Structured Theories and Institutions

Francisco Durán and José Meseguer

Computer Science Laboratory, SRI International
Menlo Park, CA 94025, USA

Abstract

Category theory provides an excellent foundation for studying structured specifications and their composition. For example, theories can be structured together in a diagram, and their composition can be obtained as a colimit. There is, however, a growing awareness, both in theory and in specification practice, that structured theories should not be viewed just as the "scaffolding" used to build unstructured theories: they should become *first-class citizens* in the specification process. Given a logic formalized as an institution $\mathcal{I}$, we therefore ask whether there is a good definition of the category of $\mathcal{I}$-structured theories, and whether they can be naturally regarded as the ordinary theories of an appropriate institution $\mathcal{S}(\mathcal{I})$ generalizing the original institution $\mathcal{I}$. We answer both question in the affirmative, and study good properties of the institution $\mathcal{I}$ inherited by $\mathcal{S}(\mathcal{I})$. We show that, under natural conditions, a number of important properties are indeed inherited, including cocompleteness of the category of theories, liberality, and extension of the basic framework by freeness constraints. The results presented here have been used as a foundation for the module algebra of the Maude language, and seem promising as a semantic basis for a generic module algebra that could be both specified and executed within the logical framework of rewriting logic.

1 Introduction

Structuring mechanisms are vital means for reusing software and for mastering the complexity of large systems at all levels, including specifications and code. Category theory provides an excellent foundation for studying structured specifications and their composition. A key contribution in the late seventies and early eighties was made by Burstall and Goguen with the Clear [4] specification language, that proposed taking colimits of theories as a systematic way of "putting theories together". Clear was based on many-sorted equational logic, but its categorical semantics was in fact logic-independent. This led Goguen and Burstall to propose the notion of *institution* as an axiomatization of a general logic, and to generalize the Clear-like operations to institutions [17,18]. These ideas have had a great theoretical and practical impact: see the bibliographies [3,16], the survey [24], and for sample references on logic-independent specification building operations, e.g., [27,10,2,25].

Typically, theory composition operations begin with theories structured in some way—for example, a diagram—and result in an unstructured, or less structured, specification as their result—for example, a colimit. That is, structured theories are often "flattened" when being composed. There are however good reasons for preserving their structure. Besides the obvious understandability and design documentation reasons, it is often very useful to consider theory-building operations whose results are structured theories. For example, refining a software design can be best understood as refining structured theories [30]; also, even when we may want to extract a flattened theory, it can be much more efficient to operate at the level of structured theories [13,12]. There are also more intrinsic reasons, namely, when the semantics associated to a structured module *essentially depends on its structure*. For example, we often want to associate to the inclusion of a parameter theory into the body of a parameterized specification a *freeness constraint*, requiring that the models of the body are free extensions of the models of the parameter; more generally, one can similarly consider other notions of constraint [26,4,28,18,15]. In practice, the need for keeping and using structure is both recognized and supported by a number of languages and systems such as, for example, languages in the Clear/OBJ tradition [4,19,9,6], SPECWARE [30], and CASL [8].

Although a number of concepts and techniques have been suggested both at the theoretical and specification language levels to keep and use the necessary amount of structure for specific purposes, the most satisfactory way of addressing the need for preserving structure is to make structured theories *first-class citizens*. In the categorical spirit, this leads to seeking a good definition of the category of structured theories, and to investigating whether structured theories can naturally be regarded as the ordinary theories of an appropriate institution. The most basic form of structured theory is that of a *hierarchy* of theory inclusions, in the sense that more complex forms of structured theories can often be *normalized* to hierarchies [13,12], perhaps keeping some additional information such as freeness constraints. Hierarchies are of course special kinds of *diagrams*, and this suggests using categories of diagrams and categorical constructions on diagrams as the theoretical basis.

The use of diagrams for structuring purposes has also been emphasized by other authors. In a limited form they were used in Clear to deal with shared structure in categorical constructions by means of *based theories* [4]. In SPECWARE [30] diagrams are first-class citizens and are used to structure and refine specifications; furthermore, an appropriate diagram category is defined in such a way that a colimit-like functor yields an operation of horizontal composition satisfying, by functoriality, the expected laws of compatibility between horizontal and vertical composition [30]. Based on the SPECWARE ideas, Dimitrakos has proposed a way of parameterizing specifications by diagrams of specifications, and of inducing an instantiation by means of a family of parallel instantiating morphisms whose sources are the components of the parameter diagram [11].

In this paper we address a number of issues about structured theories that, as far as we know, have not been systematically studied before. The most basic issue is: given an institution $\mathcal{I}$, can we naturally associate to it another institution $\mathcal{S}(\mathcal{I})$ whose ordinary theories are the structured theories of $\mathcal{I}$? We answer this question in the affirmative, and then proceed to study to what extent good properties of the institution $\mathcal{I}$ are also inherited by $\mathcal{S}(\mathcal{I})$. We show that, under natural conditions, a number of important properties are indeed inherited, including cocompleteness of the category of theories, liberality, and extension of the basic framework by freeness constraints.

We have used the present work as the theoretical foundation for the module algebra of the Maude specification language [6]. In this module algebra, structured theories are first-class citizens, and module operations result in other structured theories [13,12]. Using the fact that rewriting logic is reflective [7,5], the entire module algebra is both specified and executed within the logic of Maude [12]. As we further explain in the conclusions, using the logic-independent semantics for structured theory compositions developed in this paper and the logical framework properties of Maude [21], we plan to generalize Maude's module algebra to an executable *generic module algebra* that could be instantiated for any logic represented in the framework.

The rest of the paper is organized as follows. Section 2 reviews some basic definitions about institutions; Section 3 gives basic results about categories of diagrams; Section 4 presents our main definitions and results about the institution $\mathcal{S}(\mathcal{I})$ of structured $\mathcal{I}$-theories and its properties; and Section 5 offers some concluding remarks.

2 Institutions

The theory of institution [18] allows us to discuss the relationship between theories and models without committing ourselves to a particular logical system.

Definition 2.1 *[17]* An *institution* $\mathcal{I}$ is a 4-tuple $(\mathbf{Sign}_{\mathcal{I}}, \mathrm{sen}_{\mathcal{I}}, \mathrm{Mod}_{\mathcal{I}}, \models)$ such that:

- $\mathbf{Sign}_{\mathcal{I}}$ is a category whose objects are called *signatures*,
- $\mathrm{sen}_{\mathcal{I}} \colon \mathbf{Sign}_{\mathcal{I}} \longrightarrow \mathbf{Set}$ is a functor associating to each signature Σ a set of Σ-*sentences*,
- $\mathrm{Mod}_{\mathcal{I}} \colon \mathbf{Sign}_{\mathcal{I}} \longrightarrow \mathbf{Cat}^{\mathrm{op}}$ is a functor mapping each signature Σ to a category whose objects are called Σ-*models*, and
- $\models$ is a function associating to each $\Sigma \in |\mathbf{Sign}_{\mathcal{I}}|$ a binary relation $\models_\Sigma \subseteq |\mathrm{Mod}_{\mathcal{I}}(\Sigma)| \times \mathrm{sen}_{\mathcal{I}}(\Sigma)$ called *satisfaction* in such a way that the following

property holds for any $M' \in |\mathrm{Mod}_{\mathcal{I}}(\Sigma')|$, $H : \Sigma \to \Sigma'$, $\varphi \in \mathrm{sen}_{\mathcal{I}}(\Sigma)$:

$$M' \models_{\Sigma'} \mathrm{sen}_{\mathcal{I}}(H)(\varphi) \iff \mathrm{Mod}_{\mathcal{I}}(H)(M') \models_{\Sigma} \varphi.$$

Given a signature Σ, a presentation of a theory is given by a set Γ of Σ-sentences. We can therefore denote a theory presentation as a pair (Σ, Γ). Given a presentation (Σ, Γ), we define the category $\mathrm{Mod}_{\mathcal{I}}(\Sigma, \Gamma)$ as the full subcategory of $\mathrm{Mod}_{\mathcal{I}}(\Sigma)$ determined by those models $M \in |\mathrm{Mod}_{\mathcal{I}}(\Sigma)|$ that satisfy all the sentences in Γ, i.e., $M \models_{\Sigma} \varphi$ for each $\varphi \in \Gamma$.

We can extend the satisfaction relation to sets of sentences as follows.

$$M \models_{\Sigma} \Gamma \text{ iff } M \models_{\Sigma} \varphi \text{ for each } \varphi \in \Gamma.$$

Then, the relation between sets of sentences and sentences given by

$$\Gamma \models_{\Sigma} \varphi \text{ iff } M \models_{\Sigma} \varphi \text{ for each } M \in |\mathrm{Mod}_{\mathcal{I}}(\Sigma, \Gamma)|$$

allows us to associate to an institution an entailment system in the sense of [22]. For any signature Σ, the *closure* of a set Γ of Σ-sentences is $\Gamma^{\bullet} = \{\varphi \mid \Gamma \models_{\Sigma} \varphi\}$. The Σ-theory presented by (Σ, Γ) is then given by $(\Sigma, \Gamma^{\bullet})$.

Given presentations of theories (Σ, Γ) and (Σ', Γ'), a theory morphism $H \colon (\Sigma, \Gamma) \to (\Sigma', \Gamma')$ is a signature morphism $H \colon \Sigma \to \Sigma'$ such that if $\varphi \in \Gamma$ then $\mathrm{sen}_{\mathcal{I}}(H)(\varphi) \in \Gamma'^{\bullet}$, that is, for all $\varphi \in \Gamma$, $\Gamma' \models_{\Sigma'} \mathrm{sen}_{\mathcal{I}}(H)(\varphi)$.

Definition 2.2 Given an institution $\mathcal{I}$, its category $\mathbf{Th}_{\mathcal{I}}$ of *theories* [1] has as objects presentations of theories (Σ, Γ) and as arrows theory morphisms. We denote by $\mathrm{sign}_{\mathcal{I}} \colon \mathbf{Th}_{\mathcal{I}} \to \mathbf{Sign}_{\mathcal{I}}$ the forgetful functor sending each theory to its underlying signature.

For any institution $\mathcal{I}$, the model functor $\mathrm{Mod}_{\mathcal{I}} \colon \mathbf{Sign}_{\mathcal{I}} \longrightarrow \mathbf{Cat}^{\mathrm{op}}$ extends to a functor $\mathrm{Mod}_{\mathcal{I}} : \mathbf{Th}_{\mathcal{I}} \to \mathbf{Cat}^{op}$, by mapping a theory (Σ, Γ) to the full subcategory $\mathrm{Mod}_{\mathcal{I}}(\Sigma, \Gamma)$ of $\mathrm{Mod}_{\mathcal{I}}(\Sigma)$. The institution $\mathcal{I}$ is called *liberal* if for each theory morphism $H : (\Sigma, \Gamma) \to (\Sigma', \Gamma')$ the functor $\mathrm{Mod}_{\mathcal{I}}(H)$ has a left adjoint. We call $\mathcal{I}$ *exact* if $\mathrm{Mod}_{\mathcal{I}} \colon \mathbf{Th}_{\mathcal{I}} \to \mathbf{Cat}^{op}$ preserves colimits.

3 Diagram Categories

The issue of whether the category $Dg(\mathcal{C})$ of diagrams over a category $\mathcal{C}$ has colimits is important, because for $\mathcal{C} = \mathbf{Sign}_{\mathcal{I}}$ this specializes to colimits of structured signatures, which can then be used to define colimits of structured theories. We show in this section that, if $\mathcal{C}$ is cocomplete, then $Dg(\mathcal{C})$ is also cocomplete. This is probably a "folklore" result. Since we are not aware of a

[1] Note that in the above definition the objects of $\mathbf{Th}_{\mathcal{I}}$ are *presentations* of theories. We follow here the terminology of Meseguer's general logics [22], instead of Goguen and Burstall's original definition [18]. In what follows, when we talk about a theory (Σ, Γ) we shall mean a theory presentation.

suitable textbook exposition to give as a reference, we include it here to make the paper self-contained.

Given a cocomplete category $\mathcal{C}$ and a small category $\mathcal{A}$, the category of functors from $\mathcal{A}$ to $\mathcal{C}$, which we denote by $\mathcal{C}^{\mathcal{A}}$, is also a cocomplete category [20]. Furthermore, in $\mathcal{C}^{\mathcal{A}}$ all colimits can be constructed pointwise.

Theorem 3.1 (Left Kan extensions [20,29]) *Let $\mathcal{B}$ be a small category, let $F\colon \mathcal{B} \to \mathcal{D}$ be a functor, and let $\mathcal{C}$ be a cocomplete category. Then, the functor $\tilde{F} = \mathcal{C}^F : \mathcal{C}^{\mathcal{D}} \to \mathcal{C}^{\mathcal{B}}$ has a left adjoint $\tilde{V}_F : \mathcal{C}^{\mathcal{B}} \to \mathcal{C}^{\mathcal{D}}$, called the* left Kan extension along *F.*

Definition 3.2 *[29]* Let $\mathcal{C}$ be a category. The *diagram category $Dg(\mathcal{C})$* has as objects functors $T\colon \mathcal{P} \to \mathcal{C}$, where $\mathcal{P}$ is a small category. If $T\colon \mathcal{P} \to \mathcal{C}$ and $T'\colon \mathcal{P}' \to \mathcal{C}$ are objects, then a morphism $(R, \varrho)\colon T \to T'$ consists of a functor $R\colon \mathcal{P} \to \mathcal{P}'$ and a natural transformation $\varrho\colon T \to T' \cdot R$.

The composition of morphisms (R, ϱ) and (R', ϱ'), as depicted in the figure below, is given by the morphism $(R' \cdot R,\ \varrho' R \cdot \varrho)$.

$$\begin{tikzcd}
\mathcal{P} \arrow[d, "R"'] \arrow[drr, "T"] \arrow[dr, phantom, "\Downarrow\varrho"] & & \\
\mathcal{P}' \arrow[d, "R'"'] \arrow[rr, "T'"] \arrow[dr, phantom, "\Downarrow\varrho'"] & & \mathcal{C} \\
\mathcal{P}'' \arrow[urr, "T''"'] & &
\end{tikzcd}$$

Theorem 3.3 *Let $\mathcal{C}$ be a cocomplete category. Then, the category $Dg(\mathcal{C})$ is also cocomplete.*

Since a category with pushouts and coproducts has all colimits, we split the proof of Theorem 3.3 in two separate lemmas. The proofs summarize the main constructions; a detailed exposition can be found in [12].

Lemma 3.4 *If $\mathcal{C}$ is cocomplete, then $Dg(\mathcal{C})$ has pushouts.*

Proof. Given small categories $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{P}_2$, diagrams D_0 in $\mathcal{C}^{\mathcal{P}_0}$, D_1 in $\mathcal{C}^{\mathcal{P}_1}$, and D_2 in $\mathcal{C}^{\mathcal{P}_2}$, and diagram morphisms $(F, \alpha) : D_0 \to D_1$ and $(J, \beta) : D_0 \to D_2$, we need to construct a pushout object D_3 in $\mathcal{C}^{\mathcal{P}_3}$, and corresponding morphisms in $Dg(\mathcal{C})$, as depicted in the following figure:

$$\begin{tikzcd}
D_2 \arrow[r, dotted, "(F',\delta)"] & D_3 \\
D_0 \arrow[u, "(J,\beta)"] \arrow[r, "(F,\alpha)"'] & D_1 \arrow[u, dotted, "(J',\gamma)"']
\end{tikzcd}$$

First, we define the small category $\mathcal{P}_3$, with $J'\colon \mathcal{P}_1 \to \mathcal{P}_3$ and $F'\colon \mathcal{P}_2 \to \mathcal{P}_3$, as the pushout of F and J in **Cat**. The intuitive idea in order to build up the desired diagram D_3 is the following: if the D_i were all in the same category $\mathcal{C}^{\mathcal{P}}$, then the pushout could be constructed pointwise. We obtain the more general construction by taking the Kan extensions along the corresponding functors to $\mathcal{P}_3$ for each of these diagrams, thus "moving" them all to $\mathcal{C}^{\mathcal{P}_3}$;

D_3 is then the pushout of the Kan-extended diagrams in $\mathcal{C}^{\mathcal{P}_3}$, which can be computed pointwise.

Lemma 3.5 *For any category $\mathcal{C}$, $Dg(\mathcal{C})$ has coproducts.*

Proof. Let $\{D_i : J_i \to \mathcal{C}\}_{i\in I}$ be an I-indexed family of diagrams for any set I. Let $\coprod_{i\in I} J_i$ be the coproduct of $\{J_i\}_{i\in I}$ in **Cat**, and let us denote the inclusion morphisms by $\iota_i : J_i \to \coprod_{i\in I} J_i$. It is easy to check that the induced functor $D : \coprod_{i\in I} J_i \to \mathcal{C}$, with the induced family of inclusion morphisms $\{(\iota_i, 1_{D_i}) : D_i \to D\}_{i\in I}$, is the coproduct of the indexed family $\{D_i\}_{i\in I}$ in $Dg(\mathcal{C})$.

4 Structured Theories

In this section we define the institution $\boldsymbol{\mathcal{S}(\mathcal{I})}$ of structured theories over a given institution $\boldsymbol{\mathcal{I}}$, and give some results about the cocompleteness of its categories of signatures and theories, the liberality of $\boldsymbol{\mathcal{S}(\mathcal{I})}$, and the addition of freeness constraints to structured theories.

4.1 The Institution of Structured Theories

A *structured signature* can be formalized as a functor $D : I \to \mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ from a small category I to the category $\mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ of signatures and signature morphisms in a given institution $\boldsymbol{\mathcal{I}}$. This is of course a quite general notion. One can specialize the concept to the more familiar concept of hierarchy of signatures by requiring that I is a finite poset and that all the arrows in the diagram are inclusions in an appropriate subcategory of inclusion morphisms. Based on this definition, we build an institution $\boldsymbol{\mathcal{S}(\mathcal{I})}$, whose theories are called *structured $\boldsymbol{\mathcal{I}}$-theories*, by defining functors $\mathrm{sen}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ and $\mathrm{Mod}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ associating to each structured signature D in $\mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ a set of D-sentences and a category of D-models, respectively. Then, we give a satisfaction relation for it and show that the satisfaction condition holds.

Definition 4.1 Let us denote by $\mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ the category $Dg(\mathbf{Sign}_{\boldsymbol{\mathcal{I}}})$ of diagrams over the category of signatures in the institution $\boldsymbol{\mathcal{I}}$. We shall call the objects of $\mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ *structured ($\boldsymbol{\mathcal{I}}$-)signatures*, and will denote each structured signature by its corresponding diagram $D : I \to \mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$. The morphisms in $\mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ are called *structured signature morphisms.*

Definition 4.2 The functor $\mathrm{sen}_{\boldsymbol{\mathcal{S}(\mathcal{I})}} : \mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}} \to \mathbf{Set}$, associating to each structured signature $D : I \to \mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ a set of sentences and to each structured signature morphism $(K, H) : D \to D'$ a corresponding translation at the level of sentences, is defined as follows:

$$\mathrm{sen}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}(D) = \coprod_{i\in I} \mathrm{sen}_{\boldsymbol{\mathcal{I}}}(D(i))$$

$$\mathrm{sen}_{\mathcal{S}(\mathcal{I})}((K,H)) = \coprod_{i\in I} \mathrm{sen}_{\mathcal{I}}(H_i)$$

We can see each of the sentences of D as a pair (i,φ), where φ is a sentence in $\mathrm{sen}_{\mathcal{I}}(D(i))$. Note that, given a structured signature morphism $(K,H)\colon D \to D'$ and a sentence (i,φ) of D, we have

$$\mathrm{sen}_{\mathcal{S}(\mathcal{I})}((K,H))((i,\varphi)) = (K(i), \mathrm{sen}_{\mathcal{I}}(H_i)(\varphi)).$$

Definition 4.3 Given a structured signature $D : I \to \mathbf{Sign}_{\mathcal{I}}$, its category of models $\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}(D)$ has as objects families $M = \{M_i\}_{i\in I}$ with M_i in $\mathrm{Mod}_{\mathcal{I}}(D(i))$, such that for each $\phi : i \to j$ in I, $\mathrm{Mod}_{\mathcal{I}}(D(\phi))(M_j) = M_i$. A morphism between two such models $f : M \to M'$ is given by a family $\{f_i\colon M_i \to M'_i\}_{i\in I}$ with f_i in $\mathrm{Mod}_{\mathcal{I}}(D(i))$ such that for each $\phi\colon i \to j$ in I, $\mathrm{Mod}_{\mathcal{I}}(D(\phi))(f_j) = f_i$.

Definition 4.4 The functor $\mathrm{Mod}_{\mathcal{S}(\mathcal{I})} : \mathbf{Sign}_{\mathcal{S}(\mathcal{I})} \to \mathbf{Cat}^{\mathrm{op}}$ assigns to each structured signature $D\colon I \to \mathbf{Sign}_{\mathcal{I}}$ its category of models $\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}(D)$, and to each structured signature morphism $(K,H)\colon D \to D'$ the forgetful functor $\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))\colon \mathrm{Mod}_{\mathcal{S}(\mathcal{I})}(D') \to \mathrm{Mod}_{\mathcal{S}(\mathcal{I})}(D)$, defined as follows:

$$\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))(\{M'_j\}_{j\in I'}) = \{\mathrm{Mod}_{\mathcal{I}}(H_i)(M'_{K(i)})\}_{i\in I}$$

$$\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))(\{f'_j\}_{j\in I'}) = \{\mathrm{Mod}_{\mathcal{I}}(H_i)(f'_{K(i)})\}_{i\in I}$$

Definition 4.5 Given a structured signature D, a D-model $M = \{M_i\}_{i\in I}$ satisfies a D-sentence (i,φ) if and only if $M_i \models_{D(i)} \varphi$. In this case, we write $M \models_D (i,\varphi)$.

Proposition 4.6 (Satisfaction Condition) *Let $D\colon I \to \mathbf{Sign}_{\mathcal{I}}$ and $D'\colon I' \to \mathbf{Sign}_{\mathcal{I}}$ be structured signatures, and let $(K,H) : D \to D'$ be a structured signature morphism. Given a D-sentence (i,φ) and a D'-model M', then*

$$\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))(M') \models_D (i,\varphi) \iff M' \models_{D'} \mathrm{sen}_{\mathcal{S}(\mathcal{I})}((K,H))((i,\varphi)).$$

Proof. $\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))(M')$ is a D-model, namely, the family of models

$$\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))(M') = \{\mathrm{Mod}_{\mathcal{I}}(H_i)(M'_{K(i)})\}_{i\in I}.$$

Since φ is a sentence in $D(i)$, we have

$$\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))(M') \models_D (i,\varphi) \iff \mathrm{Mod}_{\mathcal{I}}(H_i)(M'_{K(i)}) \models_{D(i)} \varphi.$$

On the other hand, since $\mathrm{sen}_{\mathcal{S}(\mathcal{I})}((K,H))((i,\varphi)) = (K(i), \mathrm{sen}_{\mathcal{I}}(H_i)(\varphi))$, and thus $\mathrm{sen}_{\mathcal{I}}(H_i)(\varphi)$ is a sentence in $D'(K(i))$, we have

$$M' \models_{D'} \mathrm{sen}_{\mathcal{S}(\mathcal{I})}((K,H))((i,\varphi)) \iff M'_{K(i)} \models_{D'(K(i))} \mathrm{sen}_{\mathcal{I}}(H_i)(\varphi).$$

By the satisfaction condition for the institution $\mathcal{I}$, we also have

$$\mathrm{Mod}_{\mathcal{I}}(H_i)(M'_{K(i)}) \models_{D(i)} \varphi \Longleftrightarrow M'_{K(i)} \models_{D'(K(i))} \mathrm{sen}_{\mathcal{I}}(H_i)(\varphi),$$

and therefore,

$$\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}((K,H))(M') \models_D (i,\varphi) \Longleftrightarrow M' \models_{D'} \mathrm{sen}_{\mathcal{S}(\mathcal{I})}((K,H))((i,\varphi)).$$

Definition 4.7 Let $\mathcal{S}(\mathcal{I})$ be the institution with:

- $\mathbf{Sign}_{\mathcal{S}(\mathcal{I})}$ as category of signatures,
- the sentence functor $\mathrm{sen}_{\mathcal{S}(\mathcal{I})}\colon \mathbf{Sign}_{\mathcal{S}(\mathcal{I})} \to \mathbf{Set}$, of Definition 4.2,
- the model functor $\mathrm{Mod}_{\mathcal{S}(\mathcal{I})}\colon \mathbf{Sign}_{\mathcal{S}(\mathcal{I})} \to \mathbf{Cat}^{\mathrm{op}}$, of Definition 4.4, and
- the satisfaction relation given in Definition 4.5, for which the satisfaction condition holds as shown in Proposition 4.6.

Note that the notion of structured $\mathcal{I}$-theory, that is, of a theory presentation in $\mathcal{S}(\mathcal{I})$, captures well the intuitive notion of structured theory encountered in actual specifications. Indeed, when a subtheory is imported, its axioms typically *are not repeated again*; they are implicitly inherited from the subtheory. This means that axioms are presented *locally*, for a specific local signature $D(i)$, corresponding to our formal notion of a pair (i,φ). It also means that at each stage in the specification only the *incremental information* of additional axioms has to be made explicit.

Since $\mathbf{Sign}_{\mathcal{S}(\mathcal{I})} = Dg(\mathbf{Sign}_{\mathcal{I}})$, there should be a close and systematic relationship between the category $\mathbf{Th}_{\mathcal{S}(\mathcal{I})}$ of structured $\mathcal{I}$-theories and the diagram category $Dg(\mathbf{Th}_{\mathcal{I}})$. We can express this relationship as an adjunction with particularly good properties.

Let $J\colon \mathbf{Th}_{\mathcal{S}(\mathcal{I})} \to Dg(\mathbf{Th}_{\mathcal{I}})$ be the functor defined on objects by the rule

$$(D,\Gamma) \mapsto D_{\Gamma^*},$$

where if $D: I \to \mathbf{Sign}_{\mathcal{I}}$ is a structured signature, then $D_{\Gamma^*}: I \to \mathbf{Th}_{\mathcal{I}}$ has $D_{\Gamma^*}(i) = (D(i),\Gamma_i^*)$ and $D_{\Gamma^*}(\phi\colon i \to j) = D(\phi)$, where

$$\Gamma_i^* = \{\varphi \in \mathrm{sen}_{\mathcal{I}}(D(i)) \mid \forall M \in \mathrm{Mod}_{\mathcal{S}(\mathcal{I})}(D,\Gamma), M_i \models_{D(i)} \varphi\}.$$

Note that then $D(\phi)\colon D_{\Gamma^*}(i) \to D_{\Gamma^*}(j)$ is a theory morphism because

$$\begin{aligned}\varphi \in \Gamma_i^* \Leftrightarrow\, & \forall M \in \mathrm{Mod}_{\mathcal{S}(\mathcal{I})}(D,\Gamma),\ M_i \models_{D(i)} \varphi\\ & \Leftrightarrow \text{(by satisfaction, with } \mathrm{Mod}_{\mathcal{I}}(\phi)(M_j) = M_i)\\ & \quad \forall M \in \mathrm{Mod}_{\mathcal{S}(\mathcal{I})}(D,\Gamma),\ M_j \models_{D(j)} \mathrm{sen}_{\mathcal{I}}(D(\phi))(\varphi)\\ & \Rightarrow \mathrm{sen}_{\mathcal{I}}(D(\phi))(\varphi) \in \Gamma_j^*\end{aligned}$$

Therefore, $\mathrm{sen}_{\mathcal{I}}(D(\phi))(\Gamma_i^*) \subseteq \Gamma_j^*$ and $D(\phi)$ is a theory morphism.

The definition of J on morphisms assigns to each theory morphism $(K,H)\colon (D,\Gamma) \to (D',\Gamma')$ in $\mathbf{Th}_{\mathcal{S}(\mathcal{I})}$ the diagram morphism $(K,\tilde{H})\colon D_{\Gamma^*} \to D'_{\Gamma'^*}$ with

$\tilde{H}_i = H_i$ for each $i \in I$, which is well-defined by observing that $\Gamma'^{\bullet} = \coprod_{j \in I'} \Gamma'^{*}_{j}$, and using the fact that (K, H) is a theory morphism, so that for each $(i, \varphi) \in \Gamma$ we have $(K(i), \mathrm{sen}_{\boldsymbol{\mathcal{I}}}(H_i)(\varphi)) \in \Gamma'^{\bullet}$.

Let $R\colon Dg(\mathbf{Th}_{\boldsymbol{\mathcal{I}}}) \to \mathbf{Th}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ be the functor defined on objects by the rule

$$D \mapsto (\mathrm{sign}_{\boldsymbol{\mathcal{I}}} \cdot D,\ \coprod_{i \in I} \mathrm{ax}(D(i))),$$

where, for (Σ, Γ) a theory, we use the notation $\mathrm{ax}(\Sigma, \Gamma) = \Gamma$.

Note that for any theory (D, Γ) in $\mathbf{Th}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ we have a natural isomorphism $RJ(D, \Gamma) \overset{\varepsilon(D,\Gamma)}{\simeq} (D, \Gamma)$. Indeed, by construction we have $RJ(D, \Gamma) = (D, \Gamma^{\bullet})$. Therefore, both theories are isomorphic, with the identity signature morphism as the isomorphism. We do not need to define R on morphisms, since such a definition follows automatically from the adjunction result below.

Proposition 4.8 *The functor $J : \mathbf{Th}_{\boldsymbol{\mathcal{S}(\mathcal{I})}} \to Dg(\mathbf{Th}_{\boldsymbol{\mathcal{I}}})$ is full and faithful, with R left adjoint to J and ε as the counit.*

Proof. By Theorem 1 of Section IV.3 in [20], if ε is a counit and is an isomorphism, J is full and faithful. So we just have to check the adjunction. A detailed proof showing that $\varepsilon : RJ \to 1_{Dg(\mathbf{Th}_{\boldsymbol{\mathcal{I}}})}$ is indeed the counit of the adjunction can be found in [12].

4.2 Cocompleteness and Liberality

Given the institution $\boldsymbol{\mathcal{S}(\mathcal{I})}$, we now present some results on the cocompleteness of its categories of signatures and theories, and on the liberality of $\boldsymbol{\mathcal{S}(\mathcal{I})}$.

Theorem 4.9 *If $\mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ is cocomplete then $\mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ is cocomplete.*

Proof. Since $\mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}} = Dg(\mathbf{Sign}_{\boldsymbol{\mathcal{I}}})$, this follows from Theorem 3.3.

By the following result from [18], it follows that, given any institution $\boldsymbol{\mathcal{I}}$, its category of theories is cocomplete if its category of signatures is cocomplete.

Theorem 4.10 [18] *If $\boldsymbol{\mathcal{I}}$ is an institution such that $\mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ is cocomplete, then $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$ is also cocomplete and the forgetful functor $\mathrm{sign}_{\boldsymbol{\mathcal{I}}} : \mathbf{Th}_{\boldsymbol{\mathcal{I}}} \to \mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ preserves colimits.*

Corollary 4.11 *If $\mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ is cocomplete then $\mathbf{Th}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ is cocomplete, and the functor $\mathrm{sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}} : \mathbf{Th}_{\boldsymbol{\mathcal{S}(\mathcal{I})}} \to \mathbf{Sign}_{\boldsymbol{\mathcal{S}(\mathcal{I})}}$ preserves colimits.*

The above result by Goguen and Burstall gives a simple criterion to show when all colimits in the category of theories exist. However, it may not be sufficient. We can have institutions with cocomplete categories of theories whose categories of signatures lack some colimits. For example, if we consider signature morphisms in many-sorted equational logic that can map *operations* to *terms* (as it is allowed for *views* in OBJ [19] and in Maude [6], for example), the category of signatures fails to be cocomplete. The following shows that, independently of the cocompleteness of $\mathbf{Sign}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}$, if $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$ is cocomplete then $\mathbf{Th}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}$ is also cocomplete.

Theorem 4.12 *If* $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$ *is cocomplete then* $\mathbf{Th}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}$ *is cocomplete.*

Proof. If $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$ is cocomplete, then $Dg(\mathbf{Th}_{\boldsymbol{\mathcal{I}}})$ is cocomplete by Theorem 3.3. Then, since ε is a natural isomorphism, any diagram Ω in $\mathbf{Th}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}$ is isomorphic to the diagram $RJ\Omega$. But, since R is a left adjoint to J, and therefore preserves colimits, and since $Dg(\mathbf{Th}_{\boldsymbol{\mathcal{I}}})$ has colimits by Theorem 3.3, we have $R(\text{colim } J\Omega) = \text{colim } (RJ\Omega) = \text{colim } (\Omega)$.

So far, we have only considered the category $\text{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D)$ of models of a structured theory D over a fixed institution $\boldsymbol{\mathcal{I}}$. We shall see in Section 4.3 how we can easily add freeness constraints to this setting. Liberality of an institution is an important property for freeness constraints.

Theorem 4.13 *If an institution* $\boldsymbol{\mathcal{I}}$ *is liberal and exact and* $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$ *is cocomplete, then* $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})$ *is liberal.*

Proof. First of all we observe that, for any $(D,\Gamma) \in |\mathbf{Th}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}|$ we have the isomorphism

$$\text{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D,\Gamma) \simeq \lim\ (\text{Mod}_{\boldsymbol{\mathcal{I}}} \cdot D_{\Gamma^*}). \qquad (\dagger)$$

Indeed, since $\Gamma^\bullet = \coprod_{i\in I} \Gamma_i^*$, (D,Γ) and $(D,\Gamma^\bullet)$ are isomorphic theories. Therefore, for each $M \in |\text{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D)|$ we have $M \models_D \Gamma$ iff $M \models_D \Gamma^\bullet$ iff for each $i \in I$, $M_i \models_{D(i)} \Gamma_i^*$. Therefore, $M \in |\text{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D,\Gamma)|$ iff

(i) $\forall i \in I$, $M_i \in |\text{Mod}_{\boldsymbol{\mathcal{I}}}(D(i),\Gamma_i^*)|$, and

(ii) $\forall \phi\colon i \to j$ in I, $\text{Mod}_{\boldsymbol{\mathcal{I}}}(D(\phi))(M_j) = M_i$,

and similarly, $f\colon M \to M'$ is a morphism in $\text{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D,\Gamma)$ iff

(i) $\forall i \in I$, $f_i\colon M_i \to M_i'$ is a morphism in $\text{Mod}_{\boldsymbol{\mathcal{I}}}(D(i),\Gamma_i^*)$, and

(ii) $\forall \phi\colon i \to j$ in I, $\text{Mod}_{\boldsymbol{\mathcal{I}}}(D(\phi))(f_j) = f_i$.

The above isomorphism ($\dagger$) then follows either by an explicit limit construction in **Cat**, or, more easily, by observing that **Cat** is monadic [1] over $\mathbf{Graph} = \mathbf{Set}^{\cdot\rightrightarrows\cdot}$ (therefore the forgetful functor creates limits [20]) which reduces such a limit construction to a construction of limits in **Set**.

But since $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$ is cocomplete, the colimit of D_{Γ^*} exists, and by exactness of $\boldsymbol{\mathcal{I}}$ we have the isomorphism

$$\lim\ (\text{Mod}_{\boldsymbol{\mathcal{I}}} \cdot D_{\Gamma^*}) \simeq \text{Mod}_{\boldsymbol{\mathcal{I}}}(\text{colim } D_{\Gamma^*}),$$

which combined with the isomorphism ($\dagger$) gives us the isomorphism

$$\text{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D,\Gamma) \simeq \text{Mod}_{\boldsymbol{\mathcal{I}}}(\text{colim } D_{\Gamma^*}).$$

Notice also that, by cocompleteness of $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$, there is a functor

$$\mathrm{colim} : Dg(\mathbf{Th}_{\boldsymbol{\mathcal{I}}}) \to \mathbf{Th}_{\boldsymbol{\mathcal{I}}}$$

such that for any morphism $(K, H) : (D, \Gamma) \to (D', \Gamma')$ in $\mathbf{Th}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}$ we have, thanks to the above isomorphism, the following commutative diagram.

$$\begin{tikzcd}
\mathrm{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D, \Gamma) \arrow[d, phantom, "\wr"] & \mathrm{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}(D', \Gamma') \arrow[l, "\mathrm{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}((K,H))"'] \arrow[d, phantom, "\wr"] \\
\mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(\mathrm{colim}\ D_{\Gamma^*}) & \mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(\mathrm{colim}\ D'_{\Gamma'^*}) \arrow[l, "\mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(\mathrm{colim}\ J(K,H))"]
\end{tikzcd}$$

By liberality of the institution $\boldsymbol{\mathcal{I}}$ there is a left adjoint to $\mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(\mathrm{colim}\ J(K, H))$, which, composed with the vertical isomorphisms, gives rise to a left adjoint to $\mathrm{Mod}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})}((K, H))$.

4.3 Freeness Constraints

One of the key motivations for making structured theories a direct object of study is dealing with *freeness constraints*, called *constraints* or *data constraints* in [18]. They are crucial for the notion of *parameterized module*, in which the model of the parameterized module's body should be a *free extension* of the model of the parameter theory. In many specification languages (e.g., [19,14,9,6,8]) this leads to a distinction between *theories*, with loose semantics, and *modules*, with initial or, more generally, free extension semantics. Both theories and modules can be parameterized, but in the case of parameterized modules, a freeness constraint between models of the parameter and models of the body is enforced.

Intuitively, freeness constraints are associated to particular theory maps appearing in the diagram of a structured theory. Suppose that $\boldsymbol{\mathcal{I}}$ is liberal and that (D, Γ) is a structured theory with $D\colon I \to \mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$, and consider a morphism $\phi\colon i \to j$ in I. Then, we can associate a freeness constraint to the theory map $D(\phi)\colon D_{\Gamma^*}(i) \to D_{\Gamma^*}(j)$ by requiring that the models M of (D, Γ), in addition to satisfying the axioms Γ, satisfy the constraint

$$M_j \simeq F_{D(\phi)}(M_i)$$

for $F_{D(\phi)} : \mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(D_{\Gamma^*}(i)) \to \mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(D_{\Gamma^*}(j))$ the left adjoint to the forgetful functor $\mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(D(\phi))\colon \mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(D_{\Gamma^*}(j)) \to \mathrm{Mod}_{\boldsymbol{\mathcal{I}}}(D_{\Gamma^*}(i))$. For example, $D_{\Gamma^*}(i)$ may be the theory `TRIV`, specifying just one sort `Elt`, and $D_{\Gamma^*}(j)$ may be the theory `LIST` with a sort `List`, specifying lists formed with data elements from the `Elt` parameter sort. Then, the freeness constraint requires that the models of `LIST` are really lists, freely generated from the data elements.

The above notion of freeness constraint should in fact be generalized somewhat, to allow an extra signature map bringing the model to the context in which the constraint is applied. This leads us to the following definition, due to Goguen and Burstall.

Definition 4.14 *[18]* Let $\mathcal{I}$ be an institution. Then a *freeness constraint* on a signature Σ is a pair

$$c = (H\colon T'' \to T',\ G\colon \mathrm{sign}(T') \to \Sigma)$$

with H a theory morphism and G a signature morphism. A Σ-model M *satisfies* c if and only if $\mathrm{Mod}_{\mathcal{I}}(G)(M)$ satisfies T' and $\mathrm{Mod}_{\mathcal{I}}(H)(\mathrm{Mod}_{\mathcal{I}}(G)(M))$ has a free extension along H such that the corresponding component of the counit of the adjunction $\varepsilon_{\mathrm{Mod}_{\mathcal{I}}(G)(M)}\colon F_H(\mathrm{Mod}_{\mathcal{I}}(H)(\mathrm{Mod}_{\mathcal{I}}(G)(M))) \to \mathrm{Mod}_{\mathcal{I}}(G)(M)$ is an isomorphism; in this case we write $M \models_\Sigma c$.

Our intuitive notion of freeness constraint in a structured theory can then be recovered by two special cases of the above definition. The case of a *parameterized module*, illustrated by the theory inclusion `TRIV` $\hookrightarrow$ `LIST` in our previous example, corresponds to freeness constraints of the form

$$(D(\phi)\colon D_{\Gamma^*}(i) \to D_{\Gamma^*}(j),\ 1_{D(j)}\colon D(j) \to D(j)),$$

whereas the case of an *unparameterized module*, like `NAT` or `BOOL`, for which we want an *initial model semantics* corresponds to a freeness constraint of the form

$$(\emptyset_{D_{\Gamma^*}(j)}\colon \emptyset \to D_{\Gamma^*}(j),\ 1_{D(j)}\colon D(j) \to D(j)),$$

where $\emptyset$ is the initial object in the category of signatures of an exact liberal institution $\mathcal{I}$ (so that $\mathrm{Mod}_{\mathcal{I}}(\emptyset)$ has only one model, let us call it also $\emptyset$) because then the initial model of $D_{\Gamma^*}(j)$ coincides with $F_{\emptyset_{D_{\Gamma^*}(j)}}(\emptyset)$.

The need for the more general notion of freeness constraint in Definition 4.14 has to do with translation of constraints by composition with signature morphisms. We think of a constraint $c = (H\colon T'' \to T',\ G\colon \mathrm{sign}(T') \to \Sigma)$ on Σ as a *sentence* associated to the signature Σ. Then, if $Q\colon \Sigma \to \Omega$ is a signature morphism, we can associate to the constraint c the following constraint on Ω:

$$\mathrm{sen}_{\mathcal{I}}(Q)(c) = (H\colon T'' \to T',\ Q\cdot G\colon \mathrm{sign}(T') \to \Omega).$$

The key point is that, as shown by Goguen and Burstall in [18], the satisfaction condition holds for freeness constraints translated along signature morphisms. Goguen and Burstall exploit this satisfaction condition to give a general construction associating to an institution $\mathcal{I}$ another institution $\boldsymbol{\mathcal{C}(\mathcal{I})}$ (Cf. [18, Proposition 23]) with the same category of signatures and the same model functor as $\mathcal{I}$, and with $\mathrm{sen}_{\mathcal{C}(\mathcal{I})}(\Sigma)$ the disjoint union of the sets $\mathrm{sen}_{\mathcal{I}}(\Sigma)$ and of the set [2] of all freeness constraints on Σ. Then, by the general result in Theorem 4.10, if $\mathbf{Sign}_{\mathcal{I}}$ is cocomplete, then $\mathbf{Th}_{\mathcal{C}(\mathcal{I})}$ is a cocomplete category.

[2] There are foundational questions about the size of the closure of a constraint theory that we will ignore here; as pointed out in [18], they can be solved, for example, by limiting the size of the category of signatures used in the original institution.

Although the construction of $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}})$ is given by Goguen and Burstall for an institution $\boldsymbol{\mathcal{I}}$ whose signatures and theories are unstructured, we have pointed out above how the notion of freeness constraint finds its natural home as additional constraints added to specific components of a structured theory. The way of explicitly combining freeness constraints and structured theories is then straightforward.

Definition 4.15 Given an institution $\boldsymbol{\mathcal{I}}$, the institution of *structured $\boldsymbol{\mathcal{I}}$-theories with freeness constraints* is by definition the institution $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$.

Theories in $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$ are pairs (D, Γ), with $D \colon I \to \mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ a diagram, and with Γ sentences of the form (i, φ), with φ either a sentence in $\mathrm{sen}_{\boldsymbol{\mathcal{I}}}(D(i))$ or a freeness constraint on the signature $D(i)$. Therefore, structured $\boldsymbol{\mathcal{I}}$-theories with freeness constraints capture the distinction between *theories* (with loose semantics) and *modules* (with initial or free extension semantics) present, as already mentioned, in many algebraic specification languages. Furthermore, they also capture the fact that such theories and modules can be combined into more general *structured specifications*, whose semantics explicitly depends on their structure.

Notice that, although they are of course related constructions, the institution $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$ defined above is different from the institution $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}}))$, that, by Theorem 4.13 and by the general construction of $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}})$ of Goguen and Burstall, can also be defined for a liberal and exact institution $\boldsymbol{\mathcal{I}}$ with $\mathbf{Th}_{\boldsymbol{\mathcal{I}}}$ cocomplete. Intuitively speaking, in $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$ the freeness constraints are *local* to specific components of structured theories, whereas in $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}}))$ the freeness constraints are *global*, in the sense of involving pairs of structured theories. Notice also that, by Theorems 4.9 and 4.10, if $\mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ is cocomplete, then both $\mathbf{Th}_{\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))}$ and $\mathbf{Th}_{\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}}))}$ are also cocomplete.

5 Concluding Remarks

We have shown that the addition of structured theories to an institution $\boldsymbol{\mathcal{I}}$ results in an institution $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})$, and that if the category of signatures $\mathbf{Sign}_{\boldsymbol{\mathcal{I}}}$ has colimits, then the categories of signatures and theories of $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})$ both have colimits. We have also shown other basic results about the category of theories of $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})$, and about the liberality of the institution $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}})$. Finally, we have presented a very simple way of adding freeness constraints to our setting, resulting in institutions $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$ and $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{I}}))$.

The notion of structured theory is useful not only for institutions, but also for other components of a logic such entailment systems or proof calculi [22], and could be naturally extended to those contexts. As already mentioned, the notions presented in this paper can be specialized to the more familiar case of finite hierarchies of theory inclusions by considering diagrams whose diagram schemes are finite posets, and assuming a subcategory of theory inclusions stable under pushouts along the lines of [10]. We think that it is also quite promising to study *heterogeneous* structured theories, involving several institutions, following the heterogenous specification ideas of Tarlecki [31].

We plan to studying further the institution $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$, which can serve as a semantic basis for an executable *generic module algebra* that could be specified and executed in Maude, and could be instantiated for one's logic of choice, generalizing Maude's module algebra, which manipulates structured rewrite theories and is expressed and executed within the reflective logical framework of rewriting logic [13,12]. This would allow endowing a specification language of choice with structured theories and with a module algebra for free. Regarding $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$, two important questions are: (1) finding appropriate "normal forms" for freeness constraints under suitable assumptions such as persistence; and (2) finding suitable inductive inference systems that, in spite of their intrinsic incompleteness, can approximate the logic of $\boldsymbol{\mathcal{S}}(\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}))$ for a given $\boldsymbol{\mathcal{I}}$.

References

[1] M. Barr and C. Wells. *Toposes, Triples and Theories.* Springer-Verlag, 1985.

[2] H. Baumeister. Unifying initial and loose semantics of parameterized specifications in an arbitrary institution. In S. Abramsky and T. Maibaum, editors, *Proc. Int. Conf. on Theory and Practice of Software Development, TAPSOFT'91, Brighton, UK, April 1991*, volume 493 of *Lecture Notes in Computer Science*, pages 103–120. Springer-Verlag, 1991.

[3] M. Bidoit, H.-J. Kreowski, P. Lescanne, F. Orejas, and D. Sannella, editors. *Algebraic System Specification and Development. A Survey and Annotated Bibliography*, volume 501 of *Lecture Notes in Computer Science.* Springer-Verlag, 1991.

[4] R. Burstall and J. Goguen. The semantics of Clear, a specification language. In D. Bjørner, editor, *Proceedings of the 1979 Copenhagen Winter School on Abstract Software Specification*, volume 86 of *Lecture Notes in Computer Science*, pages 292–332. Springer-Verlag, 1980.

[5] M. Clavel. *Reflection in General Logics and in Rewriting Logic with Applications to the Maude Language.* PhD thesis, University of Navarre, 1998.

[6] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. Quesada. Maude: Specification and programming in rewriting logic. SRI International, January 1999. Available at `http://maude.csl.sri.com`.

[7] M. Clavel and J. Meseguer. Reflection and strategies in rewriting logic. In Meseguer [23]. Available at `http://www.elsevier.nl/locate/entcs/volume4.html`.

[8] CoFI Task Group on Language Design. CASL—The common algebraic specification language, version 1.0. Available at `http://www.brics.dk/Projects/CoFI`, October 1998.

[9] R. Diaconescu and K. Futatsugi. *CafeOBJ Report.* AMAST Series. World Scientific, 1998.

[10] R. Diaconescu, J. Goguen, and P. Stefaneas. Logical support for modularisation. In G. Huet, G. Plotkin, and C. Jones, editors, Proceedings *of Workshop on Logical Frameworks (Edinburgh, United Kingdom, May 1991)*, pages 83–130. Cambridge University Press, 1991.

[11] T. Dimitrakos. Parameterising (algebraic) specifications on diagrams. In *Automated Software Engineering–ASE'98, 13th IEEE International Conference*, 1998.

[12] F. Durán. *A Reflective Module Algebra with Applications to the Maude Language.* PhD thesis, University of Málaga, 1999.

[13] F. Durán and J. Meseguer. An extensible module algebra for Maude. volume 15 of *Electronic Notes in Theoretical Computer Science.* Elsevier, 1998. Available at `http://www.elsevier.nl/locate/entcs/volume15.html`.

[14] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1. Equations and Initial Semantics.* Springer-Verlag, 1985.

[15] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 2. Module Specifications and Constraints.* Springer-Verlag, 1990.

[16] M. Gogolla and M. Cerioli. What is an Abstract Data Type after all? (A bibliography on the workshops on abstract data types). In E. Astesiano, G. Reggio, and A. Tarlecki, editors, *Recent Trends in Data Type Specification*, volume 906 of *Lecture Notes in Computer Science*, pages 499–523. Springer Verlag, 1995.

[17] J. Goguen and R. Burstall. Introducing institutions. In E. Clarke and D. Kozen, editors, *Proc. Logics of Programming Workshop*, volume 164 of *Lecture Notes in Computer Science*, pages 221–256. Springer-Verlag, 1984.

[18] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[19] J. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ. Technical Report SRI-CSL-92-03, Computer Science Laboratory, SRI International, March 1992. To appear in Joseph Goguen and Grant Malcolm, editors, *Applications of Algebraic Specification Using OBJ*, Academic Press.

[20] S. M. Lane. *Categories for the Working Mathematician.* Springer-Verlag, 1971.

[21] N. Martí-Oliet and J. Meseguer. Rewriting logic as a logical and semantic framework. In Meseguer [23]. Available at `http://www.elsevier.nl/locate/entcs/volume4.html`.

[22] J. Meseguer. General logics. In H.-D. Ebbinghaus et al., editors, *Logic Colloquium'87*, pages 275–329. North-Holland, 1989.

[23] J. Meseguer, editor. *Proc. 1st Intl. Workshop on Rewriting Logic and its Applications, Asilomar, California, U.S.A*, volume 4 of *Electronic Notes in Theoretical Computer Science*. Elsevier, September 1996. Available at `http://www.elsevier.nl/locate/entcs/volume4.html`.

[24] J. Meseguer and N. Martí-Oliet. From abstract data types to logical frameworks. In E. Astesiano, G. Reggio, and A. Tarlecki, editors, *Recent Trends in Data Type Specification*, volume 906 of *Lecture Notes in Computer Science*, pages 499–523. Springer Verlag, 1995.

[25] F. Orejas, E. Pino, and H. Ehrig. Institutions for logic programming. *Theoretical Computer Science*, 173:485–511, 1997.

[26] H. Reichel. Initially - restricting algebraic theories. In P. Dembínski, editor, *Proceedings of the 9th Symposium on Mathematical Foundations of Computer Science*, volume 88 of *Lecture Notes in Computer Science*, pages 504–514. Springer-Verlag, 1980.

[27] D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76(2/3):165–210, 1988.

[28] D. Sannella and M. Wirsing. A kernel language for algebraic specification and implementation. In M. Karpinski, editor, *Proceedings of the 1983 International Conference on Foundations of Computation Theory. Borgholm, Sweden, August 21-27, 1983*, volume 158 of *Lecture Notes in Computer Science*, pages 415–427. Springer-Verlag, 1983.

[29] H. Schubert. *Categories.* Springer-Verlag, 1972.

[30] Y. Srinivas and R. Jüllig. SPECWARE: Formal support for composing software. In B. Moeller, editor, *Proceedings of the Conference on Mathematics of Program Construction*, volume 947 of *Lecture Notes in Computer Science*, pages 399–422. Springer-Verlag, 1995.

[31] A. Tarlecki. Towards heterogeneous specifications. In *Proc. Workshop on Frontiers of Combining Systems FroCoS'98*, Applied Logic Series. Kluwer Academic Publishers, 1998.