# Specifying with Defaults: Compositional Semantics

F. Miguel Dionísio* and Udo W. Lipeck†

**Abstract**

We present an abstract specification theory that formalizes *non-monotonic* composition constructs from specification languages providing explicit non-monotonic mechanisms as a specification facility.

This theory generalizes the institutional framework from Goguen and Burstall by adding defeasibility mechanisms to a given institution. The denotation of a specification module consists of *defaults* (formulas organized by priority) that are assumed to be true in the absence of explicit information to the contrary. In other words defaults are assumed to be true unless they are *overridden* by other defaults of higher priority. Formulas that cannot be overriden are called axioms. Such structures of axioms and prioritized defaults are called *hierarchic specifications*.

The abstract specification theory of hierarchic specifications formalizes specification operations that use *overriding* (i.e. contradiction) of previously specified properties. The specification process improves in modularity and reusability when such mechanisms are available. This theory gives formal grounds to "the requirement to re-use specification modules as far as possible, i.e. not only to include the same components in different contexts, but also to prefer modification of given parts over new definitions. To reduce development costs, software should be designed in a "differential" way - select a module from the library, refine it by adding new functions, and modify it by *overriding* some old ones"[9].

The abstract specification theory of hierarchic specifications consists in formalizing, independently of the underlying logic, the structuring operations of hierarchic specifications. These operations are defined both on the syntactical and semantical levels by canonical constructions on corresponding syntactical and semantical *categories*, and account for the modular construction of hierarchic specifications by combining, reusing and modifying (with overriding) previously specified modules.

## Introduction

### Classical Specification Theory

The theory of institutions from Goguen and Burstall ([20]) constitutes an "abstract specification theory"[1] in that it provides a formal description of structuring operations among specifications. Such specification operations provide the means to control the complexity of a system, by describing it as a combination of simpler parts: "Complexity is a fundamental problem in programming methodology: large programs, and their large specifications, are very difficult to produce, to understand, to get right, and to modify. A basic strategy for defeating complexity is to break large systems into smaller pieces that can be understood separately, and that when put back together give the original system"[19].

*Dept. of Mathematics, I.S.T., Technical University Lisbon

†Inst. für Informatik, Uni. Hannover

[1] The term "abstract specification theory" is taken from [16].

The basic idea motivating the institutional framework is that specifications, i.e. rigorous descriptions of parts of a system, denote logical theories. Structuring operations denote canonical operations among those theories. The formalization of a specification language consists in the choice of the appropriate logic (institution) and the corresponding characterization of the structuring operations.

The chosen paradigm dictates the choice of the underlying logic. For instance temporal logic(s) are used to give semantics to object oriented specification languages ([31, 15, 16, 32, 30, 12, 29, 33]). The main contribution of the theory of institutions is the fact that the compositional constructs are *independent* of the underlying logic. The specification language Clear ([11]) can be used to build large specifications from theories from *any* logical system.

### Non-monotonic Specification Theory

We present an abstract specification theory that formalizes *non-monotonic* composition constructs, and extends the institutional framework.

There are several reasons for using non-monotonic formalisms in (the semantics of) specification languages.

The first is that actual systems, reasoning in the presence of incomplete information, use such mechanisms: planning systems, diagnose systems and truth maintenance systems, for instance.

The second is that non-monotonic logics provide the formalization of the way actual systems store and process their information: the several database and knowledge base completions modeling the fact that in such systems only positive information is kept; the theory of belief revision setting the general rules for the addition of new information (inconsistent with the previous knowledge state); the frame rule modeling the minimal change of properties after the occurrence of an action.

And, finally, the third is that the specification process improves in modularity and reusability if such mechanisms are available. Non-monotonic formalisms give formal grounds to "the requirement to re-use specification modules as far as possible, i.e. not only to include the same components in different contexts, but also to prefer modification of given parts over new definitions. To reduce development costs, software should be designed in a "differential" way - select a module from the library, refine it by adding new functions, and modify it by *overriding* some old ones"[9].

The form of reasoning known as default reasoning is fruitful in formalizing the non-monotonic aspects referred. Default reasoning is reasoning in the presence of incomplete information: in the absence of evidence to the contrary, assume the "default". For instance we can assume (and specify) that, by default, a book (in a library) can be borrowed. If, however, this book is a reserved book this conclusion can be defeated by explicit information stating that *reserved books* cannot be borrowed.

Defaults are formalized here by formulas organized by priority. A default with more important priority overrides a conflicting default of less important priority: from the point of view of the later the more important default is "evidence to the contrary". Axioms are formulas that cannot be overridden. Prioritized defaults have been introduced in [10] and further studied in [5, 4, 26, 27]. Their impact in specification is stated in [7, 9].

This structure of axioms and prioritized defaults (and the corresponding semantics) has been used to model different database completions (see [6, 3]), the frame rule (see [10, 8]) and the "taxonomic" structure of classes and subclasses (see [26, 27, 28]). The operation of adding new information at a new most important priority level (recall the specification of *reserved books*) is a belief revision ([18, 17]) operator (see [26]).

Axioms and prioritized defaults are the modularization units used in the theory of composition developed here. This means that we want to formalize specification languages that use default mechanisms and we take prioritized defaults and axioms as the denotation of such a specification modules. Constructions involving specification modules are interpreted as operations involving the corresponding denotations.

For instance the specification of *reserved books* is obtained from the module *books* (reuses it). A new priority level is added, more important than those of *books*, with the formula stating that *reserved books* cannot be lent. All other properties of *book* will hold for *reserved books* since they are not contradicted by the more important formula. Only the difference between *reserved books* and *books* must be stated. This construction is given by a canonical operation involving the specification *books* and the "difference" between *reserved books* and *books*.

### Organization

The paper is organized as follows. Firstly, in section 1 we define the notion of theory and establish the Galois connection properties of hierarchic specifications. These are fundamental for the definition of composition of hierarchic specifications, presented in section 2. Composition of hierarchic specifications comprehends the definition of the syntactic and semantic categories that formalize it, the relation between these categories and the establishment of sufficient conditions for the existence of composition constructions. In section 3 we conclude the paper.

## 1 Theories and Galois Connections

The first step towards an *abstract* theory of composition is the definition of the notion of *theory* of the syntactical entities to be composed, in our case hierarchic specifications. Theories are taken here as the syntactical counterpart of the semantics. In the classical case the theory of a presentation $A$ (set of formulas) is the set of formulas entailed by $A$. The theory of $A$ has the same semantics as $A$ (the same class of models) and has the property that it is the biggest (most formulas) presentation having the same semantics as $A$. In this way the theory of $A$ can alternatively be characterized as a special representative of the equivalence class of the presentations having the same semantics as $A$.

This is the property that we choose for defining the notion of theory for hierarchic specifications: we look for a special representative of the equivalence class of the hierarchic specifications having the same *semantics* as a given one. Such theories are in a one to one relation to the semantics of hierarchic specifications and provide the needed abstraction. When defining composition of hierarchic specifications these will be compared via their theories, or, which is the same, via their semantics.

It is clear from these considerations that the notion of *semantics* (of hierarchic specifications) is the key to abstraction. In the classical case the semantics of a presentation is class of its models. An important property of this semantics is the fact that it is *denotational* w.r.t. union of presentations. This means that the semantics of the union of presentations depends only on the semantics of each argument presentation (the semantics of the union is the intersection of the classes of models). This is a necessary property of any theory of composition: the syntactical constructions (union in this case) must have a corresponding semantic construction.

In the case of hierarchic specifications the "standard" semantics, the lexicographic preference (presented bellow in section 1.2), does not have enough information to account for the syntactic operations we are concerned with: union of axioms and defaults and "addition" of priority levels. For this reason a new semantics (with more structure) of hierarchic specifications is introduced.

Moreover, a Galois connection relating the syntactical and semantical constructions of hierarchic specifications is displayed. This generalizes the corresponding property of classical presentations and means that relations between hierarchic specifications ("inclusion" see below) are mirrored by relations between the corresponding semantics (again "inclusion", but in the opposite direction). Moreover operations among hierarchic specifications ("unions" and "intersections") are also mirrored by operations among the corresponding semantics (unions are mapped to intersections and intersections to unions). The reverse is also true: relations and operations among the semantics of hierarchic specifications are also mirrored by relations and operations among the corresponding theories. The Galois connection properties are fundamental for formalizing composition of presentations.

We begin by recalling needed institutional notions. Then we study the special case of specifications (i.e. hierarchic specifications with one only priority level) and define the corresponding notion of theory. The Galois connection is displayed for this case. Finally, we generalize these notions for hierarchic specifications. The properties of specifications and the corresponding theory of composition have been presented in [13]. The properties of hierarchic specifications and the corresponding theory of composition have been defined in [14].

### Institutions

We recall the definitions of institution ([20]), presentation model and theory (of a presentation) in the following.

Note that $\mathsf{Set}$ is the category of sets and functions, $\mathsf{Cat}$ the category[2] of categories and functors between them (and $\mathsf{Cat}^{op}$ its opposite quasicategory). Given a category (or quasicategory) $\mathcal{C}$, the class of its objects is denoted by $|\mathcal{C}|$.

**Definition 1** An *institution* consists of

- a category $\mathsf{Sign}$ whose objects are called *signatures*,
- a functor $\mathsf{Sen} : \mathsf{Sign} \to \mathsf{Set}$ that assigns to each signature the set of its *formulas*,
- a functor $\mathsf{Mod} : \mathsf{Sign} \to \mathsf{Cat}^{op}$ giving for each signature $\Sigma$ a category whose objects are called $\Sigma$-*interpretation structures* and whose morphisms are the $\Sigma$-interpretation structure morphisms, and
- a relation $\models_\Sigma \subseteq |\mathsf{Mod}(\Sigma)| \times \mathsf{Sen}(\Sigma)$, called $\Sigma$-*satisfaction* such that for every morphism $\phi : \Sigma_1 \to \Sigma_2$ the *Satisfaction Condition*

$$m_2 \models_{\Sigma_2} \mathsf{Sen}(\phi)(f) \text{ iff } \mathsf{Mod}(\phi)(m_2) \models_{\Sigma_1} f$$

  holds for each model $m_2$ of $|\mathsf{Mod}(\Sigma_2)|$ and each formula $f$ of $\mathsf{Sen}(\Sigma_1)$. ■

Given an institution $\mathcal{I}$ a *presentation* is a pair $P = (\Sigma, A)$ where $\Sigma \in |\mathsf{Sign}^{\mathcal{I}}|$ is a signature from $\mathcal{I}$ and $A \subseteq \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is a set of formulas from $\Sigma$. A $\Sigma$-presentation $A$ is a presentation $(\Sigma, A)$. The interpretation structures that satisfy all formulas in a given presentation are said to *satisfy* the presentation and called *models* of the presentation. Presentations and their models are related by the function $^\bullet$ that assigns to a $\Sigma$-presentation $A$ the class $A^\bullet$ of all $\Sigma$-interpretation structures that are models of $A$. In the reverse direction the function $^\bullet$ assigns to a class $\mathcal{M} \subseteq |\mathsf{Mod}^{\mathcal{I}}(\Sigma)|$ of $\Sigma$-interpretation structures the set $\mathcal{M}^\bullet$ of all $\Sigma$-formulas that are satisfied in each interpretation structure from $\mathcal{M}$. $\mathcal{M}^\bullet$ is called the *theory* of $\mathcal{M}$. The *theory* of a $\Sigma$-presentation $A$ is the set $A^{\bullet\bullet}$ of the $\Sigma$-formulas satisfied in the models of $A$.

These two functions form a Galois connection (see [20]).

## 1.1 Specifications

Specifications are a special case of hierarchic specifications with one only priority level (of defaults). The corresponding concepts are simplified versions of those needed for hierarchic specifications. They are a good introduction to the concepts needed later.

Specifications consist in two sets of formulas (of a given signature), the set axioms and the set of defaults. The semantic of a specification is a preference relation (i.e. a pre-order) that organizes the models of the axioms by how well they satisfy the defaults. In this way the "best" models are the maximal ones in this preference relation. The maximal models are those models of the axioms that satisfy most defaults. The relation between syntax (specifications) and semantics (their preference relations) takes the form of a Galois connection. This connection generalizes the Galois connection for the classic case: more axioms imply less models. But the new structure of defaults obeys a formally similar rule: addition of defaults implies less relations of preference between those models.

### Syntax and Semantics

The definition of specification and the corresponding semantics is stated formally in the following.

[2] In fact it is a quasicategory (see [1]). But this distinction is not relevant to the constructions presented here and we will not refer it further.

**Definition 2** A *specification* (from an institution $\mathcal{I}$) is a triple $S = (\Sigma, A, D)$ where $\Sigma \in |\mathsf{Sign}^{\mathcal{I}}|$ is a signature from $\mathcal{I}$, $A \subseteq \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is a set of formulas from $\Sigma$, the set of *axioms* from $S$ and $D \subseteq \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is a set of formulas from $\Sigma$, the set of *defaults* from $S$. A $\Sigma$-*specification* $(A, D)$ is a specification $(\Sigma, A, D)$. ■

The semantics of a specification is a relation on the models of its axioms, representing that some of these models are better than other since they satisfy more of the defaults ([22, 34, 7]). This relation is a pre-order.

**Definition 3** A *pre-order* (from $\mathcal{I}$) is a triple $\mathcal{R} = (\Sigma, \mathcal{M}, \sqsubseteq)$ where $\Sigma \in |\mathsf{Sign}^{\mathcal{I}}|$ is a signature from $\mathcal{I}$, $\mathcal{M} \subseteq |\mathsf{Mod}^{\mathcal{I}}(\Sigma)|$ is a class of interpretation structures of the signature $\Sigma$, $\sqsubseteq\, \subseteq \mathcal{M} \times \mathcal{M}$ is a reflexive and transitive relation among those interpretation structures. A $\Sigma$-*pre-order* $(\mathcal{M}, \sqsubseteq)$ is a pre-order $(\Sigma, \mathcal{M}, \sqsubseteq)$. ■

The pre-order induced by a specification relates the models of the axioms by how well they satisfy the defaults.

**Definition 4** The pre-order induced by a specification $S = (\Sigma, A, D)$ is the pre-order $S^\star = (\Sigma, A^\bullet, \sqsubseteq)$ with the same signature as $S$, the models of the axioms from $S$ as class of interpretation structures, and the relation $\sqsubseteq$ among those models defined by $m \sqsubseteq n$ iff for all $d \in D$, if $m \vDash d$ then $n \vDash d$. ■

## Consequences

The equivalence classes of the preference $S^\star$ of a specification $S$ are those classes of models of the axioms that satisfy precisely the same defaults. The maximal equivalence classes of $S^\star$ are the classes of models of the axioms that satisfy most defaults. The set of formulas $\mathcal{E} = [m]^\bullet$ of the formulas holding in all interpretation structures from a *maximal* equivalence class $[m]$ of $S^\star$ is called an *extension* of $S$. A formula belonging to all extensions of $S$ is called a *skeptical* consequence of $S$. A formula belonging to some extension of $S$ is called a *credulous* consequence of $S$. The usual notion of consequence is the notion of skeptical consequence ([22, 34, 21]). The notion of credulous consequence, introduced in [24], is important in the context of abduction ([23]). We do not commit to one or the other since both can be derived from the chosen semantics.

## Theories and Galois Connection

The theory of a specification $S$ is a special representative of the equivalence class of those specifications having the same preference ($S^\star$) as $S$. This theory is the "biggest" of such specifications, i.e. the one with most axioms and defaults.

A more interesting characterization can be motivated through an analogy with the classical case. The theory of a presentation $A$ is the set of such formulas that can be added to $A$ without changing the class of models. So we are now looking for formulas (both axioms and defaults) that can be added to a specification $S$ without changing its semantics, namely its preference relation. On the axiom side the situation is like the classical case: these are the classical consequences of the axioms in $S$. The new defaults are called "implicit" in $S$ and are the formulas that can be added as defaults to $S$ without changing its preference. Such formulas $d$ must satisfy the following property: if $m \sqsubseteq n$ and $m \vDash d$ then $n \vDash d$ (where $\sqsubseteq$ is the preference of $S$). Otherwise (see the definition 4 of preference relation) when adding the default $d$ to $S$ the new preference would be different from $S^\star$ since $m$ and $n$ would no longer be related.

In general a sentence $d$ is an implicit default of a pre-order $\mathcal{R}$ if, whenever it is satisfied by an interpretation structure $m$ from $\mathcal{R}$ it is also satisfied by all interpretation structures better (according to $\mathcal{R}$) than $m$.

**Definition 5** The set of *defaults implicit* in a $\Sigma$-pre-order $\mathcal{R} = (R, \sqsubseteq)$, denoted by $\mathcal{R}^\circ$, is the set of $\Sigma$-formulas

$$\mathcal{R}^\circ = \{d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma) \ : \ \text{for all } m_1, m_2 \in R \text{ if } m_1 \sqsubseteq m_2 \text{ and } m_1 \vDash d \text{ then } m_2 \vDash d\}. \quad ■$$

We can now assign to a specification a pre-order and to a pre-order a specification.

**Definition 6** The function $^\star$ assigns to a $\Sigma$-specification $S$ its induced *preference relation*, the $\Sigma$-pre-order $S^\star$, and the function $^\star$ assigns to a $\Sigma$-pre-order $\mathcal{R}$ the specification $\mathcal{R}^\star = (|\mathcal{R}|^\bullet, \mathcal{R}^\circ)$. The specification $\mathcal{R}^\star$ is called the *theory* of $\mathcal{R}$. The specification $S^{\star\star}$ is called the *theory* of the specification $S$. ■

As in the classical case the functions $^\star$ relating specifications with their semantics form a Galois connection. This is stated formally in the following theorem. We note the formal similarity between the axiom and default parts: when adding an axiom $a$ to $S$ the new class of models is obtained from the class of models of $S$ by deleting those that do not satisfy $a$. When adding a default $d$ the new preference relation is obtained from that of $S$ by deleting the relation pairs $(m, n)$ that "do not satisfy" the default $d$: i.e. such that $m \vDash d$ but $n \nvDash d$.

The binary relation $\Subset$ denotes inclusion of the sets of axioms and defaults when relating specifications. When when relating pre-orders it denotes inclusion of classes of models and relations (seen as classes of relation pairs). The operations $\Cup$ and $\Cap$ are pairwise union and intersection. The pairwise union of pre-orders is the union of the classes of models and the *transitive* closure of the union of the pre-orders (in order to assure that the result is a pre-order).

**Theorem 7** Let $S, S'$ and $S_n, n \in N$, be $\Sigma$-specifications and $\mathcal{R}, \mathcal{R}'$ and $\mathcal{R}_n, n \in N$, be $\Sigma$-pre-orders ($N$ is some set of indexes). Then

1. $S \Subset S'$ implies $S^\star \Supset S'^\star$,
2. $\mathcal{R} \Subset \mathcal{R}'$ implies $\mathcal{R}^\star \Supset \mathcal{R}'^\star$,
3. $S \Subset S^{\star\star}$ and
4. $\mathcal{R} \Subset \mathcal{R}^{\star\star}$.

The following properties are implied from these. (a) $S^\star = S^{\star\star\star}$, (b) $\mathcal{R}^\star = \mathcal{R}^{\star\star\star}$, (c) $(\Cup_n S_n)^\star = \Cap_n S_n^\star$, (d) $(\Cup_n \mathcal{R}_n)^\star = \Cap_n \mathcal{R}_n^\star$, (e) $(\Cap_n S_n^{\star\star})^\star = (\Cup_n S_n^\star)^{\star\star}$ and (f) $(\Cap_n \mathcal{R}_n^{\star\star})^\star = (\Cup_n \mathcal{R}_n^\star)^{\star\star}$.

**Proof** Proofs can be found in [14]. ✓

The Galois connection properties imply a one to one relation between theories and the semantics of specifications: $S_1^\star = S_2^\star$ iff $S_1^{\star\star} = S_2^{\star\star}$ and also that the theory of $S$ is the biggest (w.r.t. $\Subset$) specification having the same preference as $S$.

## 1.2 Hierarchic Specifications

In this section we generalize the concepts and properties of specifications to hierarchic specifications. We note that the "standard" semantics of hierarchic specifications, its lexicographic preference, does not have enough information to explain composition. For this purpose we introduce new, more structured semantics of hierarchic specifications. With these notions we define the theory of an hierarchic specification and display a corresponding Galois connection.

### Syntax

Hierarchic specifications consist of axioms and sets of defaults organized by priorities. The priority structure is a partial order.

**Definition 8** A *hierarchic specification* is a tuple $S = (\Sigma, A, (H, \preceq), \Delta)$ consisting of: a signature $\Sigma \in |\mathsf{Sign}^\mathcal{I}|$, a set of axioms $A \subseteq \mathsf{Sen}^\mathcal{I}(\Sigma)$, a well-founded partial order $(H, \preceq)$ of priority, with *non-empty* set of priority levels $H$ and a function $\Delta$ assigning to each priority level $h \in \mathcal{H}$ a set of defaults $\Delta(h) \subseteq \mathsf{Sen}^\mathcal{I}(\Sigma)$.

To refer to the set of defaults from $S$ at level $h$ we use the notation $\mathsf{df}(S, h) = \Delta(h)$. ■

Following [26] we adopt the convention that the lower priority levels are the most important. This is justified by the structure of classes and subclasses in object oriented specifications: the lower priority levels correspond to subclasses. Its defaults have usually more priority than those of the superclasses since they express more specific properties.

### Lexicographic Semantics

The overall meaning of a hierarchic specification has to give preference to the defaults of lower (better) priority levels. This is achieved formally by assigning to a hierarchic specification a pre-order among the models of its axioms that compares them according to how well they satisfy the defaults. This is now meant in the sense that models are preferred because either they satisfy more defaults or they strictly satisfy better defaults. The intended preference organizes the models in a way reminiscent of the lexicographic ordering of words in dictionaries (see Ryan [25, 26]). We define the *lexicographic preference* induced by a hierarchic specification.

**Definition 9** The *lexicographic preference* induced by a hierarchic specification $S$ is the pre-order $\mathsf{lex}^\circ(S)$ with: the same signature as $S$, the models of the axioms from $S$ as class of interpretation structures, and the relation $\sqsubseteq^\circ$ among those models defined by $m \sqsubseteq^\circ n$ iff for every priority level $h \in H$ if $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$ then there is $h' \prec h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$ (note that this last inclusion is strict). The set $\mathsf{df}(S,h)(m)$ is the set of defaults from $S$ at level $h$ that are satisfied by the interpretation structure $m$. ■

The lexicographic preference of an hierarchic specification $S$ has been shown (in [2]) to be canonical in the sense that it is the only relation among those models satisfying certain conditions. The consequences of a hierarchic specification are defined in terms of the maximal equivalence classes of $\mathsf{lex}^\circ(S)$ in the same way as for specifications.

## 1.3 Semantics for Composition

The lexicographic preference just presented is the main semantics for defining logical properties of a hierarchic specification. However, it has not enough information to account for composition of hierarchic specifications. This means that hierarchic specifications having the same lexicographic preference do not, in general, behave in the same way with respect to composition. This is illustrated in the following example.

**Example 10** Consider two hierarchic specifications $S_1$ and $S_2$ with the same priority structure consisting of three priority levels related as follows: $h_1 \prec h_2 \prec h_3$. Assume that $S_1$ has the set $\{d\}$ of defaults at level $h_1$ and the empty set of defaults in the other levels. Let $S_2$ have the same set $\{d\}$ of defaults at level $h_3$ and the empty set of defaults in the other levels. The hierarchic specifications $S_1$ and $S_2$ have the same lexicographic preference (since there is one only default $d$). Consider now that we add, in both specifications, the default $\neg d$ at level $h_2$. In $S_1$ this has no effect: the previous default $d$ overrides $\neg d$. But in $S_2$ the opposite happens: $d$ is overridden by $\neg d$. The resulting specifications have different lexicographic preferences. △

In the following we present two equivalent semantics, the *hierarchies of lexicographic and differential preferences* that have enough structure to explain composition of hierarchic specifications.

From any of these semantics the lexicographic preference of the original specification can be derived. This means that the new semantics still contains the logical information. The new structure is needed to explain the compositional operations of hierarchic specifications. We have shown in [14] that the new semantics are as abstract as possible, i.e. they have the structure needed to explain composition and not more.

We define firstly hierarchy of pre-orders, that consist of a class of interpretation structures and pre-orders organized by priority.

**Definition 11** A *hierarchy of pre-orders* is a tuple $\mathcal{H} = (\Sigma, \mathcal{M}, (H, \preceq), \Theta)$ consisting of: a signature $\Sigma \in |\mathsf{Sign}^{\mathcal{I}}|$, a class of interpretation structures $\mathcal{M} \subseteq \mathsf{Mod}^{\mathcal{I}}(\Sigma)$, a well-founded partial order $(H, \preceq)$ of priority, with *non-empty* set of priority levels $H$ and a function $\Theta$ assigning to each priority level $h \in H$ a reflexive and transitive binary relation $\Theta(h) \subseteq \mathcal{M} \times \mathcal{M}$ over the class of interpretation structures $\mathcal{M}$. A $\Sigma$-*hierarchy of pre-orders* $(\mathcal{M}, (H, \preceq), \Theta)$ is the hierarchy $(\Sigma, \mathcal{M}, (H, \preceq), \Theta)$. ■

The first structured semantics of hierarchic specifications is its hierarchy of lexicographic preferences. In short the hierarchy of lexicographic preferences of a hierarchic specification $S$ assigns to each priority level $h$ the lexicographic preference of the substructure of $S$ obtained by restricting $S$ to the levels $h$ and under $h$. In this way each priority level $h$ has the information of the interaction of the defaults from $h$ with those at more important levels.

**Definition 12** The *hierarchy of lexicographic preferences* induced by a hierarchic specification $S$, denoted by $S^{\oplus}$ is the hierarchy of pre-orders with:

- the same signature as $S$,
- the models of the axioms from $S$ as class of interpretation structures,
- the same well-founded partial order $(H \preceq)$ as $S$,
- the function $\sqsubseteq^{\oplus}$ that to each priority level $h \in H$ assigns the relation $\sqsubseteq_h^{\oplus}$ defined by $m \sqsubseteq_h^{\oplus} n$ iff for every priority level $h' \preceq h$ if $\mathsf{df}(S, h')(m) \not\subseteq \mathsf{df}(S, h')(n)$ then there is $h'' \prec h'$ with $\mathsf{df}(S, h'')(m) \subset \mathsf{df}(S, h'')(n)$ (this last inclusion is strict).

  Recall that the set $\mathsf{df}(S, h)(m)$ is the set of defaults from $S$ at level $h$ that are satisfied by the interpretation structure $m$. ■

Clearly the lexicographic preference of $S$ can be derived from the hierarchy of lexicographic preferences of $S$. The lexicographic preference of $S$ is the preference assigned by $S^{\oplus}$ to the existing or imagined top priority level (since it computes the interaction of all levels). This preference is the intersection of all partial preferences occurring in $S^{\oplus}$.

The hierarchy of differential preferences has the same information content as the hierarchy of lexicographic preferences (see [14]) in the sense that each of them can be derived from the other. This information, however is displayed in a way technically convenient for the purpose of composition[3]. We motivate this semantics as follows. The composition operations of hierarchic specifications comprehend addition of axioms and defaults at each priority level (and also combinations of priority structure). The addition of axioms is treated classically. The addition of defaults at a particular priority level is the interesting operation in this context. Let $h$ be a priority level from a hierarchic specification $S$ and $d$ a formula that we want to add at level $h$. The question is how the semantics of $S + d$ relates to the semantics of $S$. Consider firstly that the semantics of $S$ is the hierarchy of lexicographic preferences just defined. At level $h$ we have the lexicographic preference corresponding to the substructure of $S$ obtained by restricting the priority levels to $h$ and the levels under $h$ (more important than $h$). Formal similarity with the concepts presented for specifications would suggest that the new preference would be obtained from this by deleting the pairs $(m, n)$ that do not agree with the new default $d$, i.e. such that $m \vDash d$ but $n \nvDash d$. This is, however, not true. In fact the defaults at more important (lower levels) have to be taken into account and it may be the case that $n$ is preferred to $m$ because it strictly satisfies better defaults than $m$. In this case the new default $d$ at level $h$ has no effect. The pairs that can be deleted are those pairs $(m, n)$ that are equivalent in more important levels (i.e. satisfy precisely the same defaults from more important levels) and do not agree with $d$.

The hierarchy of differential preferences assigns to each priority level $h$ the following preference. Firstly only interpretation structures that are equivalent at levels more important than $h$ can be related. Two such levels $m$ and $n$ are related iff if $m \vDash d'$ then $n \vDash d'$ for every default $d'$ from the *local set of defaults at level* $h$. In this way the addition of a default $d$ at level $h$ has the effect of deleting all those pairs (of the differential preference) that do not agree with $d$. This implies a formal similarity with the previous constructions for specifications that is of convenience.

**Definition 13** The *hierarchy of differential preferences* of a hierarchic specification $S$, denoted by $S^{\ominus}$ is the hierarchy of pre-orders with:

- the same signature as $S$,
- the models of the axioms from $S$ as class of interpretation structures,
- the same well-founded partial order $(H \preceq)$ as $S$,
- the function $\sqsubseteq^{\ominus}$ that to each priority level $h \in H$ assigns the relation $\sqsubseteq_h^{\ominus}$ defined by $m \sqsubseteq_h^{\ominus} n$ iff $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ and $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for every priority level $h' \prec h$. ■

[3]Note that the theory that we develop here on top of the hierarchy of differential preferences could be rewritten in terms of the hierarchy of lexicographic preferences.

## 1.4 Theories and Galois Connection

Having defined the semantics of hierarchic specifications as its hierarchy of differential preferences we need now the reverse direction, namely to assign to such a semantics a hierarchic specification. The motivation for this other direction is that the hierarchic specification assigned to $S^{\ominus}$ should be the *theory* of $S$, a canonical representative of the equivalence class of the hierarchic specifications having the same semantics as $S$. This theory should be the biggest hierarchic specification among such equivalent specifications. This means the hierarchic specification having the most axioms and, at each priority level, the most defaults. These are the defaults that can be added at each level without changing the semantics of a specification, i.e. the *defaults implicit* at each priority level $h$, and correspond to the defaults that do not change the differential preference *locally* at that level (i.e. the defaults implicit in the differential preference at level $h$ in the sense of definition 5). The reason why the interaction of lower levels need not be taken into account is that such interaction is already coded in the differential preference: interpretation structures can only be related by preference if they were equivalent at levels below.

**Definition 14** Let $\mathcal{H}$ be a $\Sigma$-hierarchy of pre-orders. The corresponding *theory*, denoted by $\mathcal{H}^{\ominus}$ is the $\Sigma$-hierarchic specification with:

- the same well-founded partial order $(H, \preceq)$ as $\mathcal{H}$,
- the $\Sigma$-formulas satisfied by each interpretation structure from $\mathcal{H}$ as set of axioms,
- the function $\Delta$ defined by $\Delta(h) = R_h^{\circ}$ that to each priority level $h \in H$ assigns the set of the defaults implicit in the $\Sigma$-pre-order $R_h$ assigned by $\mathcal{H}$ to the level $h$.

The *theory* of a hierarchic specification $S$ is the hierarchic specification $S^{\ominus\ominus}$. ∎

The functions assigning to a hierarchic specification its hierarchy of differential preferences and to a hierarchy of pre-orders form again a Galois connection, if the hierarchies of pre-orders are restricted to *hierarchies of differential pre-orders* (any hierarchy of differential preferences is an hierarchy of differential pre-orders). Hierarchies of differential pre-orders satisfy the property that the pre-orders at each level are contained in the equivalence classes of pre-orders at lower levels.

**Definition 15** Let $\mathcal{H}$ be a hierarchy of pre-orders with priority structure $(H, \preceq)$. $\mathcal{H}$ is said to be a *hierarchy of differential pre-orders* iff $\equiv_{h'} \supseteq \sqsubseteq_h$ for every $h' \prec h$. The relation $\equiv_{h'}$ is the equivalence of the pre-order at level $h'$ and $\sqsubseteq_h$ is the pre-order at level $h$. ∎

The Galois connection is stated formally in the next theorem.

The relation $\Subset$ among hierarchic specifications of the same priority structure denotes inclusion of the sets of axioms and pointwise (at each priority level) inclusion of sets of defaults. When relating hierarchies of pre-orders of the same priority structure it denotes inclusion of classes of models and pre-orders at each level (seen as classes of relation pairs). The operations $\Cup$ and $\Cap$ are pointwise union and intersection, i.e. union of the axioms and, at each priority level, of the defaults from the argument hierarchic specifications at that level. The situation with union and intersection of hierarchies of pre-orders is similar. Care again has to be taken with the union of pre-orders (at each level): the transitive closure of the union is taken in order to assure that the result is a pre-order.

**Theorem 16** Let $S, S'$ be $\Sigma$-hierarchic specifications and $\mathcal{H}, \mathcal{H}'$ be $\Sigma$-hierarchies of *differential* pre-orders. Then

1. $S \Subset S'$ implies $S^{\ominus} \Supset S'^{\ominus}$,
2. $\mathcal{H} \Supset \mathcal{H}'$ implies $\mathcal{H}^{\ominus} \Subset \mathcal{H}'^{\ominus}$,
3. $S \Subset S^{\ominus\ominus}$ and
4. $\mathcal{H} \Subset \mathcal{H}^{\ominus\ominus}$.

The following properties are implied from these. Let $S_n, n \in N$ be $\Sigma$-specifications and $\mathcal{H}_n, n \in N$ be $\Sigma$-hierarchies of *differential* pre-orders with the same partial order ($N$ is some set of indexes). Then (a) $S^{\ominus} = S^{\ominus\ominus\ominus}$, (b) $\mathcal{H}^{\ominus} = \mathcal{H}^{\ominus\ominus\ominus}$, (c) $(\Cup_n S_n)^{\ominus} = \Cap_n S_n^{\ominus}$, (d) $(\Cup_n \mathcal{H}_n)^{\ominus} = \Cap_n \mathcal{H}_n^{\ominus}$, (e) $(\Cap_n S_n^{\ominus\ominus})^{\ominus} = (\Cup_n S_n^{\ominus})^{\ominus\ominus}$, (f) $(\Cap_n \mathcal{H}_n^{\ominus\ominus})^{\ominus} = (\Cup_n \mathcal{H}_n^{\ominus})^{\ominus\ominus}$.

The Galois connection properties imply a one to one relation between theories and the semantics of hierarchic specifications: $S_1^{\ominus} = S_2^{\ominus}$ iff $S_1^{\ominus\ominus} = S_2^{\ominus\ominus}$. Since the hierarchy of differential preferences is equivalent to the hierarchy of lexicographic preferences (see [14]) we also have $S_1^{\oplus} = S_2^{\oplus}$ iff $S_1^{\ominus\ominus} = S_2^{\ominus\ominus}$. This means that the notion of theory is the same independently of which of these two semantics is chosen. Moreover the theory $S^{\ominus\ominus}$ of $S$ is the biggest (w.r.t. $\Subset$) specification having the same hierarchy of differential (or lexicographic) preferences as $S$.

# 2 Composition of Hierarchic Specifications

In the previous section we have provided hierarchic specifications with a notion of theory. This notion corresponds to an abstraction of the particular way such specifications are written since it declares equivalent the specifications having the same theory (or equivalently the same semantics). Moreover a notion of inclusion (of meaning) between specifications (defined by inclusion of theories) has been defined.

We extend these notions to account for composition of hierarchic specifications. Special care is taken in order that the syntactical concepts and operations defined have a corresponding semantics. The formalizations generalize the classical theory of composition of presentations ([20]) to hierarchic specifications and are inspired by the formalization of composition of hierarchic specifications presented in [9] .

To the notion of equivalence of specifications is added a notion of *independence of representation*. This corresponds to identify as equivalent specifications that only differ because they have been written with different symbols. Independence of representation is formalized by the notion of isomorphism in the category $\mathsf{hieSpec}$ of hierarchic specifications. Renaming of priority level names is also taken into account.

*Composition of hierarchic specifications* is understood as the addition of syntactical entities such as axioms, defaults, priority levels and relations between them. This composition is formalized by canonical constructions in the category $\mathsf{hieSpec}$ of hierarchic specifications. The constructions depend on the existence of signature symbols that are used in the axioms and defaults of the specifications result (i.e. they depend on the constructions of the underlying category $\mathsf{Sign}$ of signatures). They depend furthermore on the existence of a priority structure that expresses the combination of the priority structures of the hierarchic specifications involved.

The category of hierarchic specifications is mirrored on the semantic side by the semantic category $\mathsf{hiePref}$. The semantics of the composition of specifications is obtained from the semantics of the parameter specifications by canonical constructions in these categories.

We begin by defining the category of partial orders, that accounts for the composition of priority structure. After we define the category of hierarchic specifications and the corresponding mirror semantic category. We then investigate the relation between the syntactic and semantic constructions and disply sufficient conditions for the existence of such constructions.

### Category of Partial Orders

Composition of hierarchic specifications is, first of all, composition of their structure of priority. Composition of partial orders is formalized by colimits in the category $\mathsf{StPart}$. In this category we restrict the partial order morphisms to those that *strictly* respect the orderings. This means that levels that are *strictly* related by priority will remain strictly related by priority. This formalization models the construction of a specification by adding to it more relations between (possibly more) priority levels (and axioms and defaults) and rejects the possibility of identifying levels strictly related by priority. However, unrelated levels may be identified .

**Definition 17** The category $\mathsf{StPart}$ of partial orders consists of:

- **Objects:** Partial orders
- **Morphisms:** A (strict) partial order morphism $\phi : (H, \preceq) \to (H', \preceq')$ is a function $\phi : H \to H'$ that respects $\prec$ (the strict relation corresponding to $\preceq$): if $h_1 \prec h_2$ then $\phi(h_1) \prec' \phi(h_1)$. ■

The category $\mathsf{StPart}$ is *not* cocomplete as opposed to the category $\mathsf{Part}$ of partial orders and partial order morphisms. In particular it does not have all co-equalizers. It is however easy to check that the colimits in $\mathsf{StPart}$, when they exist, coincide with the corresponding colimits in $\mathsf{Part}$.

### Category of Hierarchic Specifications

The category of hierarchic specifications relates hierarchic specifications via their theories by axiom and default preserving signature and partial order morphisms.

Recall that, given a signature $\sigma : \Sigma_1 \to \Sigma_2$, the function $\mathsf{Sen}(\sigma) : \mathsf{Sen}(\Sigma_1) \to \mathsf{Sen}(\Sigma_2)$ sends each formula from the language $\mathsf{Sen}(\Sigma_1)$ to a formula in the language $\mathsf{Sen}(\Sigma_2)$. We will denote the function $\mathsf{Sen}(\sigma)$ by $\hat{\sigma}$. With this convention and letting $\mathsf{ax}(S)$ denote the set of axioms from $S$ and $\mathsf{df}(S,h)$ the set of defaults from $S$ at level $h$ the definition of the category $\mathsf{hieSpec}$ follows.

**Definition 18** The category $\mathsf{hieSpec}$ of hierarchic specifications consists of:

- **Objects:** All hierarchic specifications,
- **Morphisms:** A hierarchic specification morphism $(\sigma,\phi) : S_1 \to S_2$ from the $\Sigma_1$-hierarchic specification $S_1$ to the $\Sigma_2$-hierarchic specification $S_2$ is: a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ from the signature of $S_1$ to that of $S_2$ and a *strict* partial order morphism $\phi : \mathsf{po}(S_1) \to \mathsf{po}(S_2)$ from the partial order of priority of $S_1$ to that of $S_2$, such that
  - $\hat{\sigma}(\mathsf{ax}(S_1^{\ominus\ominus})) \subseteq \mathsf{ax}(S_2^{\ominus\ominus})$, i.e. the image by $\hat{\sigma}$ of the axioms from the theory of $S_1$ (the consequences of the axioms from $S_1$) are axioms from the theory of $S_2$ and
  - $\hat{\sigma}(\mathsf{df}(S_1^{\ominus\ominus},h)) \subseteq \mathsf{df}(S_2^{\ominus\ominus},\phi(h))$ for each level $h \in \mathsf{po}(S_1)$, i.e. the image by $\hat{\sigma}$ of the defaults from the theory of $S_1$ at level $h$ (the defaults implicit in $S_1$ at level $h$) have to be defaults from the theory of $S_2$ at level $\phi(h)$. ∎

### Categories of Hierarchies of Pre-orders

We define the category of hierarchies of pre-orders and the category of those hierarchies of pre-orders that are the semantics (hierarchies of differential preferences) of some hierarchic specification.

Firstly we have to define the notion of the reduct of a pre-order w.r.t. a signature morphism.

**Definition 19** Recall that, given a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ and an interpretation structure $m_2 \in \mathsf{Mod}(\Sigma_2)$ the *reduct* of $m_2$ w.r.t. $\sigma : \Sigma_1 \to \Sigma_2$ is the $\Sigma_1$-interpretation structure $\mathsf{Mod}(\sigma)(m_2)$. The reduct $\mathsf{Mod}(\sigma)(m_2)$ will be denoted by $\breve{\sigma}(m_2)$.

Furthermore, given a $\Sigma_2$-pre-order $R = (|R|, \sqsubseteq)$ the *reduct relation* denoted by $\breve{\sigma}(R)$ is the pair $(\breve{\sigma}(|R|), \breve{\sigma}(\sqsubseteq))$ where $\breve{\sigma}(|R|)$ is the class $\{\breve{\sigma}(m) : m \in |R|\}$ of the reducts of the interpretation structures participating in $R$ and $\breve{\sqsubseteq} = \breve{\sigma}(\sqsubseteq)$ is the smallest pre-order[4] among those interpretation structures that satisfies $\breve{\sigma}(m_2) \mathrel{\breve{\sqsubseteq}} \breve{\sigma}(n_2) \quad \text{if} \quad m_2 \sqsubseteq n_2$, that relates the reducts of the $\Sigma_2$-interpretation structures from $|R|$ if they were related by $R$. ∎

Let $|\mathcal{H}|$ denote the class of interpretation structures participating in the hierarchy of pre-orders $\mathcal{H}$ and $\mathsf{rl}(\mathcal{H},h)$ denote the pre-order assigned by $\mathcal{H}$ to the level $h$. The definition of the category $\mathsf{hiePre}$ of hierarchies of pre-orders follows.

**Definition 20** The category $\mathsf{hiePre}$ of $\Sigma$-hierarchies of pre-orders consists of:

- **Objects:** All hierarchies of pre-orders,
- **Morphisms:** A morphism $\overleftarrow{(\sigma,\phi)} : \mathcal{H}_2 \to \mathcal{H}_1$ from the $\Sigma_2$-hierarchy of pre-orders $\mathcal{H}_2$ to the $\Sigma_1$-hierarchy of pre-orders $\mathcal{H}_1$ is a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ from the signature of $\mathcal{H}_1$ to that of $\mathcal{H}_2$ and a partial order morphism $\phi : \mathsf{po}(\mathcal{H}_1) \to \mathsf{po}(\mathcal{H}_2)$ from the partial order of $\mathcal{H}_1$ to that of $\mathcal{H}_2$ such that

[4] The smallest relation that satisfies $\breve{\sigma}(m_2) \mathrel{\breve{\sqsubseteq}} \breve{\sigma}(n_2) \ \text{if} \ m_2 \sqsubseteq n_2$, is not, in general, a $\Sigma_1$-pre-order, since it may fail to be transitive. The reduct pre-order $\breve{\sigma}(R)$ is the transitive closure of this relation.

- $\breve{\sigma}(|\mathcal{H}_2|) \subseteq |\mathcal{H}_1|$, i.e. the reducts of the interpretation structures participating in $\mathcal{H}_2$ are interpretation structures participating in $\mathcal{H}_1$ and
- $\breve{\sigma}(\mathsf{rl}(\mathcal{H}_2, \phi(h_1))) \subseteq \mathsf{rl}(\mathcal{H}_1, h_1)$ for every $h_1 \in |\mathsf{po}(\mathcal{H}_1)|$, i.e. whenever two interpretation structures are related by the pre-order from $\mathcal{H}_2$ at level $\phi(h_1)$ their reducts are related by the pre-order from $\mathcal{H}_1$ at the level $h_1$. Note that this implies the reducts to be related in all pre-orders from levels $h'_1$ with $\phi(h'_1) = \phi(h_1)$. ■

The category of those hierarchies of pre-orders that are the differential semantics of some specification is defined as expected.

**Definition 21** The category $\mathsf{hiePref}$ is the full subcategory of $\mathsf{hiePre}$ with objects the $\Sigma$-hierarchies of pre-orders $\mathcal{H}$ such that $\mathcal{H} = \mathcal{H}^{\ominus\ominus}$. ■

### Syntax and Semantics

Hierarchic specifications are related in the category $\mathsf{hieSpec}$ via their theories, or equivalently, via their meaning. This fact is made explicit by the following lemma: hierarchic specifications are related in $\mathsf{hieSpec}$ precisely in the same way that their semantics are related in $\mathsf{hiePref}$ .

**Lemma 22** There is a $\mathsf{hieSpec}$ morphism $(\sigma, \phi) : S_1 \to S_2$ iff there is a $\mathsf{hiePre}$ (or equivalently a $\mathsf{hiePref}$) morphism $\overleftarrow{(\sigma, \phi)} : S_2^{\ominus} \to S_1^{\ominus}$.

The syntactic and semantic categories (and also their constructions) are related by the functors $\mathsf{hieSem} : \mathsf{hieSpec} \to \mathsf{hiePref}^{\mathsf{op}}$ and $\mathsf{hieSyn} : \mathsf{hiePref}^{\mathsf{op}} \to \mathsf{hieSpec}$. Their definition follows.

**Definition 23** The functor $\mathsf{hieSem} : \mathsf{hieSpec} \to \mathsf{hiePref}^{\mathsf{op}}$ associates to each hierarchic specification $S$ its hierarchy of differential preferences $S^{\ominus}$ and to each $\mathsf{hieSpec}$ morphism $(\sigma, \phi) : S_1 \to S_2$ the $\mathsf{hiePref}$ morphism $\overleftarrow{(\sigma, \phi)} : S_2^{\ominus} \to S_1^{\ominus}$.

The functor $\mathsf{hieSyn} : \mathsf{hiePref}^{\mathsf{op}} \to \mathsf{hieSpec}$ associates to each hierarchy of differential preferences $S^{\ominus}$ its theory $S^{\ominus\ominus}$ and to each $\mathsf{hiePref}$ morphism $\overleftarrow{(\sigma, \phi)} : S_2^{\ominus} \to S_1^{\ominus}$ the $\mathsf{hieSpec}$ morphism $(\sigma, \phi) : S_1^{\ominus\ominus} \to {S'}_2^{\ominus\ominus}$.

The semantics of the composition of hierarchic specifications is obtained by combining the semantics of the argument specifications. That is to each colimit of a diagram $\mathcal{D}$ in $\mathsf{hieSpec}$ there corresponds a limit in $\mathsf{hiePref}$ involving the semantics of the hierarchic specifications in $\mathcal{D}$. The limit of the image of $\mathcal{D}$ via $\mathsf{hieSem}$. The reverse is also true. In this way it is possible to define on the semantics side a composition of hierarchic specifications and then check what is its syntactical expression (in particular the resulting hierarchic specification).

**Theorem 24** The image by $\mathsf{hieSyn}$ of a limit in $\mathsf{hiePref}$ is a colimit in $\mathsf{hieSpec}$ and the image by $\mathsf{hieSem}$ of a colimit in $\mathsf{hieSpec}$ is a limit in $\mathsf{hiePref}$.

### Existence of Constructions

The category $\mathsf{StPart}$ of partial orders is not cocomplete. Therefore, $\mathsf{hieSpec}$ does not have all colimits and $\mathsf{hiePref}$ does not have all limits. However, when the underlying category $\mathsf{Sign}$ of signatures is cocomplete, that is the only reason why such constructions may not exist. In this case all colimits of (small) diagrams exist in $\mathsf{hieSpec}$ provided that the corresponding combination of partial orders exists. The same is true for the limits in $\mathsf{hiePref}$.

Existence of colimits in $\mathsf{hieSpec}$ is presented in the following theorem.

**Theorem 25** Let $\mathcal{D}$ be a diagram in $\mathsf{hieSpec}$. Then $\mathcal{D}$ has a colimit in $\mathsf{hieSpec}$ if the category $\mathsf{Sign}$ of signatures is cocomplete and the diagram $\mathcal{P}(\mathcal{D})$ has a colimit in $\mathsf{StPart}$, where $\mathcal{P} : \mathsf{hieSpec} \to \mathsf{StPart}$ is the forgetful functor sending each hierarchic specification to its priority structure (i.e. the composition of the corresponding partial orders of priority is defined).

Existence of limits in $\mathsf{hiePre}$ and $\mathsf{hiePref}$ is assured by theorems 24 and 25.

# 3 Conclusions

We have presented an institution independent theory of composition of hierarchic specifications that generalizes the classical theory from Goguen and Burstall. Composition is formalized on the syntactic and semantic levels by canonical operations on appropriate categories. For this purpose a new semantics of hierarchic specifications, the hierarchy of differential preferences, has been introduced. A corresponding notion of theory has been defined and a Galois connection between hierarchic specifications and their semantics has been established.

The category of hierarchic specifications is obtained by comparing hierarchic specifications via their theories by axiom and default preserving signature and priority structure morphisms. Canonical (co-)constructions in this category represent compositions of hierarchic specifications and have been shown to exist provided that both a result signature and a result priority structure exist.

The category of hierarchic specifications is mirrored on the semantic side by the category of their semantics. This implies, in particular, that the composition of hierarchic specifications can be interpreted on the semantic side by corresponding semantic constructions. In other words we have provided a denotational semantics for the composition of hierarchic specifications.

The theory just presented formalizes structuring constructs from specification languages providing explicit non-monotonic mechanisms. The identification of the most important structuring constructs (such as parameterization in this setting of hierarchic specifications) is the main subject of future work.

## References


[1] J. Adámek, H. Herrlich, and G. Strecker. *Abstract and Concrete Categories.* John Wiley & Sons, New York, 1990.

[2] H. Andréka, M. Ryan, and P.-Y. Schobbens. Operators and laws for combining preference relations. In R. J. Wieringa and R. Feenstra, editors, *IS-CORE'94 - Selected papers*, pages 191–206. World Scientific Publishers, 1995.

[3] S. Braß. Beginnings of a theory of general database completions. In Serge Abiteboul and Paris C. Kanellakis, editors, *Third International Conference on Database Theory (ICDT'90)*, number 470 in LNCS, pages 349–363. Springer, 1990.

[4] S. Braß. Deduction with supernormal defaults. In P. Schmitt G. Brewka, K. Jantke, editor, *Nonmonotonic and Inductive Logic - Second International Workshop, 1991*, pages 153–174, Berlin, 1992. Springer.

[5] S. Braß. *Defaults in Deduktiven Datenbanken.* PhD thesis, University of Hannover, 1992.

[6] S. Braß and U. W. Lipeck. Specifying closed world assumptions for logic databases. In János Demetrovics and Bernhard Thalheim, editors, *2nd Symp. on Mathematical Fundamentals of Database Syst. (MFDBS'89)*, number 364 in LNCS, pages 68–84. Springer, 1989.

[7] S. Braß and U. W. Lipeck. Semantics of inheritance in logical object specifications. In Claude Delobel, Michael Kifer, and Yoshifumi Masunaga, editors, *Deductive and Object-Oriented Databases, 2nd Int. Conf. (DOOD'91)*, number 566 in LNCS, pages 411–430. Springer, 1991.

[8] S. Braß, U. W. Lipeck, and P. Resende. Specification of object behaviour with defaults. In U. W. Lipeck and G. Koschorreck, editors, *Proc. of the Int. Workshop on Information Systems — Correctness and Reusability (ISCORE'93)*, Informatik-Bericht 01/93, Universität Hannover, pages 155–177, 1993.

[9] S. Braß, M. Ryan, and U. W. Lipeck. Hierarchical defaults in specifications. In G. Saake and A. Sernadas, editors, *Information Systems — Correctness and Reusability, Workshop IS-CORE '91*, number 91-03 in Informatik-Bericht, pages 179–201. TU Braunschweig, 1991.

[10] G. Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1991.

[11] R. M. Burstall and J. A. Goguen. Putting theories toghether to make specifications. In Raj Reddy, editor, *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1045–1058, Cincinnati, Ohio, 1977. Department of Computer Science, Carnegie-Mellon University.

[12] C. Caleiro. On the relationship between operational and denotational semantics of temporal logic specification of object behaviour. In R. J. Wieringa and R. Feenstra, editors, *IS-CORE'94 - Selected papers*, pages 69–83. World Scientific Publishers, 1995.

[13] F. M. Dionísio, U. W. Lipeck, and S. Braß. Composition of default specifications. In R. J. Wieringa and R. Feenstra, editors, *IS-CORE'94 - Selected papers*, pages 207–221. World Scientific Publishers, 1995.

[14] F. Dionísio. *Composition of Hierarchic Default Specifications.* PhD thesis, University of Hannover, 1997. Available by ftp at ftp://yoda.math.ist.utl.pt/pub/DionisioFM/phd.ps.

[15] J. Fiadeiro and T. Maibaum. Describing, structuring and implementing objects. In J. W. de Bakker, W. P. de Roever, , and G. Rozenberg, editors, *Foundations of Object-Oriented Languages*, number 489 in LNCS, pages 275–310. Springer, 1991.

[16] J. Fiadeiro and T. Maibaum. Temporal theories as modularization units for concurrent system specification. *Formal Aspects of Computing*, 4:239–272, 1992.

[17] P. Gärdenfors, editor. *Belief Revision.* Cambridge Press, Cambridge, England, 1992.

[18] P. Gärdenfors. Belief revision: An introduction. In Peter Gärdenfors, editor, *Belief Revision*, number 29 in Cambridge Tracts in Theoretical Computer Science, pages 1–28. Cambridge University Press, 1992.

[19] J. A. Goguen. A categorial manifesto. Technical report prg-72, Programming Research Group, University of Oxford, 1989.

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

[21] D. Makinson. Five faces of minimality. *Studia Logica*, 52:339–379, 1993.

[22] J. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[23] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[24] R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 55–76. Plenum Press, New York, 1978.

[25] M. Ryan. Defaults and revision in structured theories. In *Proc. Sixth Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 362–373, Los Alamitos, CA, 1991. IEEE Computer Society Press.

[26] M. Ryan. *Ordered Presentations of Theories - Default Reasoning and Belief Revision*. PhD thesis, University of London, Imperial College, London, 1992.

[27] M. Ryan. Representing defaults as sentences with reduced priority. In B. Nebel and W. Swartout, editors, *Proc. Second International Conference on Principles of Knowlage Representation and Reasoning (KR'92)*. Morgan Kaufmann, 1992.

[28] M. Ryan. Defaults in specifications. In A. Finkelstein, editor, *Proc. IEEE International Symposium on Requirements Engeneering (RE'93)*, pages 142–149, 1993.

[29] G. Saake, A. Sernadas, and C. Sernadas. Evolving object specifications. In R. J. Wieringa and R. Feenstra, editors, *IS-CORE'94 - Selected papers*, pages 84–99. World Scientific Publishers, 1995.

[30] A. Sernadas, J. F. Costa, and C. Sernadas. An institution of object behaviour. In H. Ehrig and F. Orejas, editors, *Recent Trends in Data Type Specification*, pages 337–350. Springer, 1994. LNCS 785.

[31] A. Sernadas, J. Fiadeiro, C. Sernadas, and H.-D. Ehrich. Abstract object types: A temporal perspective. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Temporal Logic in Specification*, pages 324–350. Springer, 1989.

[32] A. Sernadas and C. Sernadas. Denotational semantics of object specification within an arbitrary temporal logic institution. In U. W. Lipeck and G. Koschorreck, editors, *Proceedings of the International Workshop on Information Systems – Correctness and Reusability (IS-CORE'93)*, number 01-93, pages 61–89, Hannover, 1993. Universität Hannover.

[33] A. Sernadas, C. Sernadas, and J. F. Costa. Object specification logic. *Journal of Logic and Computation*, 5(5):603–630, 1995.

[34] Y. Shoham. Nonmonotonic logics: meaning and utility. In *Proceedings of IJCAI-87*, pages 388–392, Milan, 1987.