\Sigma_2}}|_{\mathbf{L}(\sigma)}$. It is routine now to verify further compatibility condition, so that we get:

Proposition 5.9. *Given any parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ and natural transformation $\alpha: \mathbf{L} \rightarrow \mathbf{L}'$, $\mathcal{F}_\alpha(\mathbf{P}) = \langle \mathbf{Sign}, \mathbf{L}', \mathbf{Mod}, \mathcal{G}' \rangle$ as defined above is a parchment with an institutional parchment morphism $\gamma_\alpha = \langle Id_{\mathbf{Sign}}, \alpha, Id_{\mathbf{Mod}}, \gamma_\alpha^{\mathcal{G}} \rangle$ from $\mathcal{F}_\alpha(\mathbf{P})$ to \mathbf{P} , where $Id_{\mathbf{Sign}}$ is the identity functor, $Id_{\mathbf{Mod}}$ is the identity natural transformation, and for $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, $(\gamma_\alpha^{\mathcal{G}})_{\Sigma, M} = (\eta_{\alpha_\Sigma})_M$.*

In general, $\mathcal{F}_\alpha(\mathbf{P})$ need not be institutional, even if \mathbf{P} is so. The problem is similar to that indicated for Prop. 5.2: new logical values freely added over one signature may become identified with some old logical values over another signature, and if $\mathbf{D}: *$ holds for those, the resulting mediating homomorphism is not closed. For typical extensions this does not happen though.

A natural transformation $\alpha: \mathbf{L} \rightarrow \mathbf{L}'$ (between functors from \mathbf{Sign} to \mathbf{FOSig}_*) is *clean* if new parts of syntax are never identified with old parts of syntax, i.e., for any signature morphism $\sigma: \Sigma_1 \rightarrow \Sigma_2$, for any symbol x (sort, operation or predicate name) in $\mathbf{L}'(\Sigma_1)$ that is not in the image of $\alpha_{\Sigma_1}: \mathbf{L}(\Sigma_1) \rightarrow \mathbf{L}'(\Sigma_1)$, the symbol $\mathbf{L}'(\sigma)(x)$ is not in the image of $\alpha_{\Sigma_2}: \mathbf{L}(\Sigma_2) \rightarrow \mathbf{L}'(\Sigma_2)$.

Proposition 5.10. *Given any institutional parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ and clean natural transformation $\alpha: \mathbf{L} \rightarrow \mathbf{L}'$, $\mathcal{F}_\alpha(\mathbf{P}) = \langle \mathbf{Sign}, \mathbf{L}', \mathbf{Mod}, \mathcal{G}' \rangle$ as defined above is an institutional parchment.*

This is promising, but we have not ensured that $\mathcal{F}_\alpha(\mathbf{P})$ is a *complete* (joint) extension of \mathbf{P} — there may be, and typically there are, new logical values of sort $*$ freely added by the construction above. To complete the extension, we need to identify these new logical values with some old ones, used already in \mathbf{P} . To carry this out, another concept is useful.

Given a parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$, a *coherent family of semantic congruences for \mathbf{P}* is a family $\langle \cong_{\Sigma, M} \rangle_{\Sigma \in |\mathbf{Sign}|, M \in \mathbf{Mod}(\Sigma)}$, where for $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, $\cong_{\Sigma, M}$ is a congruence on $\mathcal{G}_\Sigma(M)$ that is preserved by the mediating homomorphisms, i.e., for any signature morphism $\sigma: \Sigma \rightarrow \Sigma'$ and $M' \in \mathbf{Mod}(\Sigma')$ with $M'|_\sigma = M$, we have $\mathcal{G}_\sigma(M')(\cong_{\Sigma, M}) \subseteq \cong_{\Sigma', M'}|_{\mathbf{L}(\sigma)}$. Given such a family, we may build another parchment $\mathbf{P}/\cong = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G}^\cong \rangle$, where for $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, $\mathcal{G}_\Sigma^\cong(M) = \mathcal{G}_\Sigma(M)/\cong_{\Sigma, M}$ and for $\sigma: \Sigma \rightarrow \Sigma'$ and $M' \in \mathbf{Mod}(\Sigma')$ with $M'|_\sigma = M$, $\mathcal{G}_\sigma^\cong(M'): \mathcal{G}_\Sigma^\cong(M) \rightarrow \mathcal{G}_{\Sigma'}^\cong(M')|_{\mathbf{L}(\sigma)}$ is defined by $\mathcal{G}_\sigma^\cong(M')([a]_{\cong_{\Sigma, M}}) = [\mathcal{G}_\sigma(M')(a)]_{\cong_{\Sigma', M'}}$ (the coherence condition ensures that this is well-defined).

Proposition 5.11. *Given any parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ and coherent family \cong of semantic congruences for \mathbf{P} , $\mathbf{P}/\cong = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G}^\cong \rangle$ as defined above is a parchment, with a parchment morphism $\gamma_\cong = \langle Id_{\mathbf{Sign}}, Id_{\mathbf{L}}, Id_{\mathbf{Mod}}, \gamma_\cong^{\mathcal{G}} \rangle$ from \mathbf{P}/\cong to \mathbf{P} , where $Id_{\mathbf{Sign}}$ is the identity functor, $Id_{\mathbf{L}}$ and $Id_{\mathbf{Mod}}$ are the*

identity natural transformations, and for any $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, $(\gamma_{\cong}^{\mathcal{G}})_{\Sigma, M} = [-]_{\cong_{\Sigma, M}}$.

The construction above simplifies considerably when the parchment is atomic: instead of considering congruences on the evaluation structures, it is sufficient to consider equivalence relations on the carriers of sort $*$ of these structures (which together with identities on other sorts then form congruences).

Now, given a family $\mathcal{P} = \langle \mathbf{P}_i = \langle \mathbf{Sign}_i, \mathbf{L}_i, \mathbf{Mod}_i, \mathcal{G}_i \rangle \rangle_{i \in \mathcal{I}}$ of parchments, consider a parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ with parchment morphisms $\gamma_i: \mathbf{P} \rightarrow \mathbf{P}_i$, $i \in \mathcal{I}$. A coherent family \cong of semantic congruences for \mathbf{P} is *complete* for \mathcal{P} , if for any signature $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, for any $a \in |\mathcal{G}_{\Sigma}(M)|_*$, for some $i \in \mathcal{I}$ and $a_i \in |(\mathcal{G}_i)_{\gamma_i^{sig}(\Sigma)}((\gamma_i^{Mod})_{\Sigma}(M))|_*$, we have $a \cong_{\Sigma, M} (\gamma_i^{\mathcal{G}})_{\Sigma}(M)(a_i)$.

Proposition 5.12. *Consider any family $\mathcal{P} = \langle \mathbf{P}_i = \langle \mathbf{Sign}_i, \mathbf{L}_i, \mathbf{Mod}_i, \mathcal{G}_i \rangle \rangle_{i \in \mathcal{I}}$ of parchments, parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ with parchment morphisms $\gamma_i: \mathbf{P} \rightarrow \mathbf{P}_i$, $i \in \mathcal{I}$, and coherent family \cong of semantic congruences for \mathbf{P} that is complete for \mathcal{P} . Then the parchment \mathbf{P}/\cong with parchment morphisms $\gamma_{\cong}; \gamma_i: \mathbf{P}/\cong \rightarrow \mathbf{P}_i$, $i \in \mathcal{I}$, is a complete joint extension for the family \mathcal{P} .*

Furthermore, a coherent family \cong of semantic congruences for \mathbf{P} is *institutional* for \mathcal{P} , if for any signature $\Sigma \in |\mathbf{Sign}|$ and $M \in \mathbf{Mod}(\Sigma)$, whenever for any $i, j \in \mathcal{I}$, with $\gamma_i^{sig}(\Sigma) = \Sigma_i$, $\gamma_j^{sig}(\Sigma) = \Sigma_j$, $(\gamma_i^{Mod})_{\Sigma}(M) = M_i$ and $(\gamma_j^{Mod})_{\Sigma}(M) = M_j$, we have $a_i \in |(\mathcal{G}_i)_{\Sigma_i}(M_i)|_*$, $a_j \in |(\mathcal{G}_j)_{\Sigma_j}(M_j)|_*$, $(\gamma_i^{\mathcal{G}})_{\Sigma, M}(a_i) \cong_{\Sigma, M} (\gamma_j^{\mathcal{G}})_{\Sigma, M}(a_j)$ and $a_i \in \mathbf{D}_{(\mathcal{G}_i)_{\Sigma_i}(M_i)}$, then $a_j \in \mathbf{D}_{(\mathcal{G}_j)_{\Sigma_j}(M_j)}$ as well. Informally: we can glue together only those “old” logical values that either both designate sentences to hold, or both designate them not to hold.

Theorem 5.13. *Consider any family $\mathcal{P} = \langle \mathbf{P}_i = \langle \mathbf{Sign}_i, \mathbf{L}_i, \mathbf{Mod}_i, \mathcal{G}_i \rangle \rangle_{i \in \mathcal{I}}$ of institutional parchments, parchment $\mathbf{P} = \langle \mathbf{Sign}, \mathbf{L}, \mathbf{Mod}, \mathcal{G} \rangle$ with institutional parchment morphisms $\gamma_i: \mathbf{P} \rightarrow \mathbf{P}_i$, $i \in \mathcal{I}$, and coherent family \cong of semantic congruences for \mathbf{P} that is complete and institutional for \mathcal{P} . Then the parchment \mathbf{P}/\cong with parchment morphisms $\gamma_{\cong}; \gamma_i: \mathbf{P}/\cong \rightarrow \mathbf{P}_i$, $i \in \mathcal{I}$, is a complete institutional joint extension for the family \mathcal{P} .*

One strength of the above result is that we show the quotient parchment to be institutional without assuming that \mathbf{P} is so. This follows since the parchments in the family are institutional, and the coherent family of congruences is complete and institutional as well.

Example 5.14. Consider a pullback of \mathbf{P}_{GOSA} and \mathbf{P}_{GRWL} over \mathbf{P}_{GMSA} via the morphisms given by Example 4.7 and the pullback construction of \mathbf{P}_{GRWL} in Example 5.7, respectively. In fact, the pullback parchment is the same as the pullback parchment for \mathbf{P}_{GOSA} and $\mathbf{P}_{\text{GPRWL}}$ over \mathbf{P}_{A} described in Example 5.4. The problem is that it is not a complete joint extension of \mathbf{P}_{GOSA} and \mathbf{P}_{GRWL} , as evaluation structures carry freely added logical values, corresponding to rewriting statements between terms with undefined values. To fix this, consider a family of equivalences on the carriers of sort $*$ that glue values of the operations *rurt*

on pairs of arguments containing \perp with ff . Since the parchment is atomic, this family extends to a family of congruences by adding identities on the carriers of other sorts. It is easy to check now that this family is coherent as well as complete and institutional for \mathbf{P}_{GOSA} and \mathbf{P}_{GRWL} . Consequently, by Thm. 5.13, quotienting the pullback parchment by this family yields an institutional complete joint extension of \mathbf{P}_{GOSA} and \mathbf{P}_{GRWL} — this is an institutional parchment $\mathbf{P}_{\text{GOSRWL}}$ that presents the institution GOSRWL , and the institutional parchment morphisms from $\mathbf{P}_{\text{GOSRWL}}$ to \mathbf{P}_{GOSA} and \mathbf{P}_{GRWL} , respectively, present the corresponding institution morphisms in the CafeOBJ cube.

Example 5.15. Consider now a pullback $\mathbf{P}_0 = \langle \mathbf{Sign}_0, \mathbf{L}_0, \mathbf{Mod}_0, \mathcal{G}_0 \rangle$ of \mathbf{P}_{GHA} and \mathbf{P}_{GRWL} over \mathbf{P}_{GMSA} via the parchment morphisms given by Example 4.8 and the pullback construction of \mathbf{P}_{GRWL} in Example 5.7, respectively. As in Example 3.13, \mathbf{Sign}_0 is the category of behavioural signatures, and $\mathbf{Mod}_0(\langle \Sigma, \text{OBS} \rangle)$ is the class of Σ -algebras with a rewriting preorder $\preceq_o \subseteq |A|_o \times |A|_o$ on observable sorts $o \in \text{OBS}$ only, preserved by observable operations. For any behavioural signature $\langle \Sigma, \text{OBS} \rangle$, the abstract syntax signature $\mathbf{L}_0(\langle \Sigma, \text{OBS} \rangle)$ extends *Logic* by Σ , operations $beq: s \times s \rightarrow *$ for all sorts s in Σ , and operations $brwt: o \times o \rightarrow *$ for observable sorts $o \in \text{OBS}$. Perhaps surprisingly, \mathbf{P}_{GHA} and \mathbf{P}_{GRWL} over \mathbf{P}_{GMSA} do not interfere, and so \mathbf{P}_0 is a pullback of \mathbf{P}_{GHA} and \mathbf{P}_{GRWL} over \mathbf{P}_{GMSA} in \mathcal{JPAR} , and in fact is their complete institutional joint extension. But we still “miss” rewritings on non-observable sorts!

So, let us add them: consider the natural inclusion $\alpha: \mathbf{L}_0 \rightarrow \mathbf{L}^{\text{GHRWL}}$, for any behavioural signature $\langle \Sigma, \text{OBS} \rangle$, $\mathbf{L}^{\text{GHRWL}}(\langle \Sigma, \text{OBS} \rangle)$ adding to $\mathbf{L}_0(\langle \Sigma, \text{OBS} \rangle)$ operations $brwt: s \times s \rightarrow *$ for non-observable sorts $s \notin \text{OBS}$. Now, by Prop. 5.9, we obtain the parchment $F_\alpha(\mathbf{P}_0)$, which is institutional by Prop. 5.10. However, it is not a complete joint extension of \mathbf{P}_{GHA} and \mathbf{P}_{GRWL} , with new logical values added for behavioural rewritings between terms of non-observable sorts. Of course, it is now our decision how to interpret such rewritings.

For any signature $\langle \Sigma, \text{OBS} \rangle$ and $\langle A, \langle \preceq_o \rangle_{o \in \text{OBS}} \rangle \in \mathbf{Mod}_0(\langle \Sigma, \text{OBS} \rangle)$, let $\preceq \subseteq |A| \times |A|$ be the largest precongruence on A such that $\preceq_o \subseteq \preceq$ for all observable sorts $o \in \text{OBS}$.¹⁵ Now, consider a family of equivalences on the carriers of sort $*$ of the evaluation structures in $F_\alpha(\mathbf{P}_0)$ that glue values of the operations $brwt$ on arguments a, b with tt if $a \preceq b$ and with ff otherwise. Since the parchment is atomic, this family extends to a family of congruences by adding identities on the carriers of other sorts. Given the conditions on behavioural signature morphisms, it is easy to check now that this family is coherent as well as complete and institutional for \mathbf{P}_{GHA} and \mathbf{P}_{GRWL} . Consequently, by Thm. 5.13, quotienting $F_\alpha(\mathbf{P}_0)$ by this family yields an institutional complete joint extension of \mathbf{P}_{GHA} and \mathbf{P}_{GRWL} — this is an institutional parchment $\mathbf{P}_{\text{GHRWL}}$ that presents institution GHRWL , and the institutional parchment morphisms from $\mathbf{P}_{\text{GHRWL}}$ to \mathbf{P}_{GHA} and \mathbf{P}_{GRWL} , respectively, present the corresponding institution morphisms in the CafeOBJ cube.

¹⁵ So that $a \preceq b$ iff, relying on the standard concepts and notation, for all contexts C of an observable sort, $C_A[a] \preceq C_A[b]$.

Example 5.16. Consider now a pullback of \mathbf{P}_{GHA} and \mathbf{P}_{GOSA} over \mathbf{P}_{GMSA} via the morphisms of Examples 4.8 and 4.7, respectively. Somewhat similarly to the initial construction in Example 5.15, a signature in the resulting parchment is a behavioural signature with ordering on the set of sorts that is non-trivial on observable sorts only, and models over such signature are order-sorted algebras over the obvious order-sorted signature extracted from it. For any such signature $\langle \Sigma, \text{OBS}, \leq \rangle$, the abstract syntax signature extends *Logic* by Σ , operations $\text{beg}: s \times s \rightarrow *$ for all sorts s in Σ , and subsort inclusions and retracts as determined by the subsorting relation on the observable sorts. No need to discuss evaluation structures — they are given by the obvious amalgamation of the evaluation structures in \mathbf{P}_{GHA} and \mathbf{P}_{GOSA} .

For the purposes of this presentation we stop at this point and set this parchment to be $\mathbf{P}_{\text{GHOSA}}$, presenting an institution that corresponds to GHOSA . Note though that we thus neglect adding subsorting on non-observable sorts — this could be done much in the style of adding rewritings on non-observable sorts in Example 5.15, except that we would need a slightly more general form of Prop. 5.9 and Thm. 5.13, with extension of signatures (and models) permitted.

Example 5.17. Finally, let $\mathbf{P}_{\text{GHOSRWL}}$ be a pullback of $\mathbf{P}_{\text{GHRWL}}$ and $\mathbf{P}_{\text{GOSRWL}}$ over \mathbf{P}_{GRWL} via the morphisms constructed in Examples 5.15 and 5.14, respectively. Equivalently, $\mathbf{P}_{\text{GHOSRWL}}$ is the limit of the parchments and their morphisms constructed so far. It is a complete joint extension of the parchments considered so far, and so by Prop. 5.5, it is the limit in \mathcal{JPAR} of the diagram constructed so far. $\mathbf{P}_{\text{GHOSRWL}}$ presents an institution that corresponds to GHOSRWL in the cube, inheriting the comments on the lack of subsorting for non-observable sorts from Example 5.16.

6 Final remarks

In this paper we study the problems of systematic combination of logical systems in the framework of the theory of institutions and their presentations as parchments. To begin with, we recall the notion of institution and institution morphism [GB92], and the construction of limits in the category they form [Tar86]. Then we introduce a new notion of model-theoretic parchment, modifying the original notions defined in [GB86] and [MTP98]. We sketch again how limits in the category of such parchments are built, and argue that they do not always offer a satisfactory way of putting logical systems together. We present a new understanding of this phenomena via Props. 5.2 and 5.3, and the new notion of a complete joint extension of a family of parchments. We suggest some simple situations when the use of limits yields a desired result, as for instance captured by Prop. 5.6. We also develop constructions that adjust such limits to a more desired form, Props. 5.9, 5.11 and Thm. 5.13.

All these developments are extensively illustrated by referring to various logical systems that underlie CafeOBJ [DF02]. We start from simple parchments that capture equational logic, order-sorted equational logic, behavioural equational

logic and rewriting logic, respectively, and show how to systematically combine and modify them to obtain the remaining logical systems of the `CafeOBJ` cube.

To keep the presentation relatively simple and hopefully understandable, in places we depart from the details of the logical systems as used in `CafeOBJ`. In particular, we deal with their ground versions only (no variables). Adding variables would be simple: in essence, we would have to multiply the sorts in the abstract syntax signatures by the sets of variables considered, and to parametrise non-logical values in the evaluation structures by valuations of variables, as is done in [Mos96]. We foresee no major difficulties with this, but it is worth spelling out the details, of course. Another departure from the logics of `CafeOBJ` is elimination of conditions in statements – adding those should pose no difficulties whatsoever, although the abstract syntax signatures and the evaluation structures would again become somewhat more complex. We also simplify the view of behavioural satisfaction, by using a set of observable sorts rather than a designated set of observer operations. The changes required to capture the more refined view of `CafeOBJ` are rather obvious as well. To deal with order-sorted algebra, we introduce explicit subsort inclusions and (partial) retracts, again somewhat departing from what is sketched in [DF02]. The combination of behavioural equations with subsorting, omitted here, requires further careful study in our view, perhaps building on [BD94].

To keep the paper to a reasonable size, we entirely omitted notions of comorphisms for institutions and parchments, even though there seems to have been a shift in the presentation of the `CafeOBJ` cube from institution morphisms to comorphisms [Dia11]. In the case of the logical systems considered here, these are simple, as all the morphisms in use are based on signature functors having left adjoints — and in such cases well-known results about duality between institution morphisms and comorphisms [FC96] carry over to parchments as well. In general, however, the use of comorphisms in this context is not immediate. First, there are formal problems: for instance, the category of institutions and their comorphisms is not cocomplete due to foundational reasons, and some size limitations have to be imposed on the institutions considered. Second, and perhaps more to the point, comorphisms capture a different intuition concerning the relationship between the institutions they link. Informally, while institution morphisms indicate how a richer logical system is built over a simpler one, institution comorphisms show how a simpler logical system is encoded in a richer one. Consequently, it is not obvious at all that comorphisms offer a proper technical framework for the modular construction of logical systems we aim at here. We leave this as a worthwhile topic for further investigation though, as comorphisms open the way to the study of parchment representations in universal logics we began in [MTP98], and link to other frameworks based on heterogeneous logical environments like HETS [MML07] and LATIN [CHK⁺11], which admit logic definitions in a modular manner [CHK⁺12].

An interesting, far-reaching and difficult problem is how to capture in our framework the operational ideas that underlie the `CafeOBJ` implementation and are closely linked with the logical systems involved.

Acknowledgements Thanks to the anonymous referees for valuable feedback.

References

- [BD94] Rod Burstall and Răzvan Diaconescu. Hiding and behaviour: an institutional approach. In: A. William Roscoe, editor, *A Classical Mind: Essays in Honour of C.A.R. Hoare*, pages 75–92. Prentice-Hall, 1994.
- [BG80] R.M. Burstall and J.A. Goguen. The semantics of Clear, a specification language. In: Dines Bjørner, editor, *Proceedings of the 1979 Copenhagen Winter School on Abstract Software Specification, Lecture Notes in Computer Science*, Vol. 86, pages 292–332. Springer, 1980.
- [BH06] Michel Bidoit and Rolf Hennicker. Constructor-based observational logic. *Journal of Logic and Algebraic Programming*, 67(1–2):3–51, 2006.
- [CGR03] Carlos Caleiro, Paula Gouveia, and Jaime Ramos. Completeness results for fibred parchments: Beyond the propositional base. In: Martin Wirsing, Dirk Pattinson, and Rolf Hennicker, editors, *Recent Trends in Algebraic Development Techniques. Selected Papers from the 16th International Workshop on Algebraic Development Techniques*, Frauenchiemsee, *Lecture Notes in Computer Science*, Vol. 2755, pages 185–200. Springer, 2003.
- [CHK⁺11] Mihai Codescu, Fulya Horozal, Michael Kohlhase, Till Mossakowski, and Florian Rabe. Project abstract: Logic atlas and integrator (LATIN). In: James H. Davenport, William M. Farmer, Josef Urban, and Florian Rabe, editors, *Intelligent Computer Mathematics 18th Symposium, Calculemus 2011, and 10th International Conference, MKM 2011, Bertinoro, Italy, July 18-23, 2011. Proceedings, Lecture Notes in Computer Science*, Vol. 6824, pages 289–291. Springer-Verlag Berlin Heidelberg, 2011.
- [CHK⁺12] Mihai Codescu, Fulya Horozal, Michael Kohlhase, Till Mossakowski, Florian Rabe, and Kristina Sojakova. Towards logical frameworks in the heterogeneous tool set HETS. In: Till Mossakowski and Hans-Jörg Kreowski, editors, *Recent Trends in Algebraic Development Techniques, 20th International Workshop, WADT 2010, Lecture Notes in Computer Science*, Vol. 7137, pages 139–159. Springer, 2012.
- [CMRS01] Carlos Caleiro, Paulo Mateus, Jaime Ramos, and Amílcar Sernadas. Combining logics: Parchments revisited. In: Maura Cerioli and Gianna Reggio, editors, *Recent Trends in Algebraic Development Techniques. Selected Papers from the 15th Workshop on Algebraic Development Techniques joint with the CoFI WG Meeting*, Genova, *Lecture Notes in Computer Science*, Vol. 2267, pages 48–70. Springer, 2001.
- [DF98] Răzvan Diaconescu and Kokichi Futatsugi. *CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification*, *AMAST Series in Computing*, Vol. 6. World Scientific, 1998. See also <http://www.ldl.jaist.ac.jp/cafeobj/>.
- [DF02] Răzvan Diaconescu and Kokichi Futatsugi. Logical foundations of CafeOBJ. *Theoretical Computer Science*, 285:289–318, 2002.
- [Dia02] Răzvan Diaconescu. Grothendieck institutions. *Applied Categorical Structures*, 10(4):383–402, 2002.
- [Dia07] Răzvan Diaconescu. A methodological guide to the CafeOBJ logic. In: D. Bjørner and M.C. Henson, editors, *Logics of Specification Languages*, *Monographs in Theoretical Computer Science*, pages 153–240. Springer-Verlag Berlin Heidelberg, 2007.

- [Dia08] Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.
- [Dia11] Răzvan Diaconescu. Grothendieck inclusion systems. *Applied Categorical Structures*, 19(5):783–802, 2011.
- [FC96] José Luiz Fiadeiro and José Félix Costa. Mirror, mirror in my hand: A duality between specifications and models of process behaviour. *Mathematical Structures in Computer Science*, 6(4):353–373, 1996.
- [GB86] Joseph A. Goguen and Rod M. Burstall. A study in the functions of programming methodology: Specifications, institutions, charters and parchments. In: David H. Pitt, Samson Abramsky, Axel Poigné, and David E. Rydeheard, editors, *Proceedings of the Tutorial and Workshop on Category Theory and Computer Programming*, Guildford, *Lecture Notes in Computer Science*, Vol. 240, pages 313–333. Springer, 1986.
- [GB92] J.A. Goguen and R.M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.
- [GM92] Joseph Goguen and José Meseguer. Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. *Theoretical Computer Science*, 105(2):217–273, 1992.
- [GR02] Joseph A. Goguen and Grigore Roşu. Institution morphisms. *Formal Aspects of Computing*, 13(3–5):274–307, 2002.
- [GWM⁺00] Joseph Goguen, Timothy Winkler, José Meseguer, Kokichi Futatsugi, and Jean-Pierre Jouannaud. Introducing OBJ3. In: Joseph Goguen and Grant Malcolm, editors, *Software Engineering with OBJ: Algebraic Specification in Action*. Kluwer, 2000.
- [Mac71] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, 1971.
- [Mes98] José Meseguer. Membership algebra as a logical framework for equational specification. In: Francesco Parisi-Presicce, editor, *Recent Trends in Data Type Specification. Selected Papers from the 12th International Workshop on Specification of Abstract Data Types*, Tarquinia, *Lecture Notes in Computer Science*, Vol. 1376, pages 18–61. Springer, 1998.
- [MML07] Till Mossakowski, Christian Maeder, and Klaus Lüttich. The heterogeneous tool set, HETS. In: Orna Grumberg and Michael Huth, editors, *Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. European Joint Conferences on Theory and Practice of Software (ETAPS 2007)*, Braga, *Lecture Notes in Computer Science*, Vol. 4424, pages 519–522. Springer, 2007. See also <http://www.informatik.uni-bremen.de/cofi/hets/>.
- [Mos96] T. Mossakowski. Using limits of parchments to systematically construct institutions of partial algebras. In: M. Haverdaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types, Lecture Notes in Computer Science*, Vol. 1130, pages 379–393. Springer Verlag, London, 1996.
- [Mos03] Till Mossakowski. Foundations of heterogeneous specification. In: Martin Wirsing, Dirk Pattinson, and Rolf Hennicker, editors, *Recent Trends in Algebraic Development Techniques. Selected Papers from the 16th International Workshop on Algebraic Development Techniques*, Frauenchiemsee, *Lecture Notes in Computer Science*, Vol. 2755, pages 359–375. Springer, 2003.

- [Mos05] Till Mossakowski. *Heterogeneous Specification and the Heterogeneous Tool Set*. Habilitation thesis, Universität Bremen, 2005.
- [MT09] Till Mossakowski and Andrzej Tarlecki. Heterogeneous logical environments for distributed specifications. In: Andrea Corradini and Ugo Montanari, editors, *Recent Trends in Algebraic Development Techniques. Selected Papers from the 19th International Workshop on Algebraic Development Techniques*, Pisa, *Lecture Notes in Computer Science*, Vol. 5486, pages 266–289. Springer, 2009.
- [MTP97] Till Mossakowski, Andrzej Tarlecki, and Wiesław Pawłowski. Combining and representing logical systems. In: Eugenio Moggi and Giuseppe Rosolini, editors, *Proceedings of the 7th International Conference on Category Theory and Computer Science*, Santa Margherita Ligure, *Lecture Notes in Computer Science*, Vol. 1290, pages 177–196. Springer, 1997.
- [MTP98] Till Mossakowski, Andrzej Tarlecki, and Wiesław Pawłowski. Combining and representing logical systems using model-theoretic parchments. In: Francesco Parisi-Presicce, editor, *Recent Trends in Data Type Specification. Selected Papers from the 12th International Workshop on Specification of Abstract Data Types*, Tarquinia, *Lecture Notes in Computer Science*, Vol. 1376, pages 349–364. Springer, 1998.
- [ST87] Donald Sannella and Andrzej Tarlecki. On observational equivalence and algebraic specification. *Journal of Computer and System Sciences*, 34:150–178, 1987.
- [ST88] Donald Sannella and Andrzej Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76(2–3):165–210, 1988.
- [ST12] Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2012.
- [SW83] Donald Sannella and Martin Wirsing. A kernel language for algebraic specification and implementation. In: Marek Karpinski, editor, *Proceedings of the 1983 International Conference on Foundations of Computation Theory*, Borgholm, *Lecture Notes in Computer Science*, Vol. 158, pages 413–427. Springer, 1983.
- [Tar86] Andrzej Tarlecki. Bits and pieces of the theory of institutions. In: David H. Pitt, Samson Abramsky, Axel Poigné, and David E. Rydeheard, editors, *Proceedings of the Tutorial and Workshop on Category Theory and Computer Programming*, Guildford, *Lecture Notes in Computer Science*, Vol. 240, pages 334–360. Springer, 1986.
- [Tar96] Andrzej Tarlecki. Moving between logical systems. In: Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Recent Trends in Data Type Specification. Selected Papers from the 11th Workshop on Specification of Abstract Data Types*, Oslo, *Lecture Notes in Computer Science*, Vol. 1130, pages 478–502. Springer, 1996.
- [Tar00] Andrzej Tarlecki. Towards heterogeneous specifications. In: Dov Gabbay and Maarten de Rijke, editors, *Frontiers of Combining Systems 2*, Studies in Logic and Computation, pages 337–360. Research Studies Press, 2000.
- [TBG91] Andrzej Tarlecki, Rod M. Burstall, and Joseph A. Goguen. Some fundamental algebraic tools for the semantics of computation. Part 3: Indexed categories. *Theoretical Computer Science*, 91(2):239–264, 1991.