What Is a Derived Signature Morphism?

Till Mossakowski[1(✉)], Ulf Krumnack[2], and Tom Maibaum[3]

[1] Otto-von-Guericke University of Magdeburg, Magdeburg, Germany
till@iws.cs.ovgu.de
[2] University of Osnabrück, Osnabrück, Germany
[3] McMaster University, Hamilton, Canada

Abstract. The notion of signature morphism is basic to the theory of institutions. It provides a powerful primitive for the study of specifications, their modularity and their relations in an abstract setting. The notion of *derived* signature morphism generalises signature morphisms to more complex constructions, where symbols may be mapped not only to symbols, but to arbitrary terms. The purpose of this work is to study derived signature morphisms in an institution-independent way. We will recall and generalize two known approaches to derived signature morphisms, introduce a third one, and discuss their pros and cons. We especially study the existence of colimits of derived signature morphisms. The motivation is to give an independent semantics to the notion of derived signature morphism, query and substitution in the context of the Distributed Ontology, Modeling and Specification Language DOL.

1 Introduction

The notion of signature morphism is basic to the theory of institutions. It provides a powerful primitive for the study of specifications, their modularity and their relations in an abstract setting. The notion of *derived* signature morphism generalises signature morphisms to more complex constructions, where symbols may be mapped not only to symbols, but to arbitrary terms. Derived signature morphisms have been introduced in [15] and studied in [5,6,16,20,21]. Recently, the notion of derived signature morphism has gained attention in the field of model-driven engineering [9], databases [8], analogies [23], and ontologies[1].

In this paper we investigate derived signature morphism and their properties. We recall and generalize two known approaches to derived signature morphisms, and introduce a third one. All current works define derived signature morphisms in specific institutions. We look for a way to formulate the concept in an *institution-independent* way. Especially we look for a semantics of derived signature morphisms in languages with institution-independent semantics. We also investigate the question to what extent we can *combine* systems along derived signature morphisms (via *colimits*).

[1] Cmp. the work on the new OMG standard. *Distributed Ontology, Modeling and Specification Language (DOL)*, see http://ontoiop.org.

M. Codescu et al. (Eds.): WADT 2014, LNCS 9463, pp. 90–109, 2015.
DOI: 10.1007/978-3-319-28114-8_6

The paper is structured as follows: Sect. 2 introduces examples from different fields. In Sect. 3 we briefly summarise some relevant notions from institution theory. The first approach to derived signature morphisms is to consider them to be ordinary signature morphisms into a definitional extension (Sect. 4). The second approach is to consider derived signature morphisms to be abstract substitutions that induce mappings on syntactic and semantic level (Sect. 5). The third approach is to consider institutional monads, which have derived signature morphisms as signature morphisms in their Kleisli institution (Sect. 5). We finish by discussing pros and cons and collecting open questions.

2 Examples

In specification theory derived signature morphisms may map between equivalent representations:

Example 1 (Boolean rings and algebras). It is well known, that Boolean rings and algebras are essentially the same thing. However, a mapping between these specifications has to cope with the fact, that the algebraic $\vee$ is an inclusive disjunction while the ring addition is an exclusive disjunction:

```
interpretation i : BooleanAlgebra to BooleanRing =
     ∧ ↦ λx,y.x·y
     ∨ ↦ λx,y.x+y+x·y
     ¬ ↦ λx.1+x
end

interpretation j : BooleanRing to BooleanAlgebra =
     · ↦ λx,y.x∧y
     + ↦ λx,y.(x∨y)∧¬(x∧y)
end
```

Note that operation symbols are mapped to λ-terms. The λ-variables open a context of variables for the subsequent terms. The number of λ-variables (or, for sorted logics, their sort string) must correspond to the arity of the operation symbol. further mote that the order λ-variables: $\lambda x, y.x$ is different from $\lambda x, y.y$.

Derived signature morphisms also play an important role in model-driven engineering (MDE). A problem that appears in practice when combining multiple models is that different models specify the same information differently.

Example 2. A related field of application is databases. Suppose we have two databases that we intend to use to store information about people, which were designed independently. We now wish to merge the information in the databases, We begin by merging the schema used to define the databases. To define the merge, we have to identify what the relationships are between the relation names and attributes of the two schema. Let the signature of DB_1 be $\langle\{Persons\}, \{Name, Gender, Age\}\rangle$, where *Persons* is the name of the database relation (table) intended to contain the information and *Name*, *Gender* and *Age* are attributes (columns) of this relation. Ditto DB_2, with signature $\langle\{\mathit{Male}\,\mathit{Female}\}, \{\mathit{Name}, \mathit{Bdate}, \mathit{Bplace}\}\rangle$, where we have two relations, *Male* and *Female*, and they both have attributes *Name*, $B(\mathit{irth})\mathit{date}$ and $B(\mathit{irth})\mathit{place}$. In order to create a suitable merge, we have to decide what matches what in the two schemas. Clearly the attribute *Gender* of DB_1 and the relations *Male* and *Female* are related, but cannot directly be matched as there is a type mismatch. If in DB_1 we define two new (derived) relations *Male* and *Female*, i.e., create a view of DB_1, both with attributes *Name* and *Age*, we will have "solved" the type mismatch problem for this aspect of the merge. These two relations can be defined as an extension of the original DB_1 schema by using an appropriate query in, say SQL. This is analogous to creating a definitional extension in FOL (see Example 5). So now we have DB_1' with the three relation names and the same attribute names.

Similarly, we can extend DB_2 with an extra attribute *Age* derived from *Bdate* via an appropriate query and obtain DB_2' with relations Male' and Female', and the extra attribute name *Age*. Now we define a span between DB_1' and DB_2', on the basis of which we can create the appropriate merged database schema by computing the colimit of the span. The database scheme DB at the apex of the span has signature $\langle \mathit{Male}, \mathit{Female}, \mathit{Name}, \mathit{Age}\rangle$. The maps connecting this scheme to DB_1' connect *Male* with *Male*, *Female* with *Female*, *Name* with *Name* and *Age* with *Age*. This is a Kleisli map between DB and DB_1, mapping DB to a definitional extension of DB_1. The maps connecting DB with DB_2' connect *Male* with Male' of DB_2', *Female* with Female', *Name* with *Name* and *Age* with *Age*. Again, this defines a Kleisli between DB and DB_2. Then, the corresponding pushout will be the "correct" merge of the two database schemas, avoiding redundancy in the merge and minimising the redundancy of data in the merged scheme.

Of course, having obtained the merged scheme, we would now want to merge the corresponding data. Database schemes correspond to theories and database instances correspond to models of theories. This framework could be used to derive the datamerge from the schema merge via amalgamation results. An alternative approach using a fibrational approach is outlined in [9].

Another field of interest are analogies. An analogy identifies common structures in the same or two different domains (source and target). In other words, an analogy basically consists of a common structural core that is instantiated in both, source and target.

$$\begin{tikzcd} & \text{Generalisation (abstract core)} \arrow[dl] \arrow[dr] & \\ \text{Source} \arrow[rr, dotted, dash, "\text{analogical relation}"] & & \text{Target} \end{tikzcd}$$

Hence, in logic an analogy can be formalised by giving a set of generalised formulas together with a pair of mappings, that map these formulas into the source and target respectively.[2] However, in many cases plain signature morphisms do not suffice to describe such a mapping, so derived signature morphisms offer a natural solution:

Example 3 (heat flow).[3] The heat flow analogy is well-known from physics education. The analogy is intended to introduce the concepts of heat and heat flow by comparing them to water and water flow. A simple description of this analogy may consist of the following observations: on the source side there are to vessels, a beaker and a vial, connected via a pipe. If the height of the water in the beaker is greater than the height of the water in the vial, water will flow and the height of the water in the beaker will decrease while the height of the water in the vial will increase. On the target side a metal bar is put into a cup of hot coffee. An ice cube is attached to the upper end of the bar. It is observed, that the coffee cools down while the ice heats up and finally melts. A logic-based representation of this description may contain the following formulas:

$$
\begin{array}{ll}
\text{(G1)} & \mathit{connected}(A, B, C) \\
\text{(G2)} & \forall t_1 : \mathit{time},\, t_2 : \mathit{time} : t_2 > t_1 \\
& \quad \wedge T(A, t_1) > T(B, t_1) \\
& \quad \rightarrow T(A, t_2) < T(A, t_1) \wedge T(B, t_2) > T(B, t_1)
\end{array}
$$

$$
\begin{array}{ll}
\text{(S1)} & \mathit{connected}(\mathit{beaker}, \mathit{vial}, \mathit{pipe}) \\
\text{(S2)} & \forall t_1 : \mathit{time},\, t_2 : \mathit{time} : t_2 > t_1 \\
& \quad \wedge\ \mathit{height}(\mathit{in}(\mathit{water}, \mathit{beaker}), t_1) \\
& \qquad > \mathit{height}(\mathit{in}(\mathit{water}, \mathit{vial}), t_1) \\
& \quad \rightarrow \mathit{height}(\mathit{in}(\mathit{water}, \mathit{beaker}), t_2) \\
& \qquad < \mathit{height}(\mathit{in}(\mathit{water}, \mathit{beaker}), t_1) \\
& \quad \wedge\ \mathit{height}(\mathit{in}(\mathit{water}, \mathit{vial}), t_2) \\
& \qquad > \mathit{height}(\mathit{in}(\mathit{water}, \mathit{vial}), t_1)
\end{array}
$$

$$
\begin{array}{ll}
\text{(T1)} & \mathit{connected}(\mathit{in}(\mathit{coffee}, \mathit{cup}), \mathit{ice\_cube}, \mathit{bar}) \\
\text{(T2)} & \forall t_1,\, t_2 : \mathit{time} : t_2 > t_1 \\
& \quad \wedge \mathit{temp}(\mathit{in}(\mathit{coffee}, \mathit{cup}), t_1) \\
& \qquad > \mathit{temp}(\mathit{ice\_cube}, t_1) \\
& \quad \rightarrow \mathit{temp}(\mathit{in}(\mathit{coffee}, \mathit{cup}), t_2) \\
& \qquad < \mathit{temp}(\mathit{in}(\mathit{coffee}, \mathit{cup}), t_1) \\
& \quad \wedge\ \mathit{temp}(\mathit{ice\_cube}, t_2) \\
& \qquad > \mathit{temp}(\mathit{ice\_cube}, t_1)
\end{array}
$$

Here it is essential, that an object on the target side is matched to a vessel on the source side (but not to the water in the vessel). Hence, the following (derived) signature morphisms should be applied:

$$
\begin{aligned}
\mathit{beaker} &\hookleftarrow A \mapsto \mathit{in}(\mathit{coffee}, \mathit{cup}) \\
\mathit{vial} &\hookleftarrow B \mapsto \mathit{ice\_cube} \\
\mathit{pipe} &\hookleftarrow C \mapsto \mathit{bar} \\
\lambda x \lambda h.\mathit{height}(\mathit{in}(\mathit{water}, x), t) &\hookleftarrow T \mapsto \lambda x \lambda h.\mathit{temp}(x, t)
\end{aligned}
$$

3 Institutions

The study of derived signature morphisms can be carried out largely independently of the nature of the underlying logical system. We use the notion of

[2] Such an approach is used by the HDTP framework, described in [23].
[3] Simplified version from [23].

institution introduced by Goguen and Burstall [13] in the late 1970s (see [6] for a recent overview). It approaches the notion of logical system from a relativistic view: rather than treating the concept of logic as eternal and given, it accepts the need for a large variety of different logical systems, and instead asks about common principles shared across logical systems. A crucial feature of institutions is that logical structure is indexed by signature, and change of signature is accounted for by signature morphisms; this is of course what we need as a prerequisite for the concept of derived signature morphism.

Definition 1. *An* institution $\mathcal{I} = (\mathbb{S}ign, \mathbf{Sen}, \mathbf{Mod}, \models)$ *consists of*

- *a category* $\mathbb{S}ign$ *of signatures and signature morphisms,*
- *a functor* $\mathbf{Sen}\colon \mathbf{Sign} \to \mathbf{Set}$,[4] *giving a set* $\mathbf{Sen}(\Sigma)$ *of* Σ-sentences *for each signature* $\Sigma \in |\mathbf{Sign}|$*, and a function* $\mathbf{Sen}(\sigma)\colon \mathbf{Sen}(\Sigma) \to \mathbf{Sen}(\Sigma')$*, denoted by* $\sigma(\_)$*, that yields* σ-translation *of* Σ*-sentences to* Σ'*-sentences for each signature morphism* $\sigma\colon \Sigma \to \Sigma'$*;*
- *a functor* $\mathbf{Mod}\colon \mathbf{Sign}^{op} \to \mathbb{CAT}$,[5] *giving a category* $\mathbf{Mod}(\Sigma)$ *of* Σ-models *for each* $\Sigma \in |\mathbf{Sign}|$*, and a functor* $\mathbf{Mod}(\sigma)\colon \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma)$*, denoted by* $\_|_\sigma$*, that yields* σ-reducts *of* Σ'*-models for each signature morphism* $\sigma\colon \Sigma \to \Sigma'$*; and*
- *for each* $\Sigma \in |\mathbf{Sign}|$*, a* satisfaction relation $\models_{\mathcal{I},\Sigma} \subseteq |\mathbf{Mod}(\Sigma)| \times \mathbf{Sen}(\Sigma)$

such that for any signature morphism $\sigma\colon \Sigma \to \Sigma'$*,* Σ*-sentence* $\varphi \in \mathbf{Sen}(\Sigma)$ *and* Σ'*-model* $M' \in |\mathbf{Mod}(\Sigma')|$*:*

$$M' \models_{\mathcal{I},\Sigma'} \sigma(\varphi) \iff M'|_\sigma \models_{\mathcal{I},\Sigma} \varphi \qquad [\textit{Satisfaction condition}]$$

The satisfaction condition expresses that truth is invariant under change of notation and context.

Example 4. The institution *Prop* of propositional logic. Signatures are sets (of propositional variables), signature morphisms are functions. Models are valuations of propositional variables into $\{T, F\}$, model reduct is just composition of the given model with the corresponding signature morphism. Sentences are formed inductively from propositional variables by the usual logical connectives. Sentence translation means replacement of propositional variables along the signature morphism. Satisfaction is the usual satisfaction of a propositional sentence under a valuation. □

Example 5. The institution $FOL^=$ of many-sorted first-order logic with equality. Signatures are many-sorted first-order signatures, consisting of a set of sort and sorted operation and predicate symbols. Signature morphisms map sorts, operation and predicate symbols in a compatible way. Models are many-sorted first-order structures. Sentences are first-order formulas. Sentence translation means replacement of symbols along the signature morphism. A model reduct interprets a symbol by first translating it along the signature morphism and then

[4] The category **Set** has all sets as objects and all functions as morphisms.

[5] $\mathbb{CAT}$ is the quasi-category of all categories, where "quasi" means that it lives in a higher set-theoretic universe.

interpreting it in the model to be reduced. Satisfaction is the usual satisfaction of a first-order sentence in a first-order structure. □

Example 6. In [17], we have sketched an institution of database schemas. We here follow a naive approach: A database schema is essentially a FOL theory where (some of) the relation symbols correspond to the database relations and some of the sorts correspond to the database attributes. There may be axioms of the theory defining concepts like keys and so on. A morphism is simply a normal theory interpretation. Database instances are then just models over a fixed universe, with different universes defining different families of instances. An interesting point to note is that the usual relation algebra operators, like join, can be seen as patterns for defining endofunctors (in the category of FOL theories and morphisms) of the theory (database extension) that create definitional extensions of the database schema to which they are applied. As queries are compositions of such operators, a query is also an endofunctor defining a definitional extension of the scheme. □

Semantic entailment in an institution is defined as usual: for $\Gamma \subseteq \mathbf{Sen}(\Sigma)$ and $\varphi \in \mathbf{Sen}(\Sigma)$, we write $\Gamma \models \varphi$, if all models satisfying all sentences in Γ also satisfy φ.

An alternative definition of institution uses so-called 'rooms' (in the terminology of [12]), which capture the Tarskian notion of satisfaction of a sentence in a model:

Definition 2. *A* room $\mathcal{R} = (S, \mathcal{M}, \models)$ *consists of*

- *a set of S of* sentences*,*
- *a category $\mathcal{M}$ of* models*, and*
- *a binary relation $\models \subseteq |\mathcal{M}| \times S$, called the* satisfaction relation*.*

Then, morphisms between rooms are of course called corridors [12]:

Definition 3. *A* corridor $(\alpha, \beta)\colon (S_1, \mathcal{M}_1, \models_1) \to (S_2, \mathcal{M}_2, \models_2)$ *consists of*

- *a sentence translation function $\alpha\colon S_1 \to S_2$, and*
- *a model reduction functor $\beta\colon \mathcal{M}_2 \to \mathcal{M}_1$, such that*

$$M_2 \models_2 \alpha(\varphi_1) \textit{ if and only if } \beta(M_2) \models_1 \varphi_1$$

*holds for each $M_2 \in |\mathcal{M}_2|$ and each $\varphi_1 \in S_1$ (*satisfaction condition*).*

Since corridors compose and there are obvious identity corridors, rooms and corridors form a category $\mathbb{R}oom$. Then, an *institution* is just a functor $\mathcal{I}\colon \mathbb{S}ign \to \mathbb{R}oom$.

Relationships between institutions (and entailment systems) are captured mathematically by 'institution morphisms', of which there are several variants, each yielding a category under a canonical composition. For the purposes of this paper, institution morphisms [14] seem technically most convenient. For the notion of institutional monad introduced below, we also need 2-cells between institution morphisms, called modifications.

We use the representation of institutions as functors introduced above.

Definition 4. *Given institutions* $I_1 : \mathbb{S}ign_1 \to \mathbb{R}oom$ *and* $I_2 : \mathbb{S}ign_2 \to \mathbb{R}oom$, *an* institution morphism $(\Phi, \rho) : I_1 \to I_2$ *consists of a functor* $\Phi : \mathbb{S}ign_1 \to \mathbb{S}ign_2$ *and a natural transformation* $\rho : I_2 \circ \Phi \to I_1$.

Given institution morphisms $(\Phi, \rho) : I_1 \to I_2$ *and* $(\Phi', \rho') : I_1 \to I_2$, *an* institution morphism modification $\theta : (\Phi, \rho) \to (\Phi', \rho')$ *is just a natural transformation* $\theta : \Phi \to \Phi'$ *such that* $\rho = \rho' \circ (I_2 \cdot \theta)$.[6]

This leads to a 2-category $\mathbb{I}ns$ *of institutions, morphisms and modifications.*

Example 7. There is an institution morphism $\mu_1 : FOL^= \to Prop$. From a first-order signature, it only keeps the nullary predicates, which become propositional variables. Also from a first-order model, only the interpretations of the nullary predicates are kept. Moreover, there is an obvious inclusion of *Prop*-sentences into $FOL^=$-sentences. The satisfaction condition is easily shown. □

Example 8. Another institution morphism $\mu_2 : FOL^= \to Prop$ keeps *all* predicates from a first-order signature as propositional variables. From a first-order model, extract a valuation by mapping a predicate to true iff it is universally true. A propositional variable is translated to a sentence stating that the corresponding predicate holds universally. Again, the satisfaction condition is easily shown. □

Example 9. The inclusions $\iota_\Sigma : (\mu_1)_\Sigma \to (\mu_2)_\Sigma$ form a modification $\iota : \mu_1 \to \mu_2$. □

4 Derived Signature Morphisms Through Definitional Extensions

In this section, we develop a very general approach to derived signature morphisms, based on two assumptions: (1) signatures are replaced by *theories*, and (2) models can be *amalgamated*. While these assumptions and the idea of letting theory morphisms be targeted in some definitional extension is folklore, surprisingly little is known about the properties of this construction.

We start with colimits, which can be seen as a tool for combining and interconnecting systems, and amalgamation, which ensures that models can be combined along colimits. Amalgamation ensures further nice logical properties, e.g. laws for modularity [7], availability of institution-independent proof calculi for structured specifications [2,19] or well-behaved semantics for architectural specifications [22].

Definition 5. *A cocone for a diagram in* $\mathbb{S}ign$ *is* (weakly) amalgamable *if it is mapped to a (weak) limit in* $\mathbb{CAT}$ *under* Mod. $\mathcal{I}$ *(or* Mod*) admits* (finite) (weak) amalgamation *if (finite) colimits exists in* $\mathbb{S}ign$ *and colimiting cocones are (weakly) amalgamable, i.e. if* Mod *maps (finite) colimits to (weak) limits.*

[6] The original notion from [4] is a lax variant of this: a morphism $\rho \to \rho' \circ (I_2 \cdot \theta)$ is given instead of equality.

An important special case is pushouts: $\mathcal{I}$ (or Mod*) has* (weak) model amalgamation for pushouts*, if pushouts exist in $\mathbb{S}ign$ and are (weakly) amalgamable. More specifically, the latter means that for any pushout*

$$\begin{tikzcd} \Sigma \arrow[r] \arrow[d] & \Sigma_1 \arrow[d] \\ \Sigma_2 \arrow[r] & \Sigma_R \end{tikzcd}$$

in $\mathbb{S}ign$ and any pair $(M_1, M_2) \in \mathsf{Mod}(\Sigma_1) \times \mathsf{Mod}(\Sigma_2)$ that is compatible *in the sense that M_1 and M_2 reduce to the same Σ-model can be* amalgamated *to a unique (or weakly amalgamated to a not necessarily unique) Σ_R-model M (i.e., there exists a (unique) $M \in \mathsf{Mod}(\Sigma_R)$ that reduces to M_1 and M_2, respectively), and similarly for model morphisms.*

This specific explanation in terms of compatible families of models that can be amalgamated also generalises to arbitrary colimits.

For example, it is well-known [21] that

Proposition 1. *Both propositional logic and many-sorted first-order logic both have model amalgamation.*

In the sequel, we work in an arbitrary but fixed institution $\mathcal{I} = (\mathbb{S}ign, \mathbf{Sen}, \mathbf{Mod}, \models)$.

Definition 6. *A* theory *is a pair $T = (\Sigma, \Gamma)$ where Γ is a set of Σ-sentences. A* theory morphism *$(\Sigma, \Gamma) \to (\Sigma', \Gamma')$ is a signature morphism $\sigma : \Sigma \to \Sigma'$ such that $\Gamma' \models_{\Sigma'} \sigma(\Gamma)$. Let $\mathbb{T}h(\mathcal{I})$ denote this category. Each theory (Σ, Γ) inherits sentences from $\mathsf{Sen}^{\mathcal{I}}(\Sigma)$, while the models are restricted to those models in $\mathsf{Mod}^{\mathcal{I}}(\Sigma)$ that satisfy all sentences in Γ. It is easy to see that $\mathcal{I}$ maps theory morphisms to corridors in this way. By taking $\mathbb{T}h(\mathcal{I})$ as "signature" category, we arrive at the institution $\mathcal{I}^{Th}$ of theories.*

Definition 7. *A theory morphism $\sigma \colon T_1 \to T_2$ is* conservative*, if each T_1-model has a σ-expansion to a T_2-model; it is* definitional*, if each model has a unique such expansion. Definitional theory morphisms are also called* definitional extensions *and are denoted as $T_1 \bullet\!\!\xrightarrow{\sigma} T_2$.*

Definition 8. *A derived theory morphism $(\sigma, \theta) \colon T_1 \to T_2$ is given by an ordinary theory morphism $\sigma \colon T_1 \to T_2'$ into a definitional extension $\theta \colon T_2 \bullet\!\!\longrightarrow T_2'$ of T_2.*

Every theory morphism $\sigma \colon T_1 \to T_2$ is a derived theory morphism with respect to the identity $id \colon T_2 \bullet\!\!\longrightarrow T_2$. We can define reducts for arbitrary derived theory morphisms by first taking the unique θ-expansion and then taking σ-reduct.

$$
\begin{tikzcd}
 & T_2' \\
T_1 \arrow[ur, "\sigma"] & T_2 \arrow[u, "\theta"']
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \mathbf{Mod}(T_2') \arrow[dl, "\mathbf{Mod}(\sigma)"'] \arrow[d, "\mathbf{Mod}(\theta)"] \\
\mathbf{Mod}(T_1) & \mathbf{Mod}(T_2) \arrow[u, dotted, bend left, "\exists!"]
\end{tikzcd}
$$

If we had based derived theory morphisms on conservative instead of definitional extensions, reducts would exist but generally would not be unique due to the possibility to have several different θ-expansions.

Example 10. In the setting of Example 1, we can construct a derived theory morphism $T_{\text{ring}} \to T_{\text{algebra}}$ via definitional extension T'_{algebra}:

$$
\begin{aligned}
\Sigma'_{\text{algebra}} &:= \Sigma_{\text{algebra}} \cup \{+\} \\
\Gamma'_{\text{algebra}} &:= \Gamma_{\text{algebra}} \cup \{x + y = (x \vee y) \wedge \neg(x \wedge y)\}
\end{aligned}
$$

One can then define an ordinary signature morphism $\sigma : \Sigma_{\text{ring}} \to \Sigma'_{\text{algebra}}$ by mapping $\cdot \mapsto \wedge$ and $+ \mapsto +$.

Note that there is a caveat: adding defined symbols generally can change the notion of model morphism. For example, consider a $FOL^{=}$-signature with a binary predicate symbol Q. Then adding a unary predicate symbol P with definition

$$P(x) \Leftrightarrow \forall y. Q(x, y)$$

is indeed a definitional extension. However, not every model morphism for Q will also preserve P. As a consequence, derived theory morphisms in general do not provide reducts for model morphisms. Hence, in the sequel, we have to make the following

General Assumption. Model morphisms are compatible with definitional extensions, which means that given a definitional extension $\sigma\colon T_1 \to T_2$, every T_1-model morphism $h_1 : M_1 \to M_1'$ has a unique expansion to a T_2-model morphism $h_2 : M_2 \to M_2'$, where M_2 is the unique expansion of M_1 and M_2' that of M_1'.

One simple way to achieve this property is to dispense with non-trivial model morphisms and use discrete model categories. Another way is to restrict formulas in derived theory morphisms to those that are compatible with all model morphisms. Both ways have certain drawbacks, but there is no easy solution to this problem.

Further note that in general a derived theory morphism $T_1 \to T_2$ does not provide a translation of T_1-sentences to T_2-sentences. This means that we will arrive at a category of theories and derived theory morphisms which form a *specification frame*, which is given by a category **Spec** of (abstract) specifications (or theories), with semantics given by a model functor $\mathbf{Mod}\colon \mathbf{Spec}^{op} \to \mathbb{CAT}$. The terminology follows [3], the concept appeared earlier as "specification logic" in [10,11]). As before, functions $\mathbf{Mod}(\sigma)$, for $\sigma\colon T_1 \to T_2$ in **Spec**, will be called reducts and denoted by $\_|_\sigma$.

Our derived theory morphisms are similar to morphisms in the category $Cospan(\mathcal{I}^{Th})$. However, the latter category has severe drawbacks: generally, only some colimits exist, see [1]. Moreover, the equivalence used for cospans, isomorphism of intermediate objects, is much too fine-grained for our purposes. In general, there are many choices for a derived signature morphism due to underdetermination of the intermediate theory T_2': arbitrary symbols may be added to the signature Σ_2' and equivalent formulations of the sentences in Γ_2' can be chosen. Such modifications do not change the essence of the morphism, i.e. induced mappings on model level. The following definition will account for this fact:

Definition 9. *Two derived theory morphisms* $(\sigma_1, \theta_1), (\sigma_2, \theta_2) \colon T_1 \to T_2$ *are* equivalent*, if their induced model reduct maps are equal.*

Proposition 2. *In am institution* $\mathcal{I}$ *with model amalgamation for pushouts, derived theory morphisms compose. This leads to a category* $\mathbb{D}er(\mathcal{I})$ *of derived theory morphisms up to equivalence.*

Proof. The composition $(\sigma_2, \theta_2) \circ (\sigma_1, \theta_1)$ is given by $(\sigma \circ \sigma_1, \theta \circ \theta_2)$, where the rhombus is a pushout:

$$\begin{tikzcd}
& & \bullet \\
& T_2' \arrow[ur, dotted, "\sigma"] & T_3' \arrow[u, dotted, "\theta"'] \\
T_1 \arrow[ur, "\sigma_1"] & T_2 \arrow[u, "\theta_1"'] \arrow[ur, "\sigma_2"] & T_3 \arrow[u, "\theta_2"']
\end{tikzcd}$$

In order to show that $\theta \circ \theta_2$ is definitional, it suffices to show that θ is (θ_2 is by definition). Let M_3' be T_3'-model. Since θ_1 is definitional, $M_3'|_{\sigma_2}$ has a unique expansion to a T_2'-model M_2'. Then the unique amalgamation of M_3' and M_2' gives a the unique desired expansion of M_3'. This shows the folklore fact that definitional extensions are preserved by pushouts.

Composition is well-defined, because different pushouts always lead to equivalent derived theory morphisms. □

Note that in case that $\theta_1 = id$, the composition simplifies to $(\sigma_2 \circ \sigma_1, \theta_2)$. Further note that for the verification of commutativity of diagrams in $\mathbb{D}er(\mathcal{I})$, it is often easier to compose model reduct maps; this avoids the computation of the above pushout.

Altogether, we arrive at

Theorem 1. *For a given institution* $\mathcal{I}$*, theories and their models together with derived theory morphisms and their reducts from a specification frame* $\mathcal{I}^{Der}$.

A central result is to establish the existence of colimits in $\mathbb{D}er(\mathcal{I})$:

Theorem 2. *In an institution with model amalgamation, the category* $\mathbb{D}er(\mathcal{I})$ *of derived theory morphisms is cocomplete.*

Proof. First note that colimits lift from signatures to theories [13], so we can assume that the category of theories $\mathbb{T}h(\mathcal{I})$ is cocomplete.

The initial theory 0 is also initial in $\mathbb{D}er(\mathcal{I})$: Given any theory T, the derived theory morphism from 0 to T is $(!_T, id_T)$. Concerning its uniqueness, note that by model amalgamation, $\mathsf{Mod}(0)$ is a singleton, which means that there all derived theory morphisms starting from 0 are equivalent.

Concerning non-empty products, given a set of theories $(T_i)_{i\in I}$, its coproduct $\coprod_I T_i$ in the category of theories lifts to $\mathbb{D}er(\mathcal{I})$. The coproduct injections in $\mathbb{D}er(\mathcal{I})$ are $T_i \xrightarrow{\amalg_i} \coprod_I T_i \xleftarrow{id} \coprod_I T_i$. To show the universal property, let $T_i \xrightarrow{\tau_i} U_i \xleftarrow{\theta_i} \bullet T$ be a cocone in $\mathbb{D}er(\mathcal{I})$. Let $(C, (\mu_i \colon U_i \to C)_{i\in I})$ be the colimit of $(\ U \bullet\!\!\xrightarrow[\theta_i]{} U_i\ )_{i\in I}$ in $\mathbb{T}h(\mathcal{I})$. Then let $\tau \colon \coprod_I T_i \to C$ be $[\mu_i]_I \circ \coprod_I \tau_i$. Pick some $i_0 \in I$. The mediating morphism from the colimit to the cocone is then $(\tau, \mu_{i_0} \circ \theta_{i_0}) \colon \coprod_I T_i \to T$ ($\mu_{i_0} \circ \theta_{i_0}$ is equal to $\mu_i \circ \theta_i$ for any $i \in I$). $\mu_{i_0} \circ \theta_{i_0}$ is definitional: any T-model M has unique θ_i-expansions $M_i \in \mathsf{Mod}(U_i)$ for $i \in I$. Then M together with the M_i form a compatible family of models for $(\ U \bullet\!\!\xrightarrow[\theta_i]{} U_i\ )_{i\in I}$. By model amalgamation, this family has a unique amalgamation to a model M_C of the colimit C such that $M_C|_{\mu_i} = M_i$. Now $\mu_{i_0} \circ \theta_{i_0}$ commutes with the cocones, because (1) given a T-model, its unique C-expansion reduces via μ_i to its unique U_i-expansion and (2) the left triangle commutes by definition of τ.

$$
\begin{tikzcd}
 & & C & \\
 & \coprod_I T_i \arrow[ur, "\tau"] & & \\
T_i \arrow[ur, "\amalg_i"] \arrow[drr, "\tau_i"'] & & & \bullet T \arrow[uul, "\mu_i\circ\theta_i=\mu_{i_0}\circ\theta_{i_0}"'] \arrow[dl, "\theta_i"] \\
 & & U_i \arrow[uuu, "\mu_i"] &
\end{tikzcd}
$$

To show its uniqueness, assume that there is another morphism $(\lambda, \theta) \colon \coprod_I T_i \to T$ with $(\lambda, \theta) \circ (\amalg_i, id) = (\tau_i, \theta_i)$, which means that the two model reduct maps from T to T_i are the same:

$$
\begin{tikzcd}
 & & U_i & \\
T_i \arrow[urr, "\tau_i"] \arrow[dr, "\amalg_i"'] & & & \bullet T \arrow[ul, "\theta_i"'] \arrow[dl, "\theta"] \\
 & \coprod_I T_i \arrow[r, "\lambda"'] & \bullet &
\end{tikzcd}
$$

But then $(\lambda, \theta) = (\tau, \mu_{i_0} \circ \theta_{i_0})$ by model amalgamation:

$$\begin{tikzcd}
& & C & \\
& \coprod_I U_i \arrow[ur, "{[\mu_i]_I}"] & & \\
\coprod_I T_i \arrow[ur, "\coprod_I \tau_i"] \arrow[rr, "\lambda"'] & & \bullet & T \arrow[uul, "\mu_{i_0}\circ\theta_{i_0}"'] \arrow[l, "\theta"]
\end{tikzcd}$$

It remains to treat coequalisers. Given a pair of parallel derived theory morphisms $(\sigma_1, \theta_1), (\sigma_2, \theta_2)\colon T \to U$, its coequaliser in $\mathbb{D}er(\mathcal{I})$ is given by (μ, id_C), which is obtained as the following colimit of theories:

$$\begin{tikzcd}
& & U_1 \arrow[dr, "\mu_1"] & \\
T \arrow[urr, "\sigma_1"] \arrow[drr, "\sigma_2"'] & U \arrow[ur, "\theta_1"] \arrow[rr, "\mu"] \arrow[dr, "\theta_2"'] & & C \\
& & U_2 \arrow[ur, "\mu_2"'] &
\end{tikzcd} \qquad (*)$$

By definitionality of θ_1 and θ_2 and model amalgamation, (μ, id_C) is an epi in $\mathbb{D}er(\mathcal{I})$. Concerning the universal property, consider any cocone in $\mathbb{D}er(\mathcal{I})$ $(\tau, \theta)\colon U \to D$. Take the pushout of theories shown in the left square

$$\begin{tikzcd}
& C \arrow[dr, "\kappa"] & \\
U \arrow[ur, "\mu"] \arrow[dr, "\tau"'] & & E \\
& V \arrow[ur, "\lambda"] & D \arrow[l, "\theta"']
\end{tikzcd}$$

Then $(\kappa, \lambda \circ \theta)\colon C \to D$ is the mediating morphism. To establish definitionality of λ, consider any V-model M. Let $M_i \in \mathsf{Mod}(U_i)$ by the unique θ_i-expansion of $M_U := M|_\tau$ $(i = 1, 2)$. Since (τ, θ) is a cocone, $(\tau, \theta) \circ (\sigma_1, \theta_1) = (\tau, \theta) \circ (\sigma_1, \theta_1)$. Now consider the D-model $M_D := M|_\theta$. Then

$$M_1|_{\sigma_1} = M_U|_{(\sigma_1,\theta_1)} = M_D|_{(\tau,\theta)\circ(\sigma_1,\theta_1)} = M_D|_{(\tau,\theta)\circ(\sigma_2,\theta_2)} = M_U|_{(\sigma_2,\theta_2)} = M_2|_{\sigma_2}$$

Let us denote this model by M_T. Thus, (M_T, M_U, M_1, M_2) is a compatible family of models for the diagram $(*)$ above (without the C), which by model amalgamation can be amalgamated to a C-model M_C. Let M_E be the amalgamation of M_C and M. Then M_E is the needed λ-expansion of M. Its uniqueness follows from those of the amalgamations.

From the diagram above we easily get that $(\kappa, \lambda \circ \theta) \circ (\mu, id_C) = (\tau, \theta)$. Uniqueness follows since (μ, id_C) is an epi in $\mathbb{D}er(\mathcal{I})$. $\square$

A natural follow-up question is whether the specification frame $\mathcal{I}^{Der}$ admits amalgamation. Under some mild assumption, the answer is positive:

Theorem 3. *$\mathcal{I}^{Der}$ admits amalgamation whenever $\mathcal{I}$ does.*

Proof. Since $\mathbb{D}er(\mathcal{I})$ inherits coproducts from $\mathcal{I}$, also amalgamation lifts. Concerning coequalisers, in the notion of diagram $(*)$ above, let M_U a U-model such that $M_U|_{(\sigma_1,\theta_1)} = M_D|_{(\tau,\theta)\circ(\sigma_1,\theta_1)}$. Let $M_i \in \mathsf{Mod}(U_i)$ be the unique θ_i-expansion of M_U $(i = 1, 2)$. Then $(M_U|_{\theta_1}, M_1, M_2, M_U)$ is a compatible family for $(*)$ without C. This family can be uniquely amalgamated to a model M_C of C, which is the desired amalgamation in $\mathbb{D}er(\mathcal{I})$. Uniqueness follows from that of the amalgamation in I and that of definitional extensions. $\square$

We have defined equivalence of derived theory morphisms using a semantic condition that is undecidable in general. However, for specific institutions, one can do better. For example, in $FOL^=$, one can restrict definitional extensions to those that are given by explicit definitions of predicate and function symbols (i.e. by equivalence to a formula or equality to a term). Then one can use syntactic equality of symbol definitions in order to decide equivalence of derived signature morphisms. This yields an efficiently decidable approximation of semantic equivalence.

A more syntactic notion of equivalence of derived theory morphisms would require definitional extensions to be monic. Then, two derived theory morphisms $(\sigma_1, \theta_1)\colon T_1 \to T_2$ and $(\sigma_2, \theta_2)\colon T_1 \to T_2$ are said to be equivalent, if there is a theory T' and commutative diagrams as follows:

$$\begin{tikzcd}
& T_1' & \\
T_1 \arrow[ur, "\sigma_1"] \arrow[r] \arrow[dr, "\sigma_2"'] & T' \arrow[u, tail] \arrow[d, tail] & T_2 \arrow[ul, tail, "\theta_1"'] \arrow[l, tail] \arrow[dl, tail, "\theta_2"] \\
& T_2' &
\end{tikzcd}$$

Under suitable assumptions, coproducts exist; however, coequalisers do not. This is why we have chosen the semantic notion of equivalence above.

5 Derived Signature Morphisms as Abstract Substitutions

The second approach to derived signature morphisms is to consider them to be a special kind of abstract substitution in the sense of [5,6]. Such Σ-substitutions generalise the idea of substitutions found in many logics to arbitrary institutions. The extension of a signature by a set of variables is expressed by a signature morphism $\chi : \Sigma \to \Sigma'$, leading to the following definition (we give a version that makes use of rooms and corridors):

Definition 10. *For any signature Σ of an institution I, with two "extensions" $\chi_1\colon \Sigma \to \Sigma_1$ and $\chi_2\colon \Sigma \to \Sigma_2$, a Σ-*substitution *$\chi_1 \to \chi_2$ is a corridor $\rho\colon I(\Sigma_1) \to I(\Sigma_2)$ that preserves Σ, i.e. the following diagram commutes*

$$\begin{tikzcd} I(\Sigma_1) \arrow[rr, "\rho"] & & I(\Sigma_2) \\ & I(\Sigma) \arrow[ul, "I(\chi_1)"] \arrow[ur, "I(\chi_2)"'] & \end{tikzcd}$$

The idea of this definition is the existence of sentence translations and model reducts between extensions of a signature. This makes it a very general concept, that covers besides classical first-order substitution also second-order substitutions in $FOL^=$ and also derived signature morphisms. [6, 99f] demonstrates this for the case of $FOL^=$:[7] for a given base signature Σ, a derived signature $\Phi(\Sigma)$ is constructed. There exists a canonical embedding $\eta : \Sigma \to \Phi(\Sigma)$. It is then shown that sentences over the derived signature can be translated to sentences over the base signature and that a model for the base signatures provide a model for the derived signature, i.e. there is a corridor ρ from $I(\Phi(\Sigma))$ to $I(\Sigma)$. In summary, such a derivation is a Σ-substitution from η to *id* (this could also be expressed simply by saying that ρ is a retraction of $I(\eta)$ in the category $\mathbb{R}oom$).

Given such a derivation (Φ, η, ρ) for a signature Σ_2, a derived signature morphism from Σ_1 to Σ_2 is defined as an ordinary signature morphism $\sigma\colon \Sigma_1 \to \Phi(\Sigma_2)$. The substitution condition assures, that sentence translation and model reduction hold for the underlying base category, i.e. Σ_1-sentences can be translated to Σ_2-sentences, and Σ_2-models can be reduced to Σ_1-models along σ by detour over $\Phi(\Sigma_2)$:

$$\begin{tikzcd} & I(\Phi(\Sigma_2)) \arrow[d, "\rho_{\Sigma_2}"] \\ I(\Sigma_1) \arrow[ur, "I(\sigma)"] & I(\Sigma_2) \end{tikzcd}$$

Analysing the above example, one can find the following ingredients that seem to be essential to introduce the concept of derivation in an institution I:

– a general way to construct *derived signatures*, i.e. a functor $\Phi : \mathbb{S}ign \to \mathbb{S}ign$
– a *canonical embedding* from the base signature to its derivation, i.e. a natural transformation $\eta\colon id \to \Phi$
– a *translation* (corridor) from the “derived logic” to “simple logic”, i.e. a natural transformation $\rho : I \circ \Phi \to I$
– compatibility of the embedding and the translation, expressed by the condition $\rho \circ (I \cdot \iota) = id$

Using the language of institution morphisms and institution morphism modifications, this amounts to saying:

Definition 11. *A* derivation *for an institution I consists of*

– *an institution morphism $T = (\Phi^T, \rho^T)\colon \mathcal{I} \to \mathcal{I}$ and*
– *an institution morphism modification $\eta\colon id \to T$*

[7] We give only a brief summary here, simplifying and adapting notation.

This allows to introduce derived signature morphisms between arbitrary signatures of $\mathbb{S}ign$. However, a shortcoming of this approach is that derived signature morphisms can not be composed in an obvious way. This will be addressed in the next section.

Before closing this section, it should be remarked that one can use derivations as an alternative way to introduce substitutions at an abstract level: reconsidering the above situation, where a signature Σ and two extensions $\chi_1 : \Sigma \to \Sigma_1$ and $\chi_2 : \Sigma \to \Sigma_2$ are given, a derivation-based *Σ-substitution* $\chi_1 \to \chi_2$ is defined as a derived signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ that preserves Σ, i.e. it makes following diagram of signature morphisms commute:

$$\begin{tikzcd}
& & \Phi(\Sigma_2) \\
\Sigma_1 \arrow[urr, "\sigma"] & & \Sigma_2 \arrow[u, "\eta_{\Sigma_2}"'] \\
& \Sigma \arrow[ul, "\chi_1"] \arrow[ur, "\chi_2"'] &
\end{tikzcd}$$

For example, in the case of $FOL^=$, consider extensions χ_1 and χ_2 of a signature Σ by first-order variables, i.e. new 0-ary operator names. Then a substitution μ replaces variables of Σ_1 by terms over Σ_2, i.e. symbols from a derivation $\Phi(\Sigma_2)$. Higher-order substitutions can be obtained by extending the signature with higher-order variables, i.e. new operators names with arity > 0. This notion of derivation-based Σ-substitution, can also be used to introduce an abstract notion of unification: given a set of Σ_1-sentences S, a unifier is a derived signature morphism $\mu : \Sigma_1 \to \Sigma_1$ (that preserves Σ) such that the induced mapping on sentence level, i.e. $\rho_{\Sigma_1} \circ \mathsf{Sen}^I(\mu) : \mathsf{Sen}^I(\Sigma_1) \to \mathsf{Sen}^I(\Sigma_1)$ maps S on a singleton set.

This notion of substitution is more specific than the above one, since every derivation-based Σ-substitution σ is a Σ-substitution in the sense of Definition 10: the induced corridor $\rho = \rho_{\Sigma_2} \circ I(\sigma)$ obviously preserves $I(\Sigma)$. The advantage of the derivation-based approach is, that it anchors the corridor ρ in a mapping on signature level in a natural way, while it stills seems general enough to cover most interesting cases.

6 Derived Signature Morphisms Through Kleisli Institutions

The approaches of the previous sections have some drawbacks: In the definitional extension approach, sentences cannot be translated along derived theory morphisms, while substitution-based derived signature morphisms do not compose. In this section, we remedy these problems by introducing for each signature Σ, a *signature of terms* $\Phi^T(\Sigma)$, where T is a suitable *monad*. Then, a derived signature morphism $\sigma\colon \Sigma_1 \to \Sigma_2$ is an ordinary signature morphism $\sigma\colon \Sigma_1 \to \Phi^T(\Sigma_2)$.

The monad needs to interact with the structure of the institution. This leads to the notion of *institutional monad*.

Definition 12. *An* institutional monad $\mathcal{T} = (T, \eta, \mu)$ *is a monad in* $\mathbb{I}ns$ *(see [18] for the notion of monad in a 2-category), which amounts to*

- *an institution* $\mathcal{I}$,
- *an institution morphism* $T = (\Phi^T, \rho^T): \mathcal{I} \to \mathcal{I}$,
- *an institution morphism modification* $\eta: id \to T$, *and*
- *an institution morphism modification* $\mu: T \times T \to T$,

such that the usual laws of a monad are satisfied:

$$\begin{tikzcd} T \arrow[r, "\eta_T"] \arrow[d, "T\eta"'] \arrow[dr, equal] & T \times T \arrow[d, "\mu"] \\ T \times T \arrow[r, "\mu"] & T \end{tikzcd} \qquad \begin{tikzcd} T \times T \times T \arrow[r, "\mu_T"] \arrow[d, "T\mu"'] & T \times T \arrow[d, "\mu"] \\ T \times T \arrow[r, "\mu"] & T \end{tikzcd}$$

By selecting the signature component only, an institutional monad $\mathcal{T}$ gives rise to an ordinary a monad, which we denote by $\mathcal{T}^{\mathbb{S}ign}$.

Example 11 (monad over the first-order logic institution). Let T be the institution morphism $(\Phi^T, \rho^T): FOL^= \to FOL^=$ with $\rho^T_\Sigma = (\alpha^T_\Sigma, \beta^T_\Sigma)$.

- $\Phi^T(\Sigma)$ adds terms $\lambda x_1 : s_1, \ldots x_n : s_n.t$ as n-ary operations and terms $\lambda x_1 : s_1, \ldots x_n : s_n.\varphi$ (where φ is a formula) as n-ary predicates;
- $\alpha^T_\Sigma: Sen(\Phi^T(\Sigma)) \to Sen(\Sigma)$ β-reduces all application of λ-term operations and predicates;
- $\beta^T_\Sigma: Mod(\Sigma) \to Mod(\Phi^T(\Sigma))$ interprets λ-term operations and predicates in $\beta^T_\Sigma(M)$ as $(a_1, \ldots, a_n) \mapsto M(t)[x_i \mapsto a_i]$
- $\eta_\Sigma: \Sigma \to \Phi^T(\Sigma)$ is the obvious inclusion;
- $\mu_\Sigma: \Phi^T(\Phi^T(\Sigma)) \to \Phi^T(\Sigma)$ collapses two levels of λ-terms into one.

The notion of Kleisli category for a monad can be generalised to institutions in following way:

Definition 13. *Given an institutional monad* $\mathcal{T} = (T: \mathcal{I} \to \mathcal{I}, \eta, \mu)$, *its* Kleisli institution $\mathcal{I}_\mathcal{T}$ *is the Kleisli object of* $\mathcal{T}$ *in* $\mathbb{I}ns$, *which amounts to*

- *the signature category of* $\mathcal{I}_\mathcal{T}$ *is the Kleisli category of the monad* $\mathcal{T}^{\mathbb{S}ign}$,
- *given a signature* $\Sigma \in \mathcal{T}^{\mathbb{S}ign}$, $\mathcal{I}_\mathcal{T}(\Sigma)$ *is just* $\mathcal{I}(\Sigma)$, *and*
- *given a signature morphism* $\sigma: \Sigma_1 \to \Sigma_2 \in \mathcal{T}^{\mathbb{S}ign}$ *(which is a signature morphism* $\sigma: \Sigma_1 \to \Phi^T(\Sigma_2)$ *in* $\mathcal{I}$*),* $\mathcal{I}_\mathcal{T}(\sigma)$ *is given by*

$$\mathcal{I}(\Sigma_1) \xrightarrow{\mathcal{I}(\sigma)} \mathcal{I}(\Phi^T(\Sigma_2)) \xrightarrow{\rho^T_{\Sigma_2}} \mathcal{I}(\Sigma_2)$$

providing sentence translation and model reduct for Kleisli morphisms.

The institution independent notions of logical consequence, theory etc. and corresponding results of course also apply to the Kleisli institution; in particular, Kleisli theory morphisms preserve logical consequence.

Contrary to the statement in [9], colimits are not necessarily lifted from the base signature category to the Kleisli signature category:

Proposition 3. *If the base institution $\mathcal{I}$ has signature coproducts, then does the Kleisli institution $\mathcal{I}_{\mathcal{T}}$. However, coequalisers (and therefore also e.g. pushouts) are generally not lifted to the Kleisli institution.*

Proof. If $\Sigma_i \xrightarrow{\amalg_i} \coprod_I \Sigma_i$ is a coproduct in the signature category of $\mathcal{I}$, then $\Sigma_i \xrightarrow{\amalg_i} \coprod_I \Sigma_i \xrightarrow{\eta_{\coprod_I \Sigma_i}} T(\coprod_I \Sigma_i)$ is a coproduct in the signature category. See also (2.1) in [24].

Concerning coequalisers, consider the category of derived signature morphisms of standard first-order logic. Take a parallel pair of arrows where a binary function symbol is mapped a) to $\lambda x, y : s.x$ and b) to $\lambda x, y : s \mapsto y$. Then there is no coequalisers, since it would have to equate $x, y \mapsto x$ with $x, y \mapsto y$. (A span with no pushout can be obtained in a similar way.) □

Note that the negative situation in Proposition 3 can be remedied in some cases. For the example given in the proof, a coequaliser exists in the category of *derived theory morphisms up to equivalence*. In this pushout, an axiom $\forall x, y : s.x = y$ is added. This category can be defined as follows:

Definition 14. *For the institution of many-sorted first-order logic, the category of* derived theory morphisms up to equivalence *has theories as objects. Morphisms are derived signature morphisms that map axioms to theorems, taken up to an equivalence. Two derived theory morphisms are equivalent iff they map a given symbol to terms that are provably equal.*

Proposition 4. *The category of derived* theory *morphisms up to equivalence for $FOL^{=}$ has colimits.*

Proof. For coproducts, use Proposition 3. The coequaliser of a pair

$$U \overset{\sigma_1}{\underset{\sigma_2}{\rightrightarrows}} V$$

is obtained in the base signature category by

$$V \xrightarrow{q} Q \xrightarrow{\eta_Q} TW$$

where Q is the quotient of V by the congruence

$$\sigma_1(s) \equiv \sigma_2(s) \ (s \in sorts(U)).$$

Then on sorts, $q \circ \sigma_1 = q \circ \sigma_2 =: q'$. Moreover, W is Q augmented by axioms

$$\forall x_1 : q'(s_1), \ldots x_n : q'(s_n).\alpha(\sigma_1(f)(x_1, \ldots, x_n) = \sigma_2(f)(x_1, \ldots, x_n))$$

for each operation symbol $f : s_1 \ldots s_n \to s$ in U and axioms

$$\forall x_1 : q'(s_1), \ldots x_n : q'(s_n).\alpha(\sigma_1(p)(x_1, \ldots, x_n) \Leftrightarrow \sigma_2(p)(x_1, \ldots, x_n))$$

for each predicate symbol $p : s_1 \ldots s_n$ in U. Recall from Example 11 that the effect of α is that all applications of $\sigma_1(f)$ (resp. $\sigma_1(p)$) to terms are β-reduced.

Now the Kleisli morphism $\eta_Q \circ q \colon V \to W$ equalises σ_1 and σ_2: for sorts, this is done by q, and for operation and predicate symbols, this follows from the axioms in W (noting that provably equal symbols are identified). Given any Kleisli morphism $h \colon V \to X$ equalising σ_1 and σ_2, define $k \colon W \to X$ by $k(q(s)) = h(s)$ on sorts, and $k(f) = h(f)$ for operation and predicate symbols. In both cases, well-definedness follows from $h \circ \sigma_1 = h \circ \sigma_2$. □

7 Conclusions

We have introduced several approaches to derived signature (resp. theory) morphisms. The first approach, using definitional extensions, is very general and works in any institution with model amalgamation for pushouts. While models can be reduced against derived signature morphisms, the drawback is that sentences cannot be translated along them. The second approach remedies this problem axiomatically: model reducts and sentence translation are required to exist. Moreover, powerful Herbrand theorems relate queries and substitutions [5,6]. The third approach is more specific about the nature of derived signature morphisms: they are obtained through a Kleisli construction in an institutional monad, which provides a more precise (abstract) description of what derived signature morphisms are.

Generally, it turns out that coproducts lift easily to the derived case, while coequalizers are more difficult. The problem is that derived signature morphisms are too powerful to admit coequalisers directly, because in a coequalisers, they can be used to equate arbitrarily complex terms. The trick to still obtain coequalisers is to pass from signature to theory morphisms and impose some suitable quotient on the latter. For the approach of definitional extensions, we can obtain coequalizers by working with theory morphisms and consider derived theory morphisms up to semantic equivalence, while a stronger (more syntactic) equivalence does not work. For the particular Kleisli institutions of the natural institutional monad for many-sorted first-order logic, we can obtain coequalizers by adding suitable equations. It is an interesting open question whether and how this can be generalised to an arbitrary institution.

There are still open questions concerning the relationship between the notion introduced via definitional extensions and the one using the Kleisli construction. One can ask, if (and under which conditions) it is possible to define a "definitional extension institutional monad", in which a derivation consists of the colimit of all "suitable" definitional extensions. It seems promising to consider syntactic definitional extensions, i.e. those that induce a mapping on the sentence level that is compatible with the model expansion. Another interesting point concerns the development of a general way to construct institutional monads, that would provide a kind of canoncial derivation. Here the idea of a charter [12] may provide a starting point.

On a more general level, this approach shows again, that notions from basic category theory (monads and Kleisli construction) can be adopted to institutions and lead to useful concepts there. It naturally leads to the question, if related notions, like the Eilenberg-Moore construction, can give raise to meaningful applications in an institutional setting as well.

References

1. nLab: Span. http://ncatlab.org/nlab/show/span
2. Borzyszkowski, T.: Logical systems for structured specifications. Theor. Comput. Sci. **286**, 197–245 (2002)
3. Cornelius, F., Baldamus, M., Ehrig, H., Orejas, F.: Abstract and behaviour module specifications. Math. Struct. Comput. Sci. **9**(1), 21–62 (1999)
4. Diaconescu, R.: Grothendieck institutions. Appl. Categorical Struct. **10**, 383–402 (2002)
5. Diaconescu, R.: Herbrand theorems in arbitrary institutions. Inf. Process. Lett. **90**, 29–37 (2004)
6. Diaconescu, R.: Institution-Independent Model Theory. Birkhäuser, Basel (2008)
7. Diaconescu, R., Goguen, J., Stefaneas, P.: Logical support for modularisation. In: Huet, G., Plotkin, G. (eds.) Proceedings of a Workshop on Logical Frameworks (1991)
8. Diskin, Z., Kadish, B.: A graphical yet formalized framework for specifying view systems. In: Manthey, R., Wolfengagen, V. (eds.) Advances in Databases and Information Systems 1997, Proceedings of the First East-European Symposium on Advances in Databases and Information Systems, ADBIS 1997, St Petersburg, 2–5 September 1997 (1997)
9. Diskin, Z., Maibaum, T., Czarnecki, K.: Intermodeling, queries, and kleisli categories. In: de Lara, J., Zisman, A. (eds.) Fundamental Approaches to Software Engineering. LNCS, vol. 7212, pp. 163–177. Springer, Heidelberg (2012)
10. Ehrig, H., Baldamus, M., Cornelius, F., Orejas, F.: Theory of algebraic module specification including behavioral semantics and constraints. In: Nivat, M., Rattray, C., Rus, T., Scollo, G. (eds.) AMAST 1991. Workshops in Computing, pp. 145–172. Springer, Heidelberg (1992)
11. Ehrig, H., Baldamus, M., Orejas, F.: New concepts of amalgamation and extension for a general theory of specifications. In: Bidoit, M., Choppy, C. (eds.) Abstract Data Types 1991 and COMPASS 1991. LNCS, vol. 655. Springer, Heidelberg (1993)
12. Goguen, J.A., Burstall, R.M.: A study in the foundations of programming methodology: specifications, institutions, charters and parchments. In: Poigné, A., Pitt, D.H., Rydeheard, D.E., Abramsky, S. (eds.) Category Theory and Computer Programming. LNCS, vol. 240, pp. 313–333. Springer, Heidelberg (1986)
13. Goguen, J.A., Burstall, R.M.: Institutions: Abstract model theory for specification and programming. J. Assoc. Comput. Mach. **39**, 95–146 (1992). Predecessor in: LNCS, vol. 164, pp. 221–256 (1984)
14. Goguen, J.A., Roşu, G.: Institution morphisms. Formal Aspects Comput. **13**, 274–307 (2002)
15. Goguen, J.A., Thatcher, J.W., Wagner, E.G.: An initial algebra approach to the specification, correctness and implementation of abstract data types. In: Yeh, R.T. (ed.) Current Trends in Programming Methodology - vol. IV: Data Structuring, pp. 80–149. Prentice-Hall (1978)

16. Honsell, F., Longley, J., Sannella, D., Tarlecki, A.: Constructive data refinement in typed lambda calculus. In: Tiuryn, J. (ed.) FOSSACS 2000. LNCS, vol. 1784, pp. 161–176. Springer, Heidelberg (2000)
17. Kutz, O., Mossakowski, T., Lücke, D.: Carnap, goguen, and the hyperontologies: logical pluralism and heterogeneous structuring in ontology design. Log. Univers. **4**(2), 255–333 (2010)
18. Lack, S.: A 2-categories companion. In: Baez, J.C., May, J.P. (eds.) Towards Higher Categories. The IMA Volumes in Mathematics and its Applications, vol. 152, pp. 105–191. Springer, New York (2010)
19. Mossakowski, T., Autexier, S., Hutter, D.: Development graphs - proof management for structured specifications. J. Logic Algebraic Program. **67**(1–2), 114–145 (2006). http://www.sciencedirect.com/science?_ob=GatewayURL&_origin=CONTENTS&_method=citationSearch&_piikey=S1567832605000810&_version=1&md5=7c18897e9ffad42e0649c6b41203f41e
20. Sannella, D.T., Burstall, R.M.: Structured theories in LCF. In: Protasi, M., Ausiello, G. (eds.) CAAP 1983. LNCS, vol. 159, pp. 377–391. Springer, Heidelberg (1983)
21. Sannella, D., Tarlecki, A.: Foundations of Algebraic Specification and Formal Software Development. Monographs in Theoretical Computer Science. Springer, Berlin (2012)
22. Schröder, L., Mossakowski, T., Tarlecki, A., Klin, B., Hoffman, P.: Amalgamation in the semantics of casl. Theor. Comput. Sci. **331**(1), 215–247 (2005)
23. Schwering, A., Krumnack, U., Kühnberger, K.U., Gust, H.: Syntactic principles of heuristic-driven theory projection. J. Cogn. Syst. Res. **10**(3), 251–269 (2009). Special Issue on Analogies - Integrating Cognitive Abilities
24. Szigeti, J.: On limits and colimits in the Kleisli category. Cahiers de Topologie et Géométrie Différentielle Catégoriques **24**(4), 381–391 (1983)