Composition of Hierarchic Default Specifications

Von dem Fachbereich Mathematik
der Universität Hannover

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
Dr. rer. nat.

genehmigte Dissertation

von

Diplom Ingenieur **Francisco Miguel Alves Campos de Sousa Dionísio**

geboren am 8.4.1961 in Beja, Portugal

1997

Referent:	Prof. Dr. U. Lipeck	
Korreferenten:	Prof. Dr. H. Pfeiffer	
	Prof. Dr. A. Sernadas,	Instituto Superior Técnico, Lissabon, Portugal
Tag der Promotion:	16.07.1997	

Zusammenfassung

In dieser Dissertation wird eine Kompositionstheorie hierarchischer Spezifikationen eingeführt, die unabhängig von der gewählten Logik ist. Hierarchische Spezifikationen sind Strukturen, die logische Formeln durch partiell geordnete Prioritätsstufen organisieren und die Spezifikation von allgemeinen Regeln mit Ausnahmen (und Ausnahmen von Ausnahmen) mit prioritisierten Formeln ("Defaults") formalisieren.

Die Komposition von hierarchischen Spezifikationen wird durch kanonische Operationen in syntaktischen und semantischen Kategorien definiert.

Diese Arbeit generalisiert die klassische Komposition von Präsentationen (Goguen und Burstall, 1989) und verleiht der syntaktischen Komposition von Hierarchischen Spezifikationen (Braß, Lipeck und Ryan, 1991) eine semantische Seite.

Die folgenden Konzepte und Eigenschaften werden mit dieser Arbeit eingeführt.

- **Minimale Semantik**. Eine neue Semantik von Hierarchischen Spezifikationen, die eine Hierarchie von Präferenzrelationen zwischen Modellen ist, wird definiert. Es wird gezeigt, daß diese Semantik die *minimale* Semantik ist, die bestimmte logische und kompositionelle Eigenschaften erfüllt.

- **Galois-Dualität**. Es wird eine Galois-Dualität ("Galois connection") für hierarchische Spezifikationen und deren Semantiken nachgewiesen. Die Galois-Dualität ist eine bijektive Abbildung zwischen *Theorien* hierarchischer Spezifikationen und deren Semantiken, die eine bijektive Abbildung zwischen syntaktischen und semantischen Operationen impliziert. Dieses ist die Grundlage der folgenden Kompositionstheorie.

- **Kategorielle Konstruktionen**. Es wird eine Kategorie (hieSpec) Hierarchischer Spezifikationen und eine entsprechende Kategorie ihrer Semantiken (hiePref) definiert. Morphismen zwischen hierarchischen Spezifikationen ergeben sich aus Morphismen zwischen den zugrundeliegenden Signaturen und Morphismen zwischen den entsprechenden Prioritätsstrukturen.

  Die Komposition hierarchischer Spezifikationen wird durch Colimiten in der Kategorie hieSpec oder durch Limiten in der semantischen Kategorie hiePref formalisiert. Es wird außerdem gezeigt, daß die Existenz von solchen kanonischen Konstruktionen auf Signaturen und Prioritätstrukturen die Existenz entsprechender Konstruktionen in hieSpec und hiePref impliziert.

Abstract

The main contribution of this work has been the establishment of an institution independent theory of composition of hierarchic specifications. Hierarchic specifications consist of formulas ("defaults") from an underlying logic (institution), organized by priority levels (related by a partial order). These formulas can be defeated when in contradiction with more reliable information (at more important priority levels). Hierarchic specifications model structures with several levels of overriding of general properties, such as those occurring in the specification of classes and subclasses.

Composition of hierarchic specifications is formalized in the syntactic and semantic levels by canonical operations in appropriate categories. This composition generalizes the classical framework of institutions (Goguen and Burstall, 1989) and defines the semantics of the syntactic composition of hierarchic specifications (Braß, Lipeck, Ryan, 1991).

The following concepts and properties have been introduced in this thesis.

- **Minimal Semantics**. A new semantics for hierarchic specifications, a hierarchy of preference relations, has been defined. This semantics is shown to be the *minimal* semantics satisfying certain logical and compositional conditions.

- **Galois Connection**. We have shown a Galois connection between hierarchic specifications and their semantics. It expresses the one to one mapping between the *theories* of hierarchic specifications and their semantics. It implies a one to one mapping between syntactic and semantic operations. It is the basic mechanism of the theory of composition.

- **Categorial Constructions**. A category (hieSpec) of hierarchic specifications and a "mirror" category of their semantics (hiePref) have been defined. Hierarchic specification morphisms consist of signature and priority structure morphisms.

  Composition of hierarchic specifications is formalized by colimits in the category hieSpec, or by limits in the semantic category hiePref.

  Existence of these constructions (both in hieSpec and hiePref) is guaranteed by existence of the corresponding signature and priority structure constructions.

Acknowledgments

I would like to thank my supervisor Udo Lipeck for his advice and support, and especially for setting the main directions of research and correcting previous drafts of this thesis. Also to Amílcar Sernadas, my previous supervisor, for his guidance through all these years.

I am also grateful to Prof. Dr. Helmut Pfeiffer that, together with Udo Lipeck and Amílcar Sernadas refereed my thesis.

Many important ideas benefit from discussions with Stefan Braß, Mark Ryan, Michael Gertz, Cristina Sernadas, José Carmo, Pierre-Yves Schobbens, José Fiadeiro and Tom Maibaum.

Thanks also to my other colleagues from Hannover and Lisbon, for their patience and support: Herren Zhou, Gerhard Koschorrek, Paula Gouveia, Mª João Coutinho, Carlos Caleiro, Jaime Ramos and Pedro Resende. And to Frau Schumann and Rita Torres for their skills reasoning in the presence of inconsistent bureaucratic information.

The ideas put forward by the Esprit Basic Research action IS-CORE (Information Systems - Correctness and Reusability) very much influenced my own research.

Much of this research has been done at the Institut für Informatik from the University Hannover. During this time I have been on leave from Departamento de Matemática, Instituto Superior Técnico, Lisbon as a grantee of Junta Nacional de Investigação Científica e Tecnológica.

This thesis is dedicated to my wife and kids.

To

Mechthild, Roman and Teresa

Contents

1 Defaults in Institutions **7**

1.1 Institutions . . . 8

1.1.1 Definition . . . 8

1.1.2 Presentations and their Semantics . . . 11

1.1.3 Theories . . . 12

1.1.4 Entailment and Monotonicity . . . 14

1.1.5 Propositional Logic . . . 15

1.2 Specifications . . . 18

1.2.1 Syntax and Semantics . . . 18

1.2.2 Theories . . . 21

1.2.3 Pre-orders and Preference Relations . . . 26

1.2.4 Special Cases . . . 30

1.2.5 Theories Revisited . . . 35

1.3 Extensions . . . 38

1.3.1 Consequences . . . 39

1.3.2 Preference . . . 41

1.3.3 Compact Institutions . . . 44

1.3.4 Selection Functions . . . 46

1.3.5 Extensions and Composition . . . 47

1.4 Final Remarks . . . 48

2 Prioritized Defaults **49**

2.1 Hierarchic Specifications . . . 50

2.1.1 Syntax and Lexicographic Preference 50
2.1.2 Semantics for Composition 57
2.1.3 Combination Operators 64
2.1.4 Relations between Semantics 68
2.2 Lexicographic Preference Revisited 72
2.2.1 Structured Semantics 72
2.2.2 Specifications 79
2.3 Theories 86
2.3.1 Definition and Galois Connection 86
2.3.2 Theories Revisited 95
2.4 Extensions 105
2.4.1 Definition 105
2.4.2 Consequences 108
2.4.3 Selection Function 108
2.4.4 Properties 109
2.5 Compact Institutions 110
2.5.1 Existence of Extensions 110
2.5.2 Selection Function 112
2.5.3 Properties 112
2.6 Final Remarks 115

3 Composition **117**
3.1 Specifications 118
3.1.1 Category of Specifications 119
3.1.2 Category of Pre-orders 121
3.1.3 Syntax and Semantics 124
3.1.4 Isomorphism and Extensions 128
3.1.5 Existence of Constructions 130
3.1.6 Canonicity of the Semantics 134
3.2 Hierarchic Specifications 139
3.2.1 Category of Partial Orders 139

3.2.2 Category of Hierarchic Specifications 142
3.2.3 Category of Hierarchies of Pre-orders 145
3.2.4 Syntax and Semantics 147
3.2.5 Isomorphism and Extensions 149
3.2.6 Existence of Constructions 150
3.2.7 Canonicity of the Semantics 154
3.2.8 Other Composition Forms 158
3.3 Final Remarks 160

4 Use of Defaults in Specification 161

4.1 Operations on Specifications 162
4.1.1 Composition 162
4.1.2 Simplification of Structure 163
4.2 The Underlying Logic 166
4.2.1 Syntax 166
4.2.2 Semantics 167
4.2.3 Built in Axioms 167
4.2.4 Decidability 168
4.3 Classes 168
4.3.1 Past, Present and Future 168
4.3.2 Occurrence and Enabling 169
4.3.3 Parameterless Actions 169
4.3.4 Creation and Destruction 169
4.3.5 Locality and the Frame Rule 169
4.3.6 Default Enabling Conditions 170
4.3.7 Sequentiality 170
4.3.8 Liveness 171
4.4 Files and Users 171
4.4.1 Signatures 171
4.4.2 Locality 174
4.4.3 Sequentiality 175

4.4.4 Creation . . . 176
4.4.5 Destruction . . . 179
4.4.6 More Actions . . . 180
4.4.7 Overall Specification . . . 182
4.5 More Files and Users . . . 182
4.5.1 Executable Files . . . 182
4.5.2 Classes and Subclasses . . . 183
4.5.3 Cristalization . . . 185
4.5.4 Communication by Action Sharing . . . 185
4.5.5 Multiplying Files and Users . . . 187
4.5.6 Concurrency . . . 188
4.5.7 Cristalization Again . . . 188
4.6 Animating the Specification . . . 189
4.7 Final Remarks . . . 190

5 Final Remarks **191**
5.1 Summary . . . 191
5.2 Further Work . . . 194

Introduction

Variety and complexity of software systems led to the development of specification languages and formal methods providing the conceptual tools for their rigorous description[1].

Variety of software systems implies variety of methods and models to describe them. They can be analyzed as structures of interrelated entities (Entity Relationship model) or as programs manipulating abstract data types (Abstract Data Type school). Static and dynamic aspects may be integrated and systems specified as communities of interacting objects (Object Orientation). Specification can alternatively concentrate in the functionalities (Feature Orientation) that such systems should provide.

Complexity of software systems demands that specification languages provide structuring mechanisms: "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"[44].

Classical Specification Theory

The theory of institutions from Goguen and Burstall ([46]) provides the formalization of classical structuring operations. This theory (and its many developments) constitutes an "abstract specification theory"[2] and has influenced the design of the languages Clear ([16]), OBJ ([39, 47]), Eqlog ([48]), FOOPS ([49]), Oblog ([86]), Gnome ([71]) and Troll ([58]).

The notion of an arbitrary logical system is formalized by an institution, using abstract model theory[3]([3]). The basic idea motivating the institutional framework is that specifications, i.e. rigorous descriptions of parts of a system, denote

[1] In spite of this, "unreliable software is the norm rather than the exception"[81].

[2] The term "abstract specification theory" is taken from [31].

[3] The semantic structure can be substituted by "syntactic" consequence: see the Π-institutions from ([34, 36]).

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 ([83, 30, 31, 84, 82, 17, 80, 85]). 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 ([16]) can be used to build large specifications from theories from *any* logical system.

Non-monotonic Specification Theory

The main concern of this thesis is to provide an abstract specification theory that formalizes non-monotonic composition constructs, thus extending 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.

The third is that the specification process itself is non-monotonic since the revision of previous oversimplified descriptions of the universe of discourse may not only add information but also contradict previously specified information.

And, finally, the fourth 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"[12].

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 lent. If, however, this book is a reserved book this conclusion can be defeated by explicit information stating that *reserved books* cannot be lent.

Defaults in this thesis are 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 [14] and further studied in [7, 6, 75, 76]. Their impact in specification is stated in [9, 12].

This structure of axioms and prioritized defaults (and the corresponding semantics) has been used to model different database completions (see [8, 5]), the frame rule (see [14, 11]) and the "taxonomic" structure of classes and subclasses (see [75, 76, 77]). The operation of adding new information at a new most important priority level (recall the specification of *reserved books*) is a belief revision ([43, 42]) operator (see [75]).

Axioms and prioritized defaults are the modularization units used in the theory of composition developed in this work. 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[4].

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 (see chapter 3) involving the specification *books* and the "difference" between *reserved books* and *books*. Chapter 4 provides further examples of specification constructions.

[4]This structure of axioms and prioritized defaults is called an hierarchic specification. Therefore specification modules denote hierarchic specifications. This formulation is unfortunate in this context. The rest of the thesis will no longer refer to specification modules, only to their denotations, the hierarchic specifications.

Purpose of the Thesis

A preliminary goal of this thesis is to investigate the properties of hierarchy specifications, that consist of axioms and prioritized defaults from an arbitrary logic (institution). The main contribution, however, is to provide an abstract specification theory using hierarchic specifications as modularization units. This framework defines the semantics of the syntactical composition of hierarchic specifications from [12] and is presented in chapter 3. It corresponds to formalize, independently of the underlying logic, the structuring operations of hierarchic specifications. These operations are formalized both on the syntactic and semantic levels, and account for the modular construction of hierarchic specifications by combining, reusing and modifying (with overriding) previously specified modules.

Outline of the Thesis

The core of this thesis is chapter 3 that contains the theory of composition of hierarchic specifications.

Chapter 1 presents specifications, a special case of hierarchic specifications (with one only priority level). Logical properties of specifications from arbitrary institutions are stated. The correspondence between specification operations and semantic operations (a Galois connection) is shown. It is the basic mechanism of the theory of composition. The Galois connection implies a notion of theory that is characterized.

Chapter 2 has the same structure as chapter 1 and extends the concepts and properties of specifications to hierarchic specifications. A new semantics of hierarchic specifications is introduced, and the correspondence between hierarchic specification operations and semantic operations is shown. The corresponding notion of theory is characterized. Hierarchic specifications are shown to have the same logical content as corresponding specifications under some conditions on the underlying logic.

Chapter 3 is divided in two parts, the first dedicated to the theory of composition of specifications and the second to the generalization of that theory to hierarchic specifications. Composition of specifications and hierarchic specifications is formalized by canonical constructions on appropriate syntactic and semantic categories. The correspondence between semantic and syntactical constructions (for specifications and hierarchic specifications) follows from the corresponding Galois connections. These constructions are shown to exist under the condition that the corresponding signature and priority structure constructions exist. Moreover it is shown that the adopted semantics of specifications and hierarchic specifications are the minimal semantics assuring composition.

Chapter 4 illustrates the use of the composition operations in specification.

Prerequisites

Knowledge of both classical logic and non-monotonic logic is needed for the understanding of the thesis. In [63] the most important formalisms are carefully introduced. The theory of institutions ([46]) uses some concepts from category theory that may be found for instance in [51, 1].

Chapter 1

Defaults in Institutions

Default reasoning formalizes the ability to assume and use in reasoning a property which is likely to be true but not certainly true. Such properties are expressed by formulas, the so called "defaults", in some underlying logic. Properties that are known to be certain are expressed also by formulas in the same underlying logic, the "axioms".

Default reasoning has been originally defined on first order logic ([73]) and after that applied to other formalisms. In particular "defaults" in temporal logic(s) have been studied in the context of specifying dynamic systems ([89, 9, 61]). Also significant are the connections between default reasoning and deontic logic in the context of system specification ([68]).

Our purpose in this chapter is to investigate the use of defaults in an arbitrary logical system, following the trend set in [12]. For that purpose we accept the concept of *institution* ([45, 46]) as a convenient formalization of the notion of "arbitrary monotonic logical system". Our aim, therefore, is not to generalize the notion of institution in order to encompass non-monotonic logics (as in [81]) but to add non-monotonic features to existing monotonic logics. Institutions are covered in section 1.1.

The notion of *specification* (with "defaults") and its *preference semantics* will be parameterized in the underlying institution and defined in section 1.2. A Galois connection is established implying that operations on specifications are mirrored by corresponding operations on the semantics.

In section 1.3 we define *extensions* and different notions of *consequence* of a specification. These are related to the preference semantics presented in section 1.2. Important properties of default reasoning in an arbitrary institution (for example the existence of *extensions*) are investigated.

We conclude the chapter in section 1.4.

1.1 Institutions

In this section the definition of *institution*, a formalization of the notion of logical system, due to Goguen and Burstall ([45, 46]) is presented. The notions of *presentation*, its *semantics* and *theory*, defined within an institution, are reviewed. Furthermore the interplay between an arbitrary presentation and its semantics (given by a Galois connection) is highlighted. A general notion of entailment is defined, and it is remarked that institutions are monotonic with respect to entailment. These concepts are illustrated by displaying the institution of propositional logic.

1.1.1 Definition

Institutions are a formalization, proposed by Goguen and Burstall in [46], of the general notion of a logical (monotonic) system. This formalization encompasses the following logical systems: equational logic, (many-sorted) first order logic (with or without equality), horn clause logic with equality, inequational logic, infinitary equational logic (continuous algebras), modal and temporal logic(s), intuitionistic logic, and the λ-calculus. The theory of institutions gives semantics to the specification languages Clear ([16]) and OBJ ([39, 47]) and has also been used in designing the programming languages Eqlog ([48]) and FOOPS ([49]).

Institutions provide a formal means to study composition of theories written in a particular logic. This study will be generalized to specifications with defaults in this thesis. Moreover the theory of institutions also provides formal account of the relation between different logics (via institution morphisms).

We begin by motivating the definition of institution. For purposes of illustration the institution of propositional logic will be analyzed in some detail in the sequel. (For other institutions, including first order logic see [45].)

On the syntactic side it is recognized that within the same logic, while keeping the connectives fixed, one may have different sets of symbols in mind. In propositional logic this amounts to decide which propositional symbols to use and in first order logic to decide which predicate and function symbols to use. Each choice of symbols is called a *signature*. Within the same logical system signatures are related to each other by signature morphisms. An inclusion morphism, for example, states that the target signature has more symbols that the source one. Signatures and signature morphisms constitute a category (called Sign).

To a given signature one must be able to assign the corresponding set of formulas. Usually this is accomplished inductively by applying the (operators on sets of formulas corresponding to the) connectives to the *atomic formulas*. This assignment of sets of formulas to signatures is abstracted by the functor Sen[1] that also accounts for the fact that relations between signatures result in relations between the corresponding languages. For example signatures related by inclusion induce languages related by inclusion.

Each signature has associated a corresponding category of *interpretation structures*. Interpretation structures are in propositional logic the assignments of truth values to the propositional symbols from the signature and their morphisms are trivial (see below). In first order logic interpretation structures are the algebras that interpret the predicate and function symbols from the signature over a carrier set (or carrier sets in the multi-sorted case). Morphisms of first order logic interpretation structures are algebra homomorphisms respecting the interpretation of function symbols and satisfaction of predicate symbols.

The assignment of signatures to the corresponding category of interpretation structures is abstracted by the functor Mod[2]. To a signature morphism the functor Mod associates a functor from the category of interpretation structures of the target signature to the category of interpretation structures of the source signature (note that the direction of the signature morphism is reversed in the semantics). This provides a way to "reduce" interpretation structures of the target signature to interpretation structures of the source signature. In the case that the signature morphism is an inclusion the "reduction" of an interpretation structure of the target signature is the "restriction" of its interpretation of symbols to the (lesser) symbols of the source signature. Interpretation structure morphisms of the target signature are translated to interpretation structure morphisms between the "reduced" interpretation structures.

Finally each formula from a given signature Σ is given meaning by stating the interpretation structures from the same signature where that formula holds. This corresponds to the usual "semantic definition of truth" from Tarski ([90]) and is formalized by the relation $\models_\Sigma$. The relations $\models_\Sigma$ for different signatures cooperate in such way that when changing formulas from one signature to formulas from another their meaning changes correspondingly. This condition is known as the *Satisfaction Condition*.

The definition of institution follows. Note that Set is the category of sets and functions, Cat the category[3] of categories and functors between them (and Cat^{op}

[1] Formulas are in [46] referred to as sentences.

[2] Interpretation structures are in [46] referred to as models.

[3] The opposite of Cat is the codomain of the functor $\mathsf{Mod}\colon \mathsf{Sign} \to \mathsf{Cat}^{op}$ that assigns to a signature the category of its interpretation structures. In first order logic the interpretation structures of some signature form a proper class, and its category is a proper category (not a small category). Therefore, at least for first order logic Cat must be the "category of all categories" and not the category of all *small* categories. The "category of all categories" is not a category but a *quasicategory* (see [1]). There are no foundational problems with quasicategories (see again [1]) and, for the constructions of this thesis, they may be seen as categories. We

its opposite category). Given a category $\mathcal{C}$, the class of its objects is denoted by $|\mathcal{C}|$.

Definition 1 An *institution* consists of

- a category 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 Σ a category whose objects are called Σ-*interpretation structures* and whose morphisms are the Σ-interpretation structure morphisms, and
- a relation $\models_\Sigma \subseteq |\mathsf{Mod}(\Sigma)| \times \mathsf{Sen}(\Sigma)$, called Σ-*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 $\mathsf{Sen}(\Sigma_1)$. ∎

The relations between the concepts constituting an institution are illustrated in the following figure 1.1.

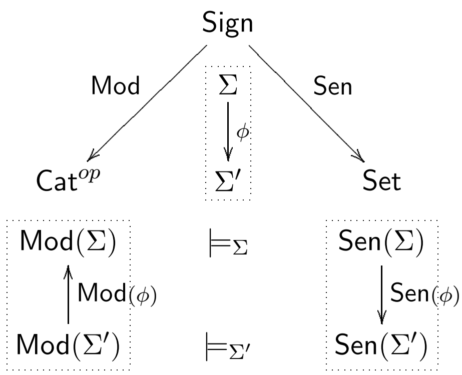

Figure 1.1: Functors Mod and Sen

will not mention this distinction further, and note only that all "semantic categories" to be introduced are quasicategories.

1.1.2 Presentations and their Semantics

In order to build a formal picture of some Universe of Discourse the "specifier" should organize it in parts, describe formally these parts in a logic considered convenient to the problem at hand (and it may be the case that different parts express themselves better in different logics) and put the formalizations together to build the overall picture.

The parts of such a specification are classically formalized by a set of formulas, a *presentation*, describing the (intended or actual) properties of such a part. These formulas are written in an appropriate signature of the chosen logic. Here we are concerned with presentations from a fixed but arbitrary institution, their semantics and properties. The concepts here presented are a necessary kernel for the generalization to specifications using defaults and their composition.

Remark 2 Throughout this section concepts are defined in the scope of

$$\mathcal{I} = (\mathsf{Sign}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \{\models^{\mathcal{I}}_{\Sigma}, \Sigma \in |\mathsf{Sign}^{\mathcal{I}}|\}),$$

a fixed but arbitrary institution.

Definition 3 A *presentation* (from the institution $\mathcal{I}$) is a pair (Σ, 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 Σ.

A Σ-*presentation* A is a presentation (Σ, A). ▪

The interpretation structures that satisfy all formulas in a given presentation are said to *satisfy* the presentation and called *models* of the presentation.

Definition 4 Let A be a Σ-presentation.

- A Σ-interpretation structure $m \in |\mathsf{Mod}^{\mathcal{I}}(\Sigma)|$ *satisfies* the Σ-presentation A, written $m \models^{\mathcal{I}}_{\Sigma} A$ iff for all $s \in A, m \models^{\mathcal{I}}_{\Sigma} s$.
- When $m \models^{\mathcal{I}}_{\Sigma} A$ then m is said a to be *model* of the Σ-presentation A.
- A class $\mathcal{M} \subseteq |\mathsf{Mod}^{\mathcal{I}}(\Sigma)|$ of Σ-interpretation structures *satisfies* the Σ-presentation A, written $\mathcal{M} \models^{\mathcal{I}}_{\Sigma} A$ iff for all $m \in \mathcal{M}, m \models^{\mathcal{I}}_{\Sigma} A$. ▪

1.1.3 Theories

The semantics of a presentation is the class of all its models[4]. The *theory* of a class of interpretation structures is the set of formulas holding in each interpretation structure of that class.

Definition 5

1. The function $^\bullet$ assigns to a Σ-presentation A the class of all Σ-interpretation structures that are models of A;

$$A^\bullet = \{m : m \in \mathsf{Mod}^{\mathcal{I}}(\Sigma) \text{ and } m \models^{\mathcal{I}}_{\Sigma} A\}.$$

2. The function $^\bullet$ assigns to a class $\mathcal{M} \subseteq |\mathsf{Mod}^{\mathcal{I}}(\Sigma)|$ of Σ-interpretation structures the set of all Σ-formulas that are satisfied in each interpretation structure from $\mathcal{M}$;

$$\mathcal{M}^\bullet = \{f : f \in \mathsf{Sen}^{\mathcal{I}}(\Sigma) \text{ and for all } m \in \mathcal{M}, m \models^{\mathcal{I}}_{\Sigma} f\}.$$

   $\mathcal{M}^\bullet$ is called the *theory* of $\mathcal{M}$. ■

These two functions form a Galois connection[5] (see [46]). This means that relations between presentations (inclusion) are mirrored by relations between the corresponding semantics (again inclusion, but in the opposite direction). Moreover operations among presentations (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 classes of interpretation structures are also mirrored by relations and operations among the corresponding theories. The Galois connection properties are fundamental for formalizing composition of presentations.

Theorem 6 Let A, A' and A_n for $n \in \mathbb{N}$, be Σ-presentations and $\mathcal{M}, \mathcal{M}'$ and $\mathcal{M}_n$ for $n \in \mathbb{N}$, be classes of Σ-interpretation structures. Then

1. $A \subseteq A'$ implies $A^\bullet \supseteq A'^\bullet$,
2. $\mathcal{M} \subseteq \mathcal{M}'$ implies $\mathcal{M}^\bullet \supseteq \mathcal{M}'^\bullet$,
3. $A \subseteq A^{\bullet\bullet}$ and

[4] In fact these models are structured by interpretation structure morphisms. But this additional information is not relevant to the problems dealt with here.

[5] A (contravariant) Galois connection ([1]) is a pair of functions $g : A \to B$ and $f : B \to A$ that respect orderings on A and B: $a \leq f(b)$ iff $b \leq g(a)$. This is equivalent to $b \leq g(f(b))$ and $a \leq f(g(a))$.

4. $\mathcal{M} \subseteq \mathcal{M}^{\bullet\bullet}$.

These imply:

1. $A^{\bullet} = A^{\bullet\bullet\bullet}$,
2. $\mathcal{M}^{\bullet} = \mathcal{M}^{\bullet\bullet\bullet}$,
3. $(\bigcup_n A_n)^{\bullet} = \bigcap_n A_n^{\bullet}$,
4. $(\bigcup_n \mathcal{M}_n)^{\bullet} = \bigcap_n \mathcal{M}_n^{\bullet}$,
5. $(\bigcap_n A_n^{\bullet\bullet})^{\bullet} = (\bigcup_n A_n^{\bullet})^{\bullet\bullet}$,
6. $(\bigcap_n \mathcal{M}_n^{\bullet\bullet})^{\bullet} = (\bigcup_n \mathcal{M}_n^{\bullet})^{\bullet\bullet}$.

Proof See [46]. Derived properties 4 and 6 above are obtained by formal similarity with 3 and 5 respectively. ✓

The same class of interpretation structures is the semantics, in general, of different presentations, and also the same presentation may be the theory of different classes of interpretation structures, too. The relation between syntax and semantics can, however, be made bijective by considering only *closed* presentations and *closed* classes of interpretation structures.

Definition 7

- The *closure* of a Σ-presentation A is the Σ-presentation $A^{\bullet\bullet}$ (i.e. $(A^{\bullet})^{\bullet}$) of the Σ-formulas that hold in each Σ-interpretation structures that are models of A. A Σ-presentation A is *closed* iff $A = A^{\bullet\bullet}$. A closed Σ-presentation is also called a Σ-*theory*. The closure of a Σ-presentation is the Σ-*theory induced* by that Σ-presentation.

- The *closure* of a class $\mathcal{M}$ of Σ-interpretation structures is the class $\mathcal{M}^{\bullet\bullet}$ (i.e. $(\mathcal{M}^{\bullet})^{\bullet}$) of the Σ-interpretation structures that satisfy the Σ-formulas satisfied in each Σ-interpretation structure from $\mathcal{M}$. A class $\mathcal{M}$ of Σ-interpretation structures is *closed* iff $\mathcal{M} = \mathcal{M}^{\bullet\bullet}$.

Clearly a presentation is closed iff it is the theory of some class of interpretation structures and a class of interpretation structures is closed iff it is the class of models of some (closed) presentation. ■

The relation between closed presentations and closed classes of interpretation structures is one to one. Also two closed presentations are related by inclusion iff their semantics are related by inclusion (reversed).

Lemma 8 Let A, A' be closed Σ-presentations and $\mathcal{M}$ and $\mathcal{M}'$ be closed classes of Σ interpretation structures. Then

- $A \subseteq A'$ iff $A^\bullet \supseteq A'^\bullet$,
- $\mathcal{M} \subseteq \mathcal{M}'$ iff $\mathcal{M}^\bullet \supseteq \mathcal{M}'^\bullet$.

Proof Trivial from the Galois connection 6 above. ✓

Note that inclusion of presentations is not a convenient way of relating presentations. In fact there may be presentations, one having more information content than the other (measured by inclusion of the respective classes of models) which are not related by inclusion. Take, for example, the presentations $\{f\}$ and $\{f \wedge f\}$, from some signature Σ from an institution where $\wedge$ is a connective interpreted as conjunction. Clearly they mean the same (have the same semantics) but are not related by inclusion (since the formulas f and $f \wedge f$ are different).

Presentations that have the same semantics should be seen as equivalent and presentations should be related by their meaning and not by the specificity of the formulas used to describe that meaning.

As seen above closed presentations and relations between them provide the needed abstraction. This fact is emphasized in the following:

Lemma 9 Let A, A' be Σ-presentations.

- Two presentations have the same semantics iff they have the same closure (or induce the same theory); $A^\bullet = A'^\bullet$ iff $A^{\bullet\bullet} = A'^{\bullet\bullet}$,
- The theory induced by a presentation is the biggest (w.r.t. inclusion) presentation having the same semantics as the original one: if $A^\bullet = A'^\bullet$ then $A' \subseteq A^{\bullet\bullet}$.

Proof Trivial from the Galois connection 6 above. ✓

1.1.4 Entailment and Monotonicity

A formula is entailed by a presentation (or is a consequence of the presentation) if it holds in all models of that presentation, i.e. if it belongs to the corresponding theory.

Definition 10 A Σ-formula f is entailed by a Σ-presentation A, written $A \vDash_\Sigma f$ iff $f \in A^{\bullet\bullet}$. ■

With this definition of entailment it is straightforward to check that institutions are monotonic:

Theorem 11 Institutions are monotonic: given Σ-presentations $A \subseteq A'$, if $A \models_\Sigma f$ then $A' \models_\Sigma f$.

Proof Straightforward from the definition of entailment and properties 1 and 2 from the Galois connection. ✓

1.1.5 Propositional Logic

We illustrate the definition of institution by constructing the institution of propositional logic[6], referred to by Π. This corresponds to defining the category $\Pi\mathsf{Sign}$ of propositional signatures and signature morphisms, the functor $\Pi\mathsf{Sen}$ associating to a signature its language, the functor $\Pi\mathsf{Mod}$ associating to a signature its category of propositional interpretation structures and, for each propositional signature $\mathcal{P}$, the propositional satisfaction relations $\models^\Pi_\mathcal{P}$.

Signatures and signature morphisms are defined as follows.

Definition 12 A *propositional logic signature* $\mathcal{P}$ is a set (of propositional symbols). A *morphism of propositional signatures* from $\mathcal{P}$ to $\mathcal{P}'$ is a function $\phi : \mathcal{P} \to \mathcal{P}'$.

Let $\Pi\mathsf{Sign}$ denote the category with propositional signatures as its objects, with propositional morphisms as its morphisms and with the obvious identities and composition[7]. ■

We now proceed to define the functor $\Pi\mathsf{Sen}$ that sends signatures to the corresponding language and signature morphisms to functions between the languages. First we define the language associated with a propositional signature (we follow [50] and [41]). This is generated from the set of atomic formulas (in this case the set of propositional symbols, i.e. the signature), together with the falsum ($\perp$) by the implication connective $\Rightarrow$. An equivalent and more usual inductive definition is:

Definition 13 Given a propositional signature $\mathcal{P}$ the set of its formulas $\Pi\mathsf{Sen}(\mathcal{P})$ is inductively defined by:

- the falsum $\perp \in \Pi\mathsf{Sen}(\mathcal{P})$ and $p \in \Pi\mathsf{Sen}(\mathcal{P})$ for every propositional symbol $p \in \mathcal{P}$,

[6] Although we cannot trace a reference giving this same example we note that is is the simplest case from those presented in [46].

[7] In fact this is the category Set of sets and functions.

- whenever $\pi_1, \pi_2 \in \Pi\mathsf{Sen}(\mathcal{P})$ then $(\pi_1 \Rightarrow \pi_2) \in \Pi\mathsf{Sen}(\mathcal{P})$,
- a formula is in $\Pi\mathsf{Sen}(\mathcal{P})$ only if it is formed by the rules above. ■

Note that the connectives (and parenthesis) are global to the institution: the formulas of different signatures differ only in the propositional symbols they use, not in the other logical symbols.

Each signature morphism $\phi : \mathcal{P} \to \mathcal{P}'$ can be extended in a unique way to a function between the sets $\Pi\mathsf{Sen}(\mathcal{P})$ and $\Pi\mathsf{Sen}(\mathcal{P}')$ (see [41]).

Definition 14 Given a propositional signature morphism $\phi : \mathcal{P} \to \mathcal{P}'$ the function $\hat{\phi} : \Pi\mathsf{Sen}(\mathcal{P}) \to \Pi\mathsf{Sen}(\mathcal{P}')$ is inductively defined by:

- $\hat{\phi}(\bot) = \bot$ and for $p \in \mathcal{P}$, $\hat{\phi}(p) = \phi(p)$,
- $\hat{\phi}((\pi_1 \Rightarrow \pi_2)) = (\hat{\phi}(\pi_1) \Rightarrow \hat{\phi}(\pi_2))$ for $\pi_1, \pi_2 \in \Pi\mathsf{Sen}(\mathcal{P})$. ■

The functor $\Pi\mathsf{Sen} : \Pi\mathsf{Sign} \to \mathsf{Set}$ is now easily defined:

Definition 15 The functor $\Pi\mathsf{Sen} : \Pi\mathsf{Sign} \to \mathsf{Set}$ sends a propositional signature $\mathcal{P}$ to its language $\Pi\mathsf{Sen}(\mathcal{P})$ and a propositional signature morphism $\phi : \mathcal{P} \to \mathcal{P}'$ to the function $\hat{\phi} : \Pi\mathsf{Sen}(\mathcal{P}) \to \Pi\mathsf{Sen}(\mathcal{P}')$.

It is straightforward to check that $\Pi\mathsf{Sen}$ is indeed a functor. ■

We now proceed to define the functor $\Pi\mathsf{Mod}$ that to a propositional signature assigns the category of its interpretation structures and to a propositional signature morphism assigns a functor (in the reverse direction) between the categories of interpretation structures of the domain and codomain signatures.

We begin by defining the interpretation structures of a propositional signature and the corresponding category (of interpretation structures).

Definition 16 For $\mathcal{P}$ a propositional signature, a $\mathcal{P}$-*interpretation structure* is a truth assignment, i.e. a function $\tau : \mathcal{P} \to \{\mathsf{true}, \mathsf{false}\}$.

There is a (trivial) propositional $\mathcal{P}$-morphism between $\tau : \mathcal{P} \to \{\mathsf{true}, \mathsf{false}\}$ and $\tau' : \mathcal{P} \to \{\mathsf{true}, \mathsf{false}\}$ iff $\tau = \tau'$[8].

Let $\Pi\mathsf{Mod}(\mathcal{P})$ denote the category with propositional $\mathcal{P}$-interpretation structures as objects and with propositional $\mathcal{P}$-morphisms as morphisms. ■

[8]Other possibility is to choose a function $m : \mathcal{P} \to \mathcal{P}$ as morphism $m : \tau \to \tau'$ satisfying the condition that, for all $p \in \mathcal{P}$ if $\tau(p) = \mathsf{true}$ then $\tau'(m(p)) = \mathsf{true}$. This is closer to the morphism condition for first order logic that is treated in [46]. The trivial identity morphisms simplify our brief illustration of institutional concepts.

Remark 17 Note that since the morphisms in $\Pi\mathsf{Mod}(\mathcal{P})$ are trivial (the identities only) this category can be identified with the class of its objects, namely the class $|\Pi\mathsf{Mod}(\mathcal{P})|$ of $\mathcal{P}$-interpretation structures. Also a functor $\mathcal{F} : \Pi\mathsf{Mod}(\mathcal{P}) \to \Pi\mathsf{Mod}(\mathcal{P}')$ is simply a function $\mathcal{F} : |\Pi\mathsf{Mod}(\mathcal{P})| \to |\Pi\mathsf{Mod}(\mathcal{P}')|$. This will be helpful in defining the functor $\Pi\mathsf{Mod}$ (see definition 18 below).

The *reduct* of a $\mathcal{P}'$-interpretation structure $\tau' : \mathcal{P}' \to \{\mathsf{true}, \mathsf{false}\}$, w.r.t. a propositional signature morphism $\phi : \mathcal{P} \to \mathcal{P}'$ is a $\mathcal{P}$-interpretation structure $\tau : \mathcal{P} \to \{\mathsf{true}, \mathsf{false}\}$. The reduct τ gives to a propositional symbol p from the (lesser) signature $\mathcal{P}$ the interpretation given by τ' to the corresponding symbol $\phi(p)$. Clearly $\tau = \tau' \circ \phi$.

Recalling remark 17 above, the functor $\Pi\mathsf{Mod}$ is defined as follows.

Definition 18 The functor $\Pi\mathsf{Mod}$ sends a propositional signature $\mathcal{P}$ to the class $|\Pi\mathsf{Mod}(\mathcal{P})|$ of $\mathcal{P}$-interpretation structures and sends each propositional signature morphism $\phi : \mathcal{P} \to \mathcal{P}'$ to the function $\Pi\mathsf{Mod}(\phi) : |\Pi\mathsf{Mod}(\mathcal{P}')| \to |\Pi\mathsf{Mod}(\mathcal{P})|$ defined by $\Pi\mathsf{Mod}(\phi)(\tau') = \tau' \circ \phi$ that assigns to each $\mathcal{P}'$-interpretation structure τ' its reduct $\tau' \circ \phi$. ■

We only have to define satisfaction to fully characterize the institution Π.

Definition 19 Given a propositional signature $\mathcal{P}$ the satisfaction relation $\models^{\Pi}_{\mathcal{P}}$ is defined as follows: $\tau \models^{\Pi}_{\mathcal{P}} \pi$, where τ is a propositional $\mathcal{P}$-model and π a $\mathcal{P}$-formula iff $\tau^{\models}(\pi) = \mathsf{true}$ where $\tau^{\models}$ is the unique extension of τ to $\mathcal{P}$-formulas defined by (see again [41]):

1. $\tau^{\models}(\perp) = \mathsf{false}$; $\tau^{\models}(p) = \tau(p)$ for $p \in \mathcal{P}$,
2. $\tau^{\models}((\pi_1 \Rightarrow \pi_2)) = \mathsf{false}$ iff $\tau^{\models}(\pi_2) = \mathsf{true}$ and $\tau^{\models}(\pi_2) = \mathsf{false}$. ■

Theorem 20 The category $\Pi\mathsf{Sign}$, the functors $\Pi\mathsf{Sen}$ and $\Pi\mathsf{Mod}$ and the propositional satisfaction relations $\models^{\Pi}_{\mathcal{P}}$, for each propositional signature $\mathcal{P}$, constitute an institution.

Proof We need only to establish the satisfaction condition. Given the signature morphism $\phi : \mathcal{P} \to \mathcal{P}'$ and a $\mathcal{P}'$-model τ' we have to check that, for every formula $f \in \Pi\mathsf{Sen}(\mathcal{P})$,

$$\tau' \models^{\Pi}_{\mathcal{P}'} \Pi\mathsf{Sen}(\phi)(f) \quad \text{iff} \quad \Pi\mathsf{Mod}(\phi)(\tau') \models^{\Pi}_{\mathcal{P}} f$$

which is equivalent to $\tau' \models^{\Pi}_{\mathcal{P}'} \hat{\phi}(f)$ iff $\tau' \circ \phi \models^{\Pi}_{\mathcal{P}} f$ from definitions 18 of $\Pi\mathsf{Mod}$ and 15 of $\Pi\mathsf{Sen}$. This in turn is equivalent (definition 19 of satisfaction) to $\tau'^{\models}(\hat{\phi}(s)) = \mathsf{true}$ iff $(\tau' \circ \phi)^{\models}(s) = \mathsf{true}$. It is easy to check, from definitions 19 and 14 that $(\tau' \circ \phi)^{\models} = \tau'^{\models} \circ \hat{\phi}$, which ends the proof. ✓

We refrain from illustrating presentations and their models and the corresponding Galois connection in the propositional institution. A final note on the notion of entailment is worthwhile: it is equivalent to the notion of consequence (closure for derivation) since this institution has as a complete set of inference rules.

1.2 Specifications

In this section we present the concepts and properties relevant to specifications built from axioms and defaults of an arbitrary institution.

Default reasoning can be added to a given institution by adding defaults to its presentations (on the syntactic side) and organizing the models of presentations with preference relations induced by those defaults (on the semantic side). A Galois connection can again be obtained for the resulting framework, expressing the interplay between constructions on specifications (i.e. presentations with defaults) and their semantics.

1.2.1 Syntax and Semantics

A specification is a set of formulas corresponding to the facts or *axioms* of the specification plus a set of formulas, the *defaults* that express properties which are likely to be true but not certainly true. Both the axioms and the defaults are written in the same language.

The formalism to be presented generalizes Poole-like defaults (see [70]) in that axioms and defaults can be written in any institution. Poole defaults, also called *supernormal defaults* ([6, 7]) or even *true defaults* can be seen as a special case of defaults in the sense of default logic from Reiter ([73]), namely the defaults of the form $\mathsf{true} : d/d$.

The formalism is simple although quite powerful ([14]) and will be extended in chapter 2 by introducing degrees of likeliness among the defaults. The resulting formalism, taken from [12], is inspired in the priorities from [6, 7, 14] and Ordered Theory Presentations from [74, 75].

A fundamental property of Poole-like defaults is that they can be assigned a preferential semantics, expressing that some models of the facts (axioms) are better than other since they satisfy more defaults. This preferential semantics has its original motivation in the semantics of Circumscription ([67]), where first order interpretation structures are related by inclusion of the carrier sets corresponding to a predicate of *abnormality*. Preferential semantics has been proposed has a basis to all non-monotonic formalisms ([88]). The semantics we present here is very close to the one proposed by Stefan Brass in [6, 7]. We will see (in chapter

3) that this semantics is particularly convenient for the study of composition of specifications.

A small example is useful to introduce the relevant concepts[9].

Example 21 Consider the specification BATMAN concerned with the flying abilities of mammals, in particular bats and humans. Bats are known to fly by default whereas humans are known not to fly by default. Non-exceptional humans are able to dream. Nothing is known about the same ability for bats. A particularly interesting individual is $\texttt{bm}$ (Batman) which is known to be both a bat and a human. This is modeled by the axiom $\texttt{Hum}(\texttt{bm}) \wedge \texttt{Bat}(\texttt{bm})$. The formulas $\texttt{Bat}(\texttt{bm}) \Rightarrow \texttt{Fl}(\texttt{bm})$, $\texttt{Hum}(\texttt{bm}) \Rightarrow \neg\texttt{Fl}(\texttt{bm})$ and $\texttt{Hum}(\texttt{bm}) \Rightarrow \texttt{Dr}(\texttt{bm})$ are used to express the default information relevant to $\texttt{bm}$. Clearly our choice of symbols is the first order logic signature $\mathsf{sg}(\text{BATMAN}) = \{\{\texttt{bm}\}_0, \{\texttt{Bat}, \texttt{Hum}, \texttt{Fl}, \texttt{Dr}\}_1\}$. The previous axiom and defaults form the specification

$$\text{BATMAN} = (\mathsf{sg}(\text{BATMAN}), \mathsf{ax}(\text{BATMAN}), \mathsf{df}(\text{BATMAN})),$$

where

$$\mathsf{ax}(\text{BATMAN}) = \{\texttt{Hum}(\texttt{bm}) \wedge \texttt{Bat}(\texttt{bm})\}$$

and

$$\mathsf{df}(\text{BATMAN}) = \{\texttt{Bat}(\texttt{bm}) \Rightarrow \texttt{Fl}(\texttt{bm}), \texttt{Hum}(\texttt{bm}) \Rightarrow \neg\texttt{Fl}(\texttt{bm}), \texttt{Hum}(\texttt{bm}) \Rightarrow \texttt{Dr}(\texttt{bm})\}.$$

△

We now proceed to define specifications.

Remark 22 Recall that concepts are defined in the scope of

$$\mathcal{I} = (\mathsf{Sign}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \{\models^{\mathcal{I}}_{\Sigma}, \Sigma \in |\mathsf{Sign}^{\mathcal{I}}|\}),$$

a fixed but arbitrary institution.

Definition 23 A *specification* (from $\mathcal{I}$) 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 Σ, the set of *axioms* from S and
- $D \subseteq \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is a set of formulas from Σ, the set of *defaults* from S.

[9]We use first order logic to formalize the examples related to BATMAN.

The projections $\mathsf{sg}(S) = \Sigma$, $\mathsf{ax}(S) = A$ and $\mathsf{df}(S) = D$ assign to a specification S its signature, its set of axioms and its set of defaults.

A Σ-*specification* (A, D) is a specification (Σ, 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 ([67, 88, 9]). This relation is a pre-order.

Definition 24 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 Σ,
- $\sqsubseteq \subseteq \mathcal{M} \times \mathcal{M}$ is a reflexive and transitive relation among those interpretation structures.

The projections $\mathsf{sg}(\mathcal{R}) = \Sigma$, $|\mathcal{R}| = \mathcal{M}$ and $\mathsf{rl}(\mathcal{R}) = \sqsubseteq$ assign to a pre-order $\mathcal{R}$ its signature, its class of interpretation structures and the relation among them, respectively.

A Σ-*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 25 The pre-order induced by a specification S, denoted by $S^\star$ is the pre-order with

- the same signature as S, $\mathsf{sg}(S^\star) = \mathsf{sg}(S)$,
- the models of the axioms from S as class of interpretation structures, $|S^\star| = \mathsf{ax}(S)^\bullet$ and
- the relation $\sqsubseteq = \mathsf{rl}(S^\star) \subseteq |S^\star| \times |S^\star|$ among those models defined by

$$m \sqsubseteq n \text{ iff for all } d \in \mathsf{df}(S), \text{ if } m \vDash^{\mathcal{I}}_{\mathsf{sg}(P)} d \text{ then } n \vDash^{\mathcal{I}}_{\mathsf{sg}(P)} d.$$

■

The preference relation associated with the specification BATMAN is displayed in the following figure 1.2. Only the interpretation structures where the axiom `Hum(bm)` $\wedge$ `Bat(bm)` holds participate in the relation. Interpretation structures satisfying precisely the same defaults are made equivalent by the preference relation corresponding to the specification BATMAN. The nodes ($\boxed{\text{label}}$) denote the equivalence classes of the interpretation structures satisfying the sets of formulas labeling them and the arrows ($\Rightarrow$) denote relations of preference among those interpretation structures (reflexive pairs are not represented).

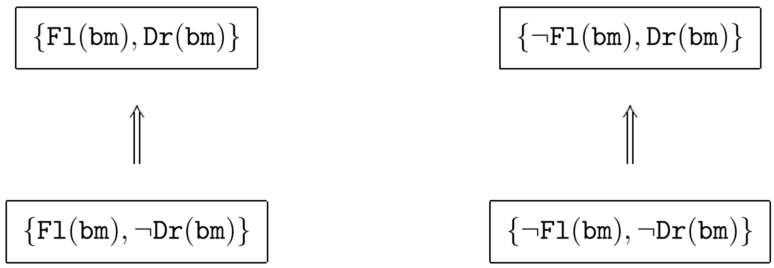

Figure 1.2: The preference relation associated with BATMAN

1.2.2 Theories

We have already seen how to obtain a pre-order from a specification. We are now concerned with the opposite direction, that of assigning a specification to a pre-order $\mathcal{R}$. This specification will also be called the *theory* of the pre-order $\mathcal{R}$. Since our framework generalizes the classical case it is expected that the theory of a pre-order $\mathcal{R}$ will have as axioms the formulas satisfied in all interpretation structures participating in $\mathcal{R}$.

For the new structure we introduce the concept of *default implicit* in a pre-order. To motivate this concept note that any default d from a specification S satisfies the property that for each $(m_1, m_2) \in S^\star$ if $m_1 \vDash d$ then $m_2 \vDash d$ (this is obvious from the definition 25 of preference relation). An implicit default from $S^\star$ is any formula that satisfies this property (clearly including the defaults from S as defaults implicit in its preference relation). The set of implicit defaults from $S^\star$ is the biggest set of formulas that can be added to the set of defaults from S without destroying any of the relations of preference displayed in $S^\star$.

Remark 26 There are formal similarities with the classical case: All axioms are satisfied in the class of their models and the theory of this class is the set of such formulas. Also this theory is the biggest set of formulas that can be added to the original axioms without changing its semantics (any formula having the same models is already in the theory).

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 27 The set of *defaults implicit* in a Σ-pre-order $\mathcal{R}$, denoted by $\mathcal{R}^\circ$, is the set of Σ-formulas

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

where $\sqsubseteq$ is $\mathsf{rl}(\mathcal{R})$. ∎

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

Definition 28

- The function * assigns to a Σ-specification S its induced *preference relation*, the Σ-pre-order $S^\star$,
- The function * assigns to a Σ-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 relation between these two operators takes again the form of a Galois connection. It generalizes the Galois connection for the classical case (presented in theorem 6).

Before presenting it we need to define inclusion, union and intersection of specifications and of pre-orders. Inclusion of specifications corresponds to inclusion of sets of axioms and sets of defaults. Union corresponds to union of the sets of axioms and the sets of defaults. Similarly for intersection. Inclusion of pre-orders corresponds to inclusion of the classes of interpretation structures and inclusion of the relations (i.e. inclusion of relations pairs). Intersection of pre-orders is intersection of the classes of interpretation structures and of the relations. Note that the union of transitive relations is not necessarily transitive. For this reason union of pre-orders is union of the classes of interpretation structures and the transitive closure of the union of the relations. In this way the union of pre-orders is itself a pre-order.

Definition 29 The relation of *inclusion* ($\Subset$) and the operations *union* ($\Cup$) and *intersection* ($\Cap$) over Σ-specifications are defined as follows. Let S and S' be Σ-specifications. Then

1. $S \Subset S'$ iff $\mathsf{ax}(S) \subseteq \mathsf{ax}(S')$ and $\mathsf{df}(S) \subseteq \mathsf{df}(S')$,
2. $S \Cup S' = (\mathsf{ax}(S) \cup \mathsf{ax}(S'), \mathsf{df}(S) \cup \mathsf{df}(S'))$,
3. $S \Cap S' = (\mathsf{ax}(S) \cap \mathsf{ax}(S'), \mathsf{df}(S) \cap \mathsf{df}(S'))$.

The relation of *inclusion* ($\Subset$) and the operations *union* ($\Cup$) and *intersection* ($\Cap$) over Σ-pre-orders are defined similarly as follows. Recall that we take the transitive closure of the union of the argument relations in order to assure that the resulting relation is reflexive and transitive.

Let $\mathcal{R}$ and $\mathcal{R}'$ be Σ-pre-orders. Then

1. $\mathcal{R} \Subset \mathcal{R}'$ iff $|\mathcal{R}| \subseteq |\mathcal{R}'|$ and $\mathsf{rl}(\mathcal{R}) \subseteq \mathsf{rl}(\mathcal{R}')$,

2. $\mathcal{R} \Cup \mathcal{R}' = (|\mathcal{R}| \cup |\mathcal{R}'|, \overline{\mathsf{rl}(\mathcal{R}) \cup \mathsf{rl}(\mathcal{R}')})$, where $\overline{\mathsf{rl}(\mathcal{R}) \cup \mathsf{rl}(\mathcal{R}')}$ is the transitive closure of the relation $\mathsf{rl}(\mathcal{R}) \cup \mathsf{rl}(\mathcal{R}')$,
3. $\mathcal{R} \Cap \mathcal{R}' = (|\mathcal{R}| \cap |\mathcal{R}'|, \mathsf{rl}(\mathcal{R}) \cap \mathsf{rl}(\mathcal{R}'))$. ∎

The Galois connection between specifications and pre-orders states that more formulas imply less models (as known classically) and, what is new, that *more defaults* imply *less relations of preference* among those models.

Moreover operations among specifications (unions or intersections of axioms *and* defaults) are mirrored by corresponding operations among pre-orders (intersections and unions).

Theorem 30 Let S, S' and $S_n, n \in N$, be Σ-specifications and $\mathcal{R}, \mathcal{R}'$ and $\mathcal{R}_n, n \in N$, be Σ-pre-orders (N is some set of indices). 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) $(\Cap_n S_n^{\star\star})^\star = (\Cup_n S_n^{\star\star})^\star$,

(f) $(\Cap_n \mathcal{R}_n^{\star\star})^\star = (\Cup_n \mathcal{R}_n^\star)^{\star\star}$.

Proof

1. Since $\mathsf{ax}(S) \subseteq \mathsf{ax}(S')$ it follows from the Galois connection for the classical case that $|S^\star| = \mathsf{ax}(S)^\bullet \supseteq \mathsf{ax}(S')^\bullet = |S'^\star|$.

   We need now to prove that $\mathsf{rl}(S^\star) \supseteq \mathsf{rl}(S'^\star)$. Take $(m, n) \in \mathsf{rl}(S'^\star)$. We show that $(m, n) \in \mathsf{rl}(S'^\star)$. By definition of preference relation (definition 25) this means that for each default $d \in \mathsf{df}(S')$ if $m \vDash d$ then also $n \vDash d$. Since the set of defaults $\mathsf{df}(S) \subseteq \mathsf{df}(S')$ then clearly for each default $d \in \mathsf{df}(S)$ if $m \vDash d$ then $n \vDash d$. But this means $(m, n) \in \mathsf{rl}(S^\star)$.

2. Since $|\mathcal{R}| \subseteq |\mathcal{R}'|$ it follows again from the Galois connection for the classical case that $\mathsf{ax}(\mathcal{R}^\star) = |\mathcal{R}|^\bullet \supseteq |\mathcal{R}'|^\bullet = \mathsf{ax}(\mathcal{R}'^\star)$.

   We need now to prove that $\mathcal{R}^\circ \supseteq \mathcal{R}'^\circ$. Take a default $d \in \mathcal{R}'^\circ$. We prove that $d \in \mathcal{R}^\circ$. By definition of implicit default (definition 27) $d \in \mathcal{R}^\circ$ iff whenever $m \vDash d$ for $m \in |\mathcal{R}|$ then for any m' such that $(m, m') \in \mathsf{rl}(\mathcal{R})$ also $m' \vDash d$. Assume $m \vDash d$ for $m \in |\mathcal{R}|$ and consider m' with $(m, m') \in \mathsf{rl}(\mathcal{R})$. We show that $m' \vDash d$. Since $|\mathcal{R}| \subseteq |\mathcal{R}'|$ we know that $m \in |\mathcal{R}'|$ and since $\mathsf{rl}(\mathcal{R}) \subseteq \mathsf{rl}(\mathcal{R}')$ we also know that $(m, m') \in \mathsf{rl}(\mathcal{R}')$. But since d is an implicit default from $\mathcal{R}'$ and $m \vDash d$ then $m' \vDash d$.

3. $\mathsf{ax}(S) \subseteq \mathsf{ax}(S^{\star\star})$ follows from the fact that $\mathsf{ax}(S^{\star\star}) = |S^\star|^\bullet = \mathsf{ax}(S)^{\bullet\bullet}$ (see definition 28) and $\mathsf{ax}(S) \subseteq \mathsf{ax}(S)^{\bullet\bullet}$ from the Galois connection for the classical case.

   What we have to check is that the defaults from S are defaults implicit in the preference relation $S^\star$. That is for each $d \in \mathsf{df}(S)$ and any interpretation structure $m \in |S^\star| = \mathsf{ax}(S)^\bullet$ if $m \vDash d$ then every m' with $(m, m') \in \mathsf{rl}(S^\star)$ also satisfies d. But it is obvious from definition 25 of preference relation induced by S that m' satisfies all defaults from S satisfied by m.

4. Note that $|\mathcal{R}^{\star\star}| = \mathsf{ax}(\mathcal{R}^\star)^\bullet = |\mathcal{R}|^{\bullet\bullet}$ (from definitions 28). Clearly $|\mathcal{R}| \subseteq |\mathcal{R}^{\star\star}| = |\mathcal{R}|^{\bullet\bullet}$ again from the Galois connection for the classical case.

   We have to check that $\mathsf{rl}(\mathcal{R}) \subseteq \mathsf{rl}(\mathcal{R}^{\star\star})$, i.e. that if $(m, n) \in \mathsf{rl}(\mathcal{R})$ then $(m, n) \in \mathsf{rl}(\mathcal{R}^{\star\star})$. Take $(m, n) \in \mathsf{rl}(\mathcal{R})$. Both $m, n \in |\mathcal{R}^{\star\star}|$ since $m, n \in |\mathcal{R}|$ and we know from above that $|\mathcal{R}| \subseteq |\mathcal{R}^{\star\star}|$.

   From definition 25 of preference relation $(m, n) \in \mathsf{rl}(\mathcal{R}^{\star\star})$ if all defaults from $\mathcal{R}^\star$ that are satisfied in m are also satisfied in n. The defaults from $\mathcal{R}^\star$ are the defaults implicit in $\mathcal{R}$ (see definition 28).

   We have now to prove that given an implicit default d from $\mathcal{R}$ if $m \vDash d$ then $n \vDash d$. Assume $m \vDash d$. But d is an implicit default from $\mathcal{R}$ precisely because whenever it would hold in an interpretation structure m participating in $\mathcal{R}$ it would also hold in all interpretation structures above m, according to $\mathcal{R}$ (see definition 27 of implicit default). This is the case with n since $(m, n) \in \mathsf{rl}(\mathcal{R})$.

Now the derived ones. Note that if, for each n, $\mathcal{R}_n \Subset \mathcal{R}$ then $\Cup_n \mathcal{R}_n \Subset \mathcal{R}$ where $\Cup_n \mathcal{R}_n$ is the transitive closure of the union of the relations in each $\mathcal{R}_n$.

(a) $[S^\star = S^{\star\star\star}]$ Since $S \Subset S^{\star\star}$ (3 above) we have from 1 that $S^\star \Supset S^{\star\star\star}$. But letting $\mathcal{R} = S^\star$ we have from 4 that $S^\star \Subset S^{\star\star\star}$.

(b) $[\mathcal{R}^\star = \mathcal{R}^{\star\star\star}]$ *mutatis mutandis* as (a),

(c) $[(\Cup_n S_n)^\star = \Cap_n S_n^\star]$ Since $S_n \Subset \Cup_n S_n$ for each $n \in N$ one has from 1 that $S_n^\star \Supset (\Cup_n S_n)^\star$ and also the intersection $\Cap_n S_n^\star \Supset (\Cup_n S_n)^\star$.

   For the other direction one has for each $n \in N$ that $S_n^\star \Supset \Cap_n S_n^\star$ and from 2, $S_n^{\star\star} \Subset (\Cap_n S_n^\star)^\star$. From 3 $S_n \Subset S_n^{\star\star} \Subset (\Cap_n S_n^\star)^\star$ and therefore $\Cup_n S_n \Subset (\Cap_n S_n^\star)^\star$.

Again from 1 we obtain $(\Cup_n S_n)^\star \Supset (\Cap_n S_n^\star)^{\star\star}$ and since from 4 $(\Cap_n S_n^\star)^{\star\star} \Supset \Cap_n S_n^\star$ we have finally $(\Cup_n S_n)^\star \Supset (\Cap_n {S_n}^\star)^{\star\star} \Supset \Cap_n S_n^\star$.

(d) $[(\Cup_n {\mathcal{R}_n}^\star) = \Cap_n \mathcal{R}_n^\star]$ *mutatis mutandis* as (c),

(e) $[(\Cap_n S_n^{\star\star})^\star = (\Cup_n S_n^{\star\star})^\star]$ Clearly $\Cap_n S_n^{\star\star} \Subset S^{\star\star}_n$ for $n \in N$ which implies, from 1, $(\Cap_n S_n^{\star\star})^\star \Supset S_n^{\star\star\star}$ and $S_n^{\star\star\star} = S_n^\star$ (the latest equality is from (a) above). Therefore $(\Cap_n S_n^{\star\star})^\star \Supset \Cup_n S_n^\star$ and from 2 and then 1 $(\Cap_n S_n^{\star\star})^{\star\star\star} \Supset (\Cup_n S_n^{\star\star})^\star$. Therefore, again from property (a) above we have $(\Cap_n S_n^{\star\star})^\star \Supset (\Cup_n S_n^{\star\star})^\star$.

For the other direction note that for $n \in N$, $\Cup_n S_n^\star \Supset S_n^\star$ and this implies, with 2 above $(\Cup_n S_n^{\star\star}) \Subset S_n^{\star\star}$. In this way $(\Cup_n S_n^\star)^\star \Subset \Cap_n S_n^{\star\star}$. From 1 we have $(\Cup_n S_n^{\star\star})^\star \Supset (\Cap_n S_n^{\star\star})^\star$.

(f) $[(\Cap_n \mathcal{R}_n^{\star\star})^\star = (\Cup_n \mathcal{R}_n^\star)^{\star\star}]$ *mutatis mutandis* as (e). ✓

The pre-orders that are the semantics of some specification are the ones closed for the operators above. Closed specifications are in a one to one relation to closed pre-orders.

Definition 31

- The *closure* of a Σ-specification S is the Σ-specification $S^{\star\star}$ (i.e. $(S^\star)^\star$). The closure $S^{\star\star}$ is also called the *theory* of S.

  A Σ-specification S is *closed* iff $S = S^{\star\star}$.

- The *closure* of a Σ-pre-order $\mathcal{R}$ is the Σ-pre-order $\mathcal{R}^{\star\star}$ (i.e. $(\mathcal{R}^\star)^\star$).

  A Σ-pre-order $\mathcal{R}$ is *closed* iff $\mathcal{R} = \mathcal{R}^{\star\star}$. A closed Σ-pre-order $\mathcal{R}$ is also called a Σ-*preference relation.* ▪

The relation between closed specifications and closed pre-orders is one to one. Moreover, semantic comparison of specifications (by inclusion of preference relation) is expressed at the syntactic level by inclusion of *closed* specifications.

Lemma 32 Let S, S' be closed Σ-specifications and $\mathcal{R}, \mathcal{R}'$ be closed Σ-pre-orders. Then

- $S \Subset S'$ iff $S^\star \Supset S'^\star$,
- $\mathcal{R} \Subset \mathcal{R}'$ iff $\mathcal{R}^\star \Supset \mathcal{R}'^\star$.

Proof Trivial from the Galois connection 30 above. ✓

We emphasize that closed specifications are canonical among the specifications having the same semantics: on one hand to equivalent specifications (having the same semantics) corresponds the same closure. On the other hand this closure is the biggest specification among equivalent ones.

Lemma 33

- Let S, S' be Σ-specifications. S and S' have the same semantics iff they have the same theory: $S^\star = S'^\star$ iff $S^{\star\star} = S'^{\star\star}$;
- Let S be a Σ-specification. $S^{\star\star}$ is the biggest specification from among those having the same semantics as S: given any Σ-specification S', if $S^\star = S'^\star$ then $S' \Subset S^{\star\star}$.

Proof Trivial from the Galois connection 30 above. ✓

Finally the Galois connection provides a way to compare preference relations on the basis of the axioms and defaults inducing them.

Lemma 34 In order to establish whether $S^{\star\star} \Subset S'^{\star\star}$ one has to check whether each axiom from S is semantically entailed by the axioms from S' and whether each default from S is an implicit default in $S'^\star$.

Proof Straightforward from the Galois properties (theorem 30). ✓

Clearly to establish equality of the preference relations one has simply to apply the lemma above in both directions.

1.2.3 Pre-orders and Preference Relations

Here we provide an alternative necessary and sufficient condition for pre-orders to be induced by a specification. There are two main advantages in this characterization. The first one is that it shows how to obtain a specification from a pre-order satisfying the new condition. The second is that this condition can be easily applied to alternative formalisms (see section 2.2.2 in chapter 2) to show that they have the same expressive power as specifications.

The new condition arises easily by noting the following. The preference relation associated with a specification S organizes the models of the axioms by satisfaction of defaults. Therefore, the class of the models better than a given one m (call it $\beta(m)$) is the class of models of the axioms satisfying additionally the defaults satisfied by m. Note also that the set of interpretation structures better than m', for m' as preferred as m coincides with $\beta(m)$. This motivates the following definitions.

Definition 35 Let $\mathcal{R}$ be a Σ-pre-order and let $m \sqsubseteq_{\mathcal{R}} m'$ denote $(m, m') \in \mathsf{rl}(\mathcal{R})$. Interpretation structures $m, m' \in |\mathcal{R}|$ such that $m \sqsubseteq_{\mathcal{R}} m'$ and $m' \sqsubseteq_{\mathcal{R}} m$ are said to be *equivalent* according to $\mathcal{R}$, written $m \equiv_{\mathcal{R}} m'$. The equivalence class of m, denoted by $[m]_{\mathcal{R}}$ is $[m]_{\mathcal{R}} = \{m' : m \equiv_{\mathcal{R}} m'\}$ the class of interpretation structures equivalent to m. ▪

We now define the class of the models better than a given one m.

Definition 36 Let $\mathcal{R}$ be a Σ-pre-order. The class $\beta_{\mathcal{R}}(m)$ of the interpretation structures from $|\mathcal{R}|$ better than m is $\beta_{\mathcal{R}}(m) = \{m' : m' \in |\mathcal{R}| \text{ and } m \sqsubseteq_{\mathcal{R}} m'\}$.

Since equivalent interpretation structures $m \equiv_{\mathcal{R}} m'$ have the same class $\beta_{\mathcal{R}}(m) = \beta_{\mathcal{R}}(m')$ the set $\beta_{\mathcal{R}}([m]_{\mathcal{R}})$ is defined by $\beta_{\mathcal{R}}([m]_{\mathcal{R}}) = \beta_{\mathcal{R}}(m)$. ▪

The set of defaults from a specification holding in one of its models is defined as follows.

Definition 37 Let S be a Σ-specification, m a model of $\mathsf{ax}(S)$ and let $\mathsf{df}(S)(m) = \{d \in \mathsf{df}(S) : m \vDash_{\Sigma} d\}$ denote the set of defaults from S that hold in m.

Since in equivalent interpretation structures $m \equiv_{S^\star} m'$ the same defaults from S hold, the set of defaults holding in one equivalence class $[m]_{S^\star}$, denoted by $\mathsf{df}(S)([m]_{S^\star})$ is defined by $\mathsf{df}(S)([m]_{S^\star}) = \mathsf{df}(S)(m)$. ▪

The class $\beta_{S^\star}(m)$ is is identified as follows.

Lemma 38 Let S be a Σ-specification and m a model of $\mathsf{ax}(S)$. Then the class $\beta_{S^\star}(m)$ of models of $\mathsf{ax}(S)$ above m is the class of models of $\mathsf{ax}(S) \cup \mathsf{df}(S)(m)$: $\beta_{S^\star}(m) = (\mathsf{ax}(S) \cup \mathsf{df}(S)(m))^\bullet$.

Proof

- Assume $m' \in \beta_{S^\star}(m)$. This means that $m' \in |S^\star| = \mathsf{ax}(S)^\bullet$ and m' satisfies at least the defaults from S satisfied by m (recall definition 25 of $S^\star$). In this way $m' \vDash \mathsf{ax}(S)$ and $m' \vDash \mathsf{df}(S)(m)$.
- Assume now that $m' \vDash \mathsf{ax}(S)$ and $m' \vDash \mathsf{df}(S)(m)$. Then $m' \in |S^\star| = \mathsf{ax}(S)^\bullet$ and since m' satisfies at least the defaults from S satisfied by m we have $(m, m') \in \mathsf{rl}(S^\star)$ and $m' \in \beta_{S^\star}(m)$. ✓

We are now concerned with the reverse direction: given a pre-order $\mathcal{R}$ we want to find (when possible) a specification S having $\mathcal{R}$ as preference relation. If there is a specification S such that $S^\star = \mathcal{R}$ then, from the lemma 38 above each class

$\beta_{\mathcal{R}}(m)$ is the class of models of the axioms from S and some defaults from S (those defaults holding in m). Therefore we will look at the sets of formulas P_m having $\beta_{\mathcal{R}}(m)$ as class of models. Each such P_m (if it exists) should consist of the looked for axioms and some of the looked for defaults. The union of all P_m should include all defaults. We see in the following that it is not enough to find such sets and care has to be taken when choosing them.

Example 39 Consider the preference relation associated with the specification $(\emptyset, \{p \wedge q\})$.

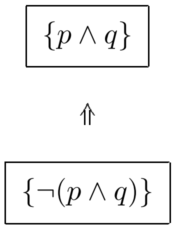

Figure 1.3: The preference relation associated with $(\emptyset, \{p \wedge q\})$.

We only have to identify with formulas two sets $\beta_{\mathcal{R}}(m)$ corresponding to the equivalence classes of the interpretation structures either not satisfying $p \wedge q$ or satisfying $p \wedge q$. The class $\beta_{\mathcal{R}}(m)$ of the models above some m that does not satisfy $p \wedge q$ is the class of all interpretation structures. Choose for it the empty set of formulas (that has $\beta_{\mathcal{R}}(m)$ as models). The class $\beta_{\mathcal{R}}(m)$ with $m \vDash p \wedge q$ is the class of models of $\{p \wedge q\}$, or equivalently the class of models of $\{p, q\}$. Therefore we have two candidates for sets of defaults: $\emptyset \cup \{p \wedge q\}$ and $\emptyset \cup \{p, q\}$. The later is not an appropriate choice since it induces a different preference relation:

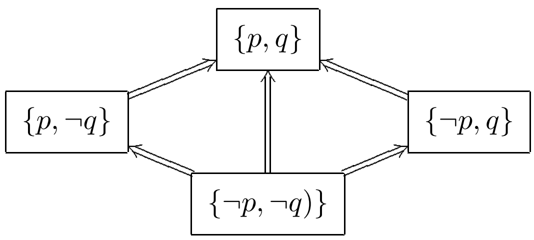

Figure 1.4: The preference relation associated with $(\emptyset, \{p, q\})$.

△

Given a pre-order $\mathcal{R}$ the axioms and defaults inducing it are found (if they exist) in the following way: the set of axioms is a set having $|\mathcal{R}|$ as models. The set of defaults is the union of the sets P_m having $\beta_{\mathcal{R}}(m)$ as models. As discussed by the previous example an additional condition must be imposed on the sets P_m. It is presented in the following definition.

Definition 40 Let $\mathcal{R}$ be a Σ-pre-order. A *coverage* of $\mathcal{R}$ is:

- a Σ-presentation $P_{|\mathcal{R}|}$ such that $P^{\bullet}_{|\mathcal{R}|} = |\mathcal{R}|$ (the models of $P_{|\mathcal{R}|}$ are the interpretation structures participating in $\mathcal{R}$) and
- a Σ-presentation P_m for each $m \in |\mathcal{R}|$ such that $P^{\bullet}_m = \beta_{\mathcal{R}}(m)$ (the models of P_m are the interpretation structures better than m);

A coverage is said to be *granular* iff each Σ-presentation P_m satisfies the following *condition of granularity*: given a formula $d \in P_m$ and $m' \vDash d$ then $m'' \vDash d$ for any m'' bigger than m' (i.e. $(m', m'') \in \mathsf{rl}(\mathcal{R})$). ■

Pre-orders having a granular coverage correspond to preference relations.

Theorem 41 A Σ-pre-order $\mathcal{R}$ is a preference relation iff it has a granular coverage.

Proof

- Assume that the pre-order $\mathcal{R}$ is a preference relation, i.e. it is the preference relation induced by some specification S: $\mathcal{R} = S^{\star}$. Then, from lemma 38 above, it has a coverage given by:
  - $P = \mathsf{ax}(S)$ and
  - $P_m = \mathsf{ax}(S) \cup \mathsf{df}(S)(m)$.

  We have to check that this coverage is granular. Take a $d \in P_m = \mathsf{ax}(S) \cup \mathsf{df}(S)(m)$. Let $m' \sqsubseteq_{\mathcal{R}} m''$ for some m' such that $m' \vDash d$. We have to see that $m'' \vDash d$. If $d \in \mathsf{df}(S)(m)$ then d is a default from S and $m' \sqsubseteq_{\mathcal{R}} m''$ iff m'' satisfies at least the defaults satisfied by m'. The case $d \in \mathsf{ax}(S)$ is even simpler.

- Now for the other direction: Assume that $(P_{|\mathcal{R}|}, \{P_m, m \in |\mathcal{R}|\})$ is a granular coverage of $\mathcal{R}$. We now see that $S = (P_{|\mathcal{R}|}, \cup_{m\in|\mathcal{R}|} P_m)$ is a specification inducing $\mathcal{R}$: $\mathcal{R} = S^{\star}$. Clearly $|\mathcal{R}| = \mathsf{ax}(S)^{\bullet}$ so we have to check that $m' \sqsubseteq_{\mathcal{R}} m''$ iff $m' \sqsubseteq_{S^{\star}} m''$.

  Assume $m' \sqsubseteq_{\mathcal{R}} m''$ with $m' \vDash d$ for a $d \in \mathsf{df}(S) = \cup_{m\in|\mathcal{R}|} P_m$. Therefore $d \in P_m$ for some m and from the coverage condition we conclude that $m'' \vDash d$. In this way m'' satisfies the defaults satisfied by m' which means $m' \sqsubseteq_{S^{\star}} m''$.

  Assume now that $m' \sqsubseteq_{S^{\star}} m''$. We have to show that $m' \sqsubseteq_{\mathcal{R}} m''$ or equivalently that $m'' \in \beta_{\mathcal{R}}(m') = P^{\bullet}_{m'}$. Since $m' \in \beta_{\mathcal{R}}(m') = P^{\bullet}_{m'}$ we have that $m' \vDash P_{m'}$. Since $m' \sqsubseteq_{S^{\star}} m''$ then m' satisfies the defaults satisfied by m', namely all formulas in $P_{m'} \subseteq \mathsf{df}(S)$. This means that $m'' \in P^{\bullet}_{m'} = \beta_{\mathcal{R}}(m')$. ✓

Remark 42 Recall that two equivalent models m, m' have equal classes $\beta_{\mathcal{R}}(m) = \beta_{\mathcal{R}}(m')$. In this way one has only to provide a P_m for each equivalence class.

1.2.4 Special Cases

The alternative characterization of preference relations has the advantage of suggesting which formulas can be taken as defaults to induce a given pre-order. These are the formulas in the sets P_m having each class $\beta_{\mathcal{R}}(m)$ as class of models. In the case each set P_m being finite and the underlying logic having conjunctions the condition of granularity in the definition 40 above need not be verified: one has to take the conjunction of the formulas in P_m as the new defaults. Before stating this property more formally we firstly define institutions having conjunctions.

Definition 43 An institution having *conjunctions* is a pair $(\mathcal{I}, \mathsf{cj})$ where $\mathcal{I}$ is an institution and cj is a family of functions, indexed on the signatures of $\mathcal{I}$ giving for any two formulas its conjunction: $\mathsf{cj} = \{\mathsf{cj}_\Sigma : \mathsf{Sen}^{\mathcal{I}}(\Sigma) \times \mathsf{Sen}^{\mathcal{I}}(\Sigma) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma) \mid \Sigma \in |\mathsf{Sign}^{\mathcal{I}}|\}$ such that for each $\Sigma \in |\mathsf{Sign}^{\mathcal{I}}|$ and each interpretation structure $m \in \mathsf{Mod}^{\mathcal{I}}(\Sigma)$ the following holds for any two formulas $f_1, f_2 \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$:

$$m \models^{\mathcal{I}}_{\Sigma} \mathsf{cj}_\Sigma(f_1, f_2) \text{ iff } m \models^{\mathcal{I}}_{\Sigma} f_1 \text{ and } m \models^{\mathcal{I}}_{\Sigma} f_2.$$

Given a finite sequence $f_1, ..., f_n$ of Σ-formulas the expression $\mathsf{cj}_\Sigma(f_1, ..., f_n)$ denotes the obvious extension of binary conjunction to the finite sequence $f_1, ..., f_n$. ∎

We now see that the task of finding the defaults inducing a particular pre-order is simplified if the underlying institution has conjunctions.

Lemma 44 Let $\mathcal{R}$ be a Σ-pre-order from an institution $(\mathcal{I}, \mathsf{cj})$ having conjunctions. If

1. its base set is the class of models of some Σ-presentation P: $|\mathcal{R}| = P^\bullet$ and
2. for each $m \in |\mathcal{R}|$ the class $\beta_{\mathcal{R}}(m)$ is the class of models of some *finite* Σ-presentation P_m: $\beta_{\mathcal{R}}(m) = P_m^\bullet$,

then $\mathcal{R}$ has a granular coverage.

Proof For each model m take

1. $\hat{P}_m = \{\mathsf{cj}_\Sigma(f_1, ..., f_n)$, where $f_1, ..., f_n$ is some enumeration of the formulas in $P_m\}$ if P_m is not empty and
2. $\hat{P}_m = P_m = \emptyset$, otherwise.

$(P, \{\hat{P}_m, m \in |\mathcal{R}|\})$ is obviously a coverage. We now see that is is granular. Each $\hat{P}_m$ has at most one formula. The only non-trivial case is when $\hat{P}_m$ has precisely one formula, $\hat{P}_m = \{\hat{p}_m\}$.

Consider an interpretation structure m' with $m' \vDash \hat{p}_m$ and m'' such that $m' \sqsubseteq_{\mathcal{R}} m''$. We have to see that $m'' \vDash \hat{p}_m$. Now since $m' \vDash \hat{p}_m$ we have that $m' \in \{\hat{p}_m\}^\bullet = \hat{P}_m^\bullet = P_m^\bullet = \beta_{\mathcal{R}}(m)$, which means that $m \sqsubseteq_{\mathcal{R}} m'$. From transitivity of $\sqsubseteq_{\mathcal{R}}$ we have $m \sqsubseteq_{\mathcal{R}} m''$ which is equivalent to $m'' \in \beta_{\mathcal{R}}(m) = \{\hat{p}_m\}^\bullet$ and proves that $m'' \vDash \hat{p}_m$. ✓

We now illustrate the previous result.

Example 45 Consider again the specification BATMAN and its preference relation displayed in figure 1.2. A coverage of this pre-order is the following:

- Clearly the interpretation structures participating in the preference relation are the models of $P = \mathsf{ax}(\text{BATMAN}) = \{\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}\}$,
- We have to provide four sets, one for each equivalence class:
  - for the equivalence class labeled by $\{\texttt{Fl(bm)}, \neg\texttt{Dr(bm)}\}$, the class of better models is the class of models of $P_1 = P \cup \{\texttt{Fl(bm)}\}$,
  - for the equivalence class labeled by $\{\texttt{Fl(bm)}, \texttt{Dr(bm)}\}$, the class of better models is the class of models of $P_2 = P \cup \{\texttt{Fl(bm)}, \texttt{Dr(bm)}\}$,
  - for the equivalence class labeled by $\{\neg\texttt{Fl(bm)}, \neg\texttt{Dr(bm)}\}$, the class of better models is the class of models of $P_3 = P \cup \{\neg\texttt{Fl(bm)}\}$,
  - for the equivalence class labeled by $\{\neg\texttt{Fl(bm)}, \texttt{Dr(bm)}\}$ the class of better models is the class of models of $P_4 = P \cup \{\neg\texttt{Fl(bm)}, \texttt{Dr(bm)}\}$.

The axioms and defaults inducing the preference of BATMAN can be rediscovered using the method suggested by theorem 41 and lemma 44 above. The axioms are those of BATMAN, namely $\{\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}\}$. There are four defaults corresponding to take the conjunctions of the sets above: $\{\texttt{Fl(bm)}, \texttt{Hum(bm)} \wedge \texttt{Bat(bm)}\}$, $\{\texttt{Fl(bm)}, \texttt{Dr(bm)}, \texttt{Hum(bm)} \wedge \texttt{Bat(bm)}\}$, $\{\neg\texttt{Fl(bm)}, \texttt{Hum(bm)} \wedge \texttt{Bat(bm)}\}$, $\{\neg\texttt{Fl(bm)}, \texttt{Dr(bm)}, \texttt{Hum(bm)} \wedge \texttt{Bat(bm)}\}$.

It is easy to check that the axiom $\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}$ is redundant. The following simplified specification also induces the preference relation associated with BATMAN:

$$\text{BATMAN2} = (\mathsf{sg}(\text{BATMAN}), \mathsf{ax}(\text{BATMAN}), \mathsf{df}(\text{BATMAN2})),$$

where

$$\mathsf{df}(\text{BATMAN2}) = \{\texttt{Fl(bm)}, \texttt{Fl(bm)} \wedge \texttt{Dr(bm)}, \neg\texttt{Fl(bm)}, \neg\texttt{Fl(bm)} \wedge \texttt{Dr(bm)}\}.$$

We will see in section 1.2.5 below that this specification is indeed equivalent to the specification BATMAN. △

Our next example is again a variation of BATMAN. We consider the case of stating that the equivalence class in which `bm` flies and dreams should be considered better than any other. A specification is found having precisely the intended pre-order as induced preference relation.

Example 46 The following figure displays a variation of the preference relation associated with BATMAN where all interpretation structures are less preferred then those where `bm` flies and dreams. (Relations resulting from transitivity are omitted.)

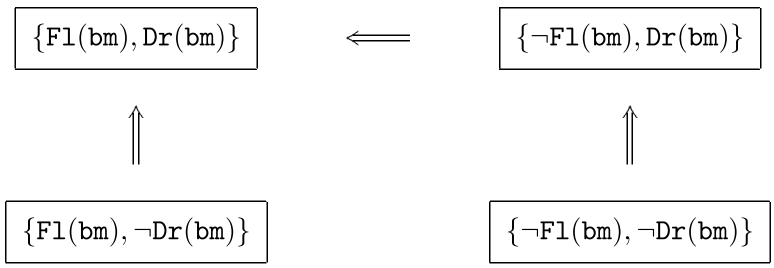

Figure 1.5: A variation of BATMAN.

Note that due to the new relations added to the preference of BATMAN the new sets $\beta(m)$ may include more interpretation structures.

The formulas characterizing each $\beta(m)$ have to have the new interpretation structures as models. Therefore they correspond to disjunctions. The arguments of these disjunctions are formulas characterizing the "old" $\beta(m)$ (i.e. corresponding to the preference of BATMAN).

A coverage of this pre-order is the following:

- The interpretation structures participating in the preference relation are the models of $P = \mathsf{ax}(\text{BATMAN}) = \{\mathtt{Hum(bm)} \wedge \mathtt{Bat(bm)}\}$,
- We have to provide four sets, one for each equivalence class:
  - for the equivalence class labeled by $\{\mathtt{Fl(bm)}, \neg\mathtt{Dr(bm)}\}$, the class of better models is the class of models of $P'_1 = P \cup \{\mathtt{Fl(bm)}\}$, as before
  - for the equivalence class labeled by $\{\mathtt{Fl(bm)}, \mathtt{Dr(bm)}\}$, the class of better models is the class of models of $P'_2 = P \cup \{\mathtt{Fl(bm)}, \mathtt{Dr(bm)}\}$, as before,
  - for the equivalence class labeled by $\{\neg\mathtt{Fl(bm)}, \neg\mathtt{Dr(bm)}\}$, the class of better models is the class of models of $P'_3 = P \cup \{\neg\mathtt{Fl(bm)} \vee (\mathtt{Fl(bm)} \wedge \mathtt{Dr(bm)})\}$, the "disjunction" of the previous P_3 and P_1,

– for the equivalence class labeled by $\{\neg\texttt{Fl}(\texttt{bm}), \texttt{Dr}(\texttt{bm})\}$ the class of better models is the class of models of $P_4' = P \cup \{\texttt{Dr}(\texttt{bm})\}$, the "disjunction" of the previous P_4 and P_1.

Therefore from theorem 41 and lemma 44 above the following specification induces the preference relation associated with this variation of BATMAN:

$$\text{BATMAN3} = (\mathsf{sg}(\text{BATMAN}), \mathsf{ax}(\text{BATMAN}), \mathsf{df}(\text{BATMAN3})),$$

where

$$\mathsf{df}(\text{BATMAN3}) = \{\texttt{Fl}(\texttt{bm}), \texttt{Fl}(\texttt{bm}) \wedge \texttt{Dr}(\texttt{bm}), \neg\texttt{Fl}(\texttt{bm}) \vee (\texttt{Fl}(\texttt{bm}) \wedge \texttt{Dr}(\texttt{bm})), \texttt{Dr}(\texttt{bm})\}.$$

△

The example suggests the following: any pre-order bigger (having more relations) than one preference relation $S^\star$ can also be expressed by a specification. The new defaults will be disjunctions of the defaults in S.

We show that this is indeed the case if S is finite and the underlying institution has conjunctions and disjunctions.

Definition 47 An institution having *disjunctions* is a pair $(\mathcal{I}, \mathsf{dj})$ where $\mathcal{I}$ is an institution and dj is a family of functions, indexed on the signatures of $\mathcal{I}$ giving for any two formulas its disjunction: $\mathsf{dj} = \{\mathsf{dj}_\Sigma : \mathsf{Sen}^\mathcal{I}(\Sigma) \times \mathsf{Sen}^\mathcal{I}(\Sigma) \to \mathsf{Sen}^\mathcal{I}(\Sigma) \mid \Sigma \in |\mathsf{Sign}^\mathcal{I}|\}$ such that for each $\Sigma \in |\mathsf{Sign}^\mathcal{I}|$ and each interpretation structure $m \in \mathsf{Mod}^\mathcal{I}(\Sigma)$ the following holds for any two formulas $f_1, f_2 \in \mathsf{Sen}^\mathcal{I}(\Sigma)$:

$$m \vDash^\mathcal{I}_\Sigma \mathsf{dj}_\Sigma(f_1, f_2) \text{ iff } m \vDash^\mathcal{I}_\Sigma f_1 \text{ or } m \vDash^\mathcal{I}_\Sigma f_2.$$

Given a finite sequence $f_1, ..., f_n$ of Σ-formulas the expression $\mathsf{dj}_\Sigma(f_1, ..., f_n)$ denotes the obvious extension of binary disjunction to the finite sequence $f_1, ..., f_n$.

An institution having *disjunctions and conjunctions* is a triple $(\mathcal{I}, \mathsf{cj}, \mathsf{dj})$ where $(\mathcal{I}, \mathsf{cj})$ is an institution having conjunctions and $(\mathcal{I}, \mathsf{dj})$ is an institution having disjunctions. ■

We now show that in such a case a pre-order bigger than a preference relation induced by a finite specification is itself a preference relation.

Lemma 48 Let $\mathcal{R}$ be a Σ-pre-order and S a *finite* Σ-specification from an institution $(\mathcal{I}, \mathsf{cj}, \mathsf{dj})$ having disjunctions and conjunctions. If

1. $|\mathcal{R}| = |S^\star|$ and

2. $\mathsf{rl}(\mathcal{R}) \supseteq \mathsf{rl}(S^\star)$,

then $\mathcal{R}$ is a preference relation, i.e. $\mathcal{R} = S'^\star$. Moreover the specification S' has the same axioms as S and the defaults from S' are implicit defaults in $S^\star$.

Proof What we need is to find a granular coverage for $\mathcal{R}$ using the specification S. Clearly $|\mathcal{R}|$ is covered by $\mathsf{ax}(S)$ since $|\mathcal{R}| = |S^\star|$. So the main part of the proof is concerned in finding a coverage for $\beta_{\mathcal{R}}(m)$ for $m \in |\mathcal{R}|$. This is done in the following way: consider the equivalence classes $E(m) = \{[m']_{S^\star} : m' \in \beta_{\mathcal{R}}(m)\}$ induced by $S^\star$ in $\beta_{\mathcal{R}}(m)$. Since S has a finite number of defaults also $E(m)$ is finite. Now $\beta_{\mathcal{R}}(m) = \cup_{[m'] \in E(m)} \beta_{S^\star}([m'])$. To see this note that on one hand each $m'' \in \beta_{\mathcal{R}}(m)$ is in $\beta_{S^\star}([m''])$ and each $n \in \beta_{S^\star}([m'])$, for $[m'] \in E(m)$ is also in $\beta_{\mathcal{R}}(m)$. This last assertion follows from $\mathsf{rl}(\mathcal{R}) \supseteq \mathsf{rl}(S^\star)$.

We now provide a coverage for $\cup_{[m'] \in E(m)} \beta_{S^\star}([m'])$.

Recall that each $\beta_{S^\star}([m'])$ is the class of models of the axioms from S plus the defaults holding in m', $P_{m'} = \mathsf{ax}(S) \cup \mathsf{df}(S)([m'])$.

Some of these $P_{m'}$ may be empty. In this case $\beta_{\mathcal{R}}(m) = \cup_{[m'] \in E(m)} \beta_{S^\star}([m'])$ is the class of all interpretation structures of the signature Σ and therefore the class of models of the empty set of Σ-formulas.

So now we assume that all $P_{m'}$ are not empty. Take the conjunction of this finite set of formulas: $\hat{p}_{m'} = \mathsf{cj}_\Sigma(f_1, ..., fn)$, where $f_1, ...f_n$ is an enumeration of $P_{m'} = \mathsf{ax}(S) \cup \mathsf{df}(S)([m'])$.

Since $\beta_{\mathcal{R}}(m)$ is a union of $\beta_{S^\star}([m'])$ it is the class of models of the disjunctions of the $\hat{p}_{m'}$. That is $\beta_{\mathcal{R}}(m)$ is the class of models of the following formula: $\mathsf{dj}_\Sigma(\hat{p}_{m_1}, ..., \hat{p}_{m_n})$ for some enumeration $[m_1], ..., [m_n]$ of the equivalence classes $[m'] \in E(m)$.

We have to check that this coverage is granular. This is however trivial since each $\beta_{\mathcal{R}}(m)$ is either the class of models of the empty set or the class of models of a single formula. Such a coverage is always granular (recall the proof of lemma 44 above).

Finally we must establish that the pre-order $\mathcal{R}$ is induced by a specification S' with the same axioms as S such that the defaults from S' are implicit defaults in $S^\star$. Now $S' = (P, \cup_{m'} P_{m'})$ by construction (see proof of the theorem 41) we have $\mathsf{ax}(S') = \mathsf{ax}(S)$. To see that the defaults from S' implicit in $S^\star$ note that $S'^\star = \mathcal{R} \supseteq S^\star$. From property 2 of the Galois connection in theorem 2 we obtain $S'^{\star\star} \subseteq S^{\star\star}$. But this implies that the defaults implicit in $S'^\star$ (including the defaults from S') are included in the defaults implicit in $S^\star$. ✓

Corollary 49 In the propositional institution Π each Σ-pre-order with finite Σ is induced by a propositional Σ-specification.

Proof Firstly given a Σ-pre-order with finite Σ its base set is the set of models of the (big) disjunction of the conjunctions of the propositional symbols holding in each of its interpretation structures and the negations of those holding not (there is only a finite number of them). Secondly consider the specification having that formula as only axiom and the following defaults: all the propositional symbols and their negations. This specification is finite.

It is straightforward to check that the preference relation induced by this specification is the relation where two of those interpretation structures are related iff they are the same. Any pre-order involving those interpretation structures, since it is reflexive, has to have more relations than this. Now use the lemma above. ✓

1.2.5 Theories Revisited

Here we are concerned with the relation that the theory of a specification S has with S itself. In particular we relate the axioms from $S^{\star\star}$ with the axioms of S and the defaults from $S^{\star\star}$ with the defaults and axioms from S.

The axioms from $S^{\star\star}$ are the formulas (classically) entailed by the set of axioms from S. The defaults from $S^{\star\star}$ are the defaults implicit in the preference relation associated with S.

Lemma 50 The theory $S^{\star\star}$ of a Σ-specification S is such that:

1. it has as axioms the formulas satisfied in each model of the axioms from S; $\mathsf{ax}(S^{\star\star}) = |S^\star|^\bullet = \mathsf{ax}(S)^\bullet$ and
2. it has as defaults the defaults implicit in the preference relation induced by S, $\mathsf{df}(S^{\star\star}) = (S^\star)^\circ$.

Proof Obvious from the definition 28 of the operators * and definition 25 of preference relation $S^\star$. ✓

We now proceed to characterize the defaults from $S^{\star\star}$. These are the defaults implicit in the preference relation $S^\star$, i.e. the formulas d that hold in all interpretation structures better than a given m whenever they hold in m.

It is straightforward that there are two types of defaults trivially implicit in $S^\star$: the formulas that hold in all models of the axioms of S and those that hold in none. In particular the tautologies and contradictions are implicit defaults of every preference relation.

For the non-trivial ones consider for the purpose of motivation that the underlying institution is first order logic and that d_1 and d_2 are defaults from a specification S written in a signature of first order logic. Given any interpretation structure m in the preference relation $S^\star$ induced by S, if m satisfies d_1 or d_2 or both then any interpretation structure better than m satisfies at least the defaults holding in m and therefore will also satisfy d_1 or d_2 or both.

From these considerations it follows easily that any conjunction or disjunction of defaults from S is a default implicit in $S^\star$. In general any formula that in the context of the axioms from S has the same models as some "disjunction" of "conjunctions" of the defaults from S is a default implicit in the preference relation induced by S. And these are all implicit defaults in $S^\star$.

If the underlying institution has disjunctions and conjunctions and the specification S is *finite* a formula d is an implicit default of S iff there is a formula

$$d' = (d_1^1 \wedge \ldots \wedge d_{n_1}^1) \vee \ldots \vee (d_1^k \wedge \ldots \wedge d_{n_k}^k) \text{ with all } d_j^i \in \mathsf{df}(S)$$

such that $\widehat{\mathsf{ax}(S)} \wedge d$ and $\widehat{\mathsf{ax}(S)} \wedge d'$ have exactly the same models ($\widehat{\mathsf{ax}(S)}$ denotes the conjunction of formulas in $\mathsf{ax}(S)$).

The property of implicit default for arbitrary specifications (from an arbitrary institution) is characterized in the next theorem, using the semantical counterparts of disjunctions and conjunctions: unions and intersections of classes of models.

Theorem 51 A Σ-formula d is an implicit default in the Σ-preference relation $S^\star$ induced by a Σ-specification S iff there is a set $\Delta \subseteq 2^{\mathsf{df}(S)}$ of subsets of the set of defaults from S such that $\mathsf{ax}(S)^\bullet \cap \{d\}^\bullet = \mathsf{ax}(S)^\bullet \cap (\bigcup_{D\in\Delta} D^\bullet)$. Moreover if the set of defaults of S is finite then Δ is finite and each set of defaults $D \in \Delta$ is also finite.

Proof

1. Assume that there exists $\Delta \subseteq 2^{\mathsf{df}(S)}$ and $\mathsf{ax}(S)^\bullet \cap \{d\}^\bullet = \mathsf{ax}(S)^\bullet \cap (\cup_{D\in\Delta} D^\bullet)$. We now see that d is an implicit default in $S^\star$. We have to prove that given an interpretation structure $m \in |S^\star|$ that satisfies d then any interpretation structure above it according to $S^\star$ also satisfies d (see definition 27 of implicit default). Clearly $m \in \mathsf{ax}(S)^\bullet \cap \{d\}^\bullet$ since $m \vDash d$ and $m \in |S^\star| = \mathsf{ax}(S)^\bullet$. Therefore, by hypothesis $m \in \cup_{D\in\Delta} D^\bullet$ which means that for some $D \in \Delta$, m satisfies each default in D: $m \vDash D$. Now recall that $D \subseteq \mathsf{df}(S)$. It is obvious, by definition 25 of preference relation induced by a specification S, that any $m' \in |S^\star|$ above m, i.e. such that $(m, m') \in \mathsf{rl}(S^\star)$, satisfies at least the defaults from S satisfied by m. In this way $m' \vDash D$ and therefore $m' \in \cup_{D\in\Delta} D^\bullet$. Since $m' \in |S^\star| = \mathsf{ax}(S)^\bullet$ we have that $m' \in \mathsf{ax}(S)^\bullet \cap (\cup_{D\in\Delta} D^\bullet) = \mathsf{ax}(S)^\bullet \cap \{d\}^\bullet$. This implies $m' \vDash d$ and proves that d is an implicit default from $S^\star$.

2. Now assume that d is an implicit default from $S^\star$. We have to characterize the class $\mathsf{ax}(S)^\bullet \cap \{d\}^\bullet$ of models of both $\mathsf{ax}(S)$ and d. Since $|S^\star| = \mathsf{ax}(S)^\bullet$ this is the class $\mathcal{S}_d = \{m : m \in |S^\star| \text{ and } m \vDash d\}$ of the interpretation structures participating in $S^\star$ that satisfy d. Let m be an interpretation structure in $\mathcal{S}_d$.

Then $m \in |S^\star| = \mathsf{ax}(S)^\bullet$ and any interpretation structure $m' \in \beta_{S^\star}(m)$ is also a model of both $\mathsf{ax}(S)$ and d. That $m' \vDash \mathsf{ax}(S)$ is obvious from the definition 36 of $\beta_{S^\star}(m)$ (since $m' \in |S^\star| = \mathsf{ax}(S)^\bullet$). That $m' \vDash d$ follows from d being a default implicit in $S^\star$: since d holds in m it holds in all m' better than m.

Since each $m \in \beta_{S^\star}(m)$ it follows that $\mathcal{S}_d = \cup_{m \in \mathcal{S}_d} \beta_{S^\star}(m)$. Instead of indexing this union on the models m of $\mathsf{ax}(S)$ that satisfy d we can index it in the corresponding equivalence classes (induced by $S^\star$) since to equivalent interpretation structures m_1 and m_2 corresponds the same $\beta_{S^\star}(m_1) = \beta_{S^\star}(m_2)$. Let $[\mathcal{S}_d]$ denote the set of equivalence classes induced by $S^\star$ in $\mathcal{S}_d$, i.e. $[\mathcal{S}_d] = \{[m] : m \in \mathcal{S}_d\}$. Then $\mathcal{S}_d = \cup_{[m] \in [\mathcal{S}_d]} \beta_{S^\star}([m])$. We now use lemma 38 that characterizes the classes $\beta_{S^\star}([m])$ to construct Δ. Since $\beta_{S^\star}([m]) = (\mathsf{ax}(S) \cup \mathsf{df}(S)(m))^\bullet$ one has $\mathcal{S}_d = \cup_{[m] \in [\mathcal{S}_d]} (\mathsf{ax}(S) \cup \mathsf{df}(S)(m))^\bullet$. It is now clear that $\Delta = \{\mathsf{df}(S)(m) : [m] \in [\mathcal{S}_d]\}$.

3. Finally we have to show that when $\mathsf{df}(S)$ is finite the Δ and each $D \in \Delta$ are finite. It is enough to check the construction of Δ in the second part of the proof above. Note that if $\mathsf{df}(S)$ is finite then $S^\star$ has a finite number of equivalence classes (never more then $2^{\#(\mathsf{df}(S))}$) and therefore $[\mathcal{S}_d]$ and Δ are finite. Moreover each $\mathsf{df}(S)(m) \subseteq \mathsf{df}(S)$ is finite. ✓

Note that when Δ is empty one obtains $\mathsf{ax}(S)^\bullet \cap \{d\}^\bullet = \emptyset$. This means that those formulas that do not hold in any model of the axioms are implicit defaults. Also if Δ has only one element, the class of models of the empty set of defaults, then $\mathsf{ax}(S)^\bullet \cap \{d\}^\bullet = \mathsf{ax}(S)^\bullet$. This means that the formulas holding in all models of the axioms are also implicit defaults.

Moreover if we take $S = S'^{\star\star}$ we conclude in particular that any conjunction or disjunction of defaults from $S = S'^{\star\star}$ is a default from $S^{\star\star} = (S'^{\star\star})^{\star\star} = S'^{\star\star}$ (this last equality is a consequence of Galois properties in theorem 30). In other words conjunctions or disjunctions (or conjunctions of disjunctions, etc ...) of implicit defaults are also implicit defaults.

We now illustrate the previous result.

Example 52 Consider again the specification BATMAN. (Recall the preference relation presented in figure 1.2).

The corresponding theory has as axioms the (classical) consequences of $\mathtt{Hum(bm)} \wedge \mathtt{Bat(bm)}$. For the implicit defaults we note firstly that any (classical) consequence of the axioms is also an implicit default. Therefore $\mathtt{Hum(bm)}$ and $\mathtt{Bat(bm)}$ are implicit defaults. But also the negations of the consequences of the axioms are (vacuously) implicit defaults. In particular $\neg\mathtt{Bat(bm)}$ and $\neg\mathtt{Hum(bm)}$.

More interesting are the implicit defaults $\mathtt{Fl(bm)}$, $\neg\mathtt{Fl(bm)}$ and $\mathtt{Dr(bm)}$. In fact $\mathtt{Hum(bm)} \wedge \mathtt{Bat(bm)}) \wedge \mathtt{Fl(bm)}$ is semantically equivalent (has the same models) to

$(\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}) \wedge (\texttt{Bat(bm)} \Rightarrow \texttt{Fl(bm)})$, where $\texttt{Bat(bm)} \Rightarrow \texttt{Fl(bm)}$ is a default from BATMAN. In this way $\texttt{Fl(bm)}$ is a default implicit in BATMAN*.

The case of $\neg\texttt{Fl(bm)}$ and $\texttt{Dr(bm)}$ are similar: $(\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}) \wedge \neg\texttt{Fl(bm)}$ is semantically equivalent to $(\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}) \wedge (\texttt{Hum(bm)} \Rightarrow \neg\texttt{Fl(bm)})$ and $(\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}) \wedge \texttt{Dr(bm)}$ is semantically equivalent to $(\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}) \wedge (\texttt{Hum(bm)} \Rightarrow \texttt{Dr(bm)})$.

Recall now the specification BATMAN2, presented in example 45. In order to prove that it induces the same preference relation as BATMAN we can check whether each default from BATMAN2 is an implicit default in BATMAN* and vice versa (since they have the same axioms, recall lemma 34).

Now this means that each default in

$$\texttt{df}(\text{BATMAN2}) = \{\texttt{Fl(bm)}, \texttt{Fl(bm)} \wedge \texttt{Dr(bm)}, \neg\texttt{Fl(bm)}, \neg\texttt{Fl(bm)} \wedge \texttt{Dr(bm)}\}.$$

has to be an implicit default in BATMAN*. All of them have either already been referred as being implicit defaults in BATMAN* or are conjunctions of implicit defaults in BATMAN*.

Finally it is interesting to note the following: The preference relation associated with the variation BATMAN3 of BATMAN presented in example 46 has more relations than the preference relation associated with BATMAN. Therefore it will have less implicit defaults (Galois property 2 in theorem 30). Therefore the defaults found to induce BATMAN3 have to be implicit defaults in the preference of BATMAN (but not the other way round since these preferences are different!).

The defaults from BATMAN3 are:

$$\texttt{df}(\text{BATMAN3}) = \{\texttt{Fl(bm)}, \texttt{Fl(bm)} \wedge \texttt{Dr(bm)}, \neg\texttt{Fl(bm)} \vee (\texttt{Fl(bm)} \wedge \texttt{Dr(bm)}), \texttt{Dr(bm)}\}.$$

Only $\neg\texttt{Fl(bm)} \vee (\texttt{Fl(bm)} \wedge \texttt{Dr(bm)})$ hasn't been referred. But it is the disjunction of implicit defaults of the preference of BATMAN. $\triangle$

1.3 Extensions

In the following we are concerned with comparing the semantics chosen for specifications (that of preference relations) to the standard consequences of a specification (credulous and skeptical), derived from the extensions of a specification.

Moreover we investigate some important properties of such consequences and of extensions in the general case of an arbitrary institution.

1.3.1 Consequences

The preference relation induced by a specification displays the fact that some models of the axioms are better than others because they satisfy more defaults. The best models (organized in maximal equivalence classes) are the models of the *extensions* of the specification.

Extensions are alternatively obtained from a specification S by extending the information given in the axioms with the information given in the defaults in two steps. First by extending maximally the axioms from S with defaults, preserving consistency (if possible). Each such maximal set of axioms and defaults will be called an extension presentation. An extension is the (classical) theory of an extension presentation ([72, 70, 8]).

First of all we have to define a general notion of consistency.

Definition 53 A Σ-presentation P with no models is said to be *inconsistent*. Otherwise it is said to be *consistent*. ▪

This definition of consistency is not akin to some logics like equational logic or λ-calculus. In fact any presentation written in these logics, even the whole Σ-language always has a model (the model with only one element serves for both logics: all equations hold and all functions are the same function). Presentations having the same models as the whole language are considered inconsistent. This is not the case with the definition above. Note however that the more interesting assumptions that some terms or functions should be taken as *different* cannot be written in these logics. Some form of negation is needed and the resulting institution will have a notion of (in)consistency corresponding to the one in definition 53 above. In this way the concepts we explore in this section are only well defined for those institutions where consistency corresponds to absence of models.

The definition of extension presentation follows.

Definition 54 Given a specification S, a $\mathsf{sg}(S)$-presentation E is an *extension presentation* of S iff

- if the presentation $\mathsf{ax}(S)$ is consistent then E is consistent and maximal among the $\mathsf{sg}(S)$-presentations E' such that $\mathsf{ax}(S) \subseteq E' \subseteq (\mathsf{ax}(S) \cup \mathsf{df}(S))$ i.e. if a consistent E' is such that $E \subseteq E' \subseteq (\mathsf{ax}(S) \cup \mathsf{df}(S))$ then $E' = E$,
- if $\mathsf{ax}(S)$ is not consistent then $E = \mathsf{ax}(S) \cup \mathsf{df}(S)$.

An *extension* $\mathcal{E} = E^{\bullet\bullet}$ of S is the theory of an extension presentation E of S. ▪

The following properties are obvious consequences of the definition of extension.

Theorem 55

1. **Consistency.** An extension presentation E of a Σ-specification S is consistent iff $\mathsf{ax}(S)$ is consistent.

2. **Maximality.** If E, E' are extension presentations of a Σ-specification S and $E \subseteq E'$ then $E = E'$;

3. **Orthogonality.** If E, E' are extension presentations of a Σ-specification S and $E \neq E'$ then $E \cup E'$ is inconsistent.

4. The previous properties also hold when E, E' are *extensions* of a Σ-specification S.

Proof Cases 1 and 2 and their versions for extensions are trivial.

For 3 we have two possibilities: if $\mathsf{ax}(S)$ is inconsistent then any two extensions are the same: the whole Σ-language. Assume therefore that $\mathsf{ax}(S)$ is consistent and so are also E and E'. Since E and E' are different their union strictly contains both. It cannot be consistent since if it were E and E' would not be maximal among the Σ-presentations E'' such that $\mathsf{ax}(S) \subseteq E'' \subseteq (\mathsf{ax}(S) \cup \mathsf{df}(S))$.

To see that this property also holds when E and E' are extensions of S note that $\mathsf{ax}(S) \cup (E \cap \mathsf{df}(S))$ and $\mathsf{ax}(S) \cup (E' \cap \mathsf{df}(S))$ are extension presentations whose closure is E and E' respectively. Moreover if E and E' are consistent and different so are $\mathsf{ax}(S) \cup (E \cap \mathsf{df}(S))$ and $\mathsf{ax}(S) \cup (E' \cap \mathsf{df}(S))$. ✓

The next example illustrates the existence of more than one extension.

Example 56 In our example it is easy to see that we cannot add consistently all defaults from BATMAN to the axiom $\mathtt{Hum(bm)} \wedge \mathtt{Bat(bm)}$. In this way two extensions are obtained, one where `bm` flies and another where he doesn't. △

Whenever there corresponds more than one extension to a default theory presentation we have to decide which formulas are to be accepted as its consequences. The standard approaches are to take as consequences the formulas that hold in either *some* ([72]) or *all* ([67, 88]) extensions.

The obvious definition of these consequence relations follows.

Definition 57 A $\mathsf{sg}(S)$-formula f is a

- *credulous consequence* of a specification S, written $S \vdash_{cr} f$, iff f belongs to some extension of S, and a

- *skeptical consequence* of a specification S, written $S \vdash_{sk} f$, iff f belongs to all extensions of S. ■

Example 58 Recall the specification BATMAN from example 21. It is straightforward to check that one cannot add all defaults to the axioms preserving consistency. Two extension presentations are obtained:

$$\{\mathtt{Hum(bm)} \wedge \mathtt{Bat(bm)}, \mathtt{Hum(bm)} \Rightarrow \mathtt{Dr(bm)}, \mathtt{Bat(bm)} \Rightarrow \mathtt{Fl(Batman)}\}$$

and

$$\{\mathtt{Hum(bm)} \wedge \mathtt{Bat(bm)}, \mathtt{Hum(bm)} \Rightarrow \mathtt{Dr(bm)}, \mathtt{Hum(bm)} \Rightarrow \neg\mathtt{Fl(Batman)}\}.$$

Therefore $\mathtt{Dr(bm)}$ is a skeptical and credulous consequence of the specification BATMAN whereas $\mathtt{Fl(bm)}$ is only a credulous consequence of this specification. △

1.3.2 Preference

The equivalence classes of the preference relation $S^\star$ correspond to classes of such models of the axioms in S that satisfy precisely the same defaults from S. The equivalence classes which are maximal are those whose models satisfy most defaults. They are the classes of models of the extensions from S.

The preference relation $S^\star$ among interpretation structures induces a partial order on the equivalence classes of $S^\star$ that will be technically useful in the following. In general, the partial order $[\mathcal{R}]$ on equivalence classes induced by a pre-order $\mathcal{R}$ is defined as follows.

Definition 59 Given a pre-order $\mathcal{R}$ its corresponding partial order $[\mathcal{R}]$ is the pre-order[10] $[\mathcal{R}] = (\mathsf{sg}([\mathcal{R}]), |[\mathcal{R}]|, \mathsf{rl}([\mathcal{R}]))$ with

1. the same signature as $\mathcal{R}$, $\mathsf{sg}([\mathcal{R}]) = \mathsf{sg}(\mathcal{R})$,

2. the equivalence classes from $\mathcal{R}$ as base set,

   $$|[\mathcal{R}]| = \{[m]_\mathcal{R} : m \in |\mathcal{R}|\},$$

   where $[m]_\mathcal{R}$ is the equivalence class of m in $\mathcal{R}$;

   $$[m]_\mathcal{R} = \{m' : \ (m, m') \in \mathsf{rl}(\mathcal{R}) \text{ and } (m', m) \in \mathsf{rl}(\mathcal{R})\}$$

   and

[10] A partial order is an anti-reflexive pre-order.

3. the relation between these elements being induced by $\mathcal{R}$,

$$\mathsf{rl}([\mathcal{R}]) = \{([m]_{\mathcal{R}}, [m']_{\mathcal{R}}) : \ (m, m') \in \mathsf{rl}(\mathcal{R})\}.$$

Since $\mathsf{rl}(\mathcal{R})$ is a pre-order it is straightforward to check that $\mathsf{rl}([\mathcal{R}])$ is indeed a partial order. ■

Maximal equivalence classes are defined in the obvious way:

Definition 60 Given a pre-order $\mathcal{R}$ an equivalence class $[m]_{\mathcal{R}} \in |[\mathcal{R}]|$ is said *maximal* (in $\mathcal{R}$ or in $[\mathcal{R}]$) iff given any other equivalence class $[m']_{\mathcal{R}} \in |[\mathcal{R}]|$ with $([m]_{\mathcal{R}}, [m']_{\mathcal{R}}) \in \mathsf{rl}([\mathcal{R}])$ then $[m]_{\mathcal{R}} = [m']_{\mathcal{R}}$. ■

The extensions of a specification S can be recovered from its associated preference relation. They correspond to maximal equivalence classes. When the specification has no models it is technically convenient to consider that its associated preference relation has a maximal equivalence class, the empty equivalence class.

Definition 61 The set of maximal equivalence classes of a pre-order $\mathcal{R}$, denoted by $\mathsf{max}(\mathcal{R})$ is defined as follows:

1. if $|\mathcal{R}| = \emptyset$ then $\mathsf{max}(\mathcal{R}) = \{\emptyset\}$,
2. otherwise $\mathsf{max}(|\mathcal{R}|) = \{[m]_{\mathcal{R}} : \ [m]_{\mathcal{R}} \in [\mathcal{R}]$ and $[m]_{\mathcal{R}}$ is maximal in $[\mathcal{R}]\}$. ■

Theorem 62 Let S be a specification. Then $\mathcal{E}$ is an extension of S iff $\mathcal{E} = M^{\bullet}$, where $M \in \mathsf{max}(S^{\star})$ is a maximal equivalence class of the preference relation $S^{\star}$ associated with S.

Proof Assume $\mathsf{ax}(S)$ is inconsistent. Then the only extension of S is the whole $\mathsf{sg}(S)$-language. Also $|S^{\star}| = \emptyset$ and $\mathsf{max}(S^{\star}) = \{\emptyset\}$. Clearly the theory of the empty set of $\mathsf{sg}(S)$-interpretation structures is the only extension of S.

Now assume $\mathsf{ax}(S)$ is consistent.

- Assume also that $[m]$ is a maximal equivalence class of $S^{\star}$. We have to prove that $[m]^{\bullet}$ is an extension. We now see that $E = \mathsf{ax}(S) \cup \mathsf{df}(S)(m)$ is an extension presentation of S. Recall from lemma 38 that $E^{\bullet} = \beta_{S^{\star}}(m)$. And since $[m]$ is a maximal equivalence class of $S^{\star}$ the interpretation structures in $\beta_{S^{\star}}(m)$ are the ones equivalent to m: $E^{\bullet} = \beta_{S^{\star}}(m) = [m]$.

  Assume now that there exists a consistent E' such that $\mathsf{ax}(S) \subseteq E \subseteq E' \subseteq (\mathsf{ax}(S) \cup \mathsf{df}(S))$. We now see that $E' = E$. Since E' is consistent it has a model m'. Clearly $E' \subseteq (\mathsf{ax}(S) \cup \mathsf{df}(S)(m'))$. This model m' satisfies E since $E \subseteq E'$.

Therefore $m' \in E^\bullet = \beta_{S^\star}(m) = [m]$. This implies that m' satisfies precisely the same defaults as m. In this way $E' \subseteq (\mathsf{ax}(S) \cup \mathsf{df}(S)(m')) = (\mathsf{ax}(S) \cup \mathsf{df}(S)(m)) = E$ and we conclude $E' = E$. Therefore E is an extension presentation.

We have already seen that $E^\bullet = [m]$. This implies $E^{\bullet\bullet} = [m]^\bullet$. That is $[m]^\bullet$ is an extension since E is an extension presentation.

- Assume now that $\mathcal{E} = E^{\bullet\bullet}$ is an extension of S. We have to prove that $\mathcal{E}^\bullet$ is a maximal equivalence class of $S^\star$.

  Since $\mathsf{ax}(S)$ is consistent $\mathcal{E}$ is consistent and has a model. Let m be an arbitrary model of $\mathcal{E}$. We firstly see that $\mathsf{ax}(S) \cup \mathsf{df}(S)(m)$ is the extension presentation E. Since m is a model of E clearly $E \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)(m)$. But since E is an extension presentation and $\mathsf{ax}(S) \cup \mathsf{df}(S)(m)$ is consistent (it has the model m) from maximality of extension presentations one has $E = \mathsf{ax}(S) \cup \mathsf{df}(S)(m)$. Since this is true for an arbitrary model of $\mathcal{E} = \mathcal{E}^{\bullet\bullet}$ we conclude that all models of E are equivalent since they satisfy precisely the same defaults from S. In this way $\mathcal{E} = E^{\bullet\bullet} = [m]$ where m is a model of E. We need only to confirm that $[m]$ is a maximal equivalence class. Since $E = \mathsf{ax}(S) \cup \mathsf{df}(S)(m)$ we conclude again from lemma 38 that $E^\bullet = \beta_{S^\star}(m) = [m]$, i.e. only the interpretation structures equivalent to m are preferable to m. This trivially implies that $[m]$ is a maximal equivalence class. ✓

The relation of skeptical and credulous consequence to maximal equivalence classes of $S^\star$ is straightforward from theorem 62 above.

Lemma 63 A $\mathsf{sg}(S)$-formula f is a

- *credulous consequence* of a specification S iff there is a maximal equivalence class $M \in \mathsf{max}(S^\star)$ with $M \vDash f$ (f holds in all $m \in M$), and a
- *skeptical consequence* of a specification S iff $M \vDash f$ for each maximal equivalence class $M \in \mathsf{max}(S^\star)$.

Proof Straightforward from theorem 62 above. ✓

We now compare our conclusions in example 58 with the preference relation associated with BATMAN.

Example 64 Recall again the specification BATMAN and its preference relation displayed in figure 1.2. The two maximal equivalence classes are the models of the two extension presentations from BATMAN:

$$\{\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}, \texttt{Hum(bm)} \Rightarrow \texttt{Dr(bm)}, \texttt{Bat(bm)} \Rightarrow \texttt{Fl(Batman)}\}$$

and

$$\{\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}, \texttt{Hum(bm)} \Rightarrow \texttt{Dr(bm)}, \texttt{Hum(bm)} \Rightarrow \neg\texttt{Fl(Batman)}\}.$$

Clearly `Dr(bm)` is both a skeptical and credulous consequence of BATMAN whereas `Fl(bm)` is only a credulous consequence of this specification. △

1.3.3 Compact Institutions

In this section we present important properties of extensions of specifications written in institutions where the compactness property holds. The most important property is that any specification has at least one extension. The credulous consequence relation is semi-monotonic: the addition of further defaults to S preserves the credulous consequences of S. The skeptical consequence relation is cumulative: the skeptical consequences of S are invariant under addition of a skeptical consequence of S (as an axiom or default) to S. Semi-monotonicity for the first order logic setting was established in [73][11]. Cumulativity is considered an important property of (non-monotonic) consequence relations ([40, 64, 13]).

Definition 65 The compactness property is said to hold in an institution $\mathcal{I}$ iff for each inconsistent Σ-presentation P from $\mathcal{I}$ there is a finite and inconsistent Σ-presentation $P' \subseteq P$. ▪

The following property of institutions where the compactness property holds is of fundamental importance: given a specification S the partial order $[S^\star]$ induced by its preference relation enjoys the property that above any equivalence class there is a maximal one.

Lemma 66 Let $\mathcal{I}$ be an institution where the compactness property holds. given a specification S from $\mathcal{I}$ with $\mathsf{ax}(S)$ consistent, let $[m]$ be an arbitrary equivalence class from $S^\star$. Then there is a maximal equivalence class $[m^\uparrow]$ of $[S^\star]$ such that $([m], [m^\uparrow]) \in \mathsf{rl}([S^\star])$.

Proof The proof uses Zorn's lemma with the restriction of the partial order $[S^\star]$ to the set $\mathcal{B}([m]) = \{[m'] : (m, m') \in \mathsf{rl}(S^\star)\}$ of the equivalence classes above $[m]$.

Let $\mathcal{C}$ be a chain in $\mathcal{B}([m])$ and consider $E = \cup_{[m'] \in \mathcal{C}} (\mathsf{ax}(S) \cup \mathsf{df}(S)([m']))$, the union of the axioms from S with the defaults holding in each equivalence class in this chain. We now see using compactness that E is consistent. If E is not consistent then there is a finite subset E' of E also inconsistent. Consider the finite set $E' \cap \mathsf{df}(S)$ of the defaults from S that belong to E'. From construction of E is is clear that the defaults in $E' \cap \mathsf{df}(S)$ belong to $\cup_{[m'] \in \mathcal{D}} \mathsf{df}(S)([m'])$, where $\mathcal{D}$ is a finite subset of $\mathcal{C}$. Since $\mathcal{D}$ is finite and $\mathcal{C}$ a chain there will be a maximum $[n] \in \mathcal{D}$. From the definition of preference relation we have that the set of defaults holding in n includes the defaults holding in less preferred interpretation structures. In this way $\mathsf{df}(S)([n]) \supseteq \cup_{[m'] \in \mathcal{D}} \mathsf{df}(S)([m'])$. This implies $(\mathsf{ax}(S) \cup \mathsf{df}(S)([n])) \supseteq \mathsf{ax}(S) \cup (\cup_{[m'] \in \mathcal{D}} \mathsf{df}(S)([m'])) \supseteq E'$. But $\mathsf{ax}(S) \cup \mathsf{df}(S)([n])$ is consistent (has a model n) contradicting the assumption of E' being inconsistent.

Since $E = \cup_{[m'] \in \mathcal{C}} (\mathsf{ax}(S) \cup \mathsf{df}(S)([m']))$ is consistent it has a model μ. Clearly μ satisfies the defaults satisfied in each equivalence class $[m'] \in \mathcal{C}$. Therefore $[\mu]$ is an upper bound of $\mathcal{C}$. From Zorn's lemma $\mathcal{B}([m])$ has a maximal element $[m^\uparrow]$.

[11] For normal default theories, that include only defaults of the form $a : b/b$.

It is obvious that $(m, m^{\uparrow}) \in \mathsf{rl}(S^{\star})$ from definition of $\mathcal{B}([m])$. We only have to prove that $[m^{\uparrow}]$ is a maximal equivalence class of $S^{\star}$. This is also straightforward since the equivalence class $[m^{\uparrow}]$ is maximal in $\mathcal{B}([m])$. ✓

The lemma above implies the following properties:

Theorem 67 Let $\mathcal{I}$ be an institution where the compactness property holds. Then

1. **Existence.** Any specification has, at least, one extension.
2. **Coverage.** Given a specification S from $\mathcal{I}$ and $D' \subseteq \mathsf{df}(S)$ if $\mathsf{ax}(S) \cup D'$ is consistent, then there exists an extension of S containing D'.
3. **Semi-monotonicity.** Given specifications S, S' from $\mathcal{I}$ with $\mathsf{ax}(S) = \mathsf{ax}(S')$ and $\mathsf{df}(S) \subseteq \mathsf{df}(S')$ then for each extension presentation E of S there is an extension presentation E' of S' such that $E \subseteq E'$. This implies that if $S \vdash_{cr} f$ then $S' \vdash_{cr} f$.
4. **Cumulativity.** Let S be a specification from $\mathcal{I}$ and f be a skeptical consequence of S, $S \vdash_{sk} f$. Let S' be the specification with either $\mathsf{ax}(S') = \mathsf{ax}(S)$ and $\mathsf{df}(S') = \mathsf{df}(S) \cup \{f\}$ or $\mathsf{ax}(S') = \mathsf{ax}(S) \cup \{f\}$ and $\mathsf{df}(S') = \mathsf{df}(S)$. (I.e. f is added either to the axioms or to the defaults.) Then E' is an extension presentation of S' iff $E' = E \cup \{f\}$ where E is an extension presentation of S. This implies that given any formula f', $S \vdash_{sk} f'$ iff $S' \vdash_{sk} f'$.

Proof

1. If $\mathsf{ax}(S)$ is inconsistent S has the whole $\mathsf{sg}(S)$-language as extension. If $\mathsf{ax}(S)$ is consistent take $m \in \mathsf{ax}(S)^{\bullet}$. From lemma 66 there is a maximal equivalence class $[m^{\uparrow}]$ of $S^{\star}$ with $(m, m^{\uparrow}) \in \mathsf{rl}(S^{\star})$, where $[m]$ is the equivalence class of m. From theorem 62 we have that $[m^{\uparrow}]^{\bullet}$ is an extension of S.
2. Like the property before, if $\mathsf{ax}(S) \cup D'$ is consistent it has a model m and there will be a maximal equivalence class $[m^{\uparrow}]$ of $S^{\star}$ with $(m, m^{\uparrow}) \in \mathsf{rl}(S^{\star})$. Since $(m, m^{\uparrow}) \in \mathsf{rl}(S^{\star})$ then any interpretation structure in $m' \in [m^{\uparrow}]$ is also preferred to m. Therefore any such m' satisfies $D' \subseteq \mathsf{df}(S)$ since it satisfies at least the defaults satisfied by m. In this way $D' \subseteq [m^{\uparrow}]^{\bullet}$. Again from theorem 62 $[m^{\uparrow}]^{\bullet}$ is an extension of S.
3. If $\mathsf{ax}(S) = \mathsf{ax}(S')$ is inconsistent this property follows trivially. For the case of a consistent $\mathsf{ax}(S) = \mathsf{ax}(S')$ the proof of semi-monotonicity uses the property 2 above: Notice that the extension presentation E consists of the axioms from S' plus a set of defaults that is consistent with these axioms. Therefore E is contained in an extension $\mathcal{E}'$ of S'. From this it follows trivially that E is contained in the extension presentation corresponding to $\mathcal{E}'$.

From definition of credulous consequence follows trivially that $S \vdash_{cr} f$ implies $S' \vdash_{cr} f$.

4. The case of inconsistent $\mathsf{ax}(S) \subseteq \mathsf{ax}(S')$ is again trivial. For consistent $\mathsf{ax}(S)$ and S' with $\mathsf{ax}(S') = \mathsf{ax}(S)$ and $\mathsf{df}(S') = \mathsf{df}(S) \cup \{f\}$, we see firstly that, given an extension presentation E of S then $E \cup \{f\}$ is an extension presentation of S'. Since f is a skeptical consequence of S the set $E \cup \{f\}$ is consistent. In fact E is consistent (property of consistency in theorem 55) and f holds in any model of E (in fact f holds in any model of some extension presentation of S). We now note that it is not possible to add consistently any other default d' from S' to $E \cup \{f\}$. This d' is also from S and consistency of $E \cup \{f\} \cup \{d'\}$ implies consistency of $E \cup \{d'\}$ contradicting maximality of E (an extension presentation of S).

   Assume now that E' is an extension of S' and consider $E' \setminus \{f\}$. This is a consistent set of axioms from S and defaults from S. Therefore (property 2 above) there is an extension presentation E of S containing $E' \setminus \{f\}$. From above $E \cup \{f\}$ is an extension presentation of S'. Since $E' \subseteq E \cup \{f\}$ it must be (property of maximality in theorem 55) $E' = E \cup \{f\}$.

   The case of S' with $\mathsf{ax}(S') = \mathsf{ax}(S) \cup \{f\}$ and $\mathsf{df}(S') = \mathsf{df}(S)$ is similar and omitted.

   Finally it is straightforward to conclude that the maximal models of the preference of S coincide with those of S'. In fact those are the models of some extension presentation of S. Since they all satisfy f they are also the models of some extension presentation of S'. Equality of the classes of maximal models trivially implies equality of skeptical consequences. ✓

1.3.4 Selection Functions

The approaches using different levels of priority on defaults imply a selection of extensions. In fact the extensions resulting from such formalisms are extensions of the original specification without the priorities (certainly not the other way round).

In this subsection our concern is to display the fact that (under certain conditions) any selection of the extensions of a specification can equivalently be expressed by another specification.

This means that given a specification and a selection of its extensions we can display another specification having precisely the selected extensions as extensions.

The idea is to construct from the preference relation associated with the original specification another pre-order. The new pre-order will have less maximal equivalence classes (the classes corresponding to the selected extensions) but will use the same interpretation structures and more relations than the original one. Using lemma 48 we can conclude that the constructed pre-order is induced by some specification.

We firstly define selection (on the semantic level). The only restriction we impose is that such a selection should not be empty.

Definition 68 Given a specification S a *selection* of S is a non-empty set (of maximal equivalence classes) $\mathcal{M} \subseteq \mathsf{max}(S^\star)$. ■

Lemma 69 Let S be a finite specification from an institution having conjunctions and disjunctions and $\mathcal{M}$ a selection of S. Then there is a specification S' such that $\mathsf{max}(S'^\star) = \mathcal{M}$. Moreover S' has the same axioms as S and all defaults from S' are defaults implicit in $S^\star$.

Proof From $S^\star$ construct an order by putting all interpretation structures in the rejected (i.e. not in $\mathcal{M}$) maximal equivalence classes less preferred than any interpretation structure in the selected maximal equivalence classes. Close this for transitivity. This pre-order clearly has the same interpretation structures as $S^\star$, more relations and its maximal equivalence classes are precisely those in $\mathcal{M}$. Now use lemma 48. ✓

1.3.5 Extensions and Composition

We have seen that from the adopted semantics (preference relation) extensions and consequences of a specification can be derived. The question we address here is why do we need all that structure.

We could, in fact, choose as semantics of a specification its skeptical consequences, or credulous or the set of all extensions. The reason the preference relation is needed is that our framework has to be able to explain composition of specifications. Although this theory will only be presented in chapter 3 we can already discuss why the alternatives based on extensions do not provide enough information.

Composition of specifications has to allow for addition of axioms (generalizing the classical case) and defaults to a given specification. A semantics able to explain these operations has to be such that by adding the same axiom or default to specifications having the same meaning (the same semantics) one obtains as result specifications having the same meaning.

This is not the case in general with any semantics based on extensions as the following example shows. Take two specifications $(\{d_1\}, \{d_2\})$ and $(\emptyset, \{d_1, d_2\})$ with d_1 and d_2 mutually consistent (i.e. $\{d_1, d_2\}$ is consistent). Both specifications have one extension given by the extension presentation $\{d_1, d_2\}$. Therefore they have the same meaning according to any notion of semantics based on extensions (including any healthy selection function). However if the underlying logic has negation they yield different results when adding the axiom $\neg d_1$ (the first one becomes inconsistent, the second one not).

A similar situation occurs when the semantics is that of a preference relation that is a strict partial order (see [7]). In such a semantics interpretation structures are also compared on the basis of how well they satisfy the defaults, only that equivalent interpretation structures are considered unrelated.

Consider now the specifications $(\emptyset, \emptyset)$ and $(\emptyset, \{d_1, \neg d_1\})$ where d_1 is neither a tautology nor a contradiction. To these specifications corresponds the same strict partial order: the empty order. However if we add d_1 as a default to both we again obtain specifications having different strict partial order semantics. (The strict partial order of the second specification remains unchanged but the one of the first one not).

That the preference relation semantics presented in section 1.2.1 is appropriate to explain composition of specifications is the subject of chapter 3.

1.4 Final Remarks

We have presented specifications and their semantics and have investigated relationships between them.

The preference semantics is quite known in the literature. We have adapted the one from S. Braß ([7]) to a general institution and taken preferences to be pre-orders. This formalization has the advantage of organizing equivalent interpretation structures in equivalence classes. Our results concerning the expression of pre-orders by specifications (sections 1.2.3 and 1.2.4) heavily depend on this property. (In fact there are similar considerations in [7]. These are not, in this respect, as fruitful as ours precisely because the relation of preference is not a pre-order).

Also the characterization of extensions as the theories of maximal equivalence classes (section 1.3.2) can only be expressed in this way in such a setting (see, however, again [7] for a characterization with strict partial orderings).

The Galois connection between specifications and their preferences together with the characterization of implicit defaults are the main results of this chapter. Clearly they generalize the corresponding properties for the classical case (see [46]) but are, in the setting of specifications with defaults, to our best knowledge, new.

Chapter 2

Prioritized Defaults

The formalism presented in the previous chapter 1 is now extended by allowing different degrees of “likeliness” or “priority” to be assigned to the defaults. More likely defaults are to be assumed prior to less likely defaults. In the case of conflicting defaults, those of higher likeliness should be assumed and those of lesser likeliness disregarded.

The specification archetype that such a formalism best models is known as the *Specificity Principle* ([75]). This is a consequence of identifying a structure of classes in the Universe of Discourse. The properties of subclasses are generally inherited from the properties of corresponding superclasses ([9, 12]). However, specific properties of subclasses may be in contradiction with some of the general ones. In this case the new specific properties should *override* the general inherited ones. In this way the structure of the subclass relationship introduces a corresponding structure of priority. Such structures of priority are formalized by *hierarchic specifications* ([12]). In this chapter we investigate hierarchic specifications, their semantics and logical properties. Furthermore the basic concepts underlying composition are defined and studied.

Hierarchic specifications (written in an arbitrary institution) and the corresponding *lexicographic preference* are defined in section 2.1. More structured semantics are needed for explaining composition. The *hierarchies of local, lexicographic and differential preferences*, introduced for this purpose, are compared. We take the hierarchy of differential preferences as semantics of a hierarchic specification. A Galois connection between hierarchic specifications and their semantics, explaining syntactic constructions via semantic operations, is established in section 2.3.

Section 2.2 deals with the lexicographic preference of *specifications*. This preference has a semantic counterpart in the *lexicographic combination* of *preferences* ([78, 2]), presented in section 2.2.1. Under some conditions (including finiteness) the lexicographic preference of a hierarchic specification can be equivalently expressed by a (flat) specification. We show this in section 2.2.2.

Having defined the semantics for composition a Galois connection between hierarchic specifications and the corresponding semantics is obtained and displayed in section 2.3.1. This connection is fundamental in defining composition of hierarchic specifications (see the next chapter 3) and, in particular, implies the definition of the *theory* of a hierarchic specification. The relations between this theory and the hierarchic specification inducing it are studied in section 2.3.2.

In section 2.4 we define *extensions* and *notions of consequence* of hierarchic specifications. These are derived from the lexicographic preference defined in section 2.1. Properties corresponding to those presented for specifications (such as the *existence* of extensions; chapter 1, section 1.3) are generalized. The special case of institutions where the compactness property holds is treated in section 2.5. We show that in such an institution any hierarchic specification has, at least, one extension.

We conclude the chapter in section 2.6.

2.1 Hierarchic Specifications

In this section we define *hierarchic specifications* and present the *semantics* needed for the purpose of *composition* of such specifications. We note in section 2.1.2 that a semantics for composition has to have more information than the corresponding *lexicographic preference*. The later, presented in section 2.1.1, displays the global preference induced by the defaults from prioritized levels on the models of the axioms. Subsection 2.1.2 is dedicated to present semantics for composition. These are the *hierarchy of lexicographic preferences* and the *hierarchy of differential preferences* which are shown to be equivalent in section 2.1.4. A third semantics, the *hierarchy of local preferences* is also presented.

In order to compare these semantics operators relating them are defined in section 2.1.3.

2.1.1 Syntax and Lexicographic Preference

The idea of assigning different priorities to defaults stems from Prioritized Circumscription from Lifschitz ([60]) and has been transported to Default Logic by Brewka ([14]). A preferential semantics of supernormal defaults structured by priorities has been given by Braß in [6, 7]. This preferential semantics corresponds to the lexicographic preference (see below) of such a specification. A corresponding concept, the lexicographic combination of (local) preferences (to be defined in section 2.2.1) is the semantics used in the Ordered Theory Presentations from

Ryan [74, 75]. The formalism we present here corresponds to the one in [12] and inherits from both [6, 7] and [74, 75].

Hierarchic specifications display the relations of priority between sets of defaults by means of priority levels, organized by a partial order. The overall effect of these defaults taking their respective priorities into account is displayed in a pre-order, the *lexicographic preference*. Hierarchic specifications and the corresponding lexicographic preference are now defined and illustrated. We begin with preliminary considerations.

Firstly we need to restrict the partial orders that provide priority levels for hierarchic specifications to *well-founded partial orders*. The condition of well-foundedness is of technical importance: it allows to prove properties of hierarchic specifications by using well-founded induction (see [91]).

Definition 70 A *well-founded relation* is a pair $(H, \prec)$ with $\prec \subseteq H \times H$ a binary relation over the set H satisfying the additional condition that there are no infinite *descending* chains $\ldots \prec h_i \ldots \prec h_1 \ldots \prec h_0$. This condition is equivalent (see [91]) to every non-empty subset H' of H having a *minimal* element, i.e. there is $h' \in H'$ such that $h \prec h'$ implies $h \notin H'$.

A *well-founded partial order* is a partial order $(H, \preceq)$ such that the corresponding strict partial order $(H, \prec)$, defined by $a \prec b$ iff $a \preceq b$ and $a \neq b$, is well-founded. An element of H is said *minimal* according to $\preceq$ iff it is minimal according to $\prec$.
■

The following notation for orders will be used throughout this chapter.

Notation 71

- Partial orders that organize priority levels are denoted by $\preceq$. The corresponding strict partial order is denoted by $\prec$. Similarly with the partial order $\subseteq$ (set inclusion) and $\subset$ (set strict inclusion).
- Pre-orders organizing interpretation structures are denoted by $\sqsubseteq$, possibly with indices: $\sqsubseteq_h^\alpha$.

  The symbol $\sqsubset_h^\alpha$ denotes the corresponding strict relation defined by $m \sqsubset_h^\alpha m'$ iff $m \sqsubseteq_h^\alpha m'$ and $m' \not\sqsubseteq_h^\alpha m$. The symbol $\equiv_h^\alpha$ denotes the corresponding equivalence relation defined by $m \equiv_h^\alpha m'$ iff $m \sqsubseteq_h^\alpha m'$ and $m' \sqsubseteq_h^\alpha m$.
- Given pre-orders $\sqsubseteq_{h_1}^\alpha, ..., \sqsubseteq_{h_n}^\alpha$ the pre-order denoted by $\sqsubseteq_{\{h_1,...,h_n\}}^\alpha$ with indices $\{h_1, ..., h_n\}$ is the intersection of the pre-orders $\sqsubseteq_{h_1}^\alpha, ..., \sqsubseteq_{h_n}^\alpha$. In other words $m \sqsubseteq_{\{h_1,...,h_n\}}^\alpha m'$ iff $m \sqsubseteq_{h_1}^\alpha m'$ and ... and $m \sqsubseteq_{h_n}^\alpha m'$.

All notions defined are parameterized in a given institution.

Remark 72 Recall that concepts are defined in the scope of

$$\mathcal{I} = (\mathsf{Sign}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \{\models^{\mathcal{I}}_{\Sigma}, \Sigma \in |\mathsf{Sign}^{\mathcal{I}}|\}),$$

a fixed but arbitrary institution.

A hierarchic specification consists of a set of axioms of a given signature plus sets of defaults from the same signature. The priorities to be assigned to each such set of defaults are given by a set of priority levels, organized by a partial order.

Note that the same formula may have different priorities (if it belongs to the sets of formulas assigned to different priority levels). This situation occurs in the case of a property that holds by default for all elements of a class, does not hold by default for those of a subclass but holds again by default for the elements of a subsubclass (the defaults formalizing this property are repeated for the elements of the subsubclass since they also belong to the class). The flying abilities of mammals, bats and newly-born bats are an example of such a situation.

Note also that the defaults of the subclass are (in general) more specific and therefore more important than those of the class. In this way the direction of the subclass relationship is *contrary* to the likeliness of the defaults. In our formalization, and for this reason, the higher a priority level is the *least* important its defaults are. That is, we choose hierarchic specifications to be formally similar to actual structured specifications (in this we adopt a convention from [75]). The disadvantage of having priority levels with reversed priority is, in our opinion, outweighed by the facility in translating an actual specification to its corresponding formalization (hierarchic specification).

We recall from [12] (with minor changes) the definition of hierarchic specification.

Definition 73 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,
- a function Δ assigning to each priority level $h \in \mathcal{H}$ a set of defaults $\Delta(h) \subseteq \mathsf{Sen}^{\mathcal{I}}(\Sigma)$.

The projections $\mathsf{sg}(S) = \Sigma$, $\mathsf{ax}(S) = A$ and $\mathsf{po}(S) = (H, \preceq)$ assign to a specification S its signature, its set of axioms and its priority partial order. Moreover $\mathsf{df}(S, h) = \Delta(h)$ assigns to S and level $h \in H$ the defaults from S at that level. ∎

The following auxiliary definitions of defaults from a hierarchic specification holding in interpretation structures are useful.

Definition 74 Let S be a hierarchic specification, $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority, $H' \subseteq H$ a set of priority levels from S and $M \subseteq \mathsf{ax}(S)^\bullet$ a class of models of the axioms of S.

- the set $\mathsf{df}(S, H') = \cup_{h' \in H'} \mathsf{df}(S, h')$ is the union of the defaults from S from levels $h' \in H'$,
- the set $\mathsf{df}(S, H')(M) = \{d \in \mathsf{df}(S, H') : M \vDash d\}$ is the set of defaults from levels in H' satisfied by each interpretation structure $m \in M$. The notations $\mathsf{df}(S, h)(M)$, $\mathsf{df}(S, H')(m)$ and $\mathsf{df}(S, h)(m)$ are used in the case of H' or M (or both) being singletons,
- when all levels H from S are involved $\mathsf{df}(S)$ abbreviates $\mathsf{df}(S, H)$ and $\mathsf{df}(S)(M)$ abbreviates $\mathsf{df}(S, H)(M)$. ■

The following example of a hierarchic specification is useful. Note that we have the *Specificity Principle* in mind: defaults of subclasses should be taken as being of higher importance (lower priority) than the defaults of their superclasses.

Example 75 We reformulate the specification BATMAN noting that bats and humans are mammals and concentrate in their flying abilities. Mammals, by default, do not fly and this behavior is inherited by humans. These dream by default. Bats, in spite of being mammals, do, by default, fly. We choose a set M of constants to denote mammals, select a set $B \subseteq M$ of constants to denote bats and a set $U \subseteq M$ to denote humans. Note that $\mathtt{bm} \in B \cap U$ (Batman is a bat and a human). Our choice of symbols is stated in the first order logic signature sg(MAMMALS) having M as set of constants (identifying mammals) and the following unary predicates: $\mathtt{Bat}, \mathtt{Hum}, \mathtt{Fl}, \mathtt{Dr}$. The set of axioms simply states that all Bs are bats and all Us are humans:

$$\mathsf{ax}(\text{MAMMALS}) = \{\mathtt{Bat(b)} : \ \mathtt{b} \in \mathtt{B}\} \cup \{\mathtt{Hum(u)} : \ \mathtt{u} \in \mathtt{U}\}.$$

The defaults are organized as displayed in the following figure 2.1. Priority levels are named $\mathtt{Mammals}, \mathtt{Bats}, \mathtt{Humans}, \mathtt{batman}$. The relation of priority between these levels is apparent from the picture. We have $\mathtt{batman} \preceq \mathtt{Bats} \preceq \mathtt{Mammals}$ and $\mathtt{batman} \preceq \mathtt{Humans} \preceq \mathtt{Mammals}$. Recall that we follow the convention that lower levels have higher likeliness or priority. To each level it is assigned a set of formulas, the defaults having that level as priority level. They state that mammals do not fly by default and this information should be inherited by humans. These dream by default. Bats on the contrary fly by default and, being at a lower level, the corresponding defaults should override the corresponding ones for mammals. The level `batman` has one default stating that `bm` does not dream[1].

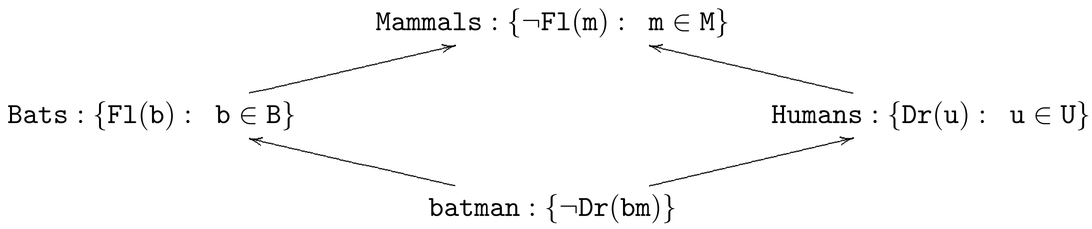

Figure 2.1: Batman Flies.

The intended overall effect of this specification is now outlined. Take a mammal which is not a bat. Then it will not fly by default. If this mammal is a human it will also dream by default. Mammals that are bats will fly by default since the corresponding default `Fl(b)` is of lower priority than the corresponding default ¬`Fl(b)` stated at the level `Mammals`. Mammals that are bats and humans, will fly and dream by default, with the one exception of `bm`: he does not dream by default (too busy fighting against crime, possibly). $\triangle$

As the previous example suggests 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 [74, 75]). We define the *lexicographic preference* induced by a hierarchic specification.

Definition 76 The *lexicographic preference* induced by a hierarchic specification S, denoted by $\mathsf{lex}^\circ(S)$ is the pre-order with

- the same signature as S, $\mathsf{sg}(\mathsf{lex}^\circ(S)) = \mathsf{sg}(S)$,
- the models of the axioms from S as class of interpretation structures, $|\mathsf{lex}^\circ(S)| = \mathsf{ax}(S)^\bullet$ and
- the relation $\sqsubseteq^\circ = \mathsf{rl}(\mathsf{lex}^\circ(S)) \subseteq |\mathsf{lex}^\circ(S)| \times |\mathsf{lex}^\circ(S)|$ 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). ■

[1]For this reason there is no logical relation, nor is it intended to be any, between this specification and BATMAN.

Proof We now show that $\sqsubseteq^\circ$ is a pre-order. Reflexivity is trivial so we proceed to prove transitivity. Assume that $m \sqsubseteq^\circ n$ and $n \sqsubseteq^\circ o$. Let $h \in H$ be an arbitrary priority levels and assume that $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(o)$. We have to show that there is $h' \prec h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(o)$. There are two possibilities: either $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$ or $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$.

1. ($\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$). In this case there is $h_1 \prec h$ such that $\mathsf{df}(S,h_1)(m) \subset \mathsf{df}(S,h_1)(n)$. Therefore the set $\{h_1 \prec h : \mathsf{df}(S,h_1)(m) \subset \mathsf{df}(S,h_1)(n)\}$ is non-empty and has a minimal element. Let h_2 be such an minimal element. Then $\mathsf{df}(S,h_2)(m) \subset \mathsf{df}(S,h_2)(n)$. If $\mathsf{df}(S,h_2)(n) \subseteq \mathsf{df}(S,h_2)(o)$ then $\mathsf{df}(S,h_2)(m) \subset \mathsf{df}(S,h_2)(n) \subseteq \mathsf{df}(S,h_2)(o)$ implies that we may choose $h' = h_2$. Otherwise there is $h_3 \prec h_2$ with $\mathsf{df}(S,h_2)(n) \subset \mathsf{df}(S,h_2)(o)$. Since $h_3 \prec h_2$ and h_2 is a minimal level under h having $\mathsf{df}(S,h_2)(m) \subset \mathsf{df}(S,h_2)(n)$ it is easy to check that $\mathsf{df}(S,h_3)(m) \subseteq \mathsf{df}(S,h_3)(n)$. Therefore $\mathsf{df}(S,h_3)(m) \subseteq \mathsf{df}(S,h_3)(n) \subset \mathsf{df}(S,h_3)(o)$ and we may choose $h' = h_3$.

2. ($\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$). Since $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(o)$ also $\mathsf{df}(S,h)(n) \not\subseteq \mathsf{df}(S,h)(o)$ and there is $h_1 \prec h$ such that $\mathsf{df}(S,h_1)(n) \subset \mathsf{df}(S,h_1)(o)$. In the case $\mathsf{df}(S,h_1)(m) \subseteq \mathsf{df}(S,h_1)(n) \subset \mathsf{df}(S,h_1)(o)$ we may choose $h' = h_1$. Otherwise we have $\mathsf{df}(S,h_1)(m) \not\subseteq \mathsf{df}(S,h_1)(n)$. The proof proceeds like the proof of the case 1 above. ✓

Example 77 Consider the hierarchic specification, written in propositional logic, with an empty set of axioms and two defaults p, q, where p is considered of lower priority than q. Its preference relation is displayed on the left of the fol-

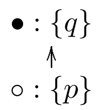

Figure 2.2: p lower than q

lowing figure 2.3 (some relations resulting from transitivity are not represented). On the right is displayed the preference relation associated with the specification

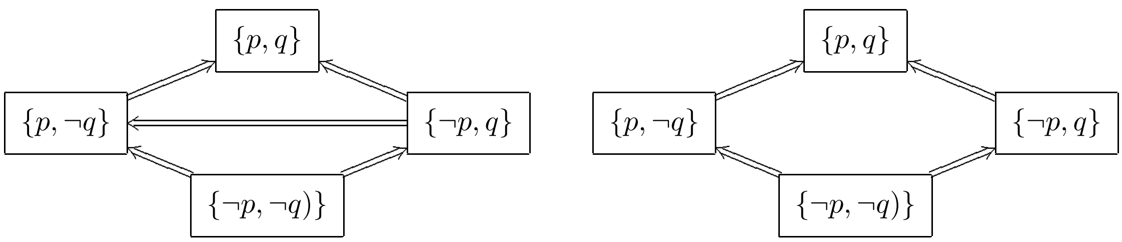

Figure 2.3: Preference of p, q related by priority (left) and not related (right)

having no axioms and the defaults p and q. The difference between both consists in the relation of preference between the models of $\{\neg p, q\}$ and those of $\{p, \neg q\}$. In fact these are unrelated when p and q are not ordered by priority, since they satisfy different defaults. This is not the case when these defaults are ordered by priority. The models of $\{p, \neg q\}$ are strictly better than the models $\{\neg p, q\}$ since they satisfy the default p better than q.

The preference relation can be described as follows. Any model of p is better than any model of $\neg p$. The default q is used to reorganize the interpretation structures that are equally good at satisfying p: from among the models of p those satisfying q are better than those not satisfying q. The same happens among the interpretation structures not satisfying p. This is better illustrated by redrawing the preference relation of p, q related by priority as follows.

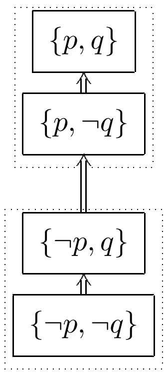

Figure 2.4: Preference of p, q related by priority

The name lexicographic ordering stems from the analogy between such orderings and the lexicographic ordering of words in dictionaries (see ([2]). In this case the position of letters in the word corresponds to priority. The order of words is decided in the first position where the words differ, i.e. in the most important priority level where they differ. The letters after that position are irrelevant: they correspond to defaults of less priority. $\triangle$

Remark 78 In the example the displayed relation of priority is not a partial order: transitivity and reflexivity are not shown. In general, the "economical" principle of stating only the needed priority relations should be followed, and the partial order of hierarchic specifications should be derived from the specified relations. This presents no formal difficulty. Any relation R "presents" a partial order among the equivalence classes of its transitive and reflexive closure (see definition 59). The hierarchic specification formalizing the actual specification will have this partial order as priority structure. The set of defaults assigned to each new level, a equivalence class of original levels, is the union of the defaults in the original levels. In fact such an equivalence class means that the original levels represent the same priority, which is the priority that should be assigned to the formulas populating them.

2.1.2 Semantics for Composition

The lexicographic preference just presented will be the main semantics for defining logical properties of a hierarchic specification (see section 2.4). 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. In particular the operation of adding defaults at a particular priority level, even in the case of specifications with the same structure of priority, may result in non-equivalent specifications. This difficulty is made explicit in the following example.

Example 79 Consider now two hierarchic specifications from propositional logic without axioms and one default, p. In the following figure are displayed the priorities assigned to p by each of these two hierarchic specifications. It is straight-

$$\begin{array}{c} \bullet : \emptyset \\ \uparrow \\ \circ : \{p\} \end{array} \qquad \begin{array}{c} \bullet : \{p\} \\ \uparrow \\ \circ : \emptyset \end{array}$$

Figure 2.5: p at different priority levels

forward to check that both have the same lexicographic preference, corresponding to the specification with the only default p (and no axioms). However, when

$$\begin{array}{c} \bullet : \emptyset \\ \uparrow \\ \circ : \{p, \neg p\} \end{array} \qquad \begin{array}{c} \bullet : \{p\} \\ \uparrow \\ \circ : \{\neg p\} \end{array}$$

Figure 2.6: $\neg p$ added at the most important priority level

adding the new default $\neg p$ to the bottom level ($\circ$) one obtains two new hierarchic specifications with different lexicographic semantics.

The lexicographic semantics of the specification on the left of figure 2.6 corresponds to the two defaults $\{p, \neg p\}$ and that of the specification from the right corresponds to $\neg p$ alone (that overrides p). △

In the following we present three different possibilities of semantics, the *hierarchies of local, lexicographic and differential preferences*. From any of these semantics the lexicographic preference of the original specification can be derived (see section 2.2.1). All of them correspond to pre-orders organized by priority levels and will be compared in the next section 2.1.4. The hierarchy of local preferences is obtained by viewing the sets of defaults at each level (plus the axioms) as independent (flat) specifications. The corresponding preferences are assigned to the original priority levels. The second and third structured semantics, the hierarchy of lexicographic preferences and the hierarchy of differential preferences are equivalent (see section 2.1.4) and are taken as the semantics for composition. The hierarchy of lexicographic preferences is obtained by associating to each priority level the lexicographic preference resulting from the interaction of the defaults at that level with the ones at lower (more important) levels. The hierarchy of differential preferences assigns to each level h the preference resulting from comparing interpretation structures with respect to the defaults at h but restricting this comparison only to the interpretation structures that were *equivalent* at lower levels. For simplicity sake we will use the hierarchy of differential preferences as semantics. All concepts can, however, be redefined for the hierarchy of lexicographic preferences.

All these structured semantics are hierarchies of pre-orders, that we now define. These organize given pre-orders over the same interpretation structures by priority.

Definition 80 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,
- a function Θ 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 $\mathcal{M}$.

The projections $\mathsf{sg}(\mathcal{H}) = \Sigma$, $|\mathcal{H}| = \mathcal{M}$ and $\mathsf{po}(\mathcal{H}) = (H\ \preceq)$ assign to a hierarchy of pre-orders its signature, its base set of interpretation structures and its partial order of priority. Moreover $\mathsf{rl}(\mathcal{H}, h) = \Theta(h)$ assigns to $\mathcal{H}$ and level $h \in H$ the pre-order from $\mathcal{H}$ at that level.

A Σ-*hierarchy of pre-orders* $(\mathcal{M}, (H, \preceq), \Theta)$ is the hierarchy $(\Sigma, \mathcal{M}, (H, \preceq), \Theta)$. ∎

We now define the hierarchy of local preferences of a hierarchic specification. This is the hierarchy of pre-orders obtained by assigning to each level h the preference relation associated with the defaults from S at that level alone (and the axioms).

Definition 81 The *hierarchy of local preferences* induced by a hierarchic specification S, denoted by $S^{\odot}$ is the hierarchy of pre-orders with:

- the same signature as S, $\mathsf{sg}(S^{\odot}) = \mathsf{sg}(S)$,
- the models of the axioms from S as class of interpretation structures, $|S^{\odot}| = \mathsf{ax}(S)^{\bullet}$,
- the same well-founded partial order $(H \preceq)$ as S, $(H \preceq) = \mathsf{po}(S^{\odot}) = \mathsf{po}(S)$,
- the function $\sqsubseteq^{\odot}$ that to each priority level $h \in H$ assigns the relation $\sqsubseteq^{\odot}_{h} = \mathsf{rl}(S^{\odot}, h) \subseteq |S^{\odot}| \times |S^{\odot}|$ defined by $m \sqsubseteq^{\odot}_{h} n$ iff $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$.

Note that each relation $\sqsubseteq^{\odot}_{h}$ is the preference relation induced by the specification $(\mathsf{ax}(S), \mathsf{df}(S, h))$ having as axioms the axioms from S and as defaults the defaults from S at level h. For this reason each $\sqsubseteq^{\odot}_{h}$ is a pre-order. ∎

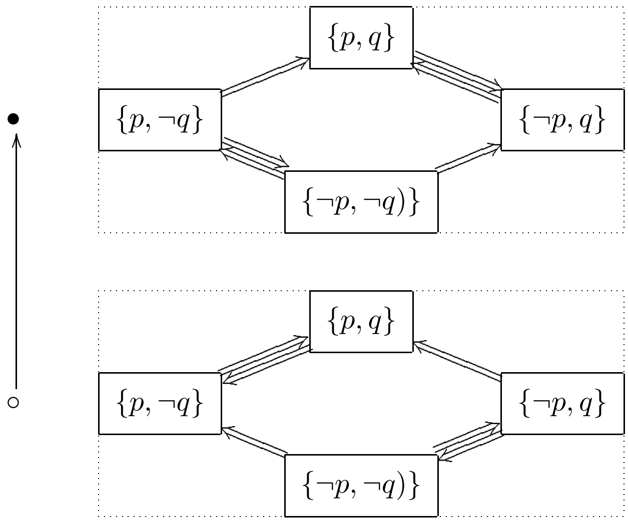

Figure 2.7: Hierarchy of Local Preferences

The hierarchy of local preferences is illustrated in the following example. Note that the defaults at each level are use to compare interpretation structures independently from the defaults from other levels.

Example 82 Recall from example 77 the specification without axioms and the following priority on the defaults p and q. The corresponding hierarchy of local

$$\begin{array}{c} \bullet : \{q\} \\ \uparrow \\ \circ : \{p\} \end{array}$$

preferences (figure 2.7) will have as class of interpretation structures all propositional interpretation structures (since the specification has no axioms).

It consists furthermore of assigning the preference of p alone to the level $\circ$ and the preference of q alone to the level $\bullet$. Therefore in level $\circ$ the models of p have preference and in level $\bullet$ those of q have preference. $\triangle$

The previous semantics does not take into account the fact that defaults exist in the context of defaults of other priority levels. This fact is corrected in the hierarchy of lexicographic preferences. This hierarchy of pre-orders assigns to each priority level the lexicographic preference associated with the defaults at that level in the context of the defaults of lower (more important) levels.

Definition 83 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, $\mathsf{sg}(S^{\oplus}) = \mathsf{sg}(S)$,
- the models of the axioms from S as class of interpretation structures, $|S^{\oplus}| = \mathsf{ax}(S)^{\bullet}$,
- the same well-founded partial order $(H \preceq)$ as S, $(H \preceq) = \mathsf{po}(S^{\oplus}) = \mathsf{po}(S)$,
- the function $\sqsubseteq^{\oplus}$ that to each priority level $h \in H$ assigns the relation $\sqsubseteq^{\oplus}_h = \mathsf{rl}(S^{\oplus}, h) \subseteq |S^{\oplus}| \times |S^{\oplus}|$ defined by $m \sqsubseteq^{\oplus}_h n$ 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).

Note that each $\sqsubseteq^{\oplus}_h$ is the lexicographic preference of the substructure of S obtained by restricting the priority levels to the set $\{h' : h' \preceq h\}$. This relation is a pre-order (see definition 76). ■

Example 84 We display the hierarchy of lexicographic preferences for the same specification of the previous example 82 in figure 2.8.

It has as class of interpretation structures all propositional interpretation structures (since the specification has no axioms). It consists furthermore of assigning the lexicographic preference of p alone to the level $\circ$ and the lexicographic preference of the whole specification (already displayed in example 77) to the level $\bullet$. The difference to the hierarchy of local preferences (shown in example 82) shows up only in the level $\bullet$.

In fact the lexicographic preference at the minimal level $\circ$ coincides with the preference associated with the defaults at $\circ$ alone (since there are no levels under $\circ$). $\triangle$

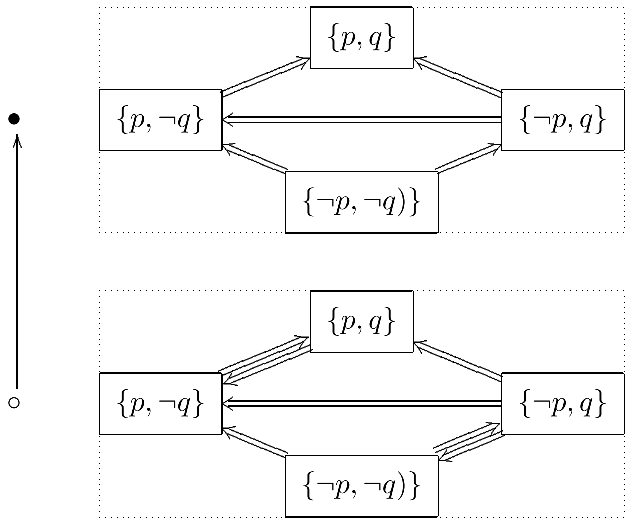

Figure 2.8: Hierarchy of Lexicographic Preferences

The previous semantics focuses on the overall effect of the defaults at some level in the context of the defaults at lower levels (and axioms). The next semantics, the hierarchy of differential preferences, is equivalent (see next section 2.1.4) to the hierarchy of lexicographic preferences, but displays the information of preference in a different way. The hierarchy of differential preferences displays at level h the preference induced by the defaults D_h in the interpretation structures that were equivalent at better levels. In fact, the defaults D_h are only relevant to compare interpretation structures that were equivalent at better hierarchy levels: interpretation structures that were strictly related by preference (resp. unrelated) at levels under h will remain strictly related by preference (resp. unrelated). This formalization also implies a simple characterization of "defaults implicit" in a level h (see the next example 86 and section 2.3.1).

Definition 85 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, $\mathsf{sg}(S^{\ominus}) = \mathsf{sg}(S)$,
- the models of the axioms from S as class of interpretation structures, $|S^{\ominus}| = \mathsf{ax}(S)^{\bullet}$ and
- the same well-founded partial order $(H \preceq)$ as S, $(H \preceq) = \mathsf{po}(S^{\ominus}) = \mathsf{po}(S)$,
- the function $\sqsubseteq^{\ominus}$ that to each priority level $h \in H$ assigns the relation $\sqsubseteq^{\ominus}_h = \mathsf{rl}(S^{\ominus}, h) \subseteq |S^{\ominus}| \times |S^{\ominus}|$ defined by

$$m \sqsubseteq^{\ominus}_h n \text{ iff } \mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n) \text{ and } \mathsf{df}(S,h')(m) = \mathsf{df}(S,h')(n)$$

for every priority level $h' \prec h$.

It is straightforward to check that each $\sqsubseteq_h^{\ominus}$ is a pre-order. ■

We illustrate this semantics by comparing a hierarchy of differential preferences with the corresponding hierarchy of lexicographic preferences.

Example 86 The previous example 84 is important in motivating the hierarchy of differential preferences (in figure 2.9).

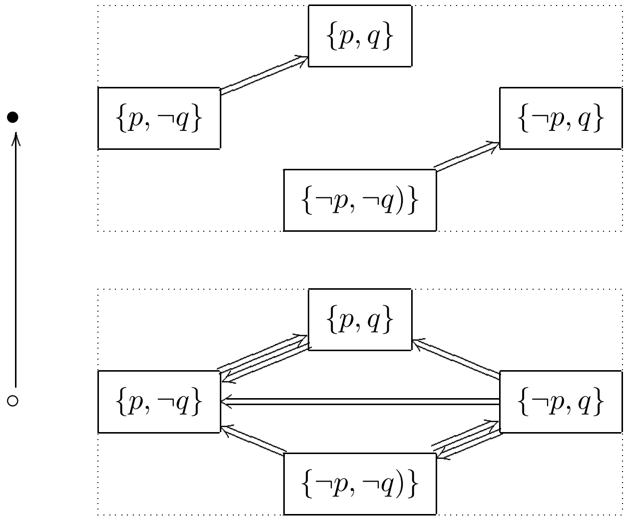

Figure 2.9: Hierarchy of Differential Preferences

Consider now the question (to be fully solved in section 2.3.1) of finding the "defaults implicit" at level • in the previous specification. This corresponds to finding the formulas that can be added to the previous specification without changing its meaning. We take as meaning of the specification the hierarchy of lexicographic preferences already displayed in example 84. Clearly q should be one of those formulas (since it is already a specified default). It is however, not true in the lexicographic preference at level • that if $m \vDash q$ then any n better that m will also satisfy q. (Since the models of $\{q, \neg p\}$ are less preferred than those of $\{\neg q, p\}$).

The condition of default implicit is obtained by restricting the interpretation structures n to those preferred to m at level • and *equivalent* to m at levels under •. But this is the same as stating that n is preferred to m according to the differential preference at level •. In this way the condition of implicit default becomes formally identical to definition 27 of chapter 1. The formal similarity with the concepts from chapter 1 will be of advantage. Note that q is a default implicit in the differential preference at level •. △

In the following we display auxiliary results that relate equivalence and strict preference of the differential and lexicographic preferences with satisfaction of defaults. These results will be used when comparing these semantics.

Lemma 87 Let S be a hierarchic specification and $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority. Let $\sqsubseteq_h^{\oplus} = \mathsf{rl}(S^{\oplus}, h)$ and $\sqsubseteq_h^{\ominus} = \mathsf{rl}(S^{\ominus}, h)$ be the relations associated respectively by the hierarchy of lexicographic preferences of S and by the hierarchy of differential preferences of S to the level $h \in H$. Then:

1. $m \equiv_h^{\ominus} n$ iff $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for every $h' \preceq h$,
2. $m \sqsubset_h^{\ominus} n$ iff $\mathsf{df}(S, h)(m) \subset \mathsf{df}(S, h)(n)$ and $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for all $h' \prec h$,
3. $m \equiv_h^{\oplus} n$ iff $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for every $h' \preceq h$,
4. $m \sqsubset_h^{\oplus} n$ iff $m \sqsubseteq_h^{\oplus} n$ and there exists $h' \preceq h$ with $\mathsf{df}(S, h')(m) \subset \mathsf{df}(S, h')(n)$ and, for every $h'' \prec h' \preceq h$, $\mathsf{df}(S, h'')(m) \subseteq \mathsf{df}(S, h'')(n)$.

Proof The proofs of the first and second properties are straightforward since by definition $m \sqsubseteq_h^{\ominus} n$ iff $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ and $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for all $h' \prec h$.

The proofs of the third and fourth properties are by well-founded induction in the well-founded partial order $(H, \preceq)$. Both proofs need the following recursive definition of $\sqsubseteq_h^{\oplus}$: $m \sqsubseteq_h^{\oplus} n$ iff for every priority level $h_1' \prec h$, $m \sqsubseteq_{h_1'}^{\oplus} n$ and if $\mathsf{df}(S, h)(m) \not\subseteq \mathsf{df}(S, h)(n)$ then there is $h' \prec h$ with $\mathsf{df}(S, h')(m) \subset \mathsf{df}(S, h')(n)$. The proof of this fact is straightforward. We omit it and only recall (from definition 83) that $m \sqsubseteq_h^{\oplus} n$ 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 $\mathsf{df}(S, h'')(m) \subset \mathsf{df}(S, h'')(n)$.

We begin with the proof of the third property.

- Let h be minimal in $(H, \preceq)$. Since there is no $h' \preceq h$ except h itself one has $m \sqsubseteq_h^{\oplus} n$ iff $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$. Therefore $m \equiv_h^{\oplus} n$ iff $\mathsf{df}(S, h)(m) = \mathsf{df}(S, h)(n)$.
- Consider a non-minimal h.
  - $\Rightarrow$ Assume that $m \equiv_h^{\oplus} n$. Then $m \sqsubseteq_{h_1'}^{\oplus} n$ and $n \sqsubseteq_{h_1'}^{\oplus} m$ (see the recursive definition of $\sqsubseteq_h^{\oplus}$) for every priority level $h' \prec h$. By induction hypothesis for every priority level $h_1' \prec h$, $\mathsf{df}(S, h_1')(m) = \mathsf{df}(S, h_1')(n)$. Therefore $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ since $m \sqsubseteq_h^{\oplus} n$ and there is no level $h_1' \prec h$ with $\mathsf{df}(S, h_1')(m) \subset \mathsf{df}(S, h_1')(n)$. In the same way we conclude $\mathsf{df}(S, h)(n) \subseteq \mathsf{df}(S, h)(m)$. Therefore $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for all h' under h, including h.

$\Leftarrow$ Assume now that $\mathsf{df}(S,h')(m) = \mathsf{df}(S,h')(n)$ for all $h' \preceq h$. It is obvious from definition of $\sqsubseteq^{\oplus}_{h}$ that $m \sqsubseteq^{\oplus}_{h} n$ and $n \sqsubseteq^{\oplus}_{h} m$. ✓

Now the proof of the fourth property.

- Let h be minimal in $(H, \preceq)$. Recall from the proof above that for minimal h, $m \sqsubseteq^{\oplus}_{h} n$ iff $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$. Therefore $m \sqsubset^{\oplus}_{h} n$ iff $\mathsf{df}(S,h)(m) \subset \mathsf{df}(S,h)(n)$. Take h' to be h. The remaining property for every $h'' \prec h' = h$ is vacuously true.
- Consider a non-minimal h.

  $\Rightarrow$ Assume that $m \sqsubset^{\oplus}_{h} n$. Then $m \sqsubseteq^{\oplus}_{h} n$ and from the recursive definition of $\sqsubseteq^{\oplus}_{h}$ also $m \sqsubseteq^{\oplus}_{h_1} n$ for every $h_1 \prec h$. Now there are two possibilities: either the relation $m \sqsubseteq^{\oplus}_{h_1} n$ is strict for some $h_1 \prec h$ or not:

  * If $m \sqsubset^{\oplus}_{h_1} n$ for some $h_1 \prec h$ we use the induction hypothesis for that h_1: there exists $h'_1 \preceq h_1 \prec h$ with $\mathsf{df}(S,h'_1)(m) \subset \mathsf{df}(S,h'_1)(n)$ and $\mathsf{df}(S,h''_1)(m) \subseteq \mathsf{df}(S,h''_1)(n)$ for every $h''_1 \prec h'_1 \preceq h_1 \prec h$. Choose $h' = h'_1$.
  * Otherwise we have $m \equiv^{\oplus}_{h_1} n$ for all $h_1 \prec h$. This implies $\mathsf{df}(S,h_1)(m) = \mathsf{df}(S,h_1)(n)$ for all $h_1 \prec h$. Therefore $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$ since $m \sqsubseteq^{\oplus}_{h} n$ and there is no $h_1 \prec h$ with $\mathsf{df}(S,h_1)(m) \subset \mathsf{df}(S,h_1)(n)$. This inclusion has to be strict for otherwise we would also have $n \sqsubseteq^{\oplus}_{h} m$. Now choose $h' = h$.

  $\Leftarrow$ Now assume that $m \sqsubseteq^{\oplus}_{h} n$ and there exists $h' \preceq h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$ and $\mathsf{df}(S,h'')(m) \subseteq \mathsf{df}(S,h'')(n)$ for every $h'' \prec h' \preceq h$. We have to prove that $m \sqsubset^{\oplus}_{h} n$, i.e. that $n \not\sqsubseteq^{\oplus}_{h} m$. To establish this we will show that $n \not\sqsubseteq^{\oplus}_{h'} m$. In fact if $h' = h$ then obviously $n \not\sqsubseteq^{\oplus}_{h} m$. If $h' \prec h$ then also $n \not\sqsubseteq^{\oplus}_{h} m$: $n \sqsubseteq^{\oplus}_{h} m$ would imply from the recursive definition of $\sqsubseteq^{\oplus}_{h}$ that $n \sqsubseteq^{\oplus}_{h'} m$ for all $h' \prec h$.

  We now show that $n \not\sqsubseteq^{\oplus}_{h'} m$. In fact neither $\mathsf{df}(S,h')(n) \subseteq \mathsf{df}(S,h')(m)$ nor there is $h'' \prec h'$ with $\mathsf{df}(S,h'')(n) \subset \mathsf{df}(S,h'')(m)$, since $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$ and $\mathsf{df}(S,h'')(m) \subseteq \mathsf{df}(S,h'')(n)$ for every $h'' \prec h'$. ✓

2.1.3 Combination Operators

In the next section 2.1.4 we compare the three structured semantics associated with a hierarchic specification. For that purpose we define here operators $\boxplus$ and $\boxminus$ that associate to a hierarchy of pre-orders a *hierarchy of lexicographic combinations* and a *hierarchy of differential combinations*. Given a hierarchic specification S the result of the operator $\boxplus$ on *any* of the corresponding structured semantics, $S^{\odot}$, $S^{\oplus}$ or $S^{\ominus}$ is the hierarchy of lexicographic preferences of S, $S^{\oplus}$. Similarly the result of the operator $\boxminus$ on*any* of $S^{\odot}$, $S^{\oplus}$ or $S^{\boxminus}$ is the hierarchy of differential preferences of S, $S^{\ominus}$. This provides a way to convert the hierarchy of lexicographic preferences to the hierarchy of differential preferences (and vice-versa) and implies equivalence of these two semantics. There is no corresponding operator for the hierarchy of local semantics. In fact different specifications with different hierarchies of local preferences may have the same $S^{\oplus}$ and $S^{\ominus}$ (see example 95 below).

The operator $\boxplus$ assigns to level h the overall effect of the pre-orders from the argument hierarchy at level h and under h. This corresponds to the *lexicographic combination* of those pre-orders and is the semantic operation corresponding to the lexicographic preference of hierarchic specifications (see section 2.2.1). As expected its definition is formally similar to that of lexicographic preference (see definition 76). Two interpretation structures m, n are related at level h if, for all $h' \preceq h$ (including h), whenever m and n are *not* related by the argument pre-order at level h' then there is a better level $h'' \prec h'$ where n is strictly related to m. The following inductive definition states this relation equivalently.

Definition 88 The *hierarchy of lexicographic combinations* associated with a hierarchy of pre-orders $\mathcal{H}$, denoted by $\mathcal{H}^{\boxplus}$ is the hierarchy of pre-orders with:

- the same signature as $\mathcal{H}$, $\mathsf{sg}(\mathcal{H}^{\boxplus}) = \mathsf{sg}(\mathcal{H})$,
- the same class of interpretation structures as $\mathcal{H}$, $|\mathcal{H}^{\boxplus}| = |\mathcal{H}|$,
- the same well-founded partial order $(H, \preceq)$ as $\mathcal{H}$, $(H, \preceq) = \mathsf{po}(\mathcal{H}^{\boxplus}) = \mathsf{po}(\mathcal{H})$,
- the function $\sqsubseteq^{\boxplus}$ that to each priority level h assigns the relation $\sqsubseteq^{\boxplus}_h = \mathsf{rl}(\mathcal{H}^{\boxplus}, h) \subseteq |\mathcal{H}^{\boxplus}| \times |\mathcal{H}^{\boxplus}|$ inductively defined in the structure of the well founded partial order as follows. Let $\sqsubseteq^{\boxplus}_{h^-} = \bigcap_{\{h' : h' \prec h\}} \sqsubseteq^{\boxplus}_{h'}$ denote the intersection of the lexicographic relations associated by $\mathcal{H}^{\boxplus}$ to the levels strictly under h and $\sqsubset^{\boxplus}_{h^-}$ the corresponding strict relation. Let $\sqsubseteq_h$ denote the relation $\mathsf{rl}(\mathcal{H}, h)$ assigned by $\mathcal{H}$ to the level h. Then
  - $m \sqsubseteq^{\boxplus}_h n$ iff $m \sqsubseteq_h n$, if h is minimal in $(H, \preceq)$;
  - $m \sqsubseteq^{\boxplus}_h n$ iff either $m \sqsubset^{\boxplus}_{h^-} n$ or ($m \sqsubseteq_h n$ and $m \sqsubseteq^{\boxplus}_{h^-} n$), otherwise. ■

Proof The proof that each $\sqsubseteq^{\boxplus}_h$ is a pre-order is by induction (in $(H, \preceq)$). Reflexivity is trivial. Transitivity results from the fact that $m \sqsubseteq^{\boxplus}_h n$ iff either $m \sqsubset^{\boxplus}_{h^-} n$ or ($m \sqsubseteq_h n$ and $m \equiv^{\boxplus}_{h^-} n$) and that $\sqsubset^{\boxplus}_{h^-}$ is transitive and $\equiv^{\boxplus}_{h^-}$ an equivalence (by the induction hypothesis). ✓

The condition defining $\sqsubseteq^{\boxplus}$ can be equivalently rewritten in a way formally similar to the definition of lexicographic preference.

Lemma 89 Let $\mathcal{H}$ be a hierarchy of pre-orders, $(H, \preceq)$ its partial order of priority and for $h \in H$ let $\sqsubseteq_h = \mathsf{rl}(\mathcal{H}, h)$ be the pre-order assigned by $\mathcal{H}$ to level h. Then $m \sqsubseteq_h^{\boxplus} n$ iff for every $h' \preceq h$ (including h) either $m \sqsubseteq_{h'} n$ or there is $h'' \prec h$ with $m \sqsubset_{h''} n$.

Proof Omitted. ✓

The definition of the *hierarchy of differential combinations* is formally similar to the hierarchy of differential preferences of a specification (see definition 85). Interpretation structures are related at level h when they are related at that level by the argument pre-order and, furthermore, are equivalent according to the argument pre-orders at levels under h.

Definition 90 The *hierarchy of differential combinations* associated with a hierarchy of pre-orders $\mathcal{H}$, denoted by $\mathcal{H}^{\boxminus}$ is the hierarchy of pre-orders with:

- the same signature as $\mathcal{H}$, $\mathsf{sg}(\mathcal{H}^{\boxminus}) = \mathsf{sg}(\mathcal{H})$,
- the same class of interpretation structures as $\mathcal{H}$, $|\mathcal{H}^{\boxminus}| = |\mathcal{H}|$,
- the same well-founded partial order $(H, \preceq)$ as $\mathcal{H}$, $(H, \preceq) = \mathsf{po}(\mathcal{H}^{\boxminus}) = \mathsf{po}(\mathcal{H})$,
- the function $\sqsubseteq^{\boxminus}$ that to each priority level h assigns the relation $\mathsf{rl}(\mathcal{H}^{\boxminus}, h) = \sqsubseteq_h^{\boxminus} \subseteq |\mathcal{H}^{\boxminus}| \times |\mathcal{H}^{\boxminus}|$ defined as follows. Let $\sqsubseteq_{h'}$ denote the relation $\mathsf{rl}(\mathcal{H}, h')$ associated by $\mathcal{H}$ to the level h' and $\equiv_{h^-}$ the equivalence defined by $m \equiv_{h^-} n$ iff for all $h' \prec h$, $m \equiv_{h'} n$, that relates interpretation structures exactly when they are equivalent according to all relations associated by $\mathcal{H}$ to the levels strictly under h. Then $m \sqsubseteq_h^{\boxminus} n$ iff $m \sqsubseteq_h n$ and $m \equiv_{h^-} n$.

Note that each $\sqsubseteq_h^{\boxminus}$ is a pre-order since it is the intersection of pre-orders. ■

We now proceed to study properties of the two operators $\boxplus$ and $\boxminus$. For that purpose the following auxiliary lemma characterizing equivalence classes of $\sqsubseteq_h^{\boxplus}$ and $\sqsubseteq_h^{\boxminus}$ is useful.

Lemma 91 Let $\mathcal{H}$ be a hierarchy of pre-orders, $(H, \preceq) = \mathsf{po}(\mathcal{H})$ its partial order of priority and $h \in H$ an arbitrary priority level. Let $\sqsubseteq_h = \mathsf{rl}(\mathcal{H}, h)$, $\sqsubseteq_h^{\boxplus} = \mathsf{rl}(\mathcal{H}^{\boxplus}, h)$ and $\sqsubseteq_h^{\boxminus} = \mathsf{rl}(\mathcal{H}^{\boxminus}, h)$ be the relations associated to level h by $\mathcal{H}$, $\mathcal{H}^{\boxplus}$ and $\mathcal{H}^{\boxminus}$. Let $\equiv_h$, $\equiv_h^{\boxplus}$ and $\equiv_h^{\boxminus}$ be the corresponding equivalence relations and $\equiv_{\{h' : h' \preceq h\}}$ the intersection of the equivalences $\equiv_{h'}$, for $h' \preceq h$ (including h). Then:

$$m \equiv_h^{\boxminus} n \text{ iff } m \equiv_{\{h' : h' \preceq h\}} n \text{ iff } m \equiv_h^{\boxplus} n.$$

Proof We show $m \equiv_h^{\boxminus} n$ iff $m \equiv_{\{h':h'\preceq h\}} n$ and $m \equiv_h^{\boxplus} n$ iff $m \equiv_{\{h':h'\preceq h\}} n$.

- That $m \equiv_h^{\boxminus} n$ iff $m \equiv_{\{h':h\preceq h\}} n$ follows easily from the definition 90 of $\sqsubseteq_h^{\boxminus}$: $m \sqsubseteq_h^{\boxminus} n$ iff $m \sqsubseteq_h n$ and $m \equiv_{h^-} n$, where h^- is the set of priority levels strictly under h. Therefore $m \equiv_h^{\boxminus} n$ iff $m \equiv_h n$ and $m \equiv_{h^-} n$, which is simply $m \equiv_{\{h':h'\preceq h\}} n$. ✓
- The proof of $m \equiv_h^{\boxplus} n$ iff $m \equiv_{\{h':h'\preceq h\}} n$ is by induction on $(H, \preceq)$:
  - if h is minimal then $m \sqsubseteq_h^{\boxplus} n$ iff $m \sqsubseteq_h n$ (definition 88 of $\sqsubseteq_h^{\boxplus}$) and $m \equiv_h^{\boxplus} n$ iff $m \equiv_h n$ iff $m \equiv_{\{h':h'\preceq h\}} n$ since $\{h' : h' \preceq h\} = \{h\}$.
  - otherwise recall from definition 88 that $m \sqsubseteq_h^{\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h n$ and $m \sqsubseteq_{h^-}^{\boxplus} n$). This implies $m \equiv_h^{\boxplus} n$ iff $m \equiv_h n$ and $m \equiv_{h^-}^{\boxplus} n$. Using the induction hypothesis we have finally $m \equiv_h^{\boxplus} n$ iff $m \equiv_h n$ and $m \equiv_{h^-} n$. ✓

The operators $\boxplus$ and $\boxminus$ are idempotent and form a bijection between hierarchies of differential combinations and hierarchies of lexicographic combinations.

Theorem 92 Given a hierarchy of pre-orders $\mathcal{H}$ then

1. $(\mathcal{H}^{\boxminus})^{\boxminus} = \mathcal{H}^{\boxminus}$,
2. $(\mathcal{H}^{\boxplus})^{\boxplus} = \mathcal{H}^{\boxplus}$,
3. $(\mathcal{H}^{\boxplus})^{\boxminus} = \mathcal{H}^{\boxminus}$ and
4. $(\mathcal{H}^{\boxminus})^{\boxplus} = \mathcal{H}^{\boxplus}$.

Proof The properties concerning signatures, interpretation structures and the partial order of priority are straightforward. The result of any of the operators on $\mathcal{H}$ is a hierarchy of pre-orders having the same signature $\mathsf{sg}(\mathcal{H})$ as $\mathcal{H}$ the same class of interpretation structures $|\mathcal{H}|$ and the same partial order of priority $(H, \preceq) = \mathsf{po}(\mathcal{H})$.

Let $\sqsubseteq_h$, $\sqsubseteq_h^{\boxminus}$ and $\sqsubseteq_h^{\boxplus}$ be the relations associated by $\mathcal{H}$, $\mathcal{H}^{\boxminus}$ and $\mathcal{H}^{\boxplus}$ to level h, respectively.

Before anything recall from lemmas 91 and 91 that $m \equiv_h^{\boxminus} n$ iff $m \equiv_h^{\boxplus} n$ iff $m \equiv_{\{h':h\preceq h\}} n$, where $\equiv_{\{h':h\preceq h\}}$ is the intersection of the equivalences $\equiv_{h'}$ from levels $h' \preceq h$ (including h).

Let $\sqsubseteq_h^{\boxminus\boxminus}$, $\sqsubseteq_h^{\boxminus\boxplus}$, $\sqsubseteq_h^{\boxplus\boxminus}$ and $\sqsubseteq_h^{\boxplus\boxplus}$ be the relations associated to level h by $(\mathcal{H}^{\boxminus})^{\boxminus}$, $(\mathcal{H}^{\boxminus})^{\boxplus}$, $(\mathcal{H}^{\boxplus})^{\boxminus}$ and $(\mathcal{H}^{\boxplus})^{\boxplus}$ respectively. The remain of the proof is by well-founded induction on $(H, \preceq)$. (The different proofs are independent although presented simultaneously.)

- Assume h is minimal in $(H, \preceq)$. Then
  - $\sqsubseteq_h^{\boxminus\boxminus}$ is by definition 90 equal to $\sqsubseteq_h^{\boxminus}$ (and this in turn equal to $\sqsubseteq_h$),
  - $\sqsubseteq_h^{\boxplus\boxplus}$ is by definition 88 equal to $\sqsubseteq_h^{\boxplus}$ (and this in turn equal to $\sqsubseteq_h$),

  - $\sqsubseteq_h^{\boxplus\boxminus}$ is again by definition 90 equal to $\sqsubseteq_h^{\boxplus}$ which coincides with $\sqsubseteq_h$ and with $\sqsubseteq_h^{\boxminus}$,
  - $\sqsubseteq_h^{\boxminus\boxplus}$ is again by definition 88 equal to $\sqsubseteq_h^{\boxminus}$ which coincides with $\sqsubseteq_h$ and with $\sqsubseteq_h^{\boxplus}$.

- Otherwise

  - From definition 90 $m \sqsubseteq_h^{\boxminus\boxminus} n$ iff $m \sqsubseteq_h^{\boxminus} n$ and $m \equiv_{h^-}^{\boxminus} n$, where $\equiv_{h^-}^{\boxminus}$ is the intersection of the equivalence relations $\equiv_{h'}^{\boxminus}$ for $h' \prec h$.

    Recall that $\equiv_{h^-}^{\boxminus}$ is the same as $\equiv_{h^-}$ (lemma 91).

    In this way the condition defining $m \sqsubseteq_h^{\boxminus\boxminus} n$ translates to $m \sqsubseteq_h n$ and $m \equiv_{h^-} n$ and $m \equiv_{h^-} n$. This is the condition defining $m \sqsubseteq_h^{\boxminus} n$.
  - From definition 88 $m \sqsubseteq_h^{\boxplus\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus\boxplus} n$ or ($m \sqsubseteq_h^{\boxplus} n$ and $m \sqsubseteq_{h^-}^{\boxplus\boxplus} n$), where $\sqsubseteq_{h^-}^{\boxplus\boxplus}$ denotes the intersection of the relations $\sqsubseteq_{h'}^{\boxplus\boxplus}$ for $h' \prec h$ and $\sqsubset_{h^-}^{\boxplus\boxplus}$ the corresponding strict relation. By the induction hypothesis this is $m \sqsubseteq_h^{\boxplus\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h^{\boxplus} n$ and $m \sqsubseteq_{h^-}^{\boxplus} n$). Recall that $m \sqsubseteq_h^{\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h n$ and $m \sqsubseteq_{h^-}^{\boxplus} n$) and substitute this above. After some tautological manipulation one obtains that $m \sqsubseteq_h^{\boxplus\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h n$ and $m \sqsubseteq_{h^-}^{\boxplus} n$), i.e. $m \sqsubseteq_h^{\boxplus\boxplus} n$ iff $m \sqsubseteq_h^{\boxplus} n$.
  - Again from definition 90 $m \sqsubseteq_h^{\boxplus\boxminus} n$ iff $m \sqsubseteq_h^{\boxplus} n$ and $m \equiv_{h^-}^{\boxplus} n$ (this means $m \equiv_{h'}^{\boxplus} n$ for all $h' \prec h$). Recalling the definition of $m \sqsubseteq_h^{\boxplus} n$ one obtains $m \sqsubseteq_h^{\boxplus\boxminus} n$ iff $m \equiv_{h^-}^{\boxplus} n$ and (either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h n$ and $m \sqsubseteq_{h^-}^{\boxplus} n$)). The conditions $m \equiv_{h^-}^{\boxplus} n$ and $m \sqsubset_{h^-}^{\boxplus} n$ are contradictory: one cannot have m, n simultaneously equivalent and strictly related according to the same relation. The condition $m \equiv_{h^-}^{\boxplus} n$ implies $m \sqsubseteq_{h^-}^{\boxplus} n$. Therefore $m \sqsubseteq_h^{\boxplus\boxminus} n$ iff $m \equiv_{h^-}^{\boxplus} n$ and $m \sqsubseteq_h n$. Recall now that $m \equiv_{h^-}^{\boxplus} n$ iff $m \equiv_{h^-} n$ (lemma 91). Then we have $m \sqsubseteq_h^{\boxplus\boxminus} n$ iff $m \equiv_{h^-} n$ and $m \sqsubseteq_h n$ iff $m \sqsubseteq_h^{\boxminus} n$.
  - From definition 88 one has $m \sqsubseteq_h^{\boxminus\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxminus\boxplus} n$ or ($m \sqsubseteq_h^{\boxminus} n$ and $m \sqsubseteq_{h^-}^{\boxminus\boxplus} n$), where $\sqsubseteq_{h^-}^{\boxminus\boxplus}$ denotes the intersection of the relations $\sqsubseteq_{h'}^{\boxminus\boxplus}$ for $h' \prec h$ and $\sqsubset_{h^-}^{\boxminus\boxplus}$ the corresponding strict relation.

    By the induction hypothesis this translates to $m \sqsubseteq_h^{\boxminus\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h^{\boxminus} n$ and $m \sqsubseteq_{h^-}^{\boxplus} n$). Note that $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h^{\boxminus} n$ and $m \sqsubseteq_{h^-}^{\boxplus} n$) is equivalent to $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h^{\boxminus} n$ and $m \equiv_{h^-}^{\boxplus} n$).

    The relation $m \sqsubseteq_h^{\boxminus} n$ is defined by $m \sqsubseteq_h n$ and $m \equiv_{h^-} n$. Substituting above we obtain $m \sqsubseteq_h^{\boxminus\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h n$ and $m \equiv_{h^-} n$ and $m \equiv_{h^-}^{\boxplus} n$). Recall that $m \equiv_{h^-} n$ iff $m \equiv_{h^-}^{\boxplus} n$ (lemma 91) and therefore $m \sqsubseteq_h^{\boxminus\boxplus} n$ iff either $m \sqsubset_{h^-}^{\boxplus} n$ or ($m \sqsubseteq_h n$ and $m \equiv_{h^-}^{\boxplus} n$) iff $m \sqsubset_h^{\boxplus} n$. ✓

2.1.4 Relations between Semantics

We now compare the three semantics of a specification S: the hierarchy of local preferences $S^{\odot}$, the hierarchy of lexicographic preferences $S^{\oplus}$ and the hierarchy of differential preferences $S^{\ominus}$. The operators $\boxplus$ and $\boxminus$ are the key to this comparison:

they assign to any of these semantics the hierarchy of lexicographic preferences and the hierarchy of differential preferences respectively. We conclude that $S^{\oplus}$ and $S^{\ominus}$ are equivalent since the operators $\boxplus$ and $\boxminus$ translate one into the other without loosing information. The hierarchy of local preferences $S^{\odot}$ is not as abstract as these two: specifications having different hierarchies of local preferences may have the same hierarchy of lexicographic preferences (and hierarchy of differential preferences). The reverse is not true.

We begin by showing (as it is expected) that $S^{\oplus}$ coincides with the result of the operation $\boxplus$ on the hierarchy of local preferences of S. Similarly $S^{\ominus}$ coincides with the result of the operation $\boxminus$ on $S^{\odot}$.

Lemma 93 Let S be a hierarchic specification. Then

1. $S^{\ominus} = (S^{\odot})^{\boxminus}$ and
2. $S^{\oplus} = (S^{\odot})^{\boxplus}$.

Proof It is obvious that $S^{\oplus}$, $S^{\odot}$, $(S^{\odot})^{\boxplus}$ have the same signature (that of S), and $(S^{\odot})^{\boxminus}$ have the same class of interpretation structures (the models of the axioms in S) and the same partial order $(H, \preceq) = \mathsf{po}(S)$. We have to show that for each level $h \in H$ interpretation structures $m, n \in \mathsf{ax}(S)^{\bullet}$ are related according to $S^{\oplus}$ iff they are related according to $S^{\odot\boxplus}$ and m, n are related according to $S^{\ominus}$ iff they are related according to $S^{\odot\boxminus}$. We begin with the proof of $S^{\ominus} = (S^{\odot})^{\boxminus}$.

1. Let $\sqsubseteq_h^{\odot}$ denote the relation $\mathsf{rl}(S^{\odot}, h)$ associated by $S^{\odot}$ to the level h and $\equiv_{h^-}^{\odot}$ the equivalence defined by $m \equiv_{h^-}^{\odot} n$ iff for all $h' \prec h$, $m \equiv_{h'}^{\odot} n$. The pre-order assigned by $(S^{\odot})^{\boxminus}$ to level h is precisely defined by $m \sqsubseteq_h^{\odot} n$ and $m \equiv_{h^-}^{\odot} n$ (see definition 90). It is obvious from definition 81 of $S^{\odot}$ that $m \sqsubseteq_h^{\odot} n$ iff $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ and that $m \equiv_{h^-}^{\odot} n$ iff $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for all $h' \prec h$. The pre-order defined by $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ and for all $h' \prec h$, $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ is the one assigned to level h by $S^{\ominus}$ (see definition 85). ✓
2. The proof of $S^{\oplus} = (S^{\odot})^{\boxplus}$ is by well-founded induction in the structure of the well founded partial order $(H, \preceq)$. Let $\sqsubseteq_h^{\odot} = \mathsf{rl}(S^{\odot}, h)$ be the relation associated with an arbitrary level $h \in H$ by $S^{\odot}$, $\sqsubseteq_h^{\oplus} = \mathsf{rl}(S^{\oplus}, h)$ be the relation associated with h by $S^{\oplus}$ and $\sqsubseteq_h^{\odot\boxplus} = \mathsf{rl}((S^{\odot})^{\boxplus}, h)$ be the relation associated with h by $(S^{\odot})^{\boxplus}$. Also let $\sqsubseteq_{h^-}^{\odot\boxplus} = \bigcap_{\{h' : h' \prec h\}} \sqsubseteq_{h'}^{\odot\boxplus}$ denote the intersection of the lexicographic relations $\sqsubseteq_{h'}^{\odot\boxplus}$ of the levels strictly under h and $\sqsubset_{h^-}^{\odot\boxplus}$ the corresponding strict relation.

   We now prove that $\sqsubseteq_h^{\oplus} = \sqsubseteq_h^{\odot\boxplus}$. Remember from the proof of lemma 87 that $\sqsubseteq_h^{\oplus}$ can be recursively defined as follows: $m \sqsubseteq_h^{\oplus} n$ iff for every priority level $h' \prec h$, $m \sqsubseteq_{h'}^{\oplus} n$ and 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 also that from definition 88 of $\boxplus$ that $m \sqsubseteq_h^{\odot\boxplus} n$ iff

- $m \sqsubseteq^{\odot}_{h} n$ if h is minimal,
- either $m \sqsubset^{\odot\boxplus}_{h^-} n$ or ($m \sqsubseteq^{\odot}_{h} n$ and $m \sqsubseteq^{\odot\boxplus}_{h^-} n$) if h is not minimal.

- Let h be minimal in $(H, \preceq)$. Since there is no $h' \preceq h$ except h itself from definition 83 of $S^{\oplus}$ we have $m \sqsubseteq^{\oplus}_{h} n$ iff $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$. This is equivalent to $m \sqsubseteq^{\odot}_{h} n$ from definition 81 of local semantics and is also equivalent to $m \sqsubseteq^{\odot\boxplus}_{h} n$ from definition 88 above.
- For non-minimal h the induction hypothesis is $\sqsubseteq^{\oplus}_{h'} = \sqsubseteq^{\odot\boxplus}_{h'}$ for h' strictly under h.
  - ⇒ Assume that $m \sqsubseteq^{\oplus}_{h} n$, i.e, for every priority level $h' \prec h$, $m \sqsubseteq^{\oplus}_{h'} n$ and if $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$ then there is $h' \prec h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$. There are two cases to check:
    - ∗ $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$. This is equivalent to $m \sqsubseteq^{\odot}_{h} n$ by definition 81 of local hierarchic semantics. Therefore $m \sqsubseteq^{\oplus}_{h} n$ becomes equivalent to: for every priority level $h' \prec h$, $m \sqsubseteq^{\oplus}_{h'} n$ and $m \sqsubseteq^{\odot}_{h} n$. Using the induction hypothesis this is: for every priority level $h' \prec h$, $m \sqsubseteq^{\odot\boxplus}_{h'} n$ and $m \sqsubseteq^{\odot}_{h} n$ which in turn is equivalent to $m \sqsubseteq^{\odot\boxplus}_{h^-} n$ and $m \sqsubseteq^{\odot}_{h} n$. This last condition implies $m \sqsubseteq^{\odot\boxplus}_{h} n$ (see definition 88).
    - ∗ $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$ and there is $h' \prec h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$. In this way the relation $\sqsubseteq^{\odot}_{h'}$ associated by the local hierarchic semantics to the level h' strictly prefers n to m (since n strictly satisfies more defaults). Recalling lemma 87 m, n cannot be equivalent according to $\sqsubseteq^{\oplus}_{h'} = \sqsubseteq^{\odot\boxplus}_{h'}$ since n strictly satisfies more defaults than m (and not the same). Moreover, in the same way as the case above, $m \sqsubseteq^{\odot\boxplus}_{h^-} n$. The two conditions imply that n is strictly preferred to m according to $\sqsubseteq^{\odot\boxplus}_{h^-}$. This last condition implies $m \sqsubseteq^{\odot\boxplus}_{h} n$ (see definition 88).
  - ⇐ Assume that $m \sqsubseteq^{\odot\boxplus}_{h} n$. This means that either $m \sqsubset^{\odot\boxplus}_{h^-} n$ or ($m \sqsubseteq^{\odot}_{h} n$ and $m \sqsubseteq^{\odot\boxplus}_{h^-} n$). In both cases $m \sqsubseteq^{\odot\boxplus}_{h^-} n$ which implies from definition 88 of $\boxplus$ that $m \sqsubseteq^{\odot\boxplus}_{h'} n$ for all $h' \prec h$. By the induction hypothesis we have $m \sqsubseteq^{\oplus}_{h'} n$ for all $h' \prec h$. Recalling the recursive definition of $\sqsubseteq^{\oplus}_{h}$ we only have to conclude in both cases that if $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$ then there is $h' \prec h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$.
    - ∗ Assume that $m \sqsubset^{\odot\boxplus}_{h^-} n$. Then there is $h' \prec h$ such that $m \sqsubset^{\odot\boxplus}_{h'} n$. Recalling lemma 87 this implies that there exists $h'' \preceq h' \prec h$ such that $\mathsf{df}(S,h'')(m) \subset \mathsf{df}(S,h'')(n)$.
    - ∗ Assume that $m \sqsubseteq^{\odot}_{h} n$ and $m \sqsubseteq^{\odot\boxplus}_{h^-} n$. This means that $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$. ✓

We conclude the comparison of the structured semantics in the following theorem. It states that $S^{\oplus}$ and $S^{\ominus}$ are equivalent semantics of S in the following sense. Another specification S' has the same hierarchy of lexicographic preference as S

iff it has the same hierarchy of differential preferences as S. Moreover specifications having the same hierarchy of local preferences have the same hierarchies of lexicographic and differential preferences.

Theorem 94 Let S, S' be hierarchic specifications. Then

1. $S^{\oplus\boxminus} = S^{\ominus}$ and $S^{\ominus\boxplus} = S^{\oplus}$,
2. $S^{\oplus} = S'^{\oplus}$ iff $S^{\ominus} = S'^{\ominus}$.
3. If $S^{\odot} = S'^{\odot}$ then $S^{\oplus} = S'^{\oplus}$ and $S^{\ominus} = S'^{\ominus}$

Proof Recall from lemma 93 that $S^{\oplus} = S^{\odot\boxplus}$ and $S^{\ominus} = S^{\odot\boxminus}$.

1. Therefore $S^{\oplus\boxminus} = S^{\odot\boxplus\boxminus} = S^{\odot\boxminus}$, i.e. $S^{\ominus}$. The last equality results from the properties of the operators $\boxplus$ and $\boxminus$ presented in theorem 92. The proof that $S^{\ominus\boxplus} = S^{\oplus}$ is formally identical to this one.
2. Trivial from the property above: if $S^{\oplus} = S'^{\oplus}$ then $S^{\oplus\boxminus} = S'^{\oplus\boxminus}$ which is equivalent to $S^{\ominus} = S'^{\ominus}$. The other direction is formally identical to this one.
3. Obvious from the fact that that $S^{\oplus} = S^{\odot\boxplus}$ and $S^{\ominus} = S^{\odot\boxminus}$. ✓

We know from item 3 in the theorem above that specifications having the same hierarchy of local preferences have the same hierarchy of lexicographic preferences (and hierarchy of differential preferences). We now see, by means of an example, the reverse is not true. In general specifications having the same hierarchy of lexicographic preferences (and hierarchy of differential preferences) do not have the same hierarchy of local preferences.

Example 95 Consider the two hierarchic specifications from propositional logic, without axioms and with a single default with priorities as displayed:

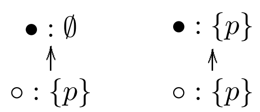

Figure 2.10: p

It is straightforward to check that both specifications have the same hierarchy of lexicographic preferences (and hierarchy of differential preferences). Intuitively this means that adding a default from a lower level (better) to a higher level does not change the meaning of the specification. However the two hierarchic specifications have different hierarchies of local preferences. To see this note that the preference associated with the default p (and no axioms) is different from the preference associated with the specification with the empty sets of defaults and axioms. △

2.2 Lexicographic Preference Revisited

The lexicographic preference of a hierarchic specification S is the structure needed to define the consequences of S (see section 2.4). We show that the structured semantics of S contains enough information to derive this preference in section 2.2.1. In this way we are sure that the semantics chosen for composition, namely the hierarchy of differential preferences of S, contains the needed logical information about S. In section 2.2.2 we show that, under some conditions including finiteness, a flat specification can be found with precisely the same lexicographic preference as a given hierarchic specification S.

2.2.1 Structured Semantics

Our concern is now to show that the lexicographic preference of a specification can be derived from any of the presented structured semantics. For that purpose we define the lexicographic combination of pre-orders. This pre-order is the semantic concept corresponding to the lexicographic preference of a specification. It has been first proposed by Lifschitz in the context of circumscription ([60]) and later generalized by Grosof ([52]) to any preferential logic. Andréka, Ryan and Schobbens show in [2] that this combination of relations organized by priority is canonical in the sense that it is the only combination of those relations satisfying certain conditions.

Definition 96 The *lexicographic combination* of the pre-orders organized by priority in the hierarchy of pre-orders $\mathcal{H}$, denoted by $\mathsf{lex}^{\square}(\mathcal{H})$ is the pre-order with:

1. the same signature as $\mathcal{H}$, $\mathsf{sg}(\mathsf{lex}^{\square}(\mathcal{H})) = \mathsf{sg}(\mathcal{H})$,
2. the class of interpretation structures from $\mathcal{H}$ as class of interpretation structures, $|\mathsf{lex}^{\square}(\mathcal{H})| = |\mathcal{H}|$ and
3. the relation $\sqsubseteq^{\square} = \mathsf{rl}(\mathsf{lex}^{\square}(\mathcal{H})) \subseteq |\mathsf{lex}^{\square}(\mathcal{H})| \times |\mathsf{lex}^{\square}(\mathcal{H})|$ defined as follows. Let $\sqsubseteq_h$ denote the relation $\mathsf{rl}(\mathcal{H}, h)$ assigned by $\mathcal{H}$ to the level h and let $(H, \preceq) = \mathsf{po}(\mathcal{H})$ be the partial order of priority from $\mathcal{H}$. Then $m \sqsubseteq^{\square} n$ iff for every priority level $h \in H$ either $m \sqsubseteq_h n$ or there is $h' \prec h$ and $m \sqsubset_{h'} n$. ▪

The lexicographic combination $\sqsubseteq^{\square}$ (of pre-orders) is a pre-order. The proof of this fact is formally identical to the proof of the corresponding property of each of the lexicographic combinations from $\boxplus$ (in definition 88).

We now see that the lexicographic combination of the (hierarchy of) local preferences of S is precisely the lexicographic preference of S.

Theorem 97 Let S be a hierarchic specification. Then the lexicographic preference of S is the lexicographic combination of the hierarchy of local preferences associated with S: $\mathsf{lex}^\circ(S) = \mathsf{lex}^\square(S^\odot)$.

Proof It is obvious that $\mathsf{lex}^\circ(S)$ has the same signature and class of interpretation structures as $\mathsf{lex}^\square(S^\odot)$. Let $(H, \preceq) = \mathsf{po}(S^\odot)$ be the partial order on the priority levels from $S^\odot$ (and S) and $\sqsubseteq_h^\odot = \mathsf{rl}(S^\odot, h)$ be the relation associated with an arbitrary level $h \in H$ by $S^\odot$.

We have to show that models $m, n \in \mathsf{ax}(S)^\bullet$ are related according to $\mathsf{lex}^\circ(S)$ iff they are related according to $\mathsf{lex}^\square(S^\odot)$.

From definition 76 of lexicographic preference $m, n \in \mathsf{ax}(S)^\bullet$ are related according to $\mathsf{lex}^\circ(S)$ when it is the case that for every level h, if $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$ then there exists $h' \prec h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$.

From definition 81 of hierarchy of local preferences $m \sqsubseteq_h^\odot n$ iff $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$. Therefore $m, n \in \mathsf{ax}(S)^\bullet$ are related according to $\mathsf{lex}^\circ(S)$ when it is the case that for every level h, if $m \not\sqsubseteq_h^\odot n$ then there exists $h' \prec h$ with $m \sqsubset_h^\odot n$. This means precisely (definition 96 of lexicographic combination) that m, n are related according to $\mathsf{lex}^\square(S^\odot)$. ✓

We now display the fact that $\mathsf{lex}^\circ(S)$ can also be derived from the hierarchy of lexicographic preferences $S^\oplus$ (and the hierarchy of differential preferences $S^\ominus$). Moreover the operation that yields $\mathsf{lex}^\circ(S)$ from $S^\oplus$ and from $S^\ominus$ is precisely the same that yields $\mathsf{lex}^\circ(S)$ from $S^\odot$: it is the lexicographic combination of the preferences in each of these hierarchies of pre-orders.

In order to show this we firstly express the lexicographic combination of the pre-orders from a hierarchy $\mathcal{H}$ as the intersection of the lexicographic preferences associated with $\mathcal{H}^\boxplus$, the hierarchy of lexicographic preferences associated with $\mathcal{H}$. In fact $\mathcal{H}^\boxplus$ assigns to each priority level h the lexicographic combination associated with the substructure of the levels under h and h. Therefore the lexicographic combination of $\mathcal{H}$ is the relation occurring at the existing or imagined "top" (least important) level in $\mathcal{H}^\boxplus$. This is the intersection of all partial lexicographic combinations occurring in $\mathcal{H}^\boxplus$.

Lemma 98 Let $\mathcal{H}$ be a hierarchy of pre-orders, $\mathcal{H}^\boxplus$ its hierarchic lexicographic semantics and $(H, \preceq) = \mathsf{po}(\mathcal{H}) = \mathsf{po}(\mathcal{H}^\boxplus)$ their partial order of priority. Then the lexicographic combination of $\mathcal{H}$, $\mathsf{lex}^\square(\mathcal{H})$ is $\mathsf{lex}^\square(\mathcal{H}) = \cap(\mathcal{H}^\boxplus)$ where $\cap(\mathcal{H}^\boxplus)$ denotes the pre-order with:

- $\mathsf{sg}(\cap(\mathcal{H}^\boxplus)) = \mathsf{sg}(\mathcal{H})$,
- the class of interpretation structures from $\mathcal{H}$ as class of interpretation structures, $|\cap(\mathcal{H}^\boxplus)| = |\mathcal{H}|$ and

- the relation $\sqsubseteq_H^{\boxplus} = \mathsf{rl}(\cap(\mathcal{H}^{\boxplus})) \subseteq |\cap(\mathcal{H}^{\boxplus})| \times |\cap(\mathcal{H}^{\boxplus})|$ where $\sqsubseteq_H^{\boxplus}$ is the intersection of all the lexicographic preferences $\sqsubseteq_h^{\boxplus}$, $h \in H$, from $\mathcal{H}^{\boxplus}$.

Moreover the lexicographic preference $\mathsf{lex}^{\circ}(S)$ of a hierarchic specification S is $\mathsf{lex}^{\circ}(S) = \cap(S^{\oplus})$ i.e. $\mathsf{lex}^{\circ}(S)$ is the intersection of the lexicographic preferences occurring in $S^{\oplus}$, the hierarchy of lexicographic preferences associated with S.

Proof Equality of signatures and classes of interpretation structures from $\cap(\mathcal{H}^{\boxplus})$ and $\mathsf{lex}^{\square}(\mathcal{H})$ are trivial. We have to show that given $m, n \in |\mathcal{H}|$ they are related by $\sqsubseteq^{\square} = \mathsf{rl}(\mathsf{lex}^{\square}(\mathcal{H}))$ iff they are related by $\sqsubseteq_H^{\boxplus}$. Let $\sqsubseteq_h = \mathsf{rl}(\mathcal{H}, h)$ be the relation associated to $\mathcal{H}$ to level h.

We have noted in lemma 89 that $m \sqsubseteq_h^{\boxplus} n$ iff for every $h' \preceq h$ (including h) either $m \sqsubseteq_{h'} n$ or there is $h'' \prec h$ with $m \sqsubset_{h''} n$. If m, n are related according to all $\sqsubseteq_h^{\boxplus}$, $h \in H$ then for all $h' \in H$ either $m \sqsubseteq_{h'} n$ or there is $h'' \prec h$ with $m \sqsubset_{h''} n$. The last condition is the definition (definition 96) of $m \sqsubseteq^{\square} n$. On the other hand if for all $h' \in H$ either $m \sqsubseteq_{h'} n$ or there is $h'' \prec h$ with $m \sqsubset_{h''} n$ then in particular this property holds for all $h' \preceq h$, for arbitrary h.

Finally we have to show that the lexicographic preference $\mathsf{lex}^{\circ}(S)$ of a hierarchic specification S is $\mathsf{lex}^{\circ}(S) = \cap(S^{\oplus})$. This is trivial recalling from theorem 97 that $\mathsf{lex}^{\circ}(S) = \mathsf{lex}^{\square}(S^{\odot})$. Therefore $\mathsf{lex}^{\square}(S^{\odot}) = \cap(S^{\odot\boxplus})$. But we know from lemma 93 that $S^{\odot\boxplus}$ is $S^{\oplus}$. ✓

We can now conclude that the lexicographic preference $\mathsf{lex}^{\circ}(S)$ of a hierarchic specification S is the lexicographic combination of any of $S^{\odot}$, $S^{\oplus}$ and $S^{\ominus}$.

Theorem 99 Let S be a hierarchic specification. Then:

$$\mathsf{lex}^{\circ}(S) = \mathsf{lex}^{\square}(S^{\odot}) = \mathsf{lex}^{\square}(S^{\oplus}) = \mathsf{lex}^{\square}(S^{\ominus}).$$

Proof Note firstly that $\mathsf{lex}^{\square}(\mathcal{H}) = \mathsf{lex}^{\square}(\mathcal{H}^{\boxplus})$ since $(\mathcal{H}^{\boxplus})^{\boxplus} = \mathcal{H}^{\boxplus}$ from theorem 92 and $\mathsf{lex}^{\square}(\mathcal{H}^{\boxplus}) = \cap(\mathcal{H}^{\boxplus\boxplus}) = \cap(\mathcal{H}^{\boxplus}) = \mathsf{lex}^{\square}(\mathcal{H})$ from lemma 98.

Therefore $\mathsf{lex}^{\square}(S^{\odot}) = \mathsf{lex}^{\square}(S^{\odot\boxplus}) = \mathsf{lex}^{\square}(S^{\oplus})$ recalling from lemma 93 that $S^{\oplus} = S^{\odot\boxplus}$. Since $S^{\ominus} = S^{\odot\boxminus}$ (lemma 93) and from theorem 92 $S^{\ominus\boxplus} = S^{\odot\boxminus\boxplus} = S^{\odot\boxplus} = S^{\oplus}$ we have $\mathsf{lex}^{\square}(S^{\ominus}) = \mathsf{lex}^{\square}(S^{\ominus\boxplus}) = \mathsf{lex}^{\square}(S^{\oplus})$. Finally recall that $\mathsf{lex}^{\circ}(S) = \mathsf{lex}^{\square}(S^{\odot})$ from theorem 97. ✓

We end this section with an example illustrating both the hierarchy of lexicographic preferences of a specification S and the corresponding lexicographic preference. We recall (see the discussion in 2.1.2) that the more structured semantics (hierarchies of lexicographic and differential preferences) are needed for composition of specifications. However, the logical content of a specification (its consequences) is derived (see next section 2.4) from the corresponding lexicographic preference. This preference displays the best possible agreement of the defaults in the specification given their priorities (see Andréka, Ryan and Schobbens [2]). Lemma 98 tells us that we can find the lexicographic preference of S by constructing the lexicographic preferences of subparts of S: for each priority level h one builds the lexicographic preference associated with the substructure of S obtained by restricting S to the levels h and under h. The lexicographic preference of S is the intersection of these preferences.

Example 100 We now illustrate a hierarchy of lexicographic preferences and the corresponding lexicographic preference. We will simplify the specification

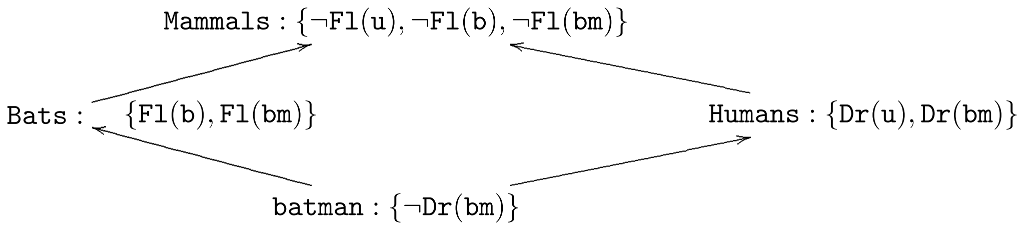

Figure 2.11: Some Mammals

MAMMALS (recall example 75) by considering only the following mammals: bm (Batman), a typical bat b (not bm) and a typical human u (also not bm). The specification will have axioms ax(MAMMALS) = {Bat(bm), Bat(b), Hum(u), Hum(bm)}, stating that bm is both a human and a bat, b is a bat and u a human. The defaults are organized by priority as displayed in figure 2.11.

We now assign to each priority level the corresponding lexicographic preference. Note that the models participating in preference relations that follow are the models of the axioms {Bat(bm), Bat(b), Hum(u), Hum(bm)}. This information is not represented in the diagrams. We also do not represent reflexive pairs of equally preferred interpretation structures and omit relations resulting from transitivity.

The lexicographic preference assigned to the level batman only has to consider the default of that level, ¬Dr(bm). It is represented in the following diagram. The models of the only default ¬Dr(bm) form an equivalence class. Interpretation structures that do not satisfy ¬Dr(bm) are strictly less preferred than the former (and also constitute an equivalence class).

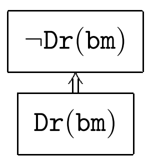

Figure 2.12: Lexicographic Preference at Level batman

The lexicographic preference assigned to the level Bats has to take into account the defaults at that level and those at the level batman (under it).

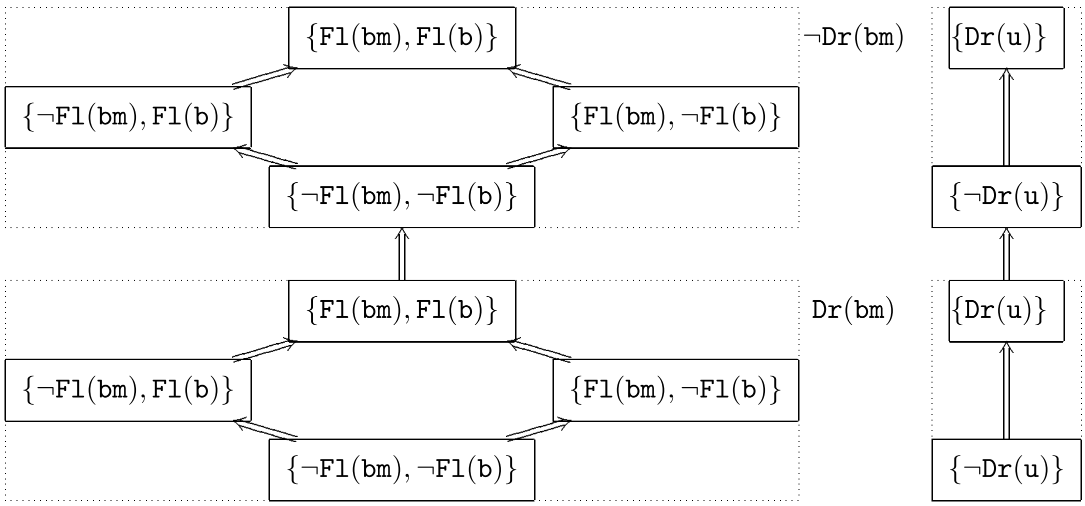

Figure 2.13: Lexicographic Preference at Levels `Bats` (left) and `Humans` (right)

Since ¬`Dr(bm)` is of lower (more important) priority the lexicographic preference is obtained by refining with the defaults from level `Bats` the models that are *equally good* at satisfying ¬`Dr(bm)`. This means that the interpretation structures in the equivalence class of the models of ¬`Dr(bm)` are now compared according to satisfaction of the defaults `Fl(b)` and `Fl(bm)` (the defaults at level `Bats`). The same for the models of `Dr(bm)`. It is important to note that any model of ¬`Dr(bm)` is preferred to any model of `Dr(bm)` since the former satisfy a "better" default.

A similar situation occurs with the preference to be associated with the level `Humans` and the defaults `Dr(u)` and `Dr(bm)`. Note that the default `Dr(bm)` has been overridden: it has no effect. It should be used to refine the relation of preference within the models of ¬`Dr(bm)` and to refine the relation of preference within the models of `Dr(bm)`. In the first case no model of ¬`Dr(bm)` satisfies `Dr(bm)` so they cannot be separated by satisfaction of `Dr(bm)`. In the second the models of `Dr(bm)` already satisfy `Dr(bm)` so they remain equivalent.

Both preferences are represented in figure 2.13. Note that we omit the formulas ¬`Dr(bm)` and `Dr(bm)`. All interpretation structures within the same outer box labeled with ¬`Dr(bm)` or `Dr(bm)` satisfy ¬`Dr(bm)` or `Dr(bm)` and were equivalent at lower levels.

Finally the preference at level `Mammals` has to consider all defaults under it. The final preference is obtained by refining the preference relation associated with the defaults at levels strictly under `Mammals` with the defaults at `Mammals`.

Before presenting this preference we display the preference corresponding to the specification without the level `Mammals`. This preference, shown in figure 2.14, is the intersection of the ones displayed up to now. (Since the preference at level batman is contained in those at levels Bats and Humans we do not have to consider it).

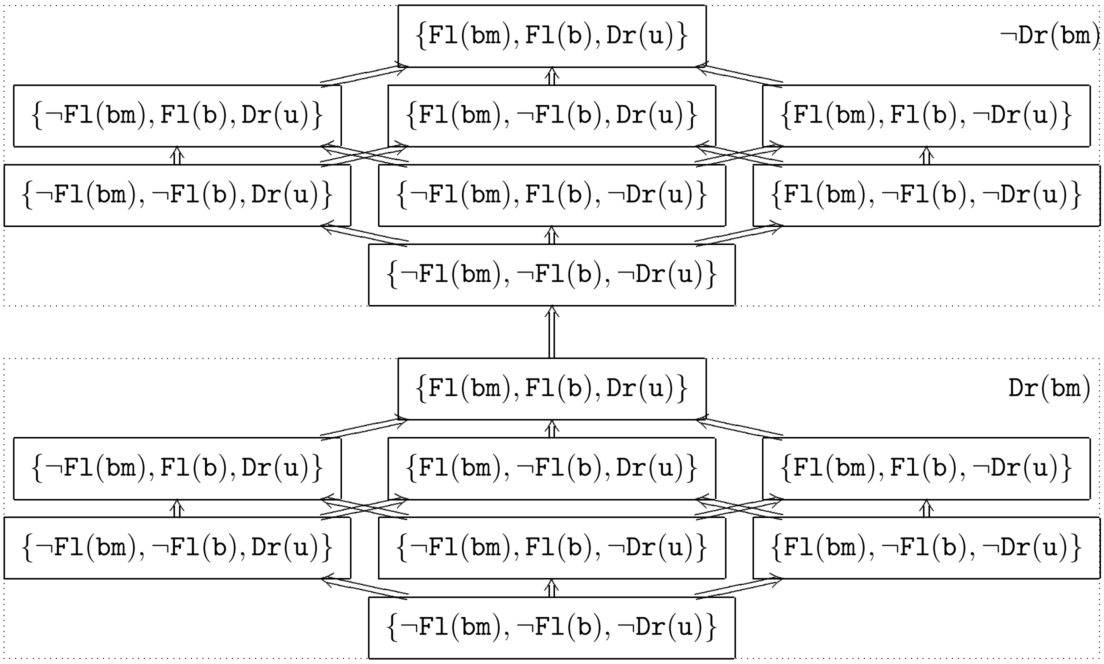

Figure 2.14: Intersection of the Preferences at Levels Bats and Humans

Note that this intersection corresponds to view the defaults Fl(b) and Fl(bm) and Dr(u) and Dr(bm) from levels Bats and Humans as of the same priority, but of higher priority than ¬Dr(bm). For the reasons already presented only the defaults Fl(b), Fl(bm) and Dr(u) are "effective".

The preference to be assigned to the level of Mammals is obtained by refining each of the equivalence classes of the preference above with the defaults at Mammals. These are ¬Fl(bm), ¬Fl(b) and ¬Fl(u). The defaults ¬Fl(bm), ¬Fl(b) have been overridden and have no effect (any of the previous equivalence classes consists of interpretation structures that cannot be separated by satisfaction of ¬Fl(bm) or ¬Fl(b)). Therefore the only "effective" default is ¬Fl(u).

The lexicographic preference associated with this version of the specification MAMMALS is the intersection of the ones presented up to now. But, since this specification has a "top" level (the level Mammals) that intersection coincides with the preference associated with this level. Therefore the lexicographic preference associated with MAMMALS is the one displayed in figure 2.15 (next page). △

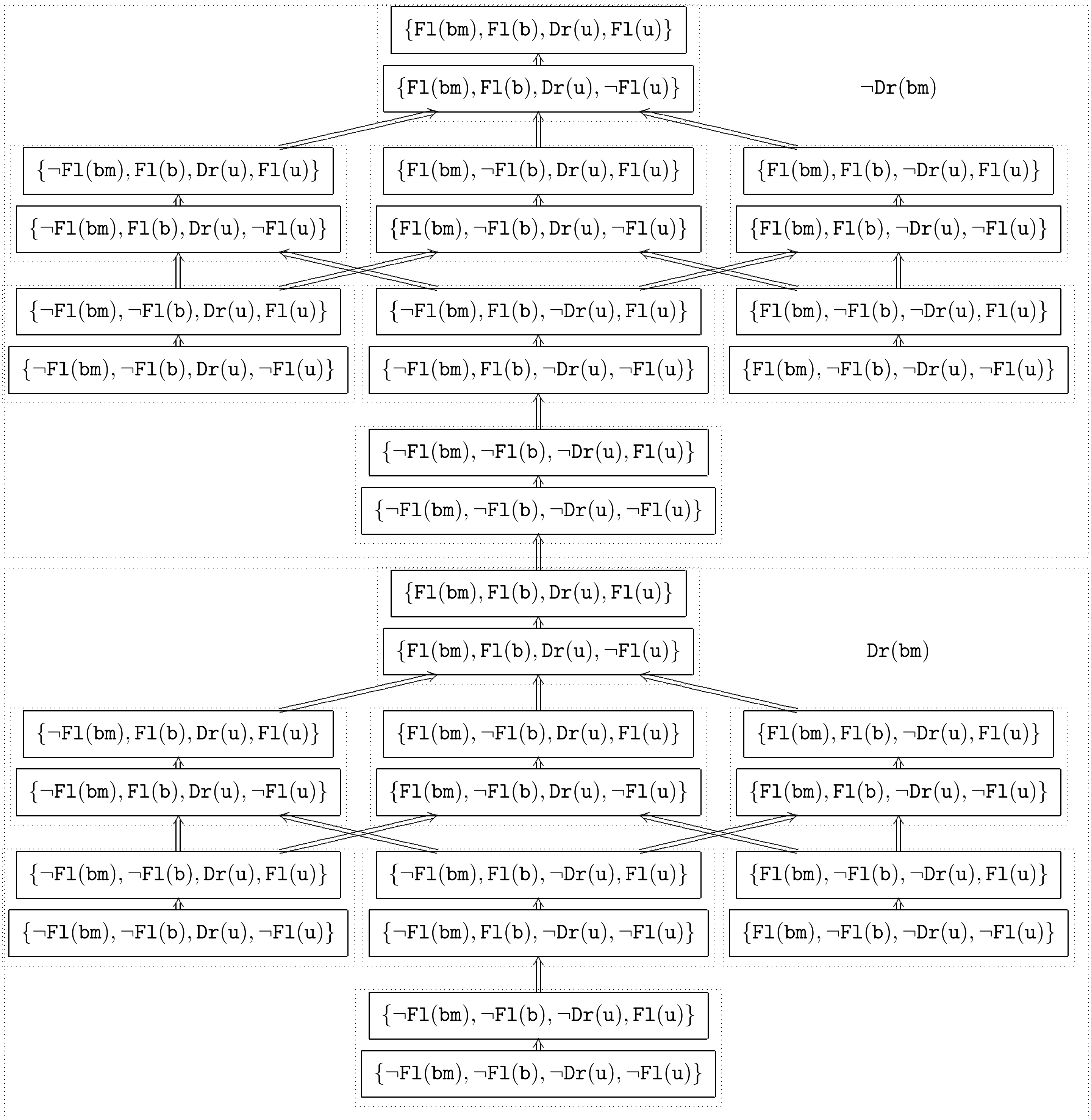

Figure 2.15: Lexicographic Preference at Level Mammals

2.2.2 Specifications

In this section we are concerned with expressing the lexicographic preference of a hierarchic specification S with a (flat) specification.

Recalling the results in section 1.2.4 of chapter 1 this is possible when the underlying institution has conjunctions and disjunctions and the pre-order to be expressed has more relations than a known preference (induced by a finite specification). It is straightforward from definition 76 that the lexicographic preference of S has *more relations* of preference than those induced by the flat specification having as defaults all the defaults from S (not related by priority). Therefore in such an institution the lexicographic semantics of a finite hierarchic specification S can always be expressed as the preference relation induced by a specification. In the following, we provide such a flat specification for a given *finite* hierarchic specification S.

Definition 101 Let S be a Σ-hierarchic specification and $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority. The specification S is said to be finite when the set $\mathsf{ax}(S)$ of its axioms, the set H of priority levels and, for each $h \in H$, the set $\mathsf{df}(S, h)$ of the defaults from S at level h are finite. ▪

We now provide the specification "equivalent" to a hierarchic specification S firstly by providing for each level h a specification equivalent to the lexicographic preference associated with the substructure of S under level h. This corresponds to give a specification that induces each preference from the hierarchy of lexicographic preferences of S. The resulting specification corresponds to the union of the partial specifications.

For this purpose we associate with each level h a set of *flattening* defaults. This means that the (flat) specification having these defaults (and the axioms of the specification) induces precisely the lexicographic preference of S at level h. These flattening defaults can be motivated as follows. Let h be an arbitrary level. If h is minimal its lexicographic preference coincides with the preference induced by the defaults at h. These are the flattening defaults for this case. If h is not minimal note that the global effect of the defaults of levels $h' \prec h$ is already encoded in the sets of flattening defaults for these levels. Let D be the union of those sets. Then D contains the information for the levels $h' \prec h$ and all defaults from D are flattening defaults from level h. New defaults are introduced in this level by the interplay between the defaults d at h and those at better levels. This interplay can be described as follows. If a model m of the axioms satisfies d then another n can be preferred to m even if n does not satisfy d. It has, however, to satisfy a default d' better than d (and m must not satisfy this d'). Furthermore the model n has to satisfy all defaults from D satisfied by m, since this means precisely that n is preferred to m at levels $h' \prec h$. Let $D(m) \subseteq D$ be the set of these "better" defaults *satisfied by* m. We rewrite the previous considerations with "formulas": if $m \vDash d \wedge D(m)$ then $n \vDash \mathcal{D}$ where $\mathcal{D} = (d \wedge D(m)) \vee d'_1 \vee d'_2 \vee ... \vee d'_n$ and $d'_1, d'_2, ..., d'_n$ are defaults better that d. That is $d'_1, d'_2, ..., d'_n \in D$. Since these better defaults must not be satisfied by m we also have $d'_1, d'_2, ..., d'_n \notin D(m)$. Finally and for this reason we can rewrite $m \vDash d \wedge D(m)$ equivalently as $m \vDash \mathcal{D}$. Therefore if $m \vDash \mathcal{D}$ then $n \vDash \mathcal{D}$. This is the condition of $\mathcal{D}$ being a (flat) default at level h and suggests that the formulas $\mathcal{D} = (d \wedge D(m)) \vee d'_1 \vee d'_2 \vee ... \vee d'_n$ for all possible $D(m) \subseteq D$ are the defaults accounting for the lexicographic preference at level h.

Definition 102 Let S be a *finite* Σ-hierarchic specification of an institution $(\mathcal{I}, \mathsf{cj}, \mathsf{dj})$ having conjunctions and disjunctions. Let $(H, \preceq) = \mathsf{po}(S)$ be its partial order of priority.

For conventional commodity, given a finite set D of Σ-formulas let $\mathsf{cj}_\Sigma(D)$ (resp. $\mathsf{dj}_\Sigma(D)$) denote the conjunction (resp. disjunction) of some enumeration $d_1, ..., d_n$ of D. Moreover allow D to be *empty* in $\mathsf{cj}_\Sigma(d, D)$ and $\mathsf{dj}_\Sigma(d, D)$. In this case $\mathsf{cj}_\Sigma(d, D)$ and $\mathsf{dj}_\Sigma(d, D)$ denote the Σ-formula d.

- The *flattening* defaults at level h are inductively defined as follows:
  - if h is minimal then the flattening defaults at h coincide with the defaults from S at that level: $\mathsf{fldf}(S, h) = \mathsf{df}(S, h)$,
  - otherwise $\mathsf{fldf}(S, h) = \mathsf{fldf}(S, h^-) \cup$

    $$\{\mathsf{dj}_\Sigma\left(\mathsf{cj}_\Sigma(d, D), \mathsf{dj}_\Sigma(\mathsf{fldf}(S, h^-) \setminus D)\right) : d \in \mathsf{df}(S, h), D \subseteq \mathsf{fldf}(S, h^-)\},$$

    where $\mathsf{fldf}(S, h^-) = \cup_{\{h' \prec h\}} \mathsf{fldf}(S, h')$ denotes the union of the flattening defaults from level h' strictly under h,
- Let $\mathsf{flat}(S, h) = (\mathsf{ax}(S), \mathsf{fldf}(S, h))$ denote the specification having the axioms from S as axioms and the flattening defaults from level h as defaults.
- Let $\mathsf{flat}(S) = (\mathsf{ax}(S), \cup_{h \in H} \mathsf{fldf}(S, h))$ denote the specification having the axioms from S as axioms and the union of the flattening defaults from all levels from S as defaults. ■

That these flattening defaults indeed induce the lexicographic preference of a specification is formally stated in the next theorem.

Theorem 103 Let S be a finite hierarchic specification from an institution $(\mathcal{I}, \mathsf{cj}, \mathsf{dj})$ having conjunctions and disjunctions and $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority. Then $(|S^\oplus|, \mathsf{rl}(S^\oplus, h))$, the lexicographic preference at level h coincides with $\mathsf{flat}(S, h)^\star$ the preference induced by the specification $\mathsf{flat}(S, h)$.

Moreover $\mathsf{lex}^\circ(S) = \mathsf{flat}(S)^\star$, i.e. the lexicographic preference associated with S is the preference associated with the specification $\mathsf{flat}(S)$.

Proof Recall the definition of the lexicographic preference at level h, $\sqsubseteq_h^\oplus$: $m \sqsubseteq_h^\oplus n$ 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). This implies the following auxiliary results:

- $m \sqsubseteq_h^\oplus n$ iff $m \sqsubseteq_{h'}^\oplus n$ for $h' \prec h$ and if $\mathsf{df}(S,h')(m) \not\subseteq \mathsf{df}(S,h')(n)$ then there is $h' \prec h$ with $\mathsf{df}(S,h')(m) \subset \mathsf{df}(S,h')(n)$
- if $m \sqsubseteq_h^\oplus n$ and $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$ then there is $h' \prec h$ with $m \sqsubset_{h'}^\oplus n$ (strict).

The first property is obvious and has been referred to in the proof of lemma 87. To see the second assume that there no is $h' \prec h$ with $m \sqsubset_{h'}^\oplus n$. Therefore $m \equiv_{h'}^\oplus n$ for all $h' \prec h$ (since $m \sqsubseteq_{h'}^\oplus n$ and none of these is strict). From lemma 87 we have that $\mathsf{df}(S,h')(m) = \mathsf{df}(S,h')(n)$ for all $h' \prec h$. From this and $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$ we conclude that $m \not\sqsubseteq_h^\oplus n$, contradicting the hypothesis.

Let $\sqsubseteq_h^{\mathsf{fl}}$ be the preference associated with $\mathsf{flat}(S,h) = (\mathsf{ax}(S), \mathsf{fldf}(S,h))$, relating the models of $\mathsf{ax}(S)$ as follows: $m \sqsubseteq_h^{\mathsf{fl}} n$ iff whenever $m \vDash d$ then $n \vDash d$ for $d \in \mathsf{fldf}(S,h)$. We have to show that $m \sqsubseteq_h^\oplus n$ iff $m \sqsubseteq_h^{\mathsf{fl}} n$. The proof is by well-founded induction in $(H, \preceq)$.

- if h is minimal then $m \sqsubseteq_h^\oplus n$ iff $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$. This is equivalent to $m \sqsubseteq_h^{\mathsf{fl}} n$ since $\mathsf{fldf}(S,h) = \mathsf{df}(S,h)$,
- if h is not minimal we use the induction hypothesis that $\sqsubseteq_{h'}^\oplus = \sqsubseteq_{h'}^{\mathsf{fl}}$ for $h' \prec h$.
  - $\Rightarrow$ Assume that $m \sqsubseteq_h^\oplus n$. This implies $m \sqsubseteq_{h'}^\oplus n$ for $h' \prec h$ and from the induction hypothesis $m \sqsubseteq_{h'}^{\mathsf{fl}} n$. Therefore, for all $d' \in \mathsf{fldf}(S,h^-)$ if $m \vDash d'$ then $n \vDash d'$ since $d' \in \mathsf{fldf}(S,h')$ for some $h' \prec h$ and $m \sqsubseteq_{h'}^{\mathsf{fl}} n$.
    The important case corresponds to verify this property for the newly introduced defaults namely $d' = \mathsf{dj}_\Sigma(\mathsf{cj}_\Sigma(d,D), \mathsf{dj}_\Sigma(\mathsf{fldf}(S,h^-) \setminus D))$ with $d \in \mathsf{df}(S,h)$ and $D \subseteq \mathsf{fldf}(S,h^-)$.
    Assume that $m \vDash d'$. Then either $m \vDash \mathsf{cj}_\Sigma(d,D)$ or $m \vDash \mathsf{dj}_\Sigma(\mathsf{fldf}(S,h^-) \setminus D)$.
    - If $m \vDash \mathsf{dj}_\Sigma(\mathsf{fldf}(S,h^-) \setminus D)$ this means that there is a $d'' \in \mathsf{fldf}(S,h^-) \setminus D$ satisfied by m. This $d'' \in \mathsf{fldf}(S,h'')$ for some $h'' \prec h$ and therefore also $n \vDash d''$ since $m \sqsubseteq_{h''}^{\mathsf{fl}} n$. From this $n \vDash \mathsf{dj}_\Sigma(\mathsf{fldf}(S,h^-) \setminus D)$ and $n \vDash d' = \mathsf{dj}_\Sigma(\mathsf{cj}_\Sigma(d,D), \mathsf{dj}_\Sigma(\mathsf{fldf}(S,h^-) \setminus D))$.
    - Assume now that $m \vDash \mathsf{cj}_\Sigma(d,D)$. We conclude that $n \vDash d''$ for $d'' \in D \subseteq \mathsf{fldf}(S,h^-)$, for the same reasons as before. If $n \vDash d$ then $n \vDash \mathsf{cj}_\Sigma(d,D)$ and $n \vDash d'$ so we may assume that $n \nvDash d$. In this case $\mathsf{df}(S,h)(m) \not\subseteq \mathsf{df}(S,h)(n)$. We now use the auxiliary result referred in the beginning of the proof. Since $m \sqsubseteq_h^\oplus n$ there is $h' \prec h$ with $m \sqsubset_{h'}^\oplus n$

(strict). From the induction hypothesis also $m \sqsubset^{\mathsf{fl}}_{h'} n$ which implies that there is $d''' \in \mathsf{fldf}(S, h')$ with $n \vDash d'''$ and $m \nvDash d'''$. In this way $d''' \notin D$. Therefore $n \vDash \mathsf{dj}_\Sigma(\mathsf{fldf}(S, h^-) \setminus D)$ and $n \vDash d'$ as intended.

$\Leftarrow$ Assume now that $m \sqsubseteq^{\mathsf{fl}}_{h} n$. Therefore $m \sqsubseteq^{\mathsf{fl}}_{h'} n$ for $h' \prec h$ since the flattening defaults from h' are contained in those of h. Therefore $m \sqsubseteq^{\oplus}_{h'} n$ from the induction hypothesis. Now we have to show that if $\mathsf{df}(S, h)(m) \not\subseteq \mathsf{df}(S, h)(n)$ then there is $h' \prec h$ with $\mathsf{df}(S, h')(m) \subset \mathsf{df}(S, h')(n)$.

That $\mathsf{df}(S, h)(m) \not\subseteq \mathsf{df}(S, h)(n)$ means that there is a $d' \in \mathsf{df}(S, h)$ such that $m \vDash d'$ and $n \nvDash d'$. Consider the set $D = \{d'' : m \vDash d'' \text{ and } d'' \in \mathsf{fldf}(S, h^-)\}$. Then $m \vDash \mathsf{cj}_\Sigma(d', D)$ and also $m \vDash \mathsf{dj}_\Sigma(\mathsf{cj}_\Sigma(d', D), \mathsf{dj}_\Sigma(\mathsf{fldf}(S, h^-) \setminus D))$. Since $m \sqsubseteq^{\mathsf{fl}}_{h} n$ and $\mathsf{dj}_\Sigma(\mathsf{cj}_\Sigma(d', D), \mathsf{dj}_\Sigma(\mathsf{fldf}(S, h^-) \setminus D))$ is one of the flattening defaults at level h also $n \vDash \mathsf{dj}_\Sigma(\mathsf{cj}_\Sigma(d', D), \mathsf{dj}_\Sigma(\mathsf{fldf}(S, h^-) \setminus D))$. The interpretation structure n, however, does not satisfy $\mathsf{cj}_\Sigma(d', D)$ since it does not satisfy d'. Therefore $n \vDash \mathsf{dj}_\Sigma(\mathsf{fldf}(S, h^-) \setminus D)$, which means that $n \vDash d''$ with $d'' \in \mathsf{fldf}(S, h^-)$. Since $d'' \notin D$ and D is the set of formulas from $\mathsf{fldf}(S, h^-)$ satisfied by m we have that $m \nvDash d''$.

Now $d'' \in \mathsf{fldf}(S, h')$ for some $h' \prec h$ and $m \sqsubseteq^{\mathsf{fl}}_{h'} n$. This relation is strict since $n \vDash d''$ and $m \nvDash d''$. Therefore $m \sqsubset^{\mathsf{fl}}_{h'} n$ and $m \sqsubset^{\oplus}_{h'} n$. From lemma 87 since $m \sqsubset^{\oplus}_{h'} n$ there is $h'' \prec h'$ with $\mathsf{df}(S, h'')(m) \subset \mathsf{df}(S, h'')(n)$.

We have only to show that $\mathsf{lex}^\circ(S)$ is the preference of $\mathsf{flat}(S) = (\mathsf{ax}(S), \cup_{h \in H} \mathsf{fldf}(S, h))$. We have seen in lemma 98 that the lexicographic preference $\mathsf{lex}^\circ(S)$ is the intersection of the lexicographic preferences in each level h of $S^\oplus$. From the Galois connection for specifications we have that this intersection corresponds to the preference of the union of argument specifications:

$$\mathsf{flat}(S)^\star = (\Cup_{h \in H} (\mathsf{ax}(S), \mathsf{fldf}(S, h)))^\star = \Cap_{h \in H} (\mathsf{ax}(S), \mathsf{fldf}(S, h))^\star$$

and this in turn is $\Cap_{h \in H}(|S^\oplus|, \mathsf{rl}(S^\oplus, h)) = \mathsf{lex}^\circ(S)$. ✓

This translation is illustrated in the next example.

Example 104 We now illustrate the flat specification corresponding to the simplified version of MAMMALS presented in example 100. The flat specification FL-MAMMALS = flat(MAMMALS) corresponding to the hierarchic specification MAMMALS will have as axioms the axioms of MAMMALS:

$$\mathsf{ax}(\text{FL-MAMMALS}) = \mathsf{ax}(\text{MAMMALS}) = \{\mathtt{Bat(bm)}, \mathtt{Bat(b)}, \mathtt{Hum(u)}, \mathtt{Hum(bm)}\},$$

stating that `bm` is both a human and a bat, b is a bat and u a human. The defaults of FL-MAMMALS are the flattening defaults from MAMMALS from all levels. The flattening defaults at each level are now shown. Note that the corresponding specifications induce the lexicographic preferences at each level.

These preferences have been displayed in example 100. For the (minimal) level $\texttt{batman}$ the flattening defaults coincide with the defaults from MAMMALS at $\texttt{batman}$. This is simply $\{\neg\texttt{Dr}(\texttt{bm})\}$. For the level $\texttt{Bats}$ we have also $\neg\texttt{Dr}(\texttt{bm})$ plus four other formulas corresponding to take $d = \texttt{Fl}(\texttt{b})$ or $d = \texttt{Fl}(\texttt{bm})$ and D to be either $D = \{\neg\texttt{Dr}(\texttt{bm})\}$ or $D = \emptyset$ in $\mathsf{dj}_\Sigma\,(\mathsf{cj}_\Sigma(d, D), \mathsf{dj}_\Sigma(\{\neg\texttt{Dr}(\texttt{bm})\} \setminus \texttt{D}))$. The flattening defaults at level $\texttt{Bats}$ are:

1. $\neg\texttt{Dr}(\texttt{bm})$, included from level $\texttt{batman}$,
2. $\texttt{Fl}(\texttt{b}) \wedge \neg\texttt{Dr}(\texttt{bm})$ with $d = \texttt{Fl}(\texttt{b})$ and $D = \{\neg\texttt{Dr}(\texttt{bm})\}$,
3. $\texttt{Fl}(\texttt{b}) \vee \neg\texttt{Dr}(\texttt{bm})$ with $d = \texttt{Fl}(\texttt{b})$ and $D = \emptyset$,
4. $\texttt{Fl}(\texttt{bm}) \wedge \neg\texttt{Dr}(\texttt{bm})$ with $d = \texttt{Fl}(\texttt{bm})$ and $D = \{\neg\texttt{Dr}(\texttt{bm})\}$,
5. $\texttt{Fl}(\texttt{bm}) \vee \neg\texttt{Dr}(\texttt{bm})$ with $d = \texttt{Fl}(\texttt{bm})$ and $D = \emptyset$.

The level $\texttt{Humans}$ is similar. The defaults are $\texttt{Dr}(\texttt{u})$ and $\texttt{Dr}(\texttt{bm})$. For $d = \texttt{Dr}(\texttt{u})$ we have:

- $\texttt{Dr}(\texttt{u}) \wedge \neg\texttt{Dr}(\texttt{bm})$ with $D = \{\neg\texttt{Dr}(\texttt{bm})\}$,
- $\texttt{Dr}(\texttt{u}) \vee \neg\texttt{Dr}(\texttt{bm})$ with $D = \emptyset$.

For $d = \texttt{Dr}(\texttt{bm})$ the construction is formally similar. Note, however, that $\texttt{Dr}(\texttt{bm})$ has been overridden and therefore should not contribute for the overall meaning of the specification. This is in fact so since the two defaults obtained for this case are false and true (and these are implicit in any specification; therefore we omit them). The flattening defaults at level $\texttt{Humans}$ are:

1. $\neg\texttt{Dr}(\texttt{bm})$, included from level $\texttt{batman}$,
2. $\texttt{Dr}(\texttt{u}) \wedge \neg\texttt{Dr}(\texttt{bm})$ with $D = \{\neg\texttt{Dr}(\texttt{bm})\}$,
3. $\texttt{Dr}(\texttt{u}) \vee \neg\texttt{Dr}(\texttt{bm})$ with $D = \emptyset$.

Finally in the level $\texttt{Mammals}$ (and for the whole specification) any of the previous defaults are flattening defaults. Let

$$\begin{aligned}\texttt{underMammals} = \{\neg\texttt{Dr}(\texttt{bm}), \quad & \texttt{Fl}(\texttt{b}) \wedge \neg\texttt{Dr}(\texttt{bm}), \texttt{Fl}(\texttt{b}) \vee \neg\texttt{Dr}(\texttt{bm}),\\ & \texttt{Fl}(\texttt{bm}) \wedge \neg\texttt{Dr}(\texttt{bm}), \texttt{Fl}(\texttt{bm}) \vee \neg\texttt{Dr}(\texttt{bm}),\\ & \texttt{Dr}(\texttt{u}) \wedge \neg\texttt{Dr}(\texttt{bm}), \texttt{Dr}(\texttt{u}) \vee \neg\texttt{Dr}(\texttt{bm})\}.\end{aligned}$$

be the set of such defaults. The new ones are obtained by substituting $\neg\texttt{Fl}(\texttt{u})$, $\neg\texttt{Fl}(\texttt{b})$ and $\neg\texttt{Fl}(\texttt{bm})$ for d. The possibilities for D are all subsets of $\texttt{underMammals}$.

We simplify this task by noting the following. The cases where some $X \wedge \neg\mathtt{Dr(bm)} \in \mathtt{D}$ and some $Y \vee \neg\mathtt{Dr(bm)} \notin \mathtt{D}$ need not to be considered (X may be true and Y may be false). In fact both the conjunctive term and the disjunctive term of each new flattening default will have $\neg\mathtt{Dr(bm)}$. In this way the whole formula is equivalent to the disjunctive term. But this is a disjunction of formulas in `underMammals` that are known to be already flattening defaults at level `Mammals`. Therefore the new default is a default implicit from those and need not be made explicit (i.e. with or without these disjunctions the preference will be the same).

So we can concentrate in the sets D not satisfying the condition above. These are the sets that either have no formula of the form $X \wedge \neg\mathtt{Dr(bm)}$ (the subsets of $\Delta_1 = \{\mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)}, \mathtt{Fl(bm)} \vee \neg\mathtt{Dr(bm)}, \mathtt{Dr(u)} \vee \neg\mathtt{Dr(bm)}\}$) or have all formulas $Y \vee \neg\mathtt{Dr(bm)}$ (the supersets of $\Delta_1 \cup \{\neg\mathtt{Dr(bm)}\}$).

For the subsets of Δ_1 and $d = \neg\mathtt{Fl(u)}$ we have, after simplifying:

1. $\neg\mathtt{Fl(u)} \wedge \mathtt{Fl(b)} \wedge \mathtt{Fl(bm)} \wedge \mathtt{Dr(u)} \vee \neg\mathtt{Dr(bm)}$,
2. $\neg\mathtt{Fl(u)} \wedge \mathtt{Fl(b)} \wedge \mathtt{Fl(bm)} \vee (\mathtt{Dr(u)} \vee \neg\mathtt{Dr(bm)})$,
3. $\neg\mathtt{Fl(u)} \wedge \mathtt{Fl(b)} \wedge \mathtt{Dr(u)} \vee (\mathtt{Fl(bm)} \vee \neg\mathtt{Dr(bm)})$,
4. $\neg\mathtt{Fl(u)} \wedge \mathtt{Dr(u)} \wedge \mathtt{Fl(bm)} \vee (\mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)})$,
5. $\neg\mathtt{Fl(u)} \wedge \mathtt{Fl(b)} \vee (\mathtt{Fl(bm)} \vee \mathtt{Dr(u)} \vee \neg\mathtt{Dr(bm)})$,
6. $\neg\mathtt{Fl(u)} \wedge \mathtt{Dr(u)} \vee (\mathtt{Fl(b)} \vee \mathtt{Fl(bm)} \vee \neg\mathtt{Dr(bm)})$,
7. $\neg\mathtt{Fl(u)} \wedge \mathtt{Fl(bm)} \vee (\mathtt{Dr(u)} \vee \mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)})$,
8. $\neg\mathtt{Fl(u)} \vee (\mathtt{Fl(bm)} \vee \mathtt{Dr(u)} \vee \mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)})$.

The supersets of

$$\Delta_1 \cup \{\neg\mathtt{Dr(bm)}\} = \{\mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)}, \mathtt{Fl(bm)} \vee \neg\mathtt{Dr(bm)}, \mathtt{Dr(u)} \vee \neg\mathtt{Dr(bm)}, \neg\mathtt{Dr(bm)}\}$$

yield the following (simplified) flattening defaults. Note that the conjunction of $\Delta_1 \cup \{\neg\mathtt{Dr(bm)}\}$ is simply $\neg\mathtt{Dr(bm)}$.

1. $(\neg\mathtt{Fl(u)} \wedge \neg\mathtt{Dr(bm)}) \vee (\neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(bm)} \vee \neg\mathtt{Dr(bm)} \wedge \mathtt{Dr(u)})$,
2. $\neg\mathtt{Fl(u)} \wedge \neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(b)}) \vee (\neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(bm)} \vee \neg\mathtt{Dr(bm)} \wedge \mathtt{Dr(u)})$,
3. $(\neg\mathtt{Fl(u)} \wedge \neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(bm)}) \vee (\neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)} \wedge \mathtt{Dr(u)})$,
4. $(\neg\mathtt{Fl(u)} \wedge \neg\mathtt{Dr(bm)} \wedge \mathtt{Dr(u)}) \vee (\neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(b)} \vee \neg\mathtt{Dr(bm)} \wedge \mathtt{Fl(bm)})$,

5. $(\neg\texttt{Fl(u)} \wedge \neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)} \wedge \texttt{Fl(b)}) \vee \neg\texttt{Dr(bm)} \wedge \texttt{Fl(bm)}$,
6. $(\neg\texttt{Fl(u)} \wedge \neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)} \wedge \texttt{Fl(bm)}) \vee \neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)}$,
7. $(\neg\texttt{Fl(u)} \wedge \neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)} \wedge \texttt{Fl(bm)}) \vee \neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)}$,
8. $\neg\texttt{Fl(u)} \wedge \neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)} \wedge \texttt{Fl(bm)} \wedge \texttt{Dr(u)}$.

The generating defaults for the cases $\neg\texttt{Fl(b)}$ and $\neg\texttt{Fl(bm)}$ are formally constructed in a similar way. However, since both defaults have been overridden, they should not contribute to the meaning of the specification. This is indeed so: each of the flattening defaults for $d = \neg\texttt{Fl(b)}$ or $d = \neg\texttt{Fl(bm)}$ can be shown to be defaults implicit in `underMammals` and therefore redundant. Take for example the flattening default $(\neg\texttt{Fl(b)} \wedge \neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)} \wedge \texttt{Fl(bm)}) \vee \neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)}$ (formally identical to the 6th of the defaults for $\texttt{Fl(u)}$). We see that it is implicit in `underMammals`. Firstly $(\neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)} \wedge \texttt{Fl(bm)}) \vee \neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)}$ is a default implicit in `underMammals` since it is the disjunction of $\neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)}$ with the conjunction of $\neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)}$ with $\neg\texttt{Dr(bm)} \wedge \texttt{Fl(bm)}$. This default $\neg\texttt{Dr(bm)}\wedge\texttt{Dr(u)}\wedge\texttt{Fl(bm)}\vee\neg\texttt{Dr(bm)}\wedge\texttt{Fl(b)}$ can be equivalently written as $(\neg\texttt{Fl(b)}\wedge \neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)} \wedge \texttt{Fl(bm)} \vee \texttt{Fl(b)} \wedge \neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)} \wedge \texttt{Fl(bm)}) \vee \neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)}$. This in turn is equivalent to the flattening default $(\neg\texttt{Fl(b)} \wedge \neg\texttt{Dr(bm)} \wedge \texttt{Dr(u)} \wedge \texttt{Fl(bm)}) \vee \neg\texttt{Dr(bm)} \wedge \texttt{Fl(b)}$. △

The translation presented in theorem 103 and illustrated in the example above can surely be optimized and simplified. As we have seen some of the flattening defaults are implicit in other and may be omitted. This includes the flattening defaults introduced by overridden defaults. It is also expected that, in some cases, the original specification can be divided in simpler "logically independent" specifications thus originating simpler translations (since less defaults are involved in each part). An example of this situation is the previous specification: the axioms and defaults for u, b and `bm` do not interfere. Therefore the specification of the simplified version of BATMAN can be divided in three, each having only the axioms and defaults involving only u or only b or only `bm`. The flattening defaults for each such case are:

1. $\{\texttt{Dr(u)}, \neg\texttt{Fl(u)} \wedge \texttt{Dr(u)}, \neg\texttt{Fl(u)} \vee \texttt{Dr(u)}\}$, for u,
2. $\{\texttt{Fl(b)}\}$, for b,
3. $\{\neg\texttt{Dr(bm)}, \texttt{Fl(bm)} \wedge \neg\texttt{Dr(bm)}, \texttt{Fl(bm)} \vee \neg\texttt{Dr(bm)}\}$, for `bm`.

It is not difficult to check that the union of (the axioms) and these defaults is equivalent (has the same preference) as the flat specification obtained above. This means that the lexicographic preference of the whole specification is, in this case, the intersection of the lexicographic preferences of these subspecifications. This is in general not true.

2.3 Theories

We define the theory of a hierarchic specification S by taking its hierarchy of differential preferences as semantics in section 2.3.1. The relation of the theory of S expressed in terms of the axioms and defaults of S itself is displayed in section 2.3.2.

2.3.1 Definition and Galois Connection

The semantics of a hierarchic specification is its hierarchy of differential preferences. Each such preference displays the "effect" of the defaults at the corresponding level h on the preference of interpretation structures made equivalent by defaults of lower levels.

We are now concerned with assigning to a hierarchic specification its *theory*. This is the hierarchic specification having at each level the biggest set of defaults that can be added to a specification without changing its semantics (i.e. without changing its hierarchy of differential preferences and, due to the equivalence between this semantics and the hierarchy of lexicographic preferences, also without changing its hierarchy of lexicographic preferences).

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, are 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 27 in chapter 1). 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.

We assign to a hierarchy of pre-orders (understood as a hierarchy of differential preferences) a corresponding theory by building the biggest specification that induces it. The operators assigning to a specification its hierarchy of differential preferences and to a hierarchy of pre-orders its theory form again a Galois connection and provide the means to compare specifications on the syntactic and semantic levels. Moreover operations of hierarchic specifications have a corresponding operation on the hierarchies of differential preferences. The reverse is also true.

Definition 105 Let $\mathcal{H}$ be a Σ-hierarchy of pre-orders. The corresponding *theory*, denoted by $\mathcal{H}^{\ominus}$ is the Σ-hierarchic specification with:

- the same well-founded partial order $(H, \preceq)$ as $\mathcal{H}$, $(H, \preceq) = \mathsf{po}(\mathcal{H}^{\ominus}) = \mathsf{po}(\mathcal{H})$,

- the Σ-formulas satisfied by each interpretation structure from $\mathcal{H}$ as set of axioms: $\mathsf{ax}(\mathcal{H}^{\ominus}) = |\mathcal{H}|^{\bullet}$,
- the function Δ defined by $\Delta(h) = (|\mathcal{H}|, \mathsf{rl}(\mathcal{H}, h))^{\circ}$ that to each priority level $h \in H$ assigns the set of the defaults implicit in the Σ-pre-order assigned by $\mathcal{H}$ to the level h. ∎

Remark 106 Recall from definition 27 in chapter 1 that $(|\mathcal{H}|, \mathsf{rl}(\mathcal{H}, h))^{\circ}$ is the set $\{d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma) :$ if $m \vDash^{\mathcal{I}}_{\Sigma} d$ then $n \vDash^{\mathcal{I}}_{\Sigma} d$, for all $m, n \in |\mathcal{H}|$ with $m \sqsubseteq_h n\}$ where $\sqsubseteq_h = \mathsf{rl}(\mathcal{H}, h)$.

Hierarchic specifications are related (by forward inclusion, see below) when one results from the other by adding either axioms, defaults at some priority level or even by adding further *priority levels*. This corresponds to inclusion of their partial orders of priority, inclusion of sets of axioms and inclusion of the sets of defaults in each priority level.

On the semantic level, as we know from the classical and flat cases, the relations of inclusion are reversed. The hierarchies of differential preferences are related by reversed inclusion of the classes of interpretation structures (which corresponds to inclusion of sets of axioms) and reversed inclusion of the differential preferences assigned to each level (and this corresponds to inclusion of the sets of defaults in each level). Note that the inclusion of the partial orders of priority is *not* reversed in the semantics: if S has more priority levels than S' then $S^{\ominus}$ also has more priority levels than $S'^{\ominus}$. For this purpose we define two types of inclusion, with inclusion of the partial order of priority either in the *same* or *reversed* direction as the inclusion of the other entities.

Definition 107 Forward inclusion, backward inclusion and pointwise inclusion are defined as follows.

1. Let S and S' be Σ-hierarchic specifications and $\mathsf{po}(S) = (H, \preceq)$ and $\mathsf{po}(S') = (H', \preceq')$ be the corresponding partial orders of priority. Then S and S' are *related by forward inclusion* written $S \overrightarrow{\Subset} S'$ (or $S' \overleftarrow{\Supset} S$) iff:
   - $\mathsf{ax}(S) \subseteq \mathsf{ax}(S')$,
   - $H \subseteq H'$ and $\preceq \subseteq \preceq'$ (i.e. if $h_1 \preceq h_2$ then $h_1 \preceq' h_2$) and
   - $\mathsf{df}(S, h) \subseteq \mathsf{df}(S', h)$ for every $h \in H$.
2. If $(H, \preceq) = (H', \preceq')$ and $S \overrightarrow{\Subset} S'$ then S and S' are said to be related by *pointwise inclusion* and this will be denoted by $S \Subset S'$ (or $S' \Supset S$).

3. Let $\mathcal{H}$ and $\mathcal{H}'$ be Σ-hierarchies of pre-orders and $\mathsf{po}(\mathcal{H}) = (H, \preceq)$ and $\mathsf{po}(\mathcal{H}') = (H', \preceq')$ be the corresponding partial orders of priority. Then $\mathcal{H}'$ and $\mathcal{H}$ are *related by backward inclusion* written $\mathcal{H}' \overleftarrow{\Subset} \mathcal{H}$ (or $\mathcal{H}' \overrightarrow{\Supset} \mathcal{H}$) iff:
   - $|\mathcal{H}'| \subseteq |\mathcal{H}|$,
   - $H' \supseteq H$ and $\preceq' \supseteq \preceq$ (i.e. if $h_1 \preceq h_2$ then $h_1 \preceq' h_2$) and
   - $\mathsf{rl}(\mathcal{H}', h) \subseteq \mathsf{rl}(\mathcal{H}, h)$ for every $h \in H$.
4. If $(H, \preceq) = (H', \preceq')$ and $\mathcal{H}' \overleftarrow{\Subset} \mathcal{H}$ then $\mathcal{H}'$ and $\mathcal{H}$ are said to be related by *pointwise inclusion* and this will be denoted by $\mathcal{H}' \Subset \mathcal{H}$ (or $\mathcal{H} \Supset \mathcal{H}'$).

Note that all these relations are partial orders. ∎

The operations of *pointwise union and intersection* generalize union and intersection to either hierarchic specifications and hierarchies of pre-orders. These operations are only defined for hierarchic specifications (or hierarchies of pre-orders) with the *same* partial order of priority.

Again care has to be taken with unions of pre-orders, since its result is not necessarily a pre-order. For this reason pointwise union of hierarchies of pre-orders assigns to level h the transitive closure of the union of the argument pre-orders at that level.

Definition 108 The operations of *pointwise union* ($\Cup$) and *pointwise intersection* ($\Cap$) are defined as follows.

Let S and S' be Σ-hierarchic specifications with the same partial order of priority $(H, \preceq) = \mathsf{po}(S) = \mathsf{po}(S')$. Then

1. the *pointwise union* of S and S', $S \Cup S'$, is the Σ-hierarchic specification with:
   - the same partial order of priority as S and S': $\mathsf{po}(S \Cup S') = (H, \preceq)$,
   - the union of the sets of axioms of S and S' as a set of axioms: $\mathsf{ax}(S \Cup S') = \mathsf{ax}(S) \cup \mathsf{ax}(S')$ and
   - the union of the sets of defaults at level h from S and S' as a set of defaults at level h: $\mathsf{df}(S \Cup S', h) = \mathsf{df}(S, h) \cup \mathsf{df}(S', h)$ for $h \in H$.
2. the *pointwise intersection* of S and S', $S \Cap S'$, is the Σ-hierarchic specification with:
   - the same partial order of priority as S and S': $\mathsf{po}(S \Cap S') = (H, \preceq)$,
   - the intersection of the sets of axioms of S and S' as a set of axioms: $\mathsf{ax}(S \Cap S') = \mathsf{ax}(S) \cap \mathsf{ax}(S')$ and

- the intersection of the sets of defaults at level h from S and S' as a set of defaults at level h: $\mathsf{df}(S \Cap S', h) = \mathsf{df}(S, h) \cap \mathsf{df}(S', h)$ for $h \in H$.

Let $\mathcal{H}, \mathcal{H}'$ be Σ-hierarchies of pre-orders with the same partial order of priority $(H, \preceq) = \mathsf{po}(\mathcal{H}) = \mathsf{po}(\mathcal{H}')$. Then

1. the *pointwise union* of $\mathcal{H}$ and $\mathcal{H}'$, $\mathcal{H} \Cup \mathcal{H}'$, is the Σ-hierarchy of pre-orders with:
   - the same partial order of priority as $\mathcal{H}$ and $\mathcal{H}'$: $\mathsf{po}(\mathcal{H} \Cup \mathcal{H}') = (H, \preceq)$,
   - the union of the sets interpretation structures of $\mathcal{H}$ and $\mathcal{H}'$ as set of interpretation structures: $|\mathcal{H} \Cup \mathcal{H}'| = |\mathcal{H}| \cup |\mathcal{H}'|$ and
   - the transitive closure union of the pre-orders at level h from $\mathcal{H}$ and $\mathcal{H}'$ as pre-order at level h: $\mathsf{rl}(\mathcal{H} \Cup \mathcal{H}', h) = \overline{\mathsf{rl}(\mathcal{H}, h) \cup \mathsf{rl}(\mathcal{H}', h)}$ for $h \in H$.
2. the *pointwise intersection* of $\mathcal{H}$ and $\mathcal{H}'$, $\mathcal{H} \Cap \mathcal{H}'$ is the Σ-hierarchy of pre-orders with:
   - the same partial order of priority as $\mathcal{H}$ and $\mathcal{H}'$: $\mathsf{po}(\mathcal{H} \Cap \mathcal{H}') = (H, \preceq)$,
   - the intersection of the sets interpretation structures of $\mathcal{H}$ and $\mathcal{H}'$ as set of interpretation structures: $|\mathcal{H} \Cap \mathcal{H}'| = |\mathcal{H}| \cap |\mathcal{H}'|$ and
   - the intersection of the pre-orders at level h from $\mathcal{H}$ and $\mathcal{H}'$ as pre-order at level h: $\mathsf{rl}(\mathcal{H} \Cap \mathcal{H}', h) = \mathsf{rl}(\mathcal{H}, h) \cap \mathsf{rl}(\mathcal{H}', h)$ for $h \in H$. ■

We present in what follows the Galois connection between hierarchic specifications and corresponding hierarchies of differential preferences. For that purpose we have to restrict the hierarchies of pre-orders to those satisfying the following additional property.

Definition 109 Let $\mathcal{H}$ be a hierarchy of pre-orders. $\mathcal{H}$ is said to be a *hierarchy of differential pre-orders* iff $\mathcal{H} = \mathcal{H}^{\boxminus}$. ■

Remark 110 The hierarchy of differential preferences $S^{\ominus}$ associated with a specification S is clearly a hierarchy of differential pre-orders. In fact $S^{\ominus} = S^{\odot^{\boxminus}}$ and therefore $S^{\ominus^{\boxminus}} = S^{\odot^{\boxminus^{\boxminus}}} = S^{\odot^{\boxminus}} = S^{\ominus}$.

The Galois connection states that to more axioms correspond less models and also that to *more defaults* at a given level corresponds *less relations* in the differential preference at that level. This implies that the operations of pointwise union and intersection of hierarchic specifications have a corresponding semantic operation.

Note that the direction of the inclusion of partial orders is not reversed when going from specifications to their semantics and vice versa. This is represented by the direction of the arrow $\rightarrowtail$ in $\overset{\rightarrowtail}{\Subset}$.

Theorem 111 Let S, S' be Σ-hierarchic specifications and $\mathcal{H}, \mathcal{H}'$ be Σ-hierarchies of *differential* pre-orders. Then

1. $S \overrightarrow{\Subset} S'$ implies $S^{\ominus} \overrightarrow{\Supset} S'^{\ominus}$,
2. $\mathcal{H} \overrightarrow{\Supset} \mathcal{H}'$ implies $\mathcal{H}^{\ominus} \overrightarrow{\Subset} \mathcal{H}'^{\ominus}$,
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 Σ-specifications and $\mathcal{H}_n, n \in N$ be Σ-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) $(\Cap_n \mathcal{H}_n^{\ominus\ominus})^{\ominus} = (\Cup_n \mathcal{H}_n^{\ominus})^{\ominus\ominus}$.

Proof

1. ($S \overrightarrow{\Subset} S'$ implies $S^{\ominus} \overrightarrow{\Supset} S'^{\ominus}$). Since S and $S^{\ominus}$ have the same partial order of priority and the same holds for S' and $S'^{\ominus}$ the property of inclusion of the partial orders of priority obviously holds.

   The property relating sets of axioms and interpretation structures is now shown. Since $S \overrightarrow{\Subset} S'$ then $\mathsf{ax}(S) \subseteq \mathsf{ax}(S')$. Therefore $|S^{\ominus}| = \mathsf{ax}(S)^{\bullet} \supseteq \mathsf{ax}(S')^{\bullet} = |S'^{\ominus}|$ from the Galois connection for the classical case (theorem 6).

   The proof for the differential preferences is by well-founded induction in the partial order of priority $(H, \preceq) = \mathsf{po}(S)$ of S. The following preliminary considerations are relevant for both cases. Let $\sqsubseteq^{\ominus}_{S,h}$ be the differential preference associated by $S^{\ominus}$ to level $h \in H$ and $\sqsubseteq^{\ominus}_{S',h}$ be the differential preference associated by $S'^{\ominus}$ to the same level. Recall (definition 85) that $m \sqsubseteq^{\ominus}_{S,h} n$ for models $m, n \in \mathsf{ax}(S)$ iff n satisfies the same defaults of levels strictly under h in $(H, \preceq)$ as m and at least as much from level h, i.e. $m \sqsubseteq^{\ominus}_{S,h} n$ iff $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$ and $\mathsf{df}(S,h_1)(m) = \mathsf{df}(S,h_1)(n)$ for $h_1 \prec h$. This can be written as $m \sqsubseteq^{\odot}_{S,h} n$ and for $h_1 \prec h, m \equiv^{\ominus}_{S,h_1} n$ where $m \sqsubseteq^{\odot}_{S,h} n$ is the condition $\mathsf{df}(S,h)(m) \subseteq \mathsf{df}(S,h)(n)$

and $m \equiv^{\ominus}_{S,h_1} n$ expresses equivalence of m and n according to $\sqsubseteq^{\ominus}_{S,h_1}$. Note finally that $\sqsubseteq^{\odot}_{S,h}$ is the preference relation associated with the (flat) Σ-specification $(\mathsf{ax}(S), \mathsf{df}(S,h))$ (recall definition 81 of $S^{\odot}$ and definition 28 of * in chapter 1). That is $\sqsubseteq^{\odot}_{S,h} = \mathsf{rl}((\mathsf{ax}(S), \mathsf{df}(S,h))^{\star})$. From the Galois connection for specifications (theorem 30) and since $(\mathsf{ax}(S), \mathsf{df}(S,h)) \Subset (\mathsf{ax}(S'), \mathsf{df}(S',h))$ we conclude that $\sqsubseteq^{\odot}_{S,h} = \mathsf{rl}((\mathsf{ax}(S), \mathsf{df}(S,h))^{\star}) \supseteq \mathsf{rl}((\mathsf{ax}(S'), \mathsf{df}(S',h))^{\star}) = \sqsubseteq^{\odot}_{S',h}$.

- Assume h is minimal in $(H, \preceq)$. Therefore $\sqsubseteq^{\ominus}_{S,h}$ is simply $\sqsubseteq^{\odot}_{S,h}$. We also know from the previous considerations that $\sqsubseteq^{\odot}_{S,h} \supseteq \sqsubseteq^{\odot}_{S',h}$.
  The level h does not have to be minimal in $(H', \preceq') = \mathsf{po}(S')$. For this reason the relation from S' at level h is given by $m \sqsubseteq^{\ominus}_{S',h} n$ iff $m \sqsubseteq^{\odot}_{S',h} n$ and for $h'_1 \prec' h, m \equiv^{\ominus}_{S',h'_1} n$. This implies $\sqsubseteq^{\ominus}_{S',h} \subseteq \sqsubseteq^{\odot}_{S',h}$. Therefore $\sqsubseteq^{\ominus}_{S',h} \subseteq \sqsubseteq^{\odot}_{S',h} \subseteq \sqsubseteq^{\odot}_{S,h} = \sqsubseteq^{\ominus}_{S,h}$.
- Let h be non-minimal in $(H, \preceq)$. Then $m \sqsubseteq^{\ominus}_{S,h} n$ iff $m \sqsubseteq^{\odot}_{S,h} n$ and for $h_1 \prec h, m \equiv^{\ominus}_{S,h_1} n$. This is the same as $\sqsubseteq^{\ominus}_{S,h} = \sqsubseteq^{\odot}_{S,h} \cap (\cap_{\{h_1 \prec h\}} \equiv^{\ominus}_{S,h_1})$. We have to compare $\sqsubseteq^{\ominus}_{S,h}$ with $\sqsubseteq^{\ominus}_{S',h} = \sqsubseteq^{\odot}_{S',h} \cap (\cap_{\{h'_1 \prec' h\}} \equiv^{\ominus}_{S',h'_1})$ and we know that $\sqsubseteq^{\odot}_{S',h} \subseteq \sqsubseteq^{\odot}_{S,h}$. Now the set $\{h'_1 \prec' h\}$ of the levels strictly under h according to $\preceq'$ can be divided into the set of levels strictly under h according to $\preceq$ plus those that are strictly under h according to $\preceq'$ but not according to $\preceq$ (recall that $H \subseteq H'$ and $\preceq \subseteq \preceq'$). Therefore $\cap_{\{h'_1 \prec' h\}} \equiv^{\ominus}_{S',h'_1}$ can be divided into $(\cap_{\{h_1 \prec h\}} \equiv^{\ominus}_{S',h_1}) \cap (\cap_{\{h'_1 \prec' h; h'_1 \not\prec h\}} \equiv^{\ominus}_{S',h'_1})$.
  From the induction hypothesis for each $h_1 \prec h$ one has $\sqsubseteq^{\ominus}_{S,h_1} \supseteq \sqsubseteq^{\ominus}_{S',h_1}$ which implies $\equiv^{\ominus}_{S,h_1} \supseteq \equiv^{\ominus}_{S',h_1}$. Therefore $\cap_{\{h_1 \prec h\}} \equiv^{\ominus}_{S,h_1} \supseteq \cap_{\{h_1 \prec h\}} \equiv^{\ominus}_{S',h_1}$ and this implies $\cap_{\{h_1 \prec h\}} \equiv^{\ominus}_{S,h_1} \supseteq \cap_{\{h'_1 \prec' h\}} \equiv^{\ominus}_{S',h'_1}$. Since $\sqsubseteq^{\odot}_{S',h} \subset \sqsubseteq^{\odot}_{S,h}$ we conclude $\sqsubseteq^{\ominus}_{S,h} = \sqsubseteq^{\odot}_{S,h} \cap (\cap_{\{h_1 \prec h\}} \equiv^{\ominus}_{S,h_1}) \supseteq \sqsubseteq^{\odot}_{S',h} \cap (\cap_{\{h'_1 \prec' h\}} \equiv^{\ominus}_{S',h'_1})$ as intended.

2. ($\mathcal{H} \overrightarrow{\Supset} \mathcal{H}'$ implies $\mathcal{H}^{\ominus} \overrightarrow{\Subset} \mathcal{H}'^{\ominus}$). Since $\mathcal{H}$ and $\mathcal{H}^{\ominus}$ have the same partial order of priority and the same holds for $\mathcal{H}'$ and $\mathcal{H}'^{\ominus}$ the property of inclusion of the partial orders of priority trivially holds. The property relating classes of interpretation structures and axioms follows from the following: since $\mathcal{H} \overrightarrow{\Supset} \mathcal{H}'$ then $|\mathcal{H}| \supseteq |\mathcal{H}'|$. This implies that $\mathsf{ax}(\mathcal{H}^{\ominus}) = |\mathcal{H}|^{\bullet} \subseteq |\mathcal{H}'|^{\bullet} = \mathsf{ax}(\mathcal{H}'^{\ominus})$ from the Galois connection for the classical case (theorem 6).

   Let $(H, \preceq) = \mathsf{po}(\mathcal{H})$ be the priority relation of $\mathcal{H}$ and $h \in H$ an arbitrary priority level. We have to show that $\mathsf{df}(\mathcal{H}^{\ominus}, h) \subseteq \mathsf{df}(\mathcal{H}'^{\ominus}, h)$ for each $h \in H$.

   Let $\mathcal{R}_h = (|\mathcal{H}|, \mathsf{rl}(\mathcal{H}, h))$ be the Σ-pre-order with the interpretation structures from $\mathcal{H}$ as interpretation structures and having as relation the relation assigned by $\mathcal{H}$ to level h. Let $\mathcal{R}'_h = (|\mathcal{H}'|, \mathsf{rl}(\mathcal{H}', h))$ be the corresponding pre-order for $\mathcal{H}'$ and level h. Then $\mathcal{H} \overrightarrow{\Supset} \mathcal{H}'$ implies $\mathcal{R}_h \Supset \mathcal{R}'_h$ and from the Galois connection for specifications (theorem 6) we have $\mathcal{R}^{\bullet}_h \Subset \mathcal{R}'^{\bullet}_h$. This means in particular that the defaults implicit in $\mathcal{R}_h$ are contained in those implicit in $\mathcal{R}'_h$. But the defaults implicit in $\mathcal{R}_h$ are precisely the defaults assigned to level h by $\mathcal{H}^{\ominus}$ and similarly the defaults implicit in $\mathcal{R}'_h$ are precisely the defaults assigned to level h by $\mathcal{H}'^{\ominus}$. Therefore $\mathsf{df}(\mathcal{H}^{\ominus}, h) \subseteq \mathsf{df}(\mathcal{H}'^{\ominus}, h)$. (see definition 27 of defaults implicit in a pre-order and definition 105 of $\mathcal{H}^{\ominus}$ and $\mathcal{H}'^{\ominus}$).

3. ($S \Subset S^{\ominus\ominus}$). Clearly S, $S^{\ominus}$ and $S^{\ominus\ominus}$ have the same partial order of priority $(H, \preceq) = \mathsf{po}(S)$.

   Note that $\mathsf{ax}(S^{\ominus\ominus}) = |S^{\ominus}|^{\bullet} = \mathsf{ax}(S)^{\bullet\bullet} \supseteq \mathsf{ax}(S)$ where the last inclusion results from the Galois connection for the classical case.

   We have to show that the defaults assigned by S to level h are contained in the defaults assigned by $S^{\ominus\ominus}$ to the same level.

   Recall (from definition 105) that $\mathsf{df}(S^{\ominus\ominus}, h)$ are the defaults implicit in the pre-order $\mathcal{R}_h = (|S^{\ominus}|, \mathsf{rl}(S^{\ominus}, h))$.

   Recall furthermore that $|S^{\ominus}| = \mathsf{ax}(S)^{\bullet}$ and note that $\mathsf{rl}(S^{\ominus}, h)$ is the relation $\sqsubseteq^{\ominus}_{S,h} = \sqsubseteq^{\odot}_{S,h} \cap (\cap_{\{h_1 \prec h\}} \equiv^{\ominus}_{S,h_1})$, where $\sqsubseteq^{\odot}_{S,h}$ is the relation defined by $m \sqsubseteq^{\odot}_{S,h} n$ iff $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ (n satisfies at least the defaults from S at level h satisfied by m). Therefore $\mathcal{R}_h = (\mathsf{ax}(S)^{\bullet}, \sqsubseteq^{\ominus}_{S,h})$.

   Consider now the (flat) Σ-specification $S_h = (\mathsf{ax}(S), \mathsf{df}(S, h))$ with the axioms of S and the defaults of S at level h. Note that the preference associated with S_h is the relation $\sqsubseteq^{\odot}_{S,h}$. Therefore $\mathcal{R}_h \Subset S_h{}^{\star}$ and from the Galois connection for specifications $S_h{}^{\star\star} \Subset \mathcal{R}_h{}^{\star}$. Remember again from the Galois connection for the classic case that the defaults from S_h are defaults of $S_h{}^{\star\star}$. Therefore $\mathsf{df}(S, h)$ are defaults of $S_h{}^{\star\star} \Subset \mathcal{R}_h{}^{\star}$. That $\mathsf{df}(S, h)$ are defaults of $\mathcal{R}_h{}^{\star}$ is the same as stating that $\mathsf{df}(S, h) \subseteq \mathsf{df}(S^{\ominus\ominus}, h)$.

4. ($\mathcal{H} \Subset \mathcal{H}^{\ominus\ominus}$). Clearly $\mathcal{H}$, $\mathcal{H}^{\ominus}$ and $\mathcal{H}^{\ominus\ominus}$ have the same partial order of priority $(H, \preceq) = \mathsf{po}(\mathcal{H})$.

   Note that $|\mathcal{H}^{\ominus\ominus}| = \mathsf{ax}(\mathcal{H}^{\ominus})^{\bullet} = |\mathcal{H}|^{\bullet\bullet} \supseteq |\mathcal{H}|$ where the last inclusion results from the Galois connection for the classical case. Let $\sqsubseteq_h$ be the relation associated by $\mathcal{H}$ to level h. The defaults implicit in $(|\mathcal{H}|, \sqsubseteq_h)$ are the formulas assigned by $\mathcal{H}^{\ominus}$ to level h. That is $\mathsf{df}(\mathcal{H}^{\ominus}, h) = (|\mathcal{H}|, \sqsubseteq_h)^{\circ}$. Let $\sqsubseteq^{\ominus\odot}_h$ be the preference on $|\mathcal{H}|$ defined by $m \sqsubseteq^{\ominus\odot}_h n$ iff for every $d \in \mathsf{df}(\mathcal{H}^{\ominus}, h)$ if $m \vDash d$ then $n \vDash d$. This is the relation of the pre-order $(|\mathcal{H}|, \sqsubseteq_h)^{\star\star}$. From the Galois connection for specifications we have $\sqsubseteq_h \subseteq \sqsubseteq^{\ominus\odot}_h$.

   The relation $\sqsubseteq^{\ominus\ominus}_h$ assigned by $\mathcal{H}^{\ominus\ominus}$ to level h is defined as follows: $m \sqsubseteq^{\ominus\ominus}_h n$ iff $\mathsf{df}(\mathcal{H}^{\ominus}, h)(m) \subseteq \mathsf{df}(\mathcal{H}^{\ominus}, h)(n)$ and $\mathsf{df}(\mathcal{H}^{\ominus}, h')(m) = \mathsf{df}(\mathcal{H}^{\ominus}, h')(n)$ for every $h' \prec h$. This is the same as $m \sqsubseteq^{\ominus\ominus}_h n$ iff $m \sqsubseteq^{\ominus\odot}_h n$ and $m \equiv^{\ominus\odot}_{h'} n$ for every $h' \prec h$. The equivalence $m \equiv^{\ominus\odot}_{h'} n$ states that $\mathsf{df}(\mathcal{H}^{\ominus}, h')(m) = \mathsf{df}(\mathcal{H}^{\ominus}, h')(n)$. From lemma 87 this is also equivalent to $m \equiv^{\ominus\ominus}_{h'} n$. Therefore the equivalence $m \equiv^{\ominus\odot}_{h'} n$ for every $h' \prec h$ can be replaced by $m \equiv^{\ominus\ominus}_{h'} n$ for every $h' \prec h$. In this way $m \sqsubseteq^{\ominus\ominus}_h n$ iff $m \sqsubseteq^{\ominus\odot}_h n$ and $m \equiv^{\ominus\ominus}_{h'} n$ for every $h' \prec h$.

   We now prove by well founded induction that $\sqsubseteq_h \subseteq \sqsubseteq^{\ominus\ominus}_h$.

   - if h is minimal in $(H, \preceq)$ then $\sqsubseteq^{\ominus\ominus}_h = \sqsubseteq^{\ominus\odot}_h$ and we already know that $\sqsubseteq_h \subseteq \sqsubseteq^{\ominus\odot}_h = \sqsubseteq^{\ominus\ominus}_h$.

- assume h is not minimal in $(H, \preceq)$. From the induction hypothesis $\sqsubseteq_{h'} \subseteq \sqsubseteq_{h'}^{\ominus\ominus}$ for $h' \prec h$. This implies $\equiv_{h'} \subseteq \equiv_{h'}^{\ominus\ominus}$ for $h' \prec h$.
  Recall that $m \sqsubseteq_h^{\ominus\ominus} n$ iff $m \sqsubseteq_h^{\ominus\odot} n$ and $m \equiv_{h'}^{\ominus\ominus} n$ for every $h' \prec h$ so we may rewrite this as $\sqsubseteq_h^{\ominus\ominus} = \sqsubseteq_h^{\ominus\odot} \cap (\cap_{h' \prec h} \equiv_{h'}^{\ominus\ominus})$. Since $\sqsubseteq_h \subseteq \sqsubseteq_h^{\ominus\odot}$ and $\equiv_{h'} \subseteq \equiv_{h'}^{\ominus\ominus}$ we conclude that $\sqsubseteq_h \cap (\cap_{h' \prec h} \equiv_{h'}) \subseteq \sqsubseteq_h^{\ominus\odot} \cap (\cap_{h' \prec h} \equiv_{h'}^{\ominus\ominus}) = \sqsubseteq_h^{\ominus\ominus}$. Note finally that since $\mathcal{H} = \mathcal{H}^{\boxminus}$ it follows easily (definition 90 of $\mathcal{H}^{\boxminus}$) that $\sqsubseteq_h = \sqsubseteq_h \cap (\cap_{h' \prec h} \equiv_{h'})$. In this way $\sqsubseteq_h = \sqsubseteq_h \cap (\cap_{h' \prec h} \equiv_{h'}) \subseteq \sqsubseteq_h^{\ominus\ominus}$ as intended.

The derived properties are proved formally as the corresponding properties for the Galois connection for specifications (theorem 6). We note only that if, for each n, $\mathcal{H}_n \Subset \mathcal{H}'$ then $\Cup_n \mathcal{H}_n \Subset \mathcal{H}'$. This is relevant since $\Cup_n \mathcal{H}_n$ has in each level the *transitive closure* of the union of the relations assigned to that level by each $\mathcal{H}_n$. ✓

As in the case of specifications some important properties are corollaries of the Galois connection and rely on the bijective relation between *theories* and their semantics. Theories are closed hierarchic specifications.

Definition 112

- The *closure* of a Σ-hierarchic specification S is the Σ-hierarchic specification $S^{\ominus\ominus}$ (i.e. $(S^{\ominus})^{\ominus}$). The closure $S^{\ominus\ominus}$ is also called the *theory* of S. A Σ-hierarchic specification S is *closed* iff $S = S^{\ominus\ominus}$.

- The *closure* of a Σ-hierarchy of pre-orders $\mathcal{H}$ is the Σ-hierarchy of pre-orders $\mathcal{H}^{\ominus\ominus}$ (i.e. $(\mathcal{H}^{\ominus})^{\ominus}$). A Σ-hierarchy of pre-orders $\mathcal{H}$ is *closed* iff $\mathcal{H} = \mathcal{H}^{\ominus\ominus}$. ▪

The relation between closed hierarchic specifications and closed hierarchies of differential pre-orders is one to one. Moreover, semantic comparison of hierarchic specifications (by pointwise inclusion of the differential preferences) is expressed at the syntactic level by inclusion of *closed* hierarchic specifications, i.e. theories.

Lemma 113 Let S, S' be closed Σ-hierarchic specifications and $\mathcal{H}, \mathcal{H}'$ be closed Σ-hierarchies of pre-orders. Then

- $S \overrightarrow{\Subset} S'$ iff $S^{\ominus} \overrightarrow{\Supset} S'^{\ominus}$,

- $\mathcal{H} \overrightarrow{\Subset} \mathcal{H}'$ iff $\mathcal{H}^{\ominus} \overrightarrow{\Supset} \mathcal{H}'^{\ominus}$.

Proof Trivial from the Galois connection in theorem 111. ✓

We emphasize that closed hierarchic specifications are canonical among the hierarchic specifications having the same semantics: on one hand equivalent specifications (having the same semantics) also have the same theory. On the other hand this theory is the biggest specification among equivalent ones.

Lemma 114

1. Let S and S' be Σ-hierarchic specifications. S and S' have the same semantics iff they have the same theory: $S^{\ominus} = S'^{\ominus}$ iff $S^{\ominus\ominus} = S'^{\ominus\ominus}$;
2. Let S be a Σ-hierarchic specification. The theory $S^{\ominus\ominus}$ is the biggest specification from among those having the same semantics as S:
   - $S^{\ominus\ominus\ominus} = S^{\ominus}$ and
   - given any Σ-hierarchic specification S', if $S^{\ominus} = S'^{\ominus}$ then $S' \Subset S^{\ominus\ominus}$.

Proof Trivial from the Galois connection in theorem 111. ✓

Notice that the fact that the theory of a specification S is the biggest among those having the same semantics means also that the set of defaults at each level from the theory of S is the biggest set of defaults that can be added to that level without changing the semantics.

Note also that equality of theories means equality of hierarchies of differential preferences (as stated above) and equality of hierarchies of lexicographic preferences. This is a simple consequence of the equivalence between the two semantics (stated in theorem 94). For the same reason the theory of a specification can be characterized as the biggest specification having the *same hierarchy of lexicographic preferences* as the original specification.

Finally, semantical comparison of specifications can be expressed at the syntactic level. In order to establish whether $S^{\ominus\ominus} \overrightarrow{\Subset} S'^{\ominus\ominus}$ one has to check inclusion of the partial orders of priority, and whether each axiom from S is semantically entailed by the axioms from S' and whether at each level h each default from S is an implicit default in the differential preference at the same level h in $S'^{\ominus}$. Recalling that the theory of S' consists precisely of this information this is formalized as follows.

Lemma 115 Let S and S' be Σ-hierarchic specifications. Then $S^{\ominus} \overrightarrow{\Supset} S'^{\ominus}$ iff $S \overrightarrow{\Subset} S'^{\ominus\ominus}$.

Proof Trivial from the Galois connection in theorem 111. ✓

Clearly to establish equality of the preference relations one simply has to apply the lemma above in both directions.

2.3.2 Theories Revisited

We are now concerned with relating the theory of a hierarchic specification S with S itself. The axioms of the theory are the formulas semantically entailed by the axioms of the specification. If the underlying logic has a sound and complete proof system these are the consequences of $\mathsf{ax}(S)$. The partial orders of priority coincide. The defaults of the theory of S at level h are the defaults implicit in the differential preference of S at level h. It is our aim in the following to present an alternative characterization of such defaults by using semantic entailment and the axioms and defaults from S. Such a characterization gives a clearer account of the defaults from the theory of S that, in some logics, can be automatically checked (see remark 124 below).

Lemma 116 The theory $S^{\ominus\ominus}$ of a Σ-hierarchic specification S is such that:

1. has as axioms the formulas satisfied in each model (the consequences) of the axioms from S; $\mathsf{ax}(S^{\ominus\ominus}) = |S^{\ominus}|^{\bullet} = \mathsf{ax}(S)^{\bullet\bullet}$ and
2. the same well-founded partial order $(H \preceq)$ as S, $(H \preceq) = \mathsf{po}(S^{\ominus\ominus}) = \mathsf{po}(S)$,
3. has as defaults at level $h \in H$ the defaults implicit in the differential preference at level h, $\mathsf{df}(S^{\ominus\ominus}, h) = (\mathsf{ax}(S)^{\bullet\bullet}, \mathsf{rl}(S^{\ominus}, h))^{\circ}$.

Proof Obvious from the definition 105 of theory of a hierarchy of pre-orders and the definition 85 of hierarchy of differential preferences. ✓

In order to characterize the set of implicit defaults in level h we only have to be able to express the corresponding differential preference as the preference of some (flat) specification. In this case we know already (chapter 1, theorem 51) how to relate the defaults with the flat specification.

A small example is useful to clarify these considerations.

Example 117 We recall again example 77 and the corresponding hierarchy of differential preferences. It is the hierarchic specification presented again in figure 2.16.

$$\bullet : \{q\}$$
$$\uparrow$$
$$\circ : \{p\}$$

Figure 2.16: p lower than q

Its hierarchy of differential preferences is displayed in the figure 2.17:

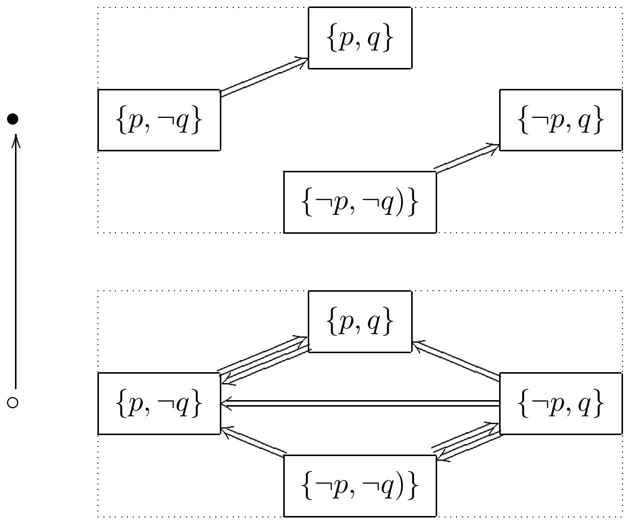

Figure 2.17: Hierarchy of Differential Preferences

The theory of this hierarchic specification will have at level $\circ$ (the lowest and most important) the defaults implicit in the corresponding differential preference. In this case we know that the differential preference at level $\circ$ is simply the preference relation induced by the flat specification $(\emptyset, \{p\})$, having as axioms the axioms of the specification and as defaults the defaults from the specification at this level. This situation is already known from chapter 1. The defaults implicit in the preference relation of $(\emptyset, \{p\})$ are those that can be expressed as disjunctions of conjunctions of the defaults in $\{p\}$, since the set of axioms is empty (in this case the implicit defaults are the formulas semantically equivalent to one of $\mathsf{true}, \mathsf{false}$ and p). And these are the defaults assigned by the theory to the level $\circ$.

We concentrate now in the differential preference at level $\bullet$. In this case we cannot identify it immediately with the preference of some specification. This differential preference relates interpretation structures that are equally good at satisfying p by how they satisfy q.

However it is not difficult to see that the equivalence relation expressing that interpretation structures are equally good at satisfying p is the preference induced by p and $\neg p$. The differential preference at level $\bullet$ compares such interpretation structures also with respect to satisfaction of q. Therefore this differential preference is the preference of the (flat) specification $(\emptyset, \{p, \neg p, q\})$.

Knowing this we can characterize all defaults implicit in the differential preference at level $\bullet$. These are the formulas that can be expressed as disjunctions of conjunctions of the defaults in $\{p, \neg p, q\}$ (and are all semantically equivalent to one of $\mathsf{true}, \mathsf{false}$, p, $\neg p$, q, $p \wedge q$, $\neg p \wedge q$, $p \vee q$, $\neg p \vee q$). And these are the defaults assigned by the theory to the level $\bullet$. $\triangle$

The previous example is an instance of the following general situation: the differential preference at some level is induced by the flat specification having as defaults not only the defaults at that level but also the defaults and negations of these from lower levels. Therefore the defaults assigned by the theory of S to a level h are the formulas corresponding to disjunctions of conjunctions (in the context of the axioms) of the previous set: the union of the defaults from S at level h with the defaults and negations of defaults of levels strictly under h.

Clearly an institution does not necessarily have all these connectives. It may, however, have formulas with a corresponding meaning, i.e. whose models are unions or intersections or complements of the models of other formulas. These, when appropriate, have to be considered when defining implicit defaults.

The characterization of implicit defaults at some level h will be in terms of the classes of models of the defaults at level h and under h. To be able to express negations of formulas (and this technically simplifies matters) we will firstly exhibit the characterization for institutions with negation.

Definition 118 An institution having *negation* is a pair $(\mathcal{I}, \mathsf{neg})$ where $\mathcal{I}$ is an institution and neg is a family of functions, indexed on the signatures of $\mathcal{I}$ giving for any formula over a signature its negation: $\mathsf{neg} = \{\mathsf{neg}_\Sigma : \mathsf{Sen}^{\mathcal{I}}(\Sigma) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma) \mid \Sigma \in |\mathsf{Sign}^{\mathcal{I}}|\}$ such that for each $\Sigma \in |\mathsf{Sign}^{\mathcal{I}}|$ and each interpretation structure $m \in \mathsf{Mod}^{\mathcal{I}}(\Sigma)$ the following holds for any formula $f \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$:

$$m \models^{\mathcal{I}}_{\Sigma} \mathsf{neg}_\Sigma(f) \text{ iff } m \not\models^{\mathcal{I}}_{\Sigma} f.$$

■

Remark 119 Note in the definition above that what is demanded is a syntactic way of expressing that "an interpretation structure does not satisfy a formula". This coincides with classical negation for propositional and first order logic. The situation in modal logics is different and depends on the modal logic and also on its formalization. We discuss briefly this question. Modal interpretation structures (or modal models) occur in one of the following two forms: either they are a structure $\mathcal{W}$ having a class of worlds, a visibility relation among them and a function assigning to atomic formulas the classes of worlds satisfying them ([56, 50, 59, 15]), or they are pairs $(\mathcal{W}, w)$ where w is a distinguished world from $\mathcal{W}$ ([19]). In the first case satisfaction of a formula in $\mathcal{W}$ is satisfaction of f in each world w from $\mathcal{W}$. In the second case $(\mathcal{W}, w) \models f$ corresponds to satisfaction only in the distinguished w. Both formalizations yield the same valid formulas but they differ in the consequences of a theory presentation T (since the models of T have different nature). Non-satisfiability of a formula f in the second formalization is expressed by negation: $(\mathcal{W}, w) \not\models f$ iff $(\mathcal{W}, w) \models \neg f$. These modal logics (give rise to institutions that) have negation in the sense of definition 118 above. This is not the case with the first formalization and in general such modal logics do not have negation in our sense. In fact $\mathcal{W} \vDash \neg f$ expresses that $\neg f$ holds in all worlds from $\mathcal{W}$ whereas $\mathcal{W} \nvDash f$ means that f does not hold in some w from $\mathcal{W}$.

Non-satisfiability of f need not be expressed by the negation connective of these modal logics. It is expressed by the formula $\neg\Box f$ in the modal logic obtained by demanding the visibility relation to be full (i.e. each world from $\mathcal{W}$ sees any other; in this case the visibility relation is redundant and may be omitted). This characterizes the modal logic $S5$ and corresponds to the view of Carnap ([18]) of necessity as truth in all worlds. See also [19].

Non-satisfiability may also be expressed in temporal logics with both a past (H) and future (G) necessity operator. The formula $\neg HGf$ expresses non-satisfiability under the (quite general) condition that given any two states s_1 and s_2 these lie both in the future of third state s_3. Satisfaction of $\neg HGf$ in s means that there is a state in the past of s that sees a state in its future where f does not hold. It is easy to check that under this assumption $\mathcal{W} \vDash \neg HGf$ iff $\mathcal{W} \nvDash f$.

The previous considerations also apply to intuitionistic logic since it may be given meaning by means of particular Kripke structures. If the first formalization is followed ([38, 51]) we obtain an institution without negation. However, using the formalization with a distinguished world, a negation connective can be introduced that expresses precisely non-satisfiability of f in $(\mathcal{W}, w)$ (see [4])[2].

We now define the set of *generating defaults* at level h, i.e. the defaults at h plus the defaults and negations of them from levels strictly under h.

Definition 120 Let S be a Σ-hierarchic specification of an institution $(\mathcal{I}, \mathsf{neg})$ having negation and $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority. The set of *generating defaults* at level h denoted by $\mathsf{gendf}(S, h)$ is the union of the defaults at level h and under h with the negations of the defaults strictly under h: $\mathsf{gendf}(S, h) = \mathsf{df}(S, h) \cup \mathsf{df}(S, h^-) \cup \mathsf{neg}_\Sigma(\mathsf{df}(S, h^-))$, where $\mathsf{df}(S, h^-)$ is the set of defaults at levels strictly under h and $\mathsf{neg}_\Sigma(\mathsf{df}(S, h^-)) = \{\mathsf{neg}_\Sigma(d) : d \in \mathsf{df}(S, h^-)\}$. ▪

The generating defaults are illustrated in the following example.

Example 121 Recall the (simplified version of the) specification MAMMALS displayed in example 100. In the next diagram (figure 2.18) the generating defaults at each level are displayed. The set of generating defaults at level `Mammals` is

$$\mathcal{M} = \{\neg\texttt{Fl(u)}, \neg\texttt{Fl(b)}, \texttt{Fl(b)}, \neg\texttt{Fl(bm)}, \texttt{Fl(bm)}, \texttt{Dr(u)}, \neg\texttt{Dr(u)}, \texttt{Dr(bm)}, \neg\texttt{Dr(bm)}\}$$

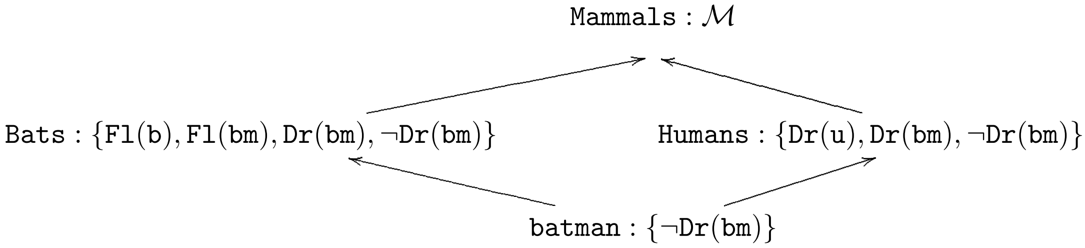

Figure 2.18: Generating Defaults in MAMMALS

$\triangle$

We now show that the differential preference at level h in $S^{\ominus}$ is the preference induced by the flat specification with axioms the axioms of S and defaults the generating defaults at level h.

Lemma 122 Let S be a Σ-hierarchic specification of an institution $(\mathcal{I}, \mathsf{neg})$ having negation and $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority. Denote by S_h the (flat) Σ-specification $S_h = (\mathsf{ax}(S), \mathsf{gendf}(S, h))$, with axioms the axioms from S and defaults the generating defaults at level $h \in H$. Then $(|S^{\ominus}|, \mathsf{rl}(S^{\ominus}, h)) = S_h^{\star}$, i.e. the pre-order corresponding to $S^{\ominus}$ at level h is the preference relation of $S_h^{\star}$.

Proof Clearly $|S^{\ominus}| = \mathsf{ax}(S)^{\bullet}$ so we may concentrate in the preference. Let $\sqsubseteq_h^{\ominus}$ denote the differential preference $\mathsf{rl}(S^{\ominus}, h)$ of S at level h and let $\sqsubseteq_{\mathsf{gen}(h)}$ denote preference associated with S_h, defined by $m \sqsubseteq_{\mathsf{gen}(h)} n$ iff for every $d \in \mathsf{gendf}(S, h)$ if $m \vDash d$ then $n \vDash d$. We have to show that $m \sqsubseteq_h^{\ominus} n$ iff $m \sqsubseteq_{\mathsf{gen}(h)} n$.

Since $\mathsf{gendf}(S, h) = \mathsf{df}(S, h) \cup \mathsf{df}(S, h^-) \cup \mathsf{neg}(\mathsf{df}(S, h^-))$ the condition defining $\sqsubseteq_{\mathsf{gen}(h)}$ is the conjunction of the conditions for $\mathsf{df}(S, h)$, $\mathsf{df}(S, h^-)$ and $\mathsf{neg}(\mathsf{df}(S, h^-))$. The last case is the only interesting one. It is: for every $\mathsf{neg}(d) \in \mathsf{neg}(\mathsf{df}(S, h^-))$ if $m \vDash \mathsf{neg}(d)$ then $n \vDash \mathsf{neg}(d)$. This is equivalent to: for every $d \in \mathsf{df}(S, h^-)$ if $n \vDash d$ then $m \vDash d$ (note that n is reversed with m) by contraposition and the definition of satisfaction of $\mathsf{neg}(d)$. This last condition is the same as $\mathsf{df}(S, h')(n) \subseteq \mathsf{df}(S, h')(m)$ for every $h' \prec h$. In this way $m \sqsubseteq_{\mathsf{gen}(h)} n$ iff $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ and $\mathsf{df}(S, h')(m) \subseteq \mathsf{df}(S, h')(n)$ and $\mathsf{df}(S, h')(n) \subseteq \mathsf{df}(S, h')(m)$ for every $h' \prec h$. This is equivalent to $\mathsf{df}(S, h)(m) \subseteq \mathsf{df}(S, h)(n)$ and $\mathsf{df}(S, h')(m) = \mathsf{df}(S, h')(n)$ for every $h' \prec h$. This condition is precisely that defining $m \sqsubseteq_h^{\ominus} n$. ✓

Since the defaults in level h of $S^{\ominus\ominus}$ are precisely those implicit in $(|S^{\ominus}|, \mathsf{rl}(S^{\ominus}, h))$ these correspond to disjunctions of conjunctions of the defaults in $\mathsf{gendf}(S, h)$ in the context of the axioms from S.

[2]Thanks to Cristina Sernadas and José Carmo for their help clarifying the question.

Theorem 123 Let S be a Σ-hierarchic specification of an institution $(\mathcal{I}, \mathsf{neg})$ having negation and $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority. The set of defaults at level $h \in H$ from the theory of S, $\mathsf{df}(S^{\ominus\ominus}, h)$ is the set of Σ-formulas d satisfying the following property:

There is a set $\Delta \subseteq 2^{\mathsf{gendf}(S,h)}$ of subsets of the set of generating defaults from S at level h such that

$$\mathsf{ax}(S)^{\bullet} \cap \{d\}^{\bullet} = \mathsf{ax}(S)^{\bullet} \cap (\cup_{D\in\Delta} D^{\bullet}).$$

Moreover if the set of generating defaults of S is finite then Δ is finite and each set of defaults $D \in \Delta$ is also finite.

Proof Trivial from lemma 122 above and from the characterization of defaults implicit in a preference relation in theorem 51. ✓

Remark 124 The condition $\mathsf{ax}(S)^{\bullet} \cap \{d\}^{\bullet} = \mathsf{ax}(S)^{\bullet} \cap (\cup_{D\in\Delta} D^{\bullet})$ is the same as stating that the "formulas" $\mathsf{ax}(S) \wedge d$ and $\mathsf{ax}(S) \wedge (\vee_{D\in\Delta} D)$ are semantically equivalent (sets denote the conjunction of their member formulas). If S is finite such formulas can be written in institutions having disjunction, conjunction and negation (recall that $\Delta \subseteq 2^{\mathsf{gendf}(S,h)}$). The test of semantical equivalence can be automated for those institutions that furthermore have a sound and complete theorem prover. This corresponds to check whether $\mathsf{ax}(S) \wedge d \vdash \mathsf{ax}(S) \wedge (\vee_{D\in\Delta} D)$ and $\mathsf{ax}(S) \wedge (\vee_{D\in\Delta} D) \vdash \mathsf{ax}(S) \wedge d$. We have, therefore, in such institutions a procedure for deciding whether $\mathsf{ax}(S)^{\bullet} \cap \{d\}^{\bullet} = \mathsf{ax}(S)^{\bullet} \cap (\cup_{D\in\Delta} D^{\bullet})$. If the underlying logic is decidable this procedure always halts. If the underlying logic is semi-decidable this procedure halts if $\mathsf{ax}(S)^{\bullet} \cap \{d\}^{\bullet} = \mathsf{ax}(S)^{\bullet} \cap (\cup_{D\in\Delta} D^{\bullet})$. The question of whether d is a default from the theory of S at level h can also be automated. One has simply to generate all subsets $\Delta \subseteq 2^{\mathsf{gendf}(S,h)}$ (recall that S is assumed to be finite and so is $\mathsf{gendf}(S, h)$) and apply to them the previous procedure). If the underlying logic is decidable so is this question. Semi-decidability is also maintained (in this case we have to test "in parallel" for all $\Delta \subseteq 2^{\mathsf{gendf}(S,h)}$ if $\mathsf{ax}(S) \wedge d$ and $\mathsf{ax}(S) \wedge (\vee_{D\in\Delta} D)$ are semantically equivalent).

As we have seen in remark 119 not all institutions have the needed negation. We proceed to generalize the characterization of the defaults from the theory of S to any such institution. This is done as follows: if the institution $\mathcal{I}$ at hand does not have negation we extend it to an institution $\widetilde{\mathcal{I}}$ with negation. We then see that the theory of S in $\mathcal{I}$ is the restriction of the theory of S in $\widetilde{\mathcal{I}}$ to the formulas (over the appropriate signature) from $\mathcal{I}$.

Definition 125 Given an institution $\mathcal{I} = (\mathsf{Sign}^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \{\models^{\mathcal{I}}_{\Sigma} : \Sigma \in |\mathsf{Sign}|\})$ its *extension by negation* is the institution $\widetilde{\mathcal{I}}$ consisting of

- the same category $\mathsf{Sign}^{\mathcal{I}}$ of signature as $\mathcal{I}$,

- the functor $\mathsf{Sen}^{\widetilde{\mathcal{I}}} \colon \mathsf{Sign}^{\mathcal{I}} \to \mathsf{Set}$ that assigns to each signature Σ the set of formulas $\mathsf{Sen}^{\mathcal{I}}(\Sigma) \cup {\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma))$, where ${\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma))$ is the set $\{{\sim}(f) : f \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)\}$ and ${\sim}(f)$ is the concatenation of the connective of negation $\sim$ with "(", f and ")". The symbol $\sim$ is appropriately chosen so that $\mathsf{Sen}^{\mathcal{I}}(\Sigma)$ and ${\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma))$ are disjoint. Furthermore given a signature morphism $\phi : \Sigma_1 \to \Sigma_2$ its image by $\mathsf{Sen}^{\widetilde{\mathcal{I}}}$ is the function $\mathsf{Sen}^{\widetilde{\mathcal{I}}}(\phi) = \widetilde{\Phi} : (\mathsf{Sen}^{\mathcal{I}}(\Sigma_1) \cup {\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma_1))) \to (\mathsf{Sen}^{\mathcal{I}}(\Sigma_2) \cup {\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma_2)))$ defined by:

  - $\widetilde{\Phi}(f_1) = \Phi(f_1)$ where $\Phi = \mathsf{Sen}^{\mathcal{I}}(\phi)$ for $f_1 \in \mathsf{Sen}^{\mathcal{I}}(\Sigma_1)$ and
  - $\widetilde{\Phi}({\sim}(f_1)) = {\sim}(\Phi(f_1))$ with Φ as above for ${\sim}(f_1) \in {\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma_1))$,

- the same functor $\mathsf{Mod} \colon \mathsf{Sign} \to \mathsf{Cat}^{op}$ as $\mathcal{I}$,

- the relations $\models^{\widehat{\mathcal{I}}}_{\Sigma} \subseteq |\mathsf{Mod}(\Sigma)| \times (\mathsf{Sen}^{\mathcal{I}}(\Sigma) \cup {\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma)))$, defined by

  - $m \models^{\widehat{\mathcal{I}}}_{\Sigma} f$ iff $m \models^{\mathcal{I}}_{\Sigma} f$ for $f \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ and
  - $m \models^{\widehat{\mathcal{I}}}_{\Sigma} {\sim}(f)$ iff $m \not\models^{\mathcal{I}}_{\Sigma} f$ for ${\sim}(f) \in {\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma))$. ∎

Proof We have to show that $\widetilde{\mathcal{I}}$ is an institution. This means checking that $\mathsf{Sen}^{\widetilde{\mathcal{I}}} : \mathsf{Sign}^{\mathcal{I}} \to \mathsf{Set}$ is a functor (which we omit for obvious) and the satisfaction condition. This condition for $f \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ holds because of the satisfaction condition of the institution $\mathcal{I}$. For ${\sim}(f) \in {\sim}(\mathsf{Sen}^{\mathcal{I}}(\Sigma))$ we proceed as follows. Given a signature morphism $\phi : \Sigma_1 \to \Sigma_2$ let $\Phi = \mathsf{Sen}^{\mathcal{I}}(\phi)$ and $\widehat{\Phi} = \mathsf{Sen}^{\widetilde{\mathcal{I}}}(\phi)$. The satisfaction condition is now $m_2 \models^{\widetilde{\mathcal{I}}}_{\Sigma_2} \widehat{\Phi}({\sim}(f))$ iff $\mathsf{Mod}^{\mathcal{I}}(\phi)(m_2) \models^{\widetilde{\mathcal{I}}}_{\Sigma_1} {\sim}(f)$.

On one hand $m_2 \models^{\widetilde{\mathcal{I}}}_{\Sigma_2} \widehat{\Phi}({\sim}(f))$ is equivalent to $m_2 \models^{\widetilde{\mathcal{I}}}_{\Sigma_2} {\sim}(\Phi(f))$ and this to $m_2 \not\models^{\mathcal{I}}_{\Sigma_2} \Phi(f)$. On the other hand $\mathsf{Mod}^{\mathcal{I}}(\phi)(m_2) \models^{\widetilde{\mathcal{I}}}_{\Sigma_1} {\sim}(f)$ is equivalent to $\mathsf{Mod}^{\mathcal{I}}(\phi)(m_2) \not\models^{\mathcal{I}}_{\Sigma_1} f$. Therefore the satisfaction condition becomes $m_2 \not\models^{\mathcal{I}}_{\Sigma_2} \Phi(f)$ iff $\mathsf{Mod}^{\mathcal{I}}(\phi)(m_2) \not\models^{\mathcal{I}}_{\Sigma_1} f$. This is the satisfaction condition for the institution $\mathcal{I}$. ✓

We now see that we can use the extended institution to determine defaults from the theory of S.

Theorem 126 Let S be a Σ-hierarchic specification of an institution $\mathcal{I}$, and $(H, \preceq) = \mathsf{po}(S)$ its partial order of priority. Let $\widetilde{\mathcal{I}}$ be the institution that extends $\mathcal{I}$ by negation. Then $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is a default from $h \in H$ from the theory of S (evaluated in $\mathcal{I}$) iff d is a default from $h \in H$ from the theory of S evaluated in $\widetilde{\mathcal{I}}$.

Proof Firstly we remark that the institution $\mathcal{I}$ extended by negation is an institution with negation. In fact $(\widetilde{\mathcal{I}}, \mathsf{neg})$ is an institution with negation where $\mathsf{neg} = \{\mathsf{neg}_\Sigma : \mathsf{Sen}^{\mathcal{I}}(\Sigma) \cup \sim(\mathsf{Sen}^{\mathcal{I}}(\Sigma)) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma) \cup \sim(\mathsf{Sen}^{\mathcal{I}}(\Sigma)) \mid \Sigma \in |\mathsf{Sign}^{\mathcal{I}}|\}$ and each neg_Σ is defined by

- $\mathsf{neg}_\Sigma(f) = \sim(f)$, $f \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ and
- $\mathsf{neg}_\Sigma(\sim(f)) = f$ for $\sim(f) \in \sim(\mathsf{Sen}^{\mathcal{I}}(\Sigma))$.

Note that neg_Σ is injective.

We now proceed to see that $S^{\ominus}$, the hierarchy of differential preferences of S is precisely the same when evaluated at $\mathcal{I}$ or $\widetilde{\mathcal{I}}$. To see this recall from lemma 93 that $S^{\ominus} = S^{\odot^{\boxminus}}$, where $S^{\odot}$ is the hierarchy of local preferences of S. One one hand $|S^{\odot}|$ is the class of models of the axioms from S. These are formulas without the new connective $\sim$ so their satisfaction coincides in $\widetilde{\mathcal{I}}$ and $\mathcal{I}$ and these two institutions have the same interpretation structures. On the other hand the preferences associated by $S^{\odot}$ to level h compare these models by satisfaction of the defaults of S at level h. These defaults also are not negated formulas and their satisfaction coincides again in both institutions. In this way $S^{\odot}$ is the same when evaluated in $\widetilde{\mathcal{I}}$ or in $\mathcal{I}$.

Identity of $S^{\odot}$ implies identity of $S^{\ominus} = S^{\odot^{\boxminus}}$ since the operator $\boxminus$ depends only on the pre-orders in the argument.

Having established that S has the same hierarchy of differential preferences in both institutions we also conclude that the pre-order $\mathcal{R} = (|S^{\ominus}|, \mathsf{rl}(S^{\ominus}, h))$ is also the same in both institutions. We now see that the defaults implicit in $\mathcal{R}$ in institution $\mathcal{I}$ are precisely the defaults from $\mathsf{Sen}^{\mathcal{I}}(\Sigma)$ that are implicit in $\mathcal{R}$ in the institution $\widetilde{\mathcal{I}}$.

Recall that d is a default implicit in $\mathcal{R}$ according to $\mathcal{I}$ iff $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ and for every $m, n \in |\mathcal{R}|$ such that $(m, n) \in \mathsf{rl}(\mathcal{R})$ whenever $m \vDash^{\mathcal{I}}_{\Sigma} d$ then $n \vDash^{\mathcal{I}}_{\Sigma} d$. Recalling that satisfaction of formulas $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ coincides in $\mathcal{I}$ and $\widetilde{\mathcal{I}}$ this condition is equivalent to: $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ and for every $m, n \in |\mathcal{R}|$ such that $(m, n) \in \mathsf{rl}(\mathcal{R})$ whenever $m \vDash^{\widetilde{\mathcal{I}}}_{\Sigma} d$ then $n \vDash^{\widetilde{\mathcal{I}}}_{\Sigma} d$. This is the condition stating that $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ and d is an implicit default in $\mathcal{R}$ according to $\widetilde{\mathcal{I}}$ and therefore $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is an implicit default in $\mathcal{R}$ according to $\mathcal{I}$ iff d is an implicit default in $\mathcal{R}$ according to $\widetilde{\mathcal{I}}$.

Therefore $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is an implicit default in $\mathcal{R}$ according to $\mathcal{I}$ iff d is an implicit default in $\mathcal{R}$ according to $\widetilde{\mathcal{I}}$. The last assertion is equivalent to $d \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ is a default from the theory of S in $\widetilde{\mathcal{I}}$ at level h. ✓

Note that the set of defaults at level h from the theory of S when evaluated in $\widetilde{\mathcal{I}}$ does not coincide, in general, with the set of defaults from the theory of S, at that level, when evaluated in $\mathcal{I}$. This is because the institution $\widetilde{\mathcal{I}}$ has more formulas (in each signature) than $\mathcal{I}$. In particular it has the newly introduced negations. What the theorem above states is that the defaults of the theory of S that are non-negated formulas coincide in both institutions. The next example helps to clarify this question.

Example 127 Consider now the specification of examples 77 and 127 now with the difference that it is seen as a specification of propositional modal logic. We have an empty set of axioms and two defaults p, q, propositional symbols, where p is considered of lower priority than q.

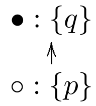

Figure 2.19: p lower than q

Its hierarchy of differential preferences is again displayed in the following diagram 2.20.

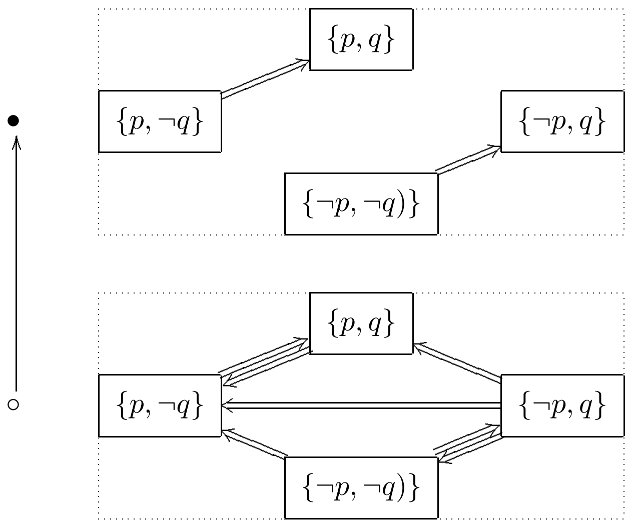

Figure 2.20: Hierarchy of Differential Preferences

This is precisely the same figure as for the propositional case. The difference is in the interpretation structures that are being compared: in this case modal interpretation structures (before valuations). At this point we have to make clear which modal interpretation structures we have in mind (recall remark 119). We consider both cases:

1. Assume that interpretation structures are structures $\mathcal{W}$ comprising a class of models, a visibility relation among them and an assignment of sets of propositional symbols to models. Satisfaction in $\mathcal{W}$ is satisfaction in each model from $\mathcal{W}$. Assume moreover that the modal logic at hand has reflexive visibility relations (modal logic T). The theory of the previous specification in this logic will have at level $\circ$ the defaults implicit in the differential preference at level $\circ$. This corresponds to the preference relation induced

by the flat specification $(\emptyset, \{p\})$. We obtain again the formulas semantically equivalent to one of $\mathsf{true}, \mathsf{false}$ and p. Among these is also $\Box p$. In fact $\mathcal{W} \vDash f$ iff f holds in all models from $\mathcal{W}$ iff $\mathcal{W} \vDash \Box f$.

At level $\bullet$ we face the problem of expressing non-satisfiability of p. Since in this logic there is no formula expressing this property we extend it with negation. We conclude that the implicit defaults at this level are the formulas semantically equivalent to disjunctions of conjunctions of the defaults in $\{p, \sim(p), q\}$ (in the extended institution). These are semantically equivalent (in the modal logic T) to one of $\mathsf{true}, \mathsf{false}$, p, $\sim(p)$, q, $p \wedge q$, $\sim(p) \wedge q$, $p \vee q$, $\sim(p) \vee q$. In our logic the formulas that can be expressed are a strict subset of these, namely those semantically equivalent to one of $\mathsf{true}, \mathsf{false}$, p, q, $p \wedge q$, $p \vee q$. These include $\Box f$ where f is any of the previous formulas.

The situation is different if the logic has negation. For example in linear temporal logic with both a past (H) and future (G) necessity operator we would obtain the formulas semantically equivalent to one of $\mathsf{true}, \mathsf{false}$, p, $\neg HGp$, q, $p \wedge q$, $(\neg HGp) \wedge q$, $p \vee q$, $(\neg HGp) \vee q$ as the defaults implicit in level $\bullet$ (see remark 119). The formula $\neg HGp$ is semantically equivalent (linear time) to $\neg Hp \vee \neg p \vee \neg Gp$ that is simpler. It states that there is a state either in the past, present or future (of the present state) where f does not hold. In this way also $\neg Hp \vee \neg p \vee \neg Gp$ is a default implicit in level $\bullet$. Note finally that both Hf and Gf are semantically equivalent to f.

2. When the modal logic T (or other) is formalized by interpretation structures with a distinguished world, the negation connective has the meaning of non-satisfiability. Therefore the implicit defaults at level $\bullet$ are simply the formulas semantically equivalent to one of $\mathsf{true}, \mathsf{false}$, p, $\neg p$, q, $p \wedge q$, $\neg p \wedge q$, $p \vee q$, $\neg p \vee q$. The defaults implicit at level $\circ$ are the formulas semantically equivalent to one of $\mathsf{true}, \mathsf{false}$ and p. We note that it is no longer the case that, in general, $\Box f$ is semantically equivalent to f. △

Remark 128 Note that the construction of theorem 126 is not helpful in finding an automatic method to determine whether a formula is a default from the theory of a finite specification S. The procedure outlined in remark 124 applies to institutions having conjunctions, disjunctions and negations and, furthermore a sound and complete proof system (preferably decidable or semi-decidable). First of all our construction of the extended institution does not add conjunctions or disjunctions of the negated formulas with non-negated ones. Secondly and more important there is no canonical way of building a proof system for the extended institution.

2.4 Extensions

The semantics assigned to a hierarchic specification S in section 2.3.1 includes information for each priority level. We might want to use this information to reason about S, for example by asking whether a certain formula is entailed already at a certain priority level. However, the usual consequences of S do no longer carry any information about the original structure of that specification. One is usually interested in knowing whether a certain formula is a skeptical or credulous consequence of S. These consequences *(skeptical and credulous)* are defined and studied in the following section . They are derived from the (maximal equivalence classes) of the lexicographic preference of S (see definition 76).

We define *extension presentations* and *extensions* in section 2.4.1 and *credulous* and *skeptical consequences* in section 2.4.2. We note that the *extensions* of the prioritized case correspond to a selection of the extensions for the flat case in section 2.4.3. Some consequences of this fact are presented in section 2.4.4.

2.4.1 Definition

The extensions of a hierarchic specification are the classical theories of some *extension presentation*. Extension presentations correspond to maximal consistent sets of axioms and defaults. The ordering on such sets is no longer inclusion, as in the flat case, since the priority ordering has to be respected. We begin by defining extensions and extension presentations on the semantic level, using the lexicographic preference $\mathsf{lex}^\circ(S)$ of S. An alternative equivalent definition of extension presentations using an ordering on sets of axioms and defaults will be presented below.

We begin with the following auxiliary lemma.

Lemma 129 Let S be a hierarchic specification and let $\sqsubseteq^\circ = \mathsf{rl}(\mathsf{lex}^\circ(S))$ be the lexicographic preference of S. Moreover let $\equiv^\circ$ and $\sqsubset^\circ$ be the corresponding equivalence and strict relations. Let $m, n \in |\mathsf{lex}^\circ(S)|$. Then:

1. $m \equiv^\circ n$ iff $\mathsf{df}(S)(m) = \mathsf{df}(S)(n)$,
2. $m \sqsubset^\circ n$ iff $m \sqsubseteq^\circ n$ and there exists $h \in H$ with $\mathsf{df}(S,h)(m) \subset \mathsf{df}(S,h)(n)$ and for every $h' \prec h$, $\mathsf{df}(S,h')(m) \subseteq \mathsf{df}(S,h')(n)$.

Proof The intended results are simple consequences of lemma 87 that presents a similar result for each lexicographic preference in the hierarchy of lexicographic preferences $S^\oplus$ and the fact that the lexicographic preference of S is the intersection of those relations (presented in lemma 98). ✓

Extension presentations are the sets of axioms and defaults holding in some maximal equivalence class of $\mathsf{lex}^\circ(S)$. Their theories are the extensions.

Definition 130 Let S be a Σ-hierarchic specification and $\mathsf{max}(\mathsf{lex}^\circ(S))$ the set of maximal equivalence classes of the lexicographic preference of S.

- a set E of Σ-formulas is an *extension presentation* of S iff $E = \mathsf{ax}(S) \cup \mathsf{df}(S)(M)$, where $\mathsf{df}(S)(M)$ is the set of defaults from S holding in all interpretation structures of some *maximal equivalence class* $M \in \mathsf{max}(\mathsf{lex}^\circ(S))$,
- an *extension* $\mathcal{E} = E^{\bullet\bullet}$ of S is the (classic) theory of some extension presentation E of S. ▪

Recall from definition 61 that, if S is inconsistent (i.e. $\mathsf{ax}(S)$ is inconsistent), $\mathsf{max}(\mathsf{lex}^\circ(S))$ contains one equivalence class, the empty equivalence class. Therefore, for that case, there will be precisely one extension presentation of S, namely $\mathsf{ax}(S) \cup \mathsf{df}(S)$, the union of the axioms with all defaults of S.

We show that an extension $\mathcal{E}$ is the set of formulas holding in some maximal equivalence class of $\mathsf{lex}^\circ(S)$.

Lemma 131 A set $\mathcal{E}$ of Σ-formulas is an extension of a Σ-hierarchic specification S iff $\mathcal{E} = M^\bullet$ where M is a maximal equivalence class of $\mathsf{lex}^\circ(S)$.

Proof We prove firstly that $E^\bullet = M$. If $\mathsf{ax}(S)$ is inconsistent the intended property follows trivially. Therefore we may assume $\mathsf{ax}(S)$ consistent. In this case $M = [m]$. We firstly note that given $[m] \in \mathsf{max}(\mathsf{lex}^\circ(S))$ then $\mathsf{df}(S)(m) = \mathsf{df}(S)([m])$, i.e. the defaults from S satisfied by all models in $[m]$ are those satisfied by m. This follows from the fact that two models are equivalent w.r.t. $\mathsf{lex}^\circ(S)$ iff they satisfy precisely the same defaults (lemma 129). Consider now a model m' of the extension presentation E induced by $[m]$ (i.e. $m \vDash E = \mathsf{ax}(S) \cup \mathsf{df}(S)([m])$). Since $m' \vDash E$ this model satisfies at least the defaults satisfied by m. Therefore $m \sqsubseteq^\circ m'$. From maximality of $[m]$ it follows $m' \in [m]$. This shows that $E^\bullet \subseteq [m]$. The reverse inclusion is obvious from $E = \mathsf{ax}(S) \cup \mathsf{df}(S)([m])$. In this way $E^\bullet = [m]$. This ends the first part of the proof.

Finally assume that $\mathcal{E}$ is an extension of S. Then $\mathcal{E} = E^{\bullet\bullet}$ where $E = \mathsf{ax}(S) \cup \mathsf{df}(S)(M)$ is an extension presentation of S and $M \in \mathsf{max}(\mathsf{lex}^\circ(S))$. From the first part of the proof $E^\bullet = M$ and $\mathcal{E} = M^\bullet$. On the other hand if $\mathcal{E} = M^\bullet$ with $M \in \mathsf{max}(\mathsf{lex}^\circ(S))$ then $M = E^\bullet$ for $E = \mathsf{ax}(S) \cup \mathsf{df}(S)(M)$. Therefore $\mathcal{E} = E^{\bullet\bullet}$. ✓

The extension presentations for the flat case (see definition 54) are the maximal (w.r.t. inclusion) sets of axioms and defaults that are consistent if the axioms are so. In the following we observe that the extension presentations in the hierarchic case are also maximal sets of axioms and defaults, but according to a different relation. This relation is similar to the lexicographic ordering of corresponding models and is motivated as follows. Consider two models m_1 and m_2 of the axioms from a hierarchic specification S such that $m_1 \sqsubseteq^\circ m_2$ (m_2 is preferred to m_1 according to the lexicographic preference of S). Then the set E_2 of axioms and defaults from S holding in m_2 is "preferred" to the set E_1 of axioms and defaults from S holding in m_1. The lexicographic preference relating m_1 and m_2 can be rewritten in terms of the sets E_1 and E_2.

Definition 132 Let S be a hierarchic specification, $(H, \preceq)$ its partial order of priority and recall that $\mathsf{df}(S)$ is the set of all defaults (from any level) from S. Let E_1, E_2 with $\mathsf{ax}(S) \subseteq E_1, E_2 \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)$ be sets of axioms and defaults from S. Then $E_1 \subseteq^\circ E_2$ iff for every level $h \in H$ if $E_1 \cap \mathsf{df}(h, S) \not\subseteq E_2 \cap \mathsf{df}(h, S)$ then there is $h' \prec h$ and $E_1 \cap \mathsf{df}(h', S) \subset E_2 \cap \mathsf{df}(h', S)$.

This relation is a partial order. ■

Proof Reflexivity is obvious. Transitivity is formally similar to the proof of the corresponding property of lexicographic preference on interpretation structures (see definition 76). Anti-symmetry is simple by well founded induction: assume that $E_1 \subseteq^\circ E_2$ and $E_2 \subseteq^\circ E_1$. If h is minimal then $E_1 \cap \mathsf{df}(h, S) \subseteq E_2 \cap \mathsf{df}(h, S)$ and $E_2 \cap \mathsf{df}(h, S) \subseteq E_1 \cap \mathsf{df}(h, S)$. Therefore $E_2 \cap \mathsf{df}(h, S) = E_1 \cap \mathsf{df}(h, S)$. The same conclusion holds for non-minimal h, since, by the induction hypothesis, $E_2 \cap \mathsf{df}(h', S) = E_1 \cap \mathsf{df}(h', S)$ and, for this reason there can be no level h' under h with $E_2 \cap \mathsf{df}(h', S) \subset E_1 \cap \mathsf{df}(h', S)$ (or vice versa). We conclude that, for all $h \in H$ $E_2 \cap \mathsf{df}(h, S) = E_1 \cap \mathsf{df}(h, S)$, implying $E_1 = E_2$. ✓

The sets E that are maximal according to the ordering above are the extension presentations of S.

Theorem 133 A set E with $\mathsf{ax}(S) \subseteq E \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)$ is an extension presentation of the hierarchic specification S iff

- If $\mathsf{ax}(S)$ is consistent then E is consistent and maximal among the consistent E' with $\mathsf{ax}(S) \subseteq E' \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)$ (i.e. given a such a consistent E' if $E \subseteq^\circ E'$ then $E = E'$),
- if $\mathsf{ax}(S)$ is not consistent then $E = \mathsf{ax}(S) \cup \mathsf{df}(S)$.

Proof Only the case of $\mathsf{ax}(S)$ consistent is non-trivial.

⇒ Assume that E is an extension presentation of S and there exists a consistent E' such that $E \subseteq^\circ E'$ (and $\mathsf{ax}(S) \subseteq E' \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)$). Let m be a model of E and m' a model of E'. Since $m \vDash E$ we have that $[m]$ is maximal according to $\mathsf{lex}^\circ(S)$

and $E = \mathsf{ax}(S) \cup \mathsf{df}([m])(=)\mathsf{ax}(S) \cup \mathsf{df}(S)(m)$. This implies that $\mathsf{df}(S,h)(m) = E \cap \mathsf{df}(S,h)$ for each h. For m' we have $\mathsf{df}(S,h)(m') \supseteq E' \cap \mathsf{df}(S,h)$ for each h. From this and $E \subseteq^\circ E'$ it follows $m \sqsubseteq^\circ m'$. Since E is an extension presentation the equivalence class of m is maximal w.r.t. $\mathsf{lex}^\circ(S)$. Therefore $m' \in [m]$ and $\mathsf{ax}(S) \cup \mathsf{df}(S)(m') = \mathsf{ax}(S) \cup \mathsf{df}(S)(m) = E$ (see lemma 129). Since $E' \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)(m') = E$ we conclude $E' \subseteq^\circ E$ and therefore $E' = E$.

$\Leftarrow$ Let E be consistent and maximal among the E' with $\mathsf{ax}(S) \subseteq E' \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)$. Let m be a model of E and consider $E_1 = \mathsf{ax}(S) \cup \mathsf{df}(S)(m)$, the set of axioms plus the defaults holding in m. Clearly $E \subseteq E_1$ and therefore $E \subseteq^\circ E_1$. Maximality of E implies $E_1 = E$. In this way any model of E satisfies (the axioms and) precisely the same defaults from S. Therefore $E^\bullet$ is an equivalence class of $\mathsf{lex}^\circ(S)$. We now see that it is maximal. Consider any m' with $m \sqsubseteq^\circ m'$ where m is a model of E. It is easy to check that $E_2 = \mathsf{ax}(S) \cup \mathsf{df}(S)m'$ is such that $E \subseteq^\circ E_2$. Again by maximality of E we have that $E_2 = E$ and this implies $m' \in E^\bullet = [m]$ for any m' with $m \sqsubseteq^\circ m'$. This means that $E^\bullet = [m]$ is a maximal equivalence class of $\mathsf{lex}^\circ(S)$. ✓

2.4.2 Consequences

The credulous and skeptical consequences of a specification S are defined in terms of the *extensions* of S. The obvious definition of these relations follow.

Definition 134 A $\mathsf{sg}(S)$-formula f is a

- *credulous consequence* of a hierarchic specification S, written $S \vdash_{cr} f$, iff f belongs to some extension of S, and a
- *skeptical consequence* of a hierarchic specification S, written $S \vdash_{sk} f$, iff f belongs to all extensions of S. ■

2.4.3 Selection Function

Extension presentations of hierarchic specifications are maximal sets of defaults (and axioms) that, furthermore, respect the relations of priority. Therefore they are also extension presentations of the specification obtained by forgetting the priority structure of the original hierarchic specification.

Lemma 135 Let E be an extension presentation of the Σ-hierarchic specification S. Then E is an extension presentation of the Σ-specification $(\mathsf{ax}(S), \mathsf{df}(S))$ consisting of the axioms from S and the defaults from all levels from S. Moreover if $\mathcal{E}$ is an extension of S then $\mathcal{E}$ is an extension of $(\mathsf{ax}(S), \mathsf{df}(S))$.

Proof The property for the extensions of S follows trivially from the corresponding property for extension presentations: Extensions are the theories of the extension presentations (both in the hierarchic and in the flat case).

Assume $\mathsf{ax}(S)$ consistent (the other case is trivial). Let E be an extension presentation of S and consider an E' such that $E \subseteq E'$ and $\mathsf{ax}(S) \subseteq E' \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)$. Therefore $E \subseteq^\circ E'$ and from theorem 133 it must be $E = E'$. We have proved that E is maximal w.r.t. inclusion among such sets E'. This is the definition 54 of extension presentation of $(\mathsf{ax}(S), \mathsf{df}(S))$. ✓

The other direction does not hold in general (see any of the examples involving batman, e.g. example 75). In this way the priority ordering has the effect of "selecting" some of the extensions of $(\mathsf{ax}(S), \mathsf{df}(S))$. This selection corresponds to check (according to theorem 133) which of the extensions of $(\mathsf{ax}(S), \mathsf{df}(S))$ are maximal according to $\subseteq^\circ$. We will see in the next section 2.5.2 (theorem 139) that, in the case of compact logics, it is enough to compare, according to $\subseteq^\circ$, the extensions of $(\mathsf{ax}(S), \mathsf{df}(S))$ among themselves.

2.4.4 Properties

The following properties of extensions and extension presentations are simple consequences of the lemma 135 above and corresponding properties for specifications.

Theorem 136

1. **Consistency.** An extension presentation E of a Σ-hierarchic specification S is consistent iff $\mathsf{ax}(S)$ is consistent.

2. **Maximality.** If E, E' are extension presentations of a Σ-hierarchic specification S and $E \subseteq E'$ then $E = E'$;

3. **Orthogonality.** If E, E' are extension presentations of a Σ-hierarchic specification S and $E \neq E'$ then $E \cup E'$ is inconsistent.

4. The previous properties also hold when E, E' are *extensions* of a Σ-specification S.

Proof Trivial from lemma 135 above since these properties hold for extension presentations of specifications (see theorem 55). One has simply to see E and E' as extension presentations of $(\mathsf{ax}(S), \mathsf{df}(S))$. Similarly for the extensions of S. ✓

2.5 Compact Institutions

In the following we are concerned with the properties of extensions in institutions where the compactness property holds. In section 2.5.1 we show that any hierarchic specification has at least one extension. Further properties of such institutions are studied in sections 2.5.2 and 2.5.3.

2.5.1 Existence of Extensions

We have seen in section 1.3.3 that the preference relation associated with specifications enjoys the property that, given a model of the axioms then there exists a maximal equivalence class above it. This property implies existence of extensions, coverage, semi-monotonicity and cumulativity (see theorem 67). The corresponding property for the lexicographic preference associated with a hierarchic specification is now presented.

Lemma 137 Let $\mathcal{I}$ be an institution where the compactness property holds. Given a hierarchic specification S from $\mathcal{I}$ with $\mathsf{ax}(S)$ consistent, let $[m]$ be an arbitrary equivalence class from the lexicographic preference $\mathsf{lex}^\circ(S)$ of S. Then there is a maximal equivalence class $[m^\uparrow]$ of $[\mathsf{lex}^\circ(S)]$ such that $([m],[m^\uparrow]) \in \mathsf{rl}([\mathsf{lex}^\circ(S)])$.

Proof[3] The proof uses Zorn's lemma with the restriction of the partial order $[\mathsf{lex}^\circ(S)]$ of the equivalence classes of the lexicographic preference $\mathsf{lex}^\circ(S)$ to the set $\mathcal{B}([m]) = \{[m'] : m \sqsubseteq^\circ m'\}$ of the equivalence classes above $[m]$. Let $\mathcal{C}$ be a chain in $\mathcal{B}([m])$. We have to exhibit a upper bound of $\mathcal{C}$ and we do this by displaying a presentation E having as class of models the looked for upper bound. This presentation is defined as follows: $E = \mathsf{ax}(S) \cup D$ where

$$D = \{d \in \mathsf{df}(S) : \exists\ [m_1] \in \mathcal{C} \text{ such that } m_1 \vDash d \text{ and } m_1' \vDash d \\ \text{for all } [m_1'] \in \mathcal{C} \text{ such that } m_1 \sqsubseteq^\circ m_1'\}.$$

Note that given an equivalence class $[m']$ of the lexicographic preference of S then any $m'' \in [m']$ satisfies precisely the same defaults from S as m'. In this way each D above is well-defined.

We now show that: 1) E is consistent and 2) the equivalence class of the models of E is an upper bound of the chain $\mathcal{C}$.

Assume that E is inconsistent. Then there is an inconsistent finite set $F \subseteq E$. Choose for each $d \in F \cap D$ an interpretation structure m_d' with $[m_d'] \in \mathcal{C}$ such that $m_d' \vDash d$

[3] With minor changes this proof is taken from [12], that proves the existence of a maximal model (not a maximal equivalence class) of hierarchic specifications from compact institutions. In [12] hierarchic specifications have *finite* priority structure. This restriction is dropped in the proof we present here.

(this m'_d exists by construction of D). Since $\mathcal{F} = \{[m'_d] : d \in F \cap D\}$ is finite and each equivalence class participates in the chain $\mathcal{C}$ there is a maximum $[n]$ of $\mathcal{F}$. By definition of D and m'_d the interpretation structure n satisfies all $d \in F \cap D$. Moreover n satisfies $\mathsf{ax}(S)$ since $[n] \in \mathcal{C}$. Therefore $n \vDash F$ contradicting the hypothesis of F being inconsistent.

We now prove that given $z \vDash E$ then $[z]$ is an upper bound of $\mathcal{C}$. For this purpose we show that $m'_1 \sqsubseteq^\circ z$ for every $[m'_1] \in \mathcal{C}$. This amounts to prove by well-founded induction in the partial order of priority $(H, \preceq) = \mathsf{po}(S)$ that if $\mathsf{df}(S,h)(m'_1) \not\subseteq \mathsf{df}(S,h)(z)$ then there is an $h' \prec h$ with $\mathsf{df}(S,h')(m'_1) \subset \mathsf{df}(S,h')(z)$, for every $h \in H$ and every $[m'_1] \in \mathcal{C}$.

Note firstly that $z \vDash \mathsf{ax}(S)$ and, therefore, participates in $\mathsf{lex}^\circ(S)$. Furthermore, the defaults from S at level h satisfied by z,$\mathsf{df}(S,h)(z)$, are contained in $D \cap \mathsf{df}(S,h)$ since $z \vDash D$.

- if h is minimal then $\mathsf{df}(S,h)(m'_1) \subseteq \mathsf{df}(S,h)(m'_2)$ for all m'_2 such that $m'_1 \sqsubseteq^\circ m'_2$. Therefore, recalling the definition of D above, $\mathsf{df}(S,h)(m'_1) \subseteq D \subseteq \mathsf{df}(S,h)(z)$.
- for non-minimal h assume that $\mathsf{df}(S,h)(m'_1) \not\subseteq \mathsf{df}(S,h)(z)$. We have to find an $h' \prec h$ such that $\mathsf{df}(S,h')(m'_1) \subset \mathsf{df}(S,h')(z)$. The condition $\mathsf{df}(S,h)(m'_1) \not\subseteq \mathsf{df}(S,h)(z)$ states that there exists a $d \in \mathsf{df}(S,h)(m'_1)$ such that $d \notin \mathsf{df}(S,h)(z)$. In particular $d \notin D$. Therefore, from definition of D, there is $[m'_2] \in \mathcal{C}$ with $m'_1 \sqsubseteq^\circ m'_2$ and $m'_2 \nvDash d$. But this implies $\mathsf{df}(S,h)(m'_1) \not\subseteq \mathsf{df}(S,h)(m'_2)$ and, from lemma 129 there is $h'_1 \prec h$ such that $\mathsf{df}(S,h'_1)(m'_1) \subset \mathsf{df}(S,h'_1)(m'_2)$ *and* $\mathsf{df}(S,h''_1)(m'_1) \subseteq \mathsf{df}(S,h''_1)(m'_2)$ for every $h''_1 \prec h'_1$. Now either $\mathsf{df}(S,h'_1)(m'_2) \subseteq \mathsf{df}(S,h'_1)(z)$ or not:
  - if $\mathsf{df}(S,h'_1)(m'_2) \subseteq \mathsf{df}(S,h'_1)(z)$ then $\mathsf{df}(S,h'_1)(m'_1) \subset \mathsf{df}(S,h'_1)(z)$ since the set $\mathsf{df}(S,h'_1)(m'_1) \subset \mathsf{df}(S,h'_1)(m'_2) \subseteq \mathsf{df}(S,h'_1)(z)$. This case ends by choosing $h' = h'_1$.
  - If $\mathsf{df}(S,h'_1)(m'_2) \not\subseteq \mathsf{df}(S,h'_1)(z)$ then, by the induction hypothesis for h'_1, there is $h''_1 \prec h'_1$ with $\mathsf{df}(S,h''_1)(m'_2) \subset \mathsf{df}(S,h''_1)(z)$. Recall that $\mathsf{df}(S,h''_1)(m'_1) \subseteq \mathsf{df}(S,h''_1)(m'_2)$. Therefore $\mathsf{df}(S,h''_1)(m'_1) \subset \mathsf{df}(S,h''_1)(z)$ and we can choose $h' = h''_1$. ✓

The previous lemma is mirrored at the syntactic level as we see in the following.

Lemma 138 Let S be a hierarchic specification with consistent $\mathsf{ax}(S)$ and E a consistent set such that $\mathsf{ax}(S) \subseteq E \subseteq \mathsf{ax}(S) \cup \mathsf{df}(S)$. Then there is an extension presentation E' of S with $E \subseteq^\circ E'$.

Proof Consider the equivalence class (w.r.t. the lexicographic preference of S) $[m]$ of a model m of E. Let $[m^\uparrow]$ be a maximal equivalence class with $m \sqsubseteq^\circ m^\uparrow$ (that exists from the lemma 137 above). The set $E' = \mathsf{ax}(S) \cup \mathsf{df}(S)([m^\uparrow])$ is an extension presentation of S (see definition 130). It is straightforward to check that $E \subseteq^\circ E'$. ✓

2.5.2 Selection Function

A corollary of the previous result is that the extension presentations of S are those extensions of the flat specification $(\mathsf{ax}(S), \mathsf{df}(S))$ that are maximal w.r.t. $\subseteq^{\circ}$ when compared among themselves.

Theorem 139 An extension presentation E of $(\mathsf{ax}(S), \mathsf{df}(S))$ is an extension presentation of S iff there is no other extension presentation $E' \neq E$ of $(\mathsf{ax}(S), \mathsf{df}(S))$ with $E \subseteq^{\circ} E'$.

Proof Only the case of consistent $\mathsf{ax}(S)$ is non-trivial. The direct implication is obvious from theorem 133: there is no $E' \neq E$ with $E \subseteq^{\circ} E'$. Assume now that E is an extension presentation of $(\mathsf{ax}(S), \mathsf{df}(S))$ and there is no other extension presentation $E' \neq E$ of $(\mathsf{ax}(S), \mathsf{df}(S))$ with $E \subseteq^{\circ} E'$. From lemma 138 above it exists an extension presentation E' of S with $E \subseteq^{\circ} E'$. This extension presentation E' of S is also an extension of $(\mathsf{ax}(S), \mathsf{df}(S))$. Therefore $E = E'$. ✓

2.5.3 Properties

The properties of existence of extensions and cumulativity of hierarchic specifications of compact logics can be concluded from lemma 137.

Theorem 140 Let $\mathcal{I}$ be an institution where the compactness property holds. Then

1. **Existence.** Any specification has, at least, one extension.

2. **Cumulativity.** Let S be a specification from $\mathcal{I}$ and f be a skeptical consequence of S, $S \vdash_{sk} f$. Let S' be a hierarchic specification obtained from S by adding f either to the axioms or to the defaults of S at some level.

   Then E' is an extension presentation of S' iff $E' = E \cup \{f\}$ where E an extension presentation of S. This implies that given any formula f', $S \vdash_{sk} f'$ iff $S' \vdash_{sk} f'$.

Proof

1. If $\mathsf{ax}(S)$ is inconsistent S has the whole $\mathsf{sg}(S)$-language as extension. If $\mathsf{ax}(S)$ is consistent take $m \in \mathsf{ax}(S)^{\bullet}$. From lemma 137 there is a maximal equivalence class $[m^{\uparrow}]$ of $\mathsf{lex}^{\circ}(S)$ with $(m, m^{\uparrow}) \in \mathsf{rl}(\mathsf{lex}^{\circ}(S))$, where $[m]$ is the equivalence class of m. From definition 130 and lemma 131 we have that $[m^{\uparrow}]^{\bullet}$ is an extension of S,

2. The case of inconsistent $\mathsf{ax}(S) \subseteq \mathsf{ax}(S')$ is trivial. We now see the case of consistent $\mathsf{ax}(S)$ and S' with $\mathsf{ax}(S') = \mathsf{ax}(S)$, $\mathsf{df}(S, h_1) = \mathsf{df}(S', h_1) \cup \{f\}$ and $\mathsf{df}(S, h) = \mathsf{df}(S', h)$ for each priority level $h \neq h_1$. Firstly we show that given an extension presentation E of S then $E \cup \{f\}$ is an extension presentation of S'. Note that E is also an extension presentation of the (flat) specification $(\mathsf{ax}(S), \mathsf{df}(S))$ (lemma 135). Therefore, from cumulativity for flat specifications in theorem 67, $E \cup \{f\}$ is an extension presentation of $(\mathsf{ax}(S'), \mathsf{df}(S'))$. We have to see that $E \cup \{f\}$ is maximal according to $\subseteq^{\circ}_{S'}$. Assume it is not. From theorem 139 this means that there exists another extension presentation E_1' of $(\mathsf{ax}(S'), \mathsf{df}(S'))$ such that $E \cup \{f\} \subseteq^{\circ}_{S'} E_1'$. But, again from cumulativity for (flat) specifications this E_1' is also $E_1' = E_1 \cup \{f\}$ with $E_1 \neq E$ an extension presentation of $(\mathsf{ax}(S), \mathsf{df}(S))$. From $E \cup \{f\} \subseteq^{\circ}_{S'} E_1 \cup \{f\}$ we conclude $E \subseteq^{\circ}_{S} E_1$, with $E \neq E_1$, contradicting the fact that E is an extension presentation of S.

   Let now E' be an extension presentation of S'. Then E' is an extension presentation of the (flat) specification $(\mathsf{ax}(S'), \mathsf{df}(S'))$. Therefore $E' = E \cup \{f\}$ where E is an extension presentation of $(\mathsf{ax}(S), \mathsf{df}(S))$. We only have to show that E is also an extension presentation of S. Assume it is not. Therefore there is an extension presentation $E_1 \neq E$ of $(\mathsf{ax}(S), \mathsf{df}(S))$ such that $E \subseteq^{\circ}_{S} E_1$ (theorem 139). This implies $E \cup \{f\} \subseteq^{\circ}_{S'} E_1 \cup \{f\}$, with $E \cup \{f\} \neq E_1 \cup \{f\}$ contradicting the hypothesis of $E' = E \cup \{f\}$ being an extension presentation of S'.

   We omit the proof for the case of consistent $\mathsf{ax}(S)$ and S' with $\mathsf{ax}(S') = \mathsf{ax}(S) \cup \{f\}$ and $\mathsf{df}(S', h) = \mathsf{df}(S, h)$ for any priority level h.

   Finally it is straightforward to conclude that the maximal models of the lexicographic preference of S coincide with those of S'. In fact those are the models of some extension presentation of S. Since they all satisfy f they are also the models of some extension presentation of S'. Equality of the classes of maximal models trivially implies equality of skeptical consequences. ✓

The property of semi-monotonicity has to be rewritten to take the richer structure of hierarchic specifications into account. Whereas in the flat case overriding was only possible by the addition of new axioms, in the hierarchic case addition of new defaults may override old ones. Overriding of defaults is, however, only possible when the new defaults are assigned better priority than those to be overridden. Therefore overriding is not possible if the new defaults are added to priority levels that are *not* under some of the original ones. This includes a) addition of defaults in the least important (maximal) priority levels; b) addition of defaults in new priority levels without better priority than any of the original priority levels; and any combination of these two mechanisms.

We now define the corresponding two relations between hierarchic specifications.

Definition 141 Let S be a Σ-hierarchic specification and $(H, \prec) = \mathsf{po}(S)$ its partial order of priority.

- A level $h \in H$ is said *maximal* (in $(H, \prec)$) iff given any $h' \in H$ with $h \prec h'$ then $h' = h$,
- S is said *weakly included* in S' iff
  - $S \Subset S'$,
  - $\mathsf{ax}(S) = \mathsf{ax}(S')$ and
  - $\mathsf{df}(S, h) = \mathsf{df}(S', h)$ for every non-maximal $h \in H$.

  ($S' = S$ except possibly at some maximal levels, where S' has more defaults than S.)
- S is said *disjointly included* in S' iff
  - $S \overrightarrow{\Subset} S'$,
  - $\mathsf{ax}(S) = \mathsf{ax}(S')$,
  - $\mathsf{df}(S, h) = \mathsf{df}(S', h)$ for every $h \in H$, where $(H, \preceq) = \mathsf{po}(S)$,
  - for all $h' \in (H' \setminus H)$ there is no $h \in H$ with $h' \preceq' h$, where $(H', \preceq') = \mathsf{po}(S')$ and
  - $h_1 \preceq h_2$ iff $h_1 \preceq' h_2$ for every $h_1, h_2 \in H$

  (S' coincides with S in the original priority levels and no new hierarchy level is of better priority than any of the original ones). ∎

Theorem 142 Let S be a hierarchic specification of a compact institution that is weakly included or disjointly included in S'. For each extension presentation (resp. extension) E of S there is an extension presentation (resp. extension) E' of S' such that $E \subseteq E'$.

Proof The proof for extensions is a simple consequence of that for extension presentations.

- We begin with the case of S being weakly included in S'. Only the case of consistent $\mathsf{ax}(S) = \mathsf{ax}(S')$ is non-trivial. Let E be an extension presentation of S. Then E is consistent and there exists a (consistent) extension presentation E' of S' with $E \subseteq^{\circ}_{S'} E'$ (from lemma 138 and the fact that $\mathsf{ax}(S') \subseteq E \subseteq \mathsf{ax}(S') \cup \mathsf{df}(S')$). We compare $E'' = E' \cap \mathsf{df}(S)$ with E according to $\subseteq^{\circ}_{S}$ and conclude that $E \subseteq^{\circ}_{S} E''$. Therefore $E = E''$ (E is an extension presentation of S) and $E' \supseteq E'' = E$.

  Let $(H, \preceq)$ be the partial order of priority of S and S' and $h \in H$. Since $E \subseteq^{\circ}_{S'} E'$ we know that either $E \cap \mathsf{df}(h, S') \subseteq E' \cap \mathsf{df}(h, S')$ or there is $h' \prec h$ and $E \cap \mathsf{df}(h', S') \subset E' \cap \mathsf{df}(h', S')$. We now see that this is equivalent to $E \cap \mathsf{df}(h, S) \subseteq$

$E'' \cap \mathsf{df}(h, S)$ or there is $h' \prec h$ and $E \cap \mathsf{df}(h', S) \subset E'' \cap \mathsf{df}(h', S)$, therefore showing that $E \subseteq^{\circ}_{S} E''$.

The condition $E \cap \mathsf{df}(h', S') \subset E' \cap \mathsf{df}(h', S')$ involves an h' that is not maximal. In this case $\mathsf{df}(h', S') = \mathsf{df}(h', S)$ and this condition becomes $E \cap \mathsf{df}(h', S) \subset E' \cap \mathsf{df}(h', S)$. Moreover $E' \cap \mathsf{df}(h', S)$ is the same as $E'' \cap \mathsf{df}(h', S)$.

The condition $E \cap \mathsf{df}(h, S') \subseteq E' \cap \mathsf{df}(h, S')$ is equivalent to $E \cap \mathsf{df}(h, S) \subseteq E'' \cap \mathsf{df}(h, S)$, for non-maximal h, for similar reasons. If h is maximal note that $E \cap \mathsf{df}(h, S')$ is the same as $E \cap \mathsf{df}(h, S)$ since E consists only of defaults from S. For the same reason $E \cap \mathsf{df}(h, S) \subseteq E' \cap \mathsf{df}(h, S')$ iff $E \cap \mathsf{df}(h, S) \subseteq E' \cap \mathsf{df}(h, S') \cap \mathsf{df}(S)$. The set $E' \cap \mathsf{df}(h, S') \cap \mathsf{df}(S)$ is $E'' \cap \mathsf{df}(h, S)$.

- The case of S being disjointly included in S' is similar. Let E be an extension presentation of S (with consistent $\mathsf{ax}(S)$) and E' an extension presentation of S' with $E \subseteq^{\circ}_{S'} E'$. We see again that $E \subseteq^{\circ}_{S} E''$, where $E'' = E' \cap \mathsf{df}(S)$. Therefore $E = E''$ and $E' \supseteq E'' = E$. Let $(H, \preceq)$ be the partial order of priority of S and $(H', \preceq')$ that of S' and $h \in H$. Since $E \subseteq^{\circ}_{S'} E'$ we have that for every $h_1 \in H'$ either $E \cap \mathsf{df}(h_1, S') \subseteq E' \cap \mathsf{df}(h_1, S')$ or there is $h_1' \prec' h_1$ and $E \cap \mathsf{df}(h_1', S') \subset E' \cap \mathsf{df}(h_1', S')$. If $h_1 \in H$ then any $h_1' \prec' h_1$ is also in H (no new level is under any of the levels from S) and $h_1' \prec h_1$ (the restriction of $\prec'$ to H is $\prec$). Therefore for every $h \in H$ either $E \cap \mathsf{df}(h, S) \subseteq E' \cap \mathsf{df}(h, S)$ or there is $h' \prec h$ and $E \cap \mathsf{df}(h', S) \subset E' \cap \mathsf{df}(h', S)$. The remaining proof is like the case of non-maximal h as before, recording that $\mathsf{df}(h, S') = \mathsf{df}(h, S)$ for every $h \in H$. ✓

We remark that the property of semi-monotonicity displayed in the theorem above also holds when S' is obtained from S by some combination of weak and disjoint inclusions.

2.6 Final Remarks

Hierarchic specifications were introduced in [12] and generalize ordered theory presentations from [74, 75, 76] and prioritized defaults from [7, 6]. Their lexicographic preference is the semantics of ordered theory presentations and prioritized defaults. For these reasons the definitions and properties of extensions presented in sections 2.1.1, 2.4 and 2.5 either repeat or slightly generalize corresponding definitions and properties (see [12, 7, 6, 75, 78]). The characterization of the selection function choosing from the extensions of the flat specification those of the hierarchic specification, presented in section 2.4.3 and 2.5.2 is, however, new. The same happens with the translation of a hierarchic specification into a flat one (section 2.2.2) with same lexicographic preference.

The most important contribution of this chapter is the identification and comparison of structured semantics needed for composition, the definition of theory of a hierarchic specification and the interplay between syntactic and semantic inclusion. These concepts have been introduced and investigated in sections 2.1 and 2.3. They generalize corresponding concepts for flat specifications and are the foundations of the theory of composition of hierarchic specifications presented in the next chapter.

Chapter 3

Composition

In the previous chapters 1 and 2 we have provided specifications and 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 also been defined.

We extend these notions in this chapter to account for composition of 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 ([46]) to specifications and hierarchic specifications and are inspired by the formalization of composition of hierarchic specifications presented in [12] .

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. In this way $S_1 = (\emptyset, \{p\})$ has the same meaning as $S_2 = (\emptyset, \{q\})$ since what is true of S_1 can be translated to a true proposition of S_2 by replacing p by q and vice versa. A more intuitive example is the use of the predicates *Pacifist* and *Pazifist* by specifiers of different languages but modeling the same universe of discourse. Independence of representation is formalized by the notion of isomorphism in the categories Spec of specifications or $\mathsf{hieSpec}$ of hierarchic specifications. In the first case only renaming of (signature) symbols in formulas is considered. In the second renaming of priority level names is also taken into account.

Moreover this chapter addresses *composition of specifications and hierarchic specifications* 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 Spec of specifications or $\mathsf{hieSpec}$ of hierarchic specifications. In the first case 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 Sign of signatures). In the second case, where the priority structure has to be taken into account, the constructions depend furthermore on the existence of a priority structure that expresses the combination of the priority structures of the hierarchic specifications involved.

The categories of specifications and hierarchic specifications are mirrored on the semantic side by semantic categories (Pref and hiePref). The semantics of the composition of specifications is obtained from the semantics of the parameter specifications by canonical constructions in these categories.

The chapter is organized as follows. In section 3.1 the concepts related to composition of specifications are presented. These include the category Spec of specifications and its semantical counterpart, the category Pref of preference relations. The one to one relation between syntactical and semantical constructions is presented. The existence of such constructions is also studied.

In section 3.2 the previous concepts are generalized to hierarchic specifications. This corresponds to add the priority structure and corresponding operations to the categories Spec and Pref, obtaining the categories hieSpec and hiePref.

We see (in 3.1.4 and 3.2.5) that isomorphic specifications or isomorphic hierarchic specifications have (up to a renaming of signature symbols) the same consequences. Since the semantics adopted to formalize composition possesses more structure than these consequences (as opposed to the classical case) we show (in 3.1.6 and 3.2.7) that the additional structure is the minimal one that assures a formal description of the intended forms of composition.

A general direction for the formalization of other composition forms is sketched in section 3.2.8. In section 3.3 we conclude the chapter.

3.1 Specifications

The concepts of theory and inclusion of specifications, presented in the previous chapters, are the basis of the theory of composition of specifications. Composition of specifications is formalized on the syntactical side by colimits in the category Spec of specifications (section 3.1.1). Composition of specifications is also interpreted on the semantic side by limits in the category Pref of preference relations (section 3.1.2). The correspondence between the two formalizations is displayed in section 3.1.3. That isomorphic specifications have the same logical content is shown in section 3.1.4. In section 3.1.5 we show that composition of specifications is always defined provided that a rich enough signature is available (i.e. the category Spec is cocomplete whenever the underlying category of signatures is cocomplete). Finally in section 3.1.6 we see that the chosen semantics is canonical among other semantics for composition.

3.1.1 Category of Specifications

In this section we introduce the category Spec with specifications as objects. Specifications are compared via the respective theories. These are related by axiom and default preserving signature morphisms. This means that the translation to the codomain signature of the axioms and defaults from the (theory of the) domain specification are axioms and defaults of the theory of the codomain specification.

The translation of a specification to another signature, given a signature morphism, is defined as follows:

Definition 143 Let S_1 be a Σ_1-specification and $\sigma : \Sigma_1 \to \Sigma_2$ a signature morphism. Recall that $\mathsf{Sen}(\sigma) : \mathsf{Sen}(\Sigma_1) \to \mathsf{Sen}(\Sigma_2)$ is a Set function sending 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 $\hat{\sigma}$. The Σ_2-specification $\hat{\sigma}(S_1)$ with axioms $\hat{\sigma}(\mathsf{ax}(S_1)) = \{\hat{\sigma}(a), a \in \mathsf{ax}(S_1)\}$ and defaults $\hat{\sigma}(\mathsf{df}(S_1)) = \{\hat{\sigma}(d), d \in \mathsf{df}(S_1)\}$, is the *translation* of S_1 into Σ_2, given $\sigma : \Sigma_1 \to \Sigma_2$. ∎

The definition of the category Spec follows.

Definition 144 The category Spec of specifications consists of:

- **Objects:** All specifications,
- **Morphisms:** A specification morphism $\sigma : S_1 \to S_2$ from the Σ_1-specification S_1 to the Σ_2-specification S_2 is a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ such that $\hat{\sigma}(S_1^{\star\star}) \Subset S_2^{\star\star}$. ∎

Proof It is straightforward to check that Spec is indeed a category. This follows from the fact that the translation of the composition of signature morphisms is the composition of the translations: $\widehat{\sigma_2 \cdot \sigma_1} = \hat{\sigma}_2 \circ \hat{\sigma}_1$ ($\cdot$ is composition in Sign and $\circ$ in Set). This property is a consequence of Sen being a functor (see definition 1 of institution). ✓

The morphism condition presented above can be equivalently rewritten by stating that the translation of S_1 (not $S_1^{\star\star}$) w.r.t. $\sigma : \Sigma_1 \to \Sigma_2$ has to be included ($\Subset$) in the theory of S_2. In other words, the image by $\sigma : \Sigma_1 \to \Sigma_2$ of the axioms from the domain specification are "implicit" axioms of the codomain specification (and this is the classical condition - see [46]) and the image by $\sigma : \Sigma_1 \to \Sigma_2$ of the defaults from the domain specification are implicit defaults of the codomain specification. This is stated formally in the following *Presentation Lemma*.

Lemma 145 Let S_1 be a Σ_1-specification, S_2 a Σ_2-specification and $\sigma : \Sigma_1 \to \Sigma_2$ a signature morphism. Then

$$\hat{\sigma}(S_1^{\star\star}) \Subset S_2^{\star\star} \text{ iff } \hat{\sigma}(S_1) \Subset S_2^{\star\star}.$$

Proof The "if" part is trivial: $\hat{\sigma}(S_1) \Subset \hat{\sigma}(S_1^{\star\star})$ since $S_1 \Subset S_1^{\star\star}$. The "only if" part is proved as follows. Since $\hat{\sigma}(S_1) \Subset S_2^{\star\star}$ then $\hat{\sigma}(S_1)^{\star\star} \Subset S_2^{\star\star}$ (recall that ${S_2}^{\star\star\star\star} = S_2^{\star\star}$). Moreover (this we prove below) $\hat{\sigma}(S_1^{\star\star})^{\star\star} = \hat{\sigma}(S_1)^{\star\star}$. Therefore $\hat{\sigma}(S_1^{\star\star}) \Subset \hat{\sigma}(S_1^{\star\star})^{\star\star} = \hat{\sigma}(S_1)^{\star\star} \Subset S_2^{\star\star}$.

To show that $\hat{\sigma}(S_1^{\star\star})^{\star\star} = \hat{\sigma}(S_1)^{\star\star}$ we need the following fact: if $S^\star = S'^\star$ then $\hat{\sigma}(S)^\star = \hat{\sigma}(S')^\star$. From this and since $S_1^{\star\star}$ and S_1 have the same preference we conclude that $\hat{\sigma}(S_1^{\star\star})$ and $\hat{\sigma}(S_1)$ also have the same preference and therefore the same theory. I.e. $\hat{\sigma}(S_1^{\star\star})^{\star\star} = \hat{\sigma}(S_1)^{\star\star}$ as wanted.

That $\hat{\sigma}(S)^\star = \hat{\sigma}(S')^\star$ if $S^\star = S'^\star$ is a simple consequence of the satisfaction condition.

- We begin by showing that $\hat{\sigma}(S)$ and $\hat{\sigma}(S')$ have the same models. Assume they do not. Then there is a model m of $\mathsf{ax}(\hat{\sigma}(S))$ that is not a model of $\mathsf{ax}(\hat{\sigma}(S'))$ (or vice versa). I.e. $m \vDash \mathsf{ax}(\hat{\sigma}(S'))$ and there is a $\hat{\sigma}(a') \in \hat{\sigma}(\mathsf{ax}(S'))$ such that $m \nvDash \hat{\sigma}(a')$. By the satisfaction condition the reduct[1] of m w.r.t. σ satisfies $\mathsf{ax}(S)$ and does not satisfy $a' \in \mathsf{ax}(S')$, contradicting the hypothesis $S^\star = S'^\star$ (in particular that they have the same models).
- Assume now that $\hat{\sigma}(S)$ and $\hat{\sigma}(S')$ have the same models but different preference relations. Then there are models m, n of $\hat{\sigma}(S)$ and $\hat{\sigma}(S')$ such that $m \sqsubseteq n$ but $m \not\sqsubseteq' n$ (or vice versa). This means that all defaults from $\mathsf{df}(\hat{\sigma}(S))$ that are satisfied by m are also satisfied by n but there is a default $\hat{\sigma}(d') \in \hat{\sigma}(\mathsf{df}(S'))$ that is satisfied by m and not by n. Again from the satisfaction condition we conclude that the reducts of m and n are related according to S but not according to S', contradicting the hypothesis $S^\star = S'^\star$. ✓

We end this section by remarking that isomorphic specifications have, up to a renaming of vocabulary, the same theory.

Lemma 146 Let $\sigma : \mathsf{sg}(S_1) \to \mathsf{sg}(S_2)$ be a signature isomorphism. Then $\sigma : S_1 \to S_2$ is a Spec-isomorphism iff $\hat{\sigma}(S_1^{\star\star}) = S_2^{\star\star}$.

Proof Let $\overline{\sigma} : \mathsf{sg}(S_2) \to \mathsf{sg}(S_1)$ be the inverse isomorphism of $\sigma : \mathsf{sg}(S_1) \to \mathsf{sg}(S_2)$. Since Sen is a functor to Set then $\hat{\sigma}$ is a Set isomorphism with inverse $\hat{\overline{\sigma}} = \hat{\sigma}^{-1}$.

- Assume that $\sigma : S_1 \to S_2$ is a Spec-isomorphism. Then $\hat{\sigma}(S_1^{\star\star}) \Subset S_2^{\star\star}$. Moreover $\overline{\sigma} : S_2 \to S_1$ is also an isomorphism and $\hat{\overline{\sigma}}(S_2^{\star\star}) \Subset S_1^{\star\star}$. This implies $S_2^{\star\star} = \hat{\sigma}(\hat{\overline{\sigma}}(S_2^{\star\star})) \Subset \hat{\sigma}(S_1^{\star\star})$.

[1] The reduct of m w.r.t. σ is the interpretation structure $\mathsf{Mod}(\sigma)(m_2)$ (see section 1.1).

- Assume that $\hat{\sigma}(S_1^{\star\star}) = S_2^{\star\star}$. We only have to check that $\sigma : S_1 \to S_2$ and $\overline{\sigma} : S_2 \to S_1$ are Spec-morphisms (the property of isomorphism follows trivially from the corresponding property in Sign). Clearly $\sigma : S_1 \to S_2$ is a Spec-morphism. Moreover $\hat{\sigma}(S_1^{\star\star}) = S_2^{\star\star}$ implies $S_1^{\star\star} = \hat{\overline{\sigma}}(\hat{\sigma}(S_1^{\star\star})) = \hat{\overline{\sigma}}(S_2^{\star\star})$ and this in turn that $\overline{\sigma} : S_2 \to S_1$ is also a Spec-morphism. ✓

3.1.2 Category of Pre-orders

In this subsection we introduce the semantical counterpart of the syntactical concepts just presented. The semantics of a specification is its associated preference relation[2].

We introduce the category PreOrder of Σ-pre-orders. Its morphisms are the semantical counterpart of the morphisms in Spec and correspond to inclusion both of the classes of interpretation structures and of the relations (of preference) among them. Recall from section 1.2.2 that addition of axioms corresponds to lessening of models and addition of defaults to lessening of relations of preference among those models. Moreover, the translation of axioms and defaults induced by signature morphisms has also a semantic expression in the notion of reduct of a pre-order (that “translates” a pre-order among interpretation structures of one signature to a pre-order of another signature).

Furthermore we identify the full subcategory Pref of PreOrder with objects the Σ-pre-orders that are the preference relations of some Σ-specification. The category Pref mirrors the category of specifications: the co-constructions in Spec are mapped to constructions in Pref. This map assigns to each operation of composition of specifications its semantic expression as an operation between the semantics of those specifications. The reverse is also true.

We begin by introducing the preliminary notion of reduct of a relation w.r.t. a signature morphism.

Definition 147 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 Σ_1-interpretation structure $\mathsf{Mod}(\sigma)(m_2)$. The reduct $\mathsf{Mod}(\sigma)(m_2)$ will be denoted by $\check{\sigma}(m_2)$.

Furthermore, given a Σ_2-pre-order $R = (|R|, \sqsubseteq)$ the *reduct relation* $\check{\sigma}(R)$ is the pair

$$(\check{\sigma}(|R|), \check{\sigma}(\sqsubseteq))$$

[2]The classical semantics of the presentation A_Σ is the full subcategory $\mathsf{Mod}(A_\Sigma)$ of $\mathsf{Mod}(\Sigma)$ of the Σ-interpretation structures that are also models of A_Σ. This structure displays the relationships between interpretation structures and may additionally be assigned to a specification. It will not, however, be of interest to us here since we are only concerned with the relationship of preference.

where $\check{\sigma}(|R|)$ is the class $\{\check{\sigma}(m) : m \in |R|\}$ of the reducts of the interpretation structures participating in R and $\check{\sqsubseteq} = \check{\sigma}(\sqsubseteq)$ is the smallest pre-order[3] among those interpretation structures that satisfies

$$\check{\sigma}(m_2) \mathrel{\check{\sqsubseteq}} \check{\sigma}(n_2) \text{ if } m_2 \sqsubseteq n_2,$$

that relates the reducts of the Σ_2-interpretation structures from $|R|$ if they were related by R. ▪

We now define the category of Σ-pre-orders. The morphism condition on the model part is the classical one, inclusion of classes of interpretation structures. On the relation part it is inclusion of relation pairs.

Definition 148 The category PreOrder of Σ-pre-orders consists of:

- **Objects:** All pre-orders,
- **Morphisms:** A pre-order morphism[4] $\overleftarrow{\sigma}\colon R' \to R$ from the Σ_2-pre-order R' to the Σ_1-pre-order R is a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ such that $\check{\sigma}(R') \Subset R$ i.e. the reduct of every interpretation structure in $|R'|$ is in $|R|$ and whenever two interpretation structures are related by R' their reducts are related by R.

Proof It is straightforward to check that PreOrder is indeed a category. This follows from the fact that the reduct w.r.t. $\sigma_2 \cdot \sigma_1$ is given by the composition $\check{\sigma}_1 \circ \check{\sigma}_2$ of the reducts w.r.t. σ_2 and σ_1 ($\check{\sigma}$ is the functor $\mathsf{Mod}(\sigma)$, $\cdot$ is composition in Sign and $\circ$ is composition of functors)[5]. This identity follows from the functor property of Mod (see definition 1 of institution). ✓ ▪

We are mainly concerned with the pre-orders that are preferences of some specification. Recall that a Σ-preference relation is a Σ-pre-order that is the preference relation of some Σ-specification. From the Galois connection in theorem 30 it follows easily that a Σ-pre-order R is a Σ-preference relation iff $R = R^{\star\star}$. In this way the category of Σ-preference relation is defined as follows.

Definition 149 The category Pref is the full subcategory of PreOrder with objects the Σ-pre-orders R such that $R = R^{\star\star}$. ▪

---

[3]The smallest relation that satisfies $\check{\sigma}(m_2) \mathrel{\check{\sqsubseteq}} \check{\sigma}(n_2)$ if $m_2 \sqsubseteq n_2$, is not, in general, a Σ_1-pre-order, since it may fail to be transitive. The reduct pre-order $\check{\sigma}(R)$ is the transitive closure of this relation.

[4]Note that to a morphism $\sigma : S_1 \to S_2$ of specifications there corresponds a PreOrder morphism $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$, in the reverse direction, as emphasized by the notation $\overleftarrow{\sigma}$.

[5]Recall that we are not concerned with the category structure of interpretation structures. In this way $\mathsf{Mod}(\sigma)$ can alternatively be seen as a function between classes of interpretation structures.

We expect two preference relations to be isomorphic when one results from the other by replacing the interpretation structures by their corresponding reducts. This is not completely true: if $\overleftarrow{\sigma}\colon R' \to R$ is an isomorphism then it is not necessarily the case that $\breve{\sigma}(R') = R$. In fact not all interpretation structures participating in R are reducts of interpretation structures participating in R'. However, these additional models do not add any extra information and $\breve{\sigma}(R')$ and R are equivalent in a sense that we define below.

Definition 150

- The Σ-interpretation structures m and m' are said Σ-*equivalent*, written $m \approx m'$ if m and m' cannot be distinguished by Σ-formulas, i.e. $m \vDash f$ iff $m' \vDash f$, for all $f \in \mathsf{Sen}(\Sigma)$.
- The Σ-pre-orders R and R' are said Σ-*equivalent*, written $R \approx R'$ iff $R \mathrel{\tilde{\Subset}} R'$ and $R' \mathrel{\tilde{\Subset}} R$ where $R \mathrel{\tilde{\Subset}} R'$ iff
  - for every $m \in |R|$ there is $m' \in |R'|$ with $m' \approx m$ and
  - for every $(m, n) \in \mathsf{rl}(R)$ there is $(m', n') \in \mathsf{rl}(R')$ with $m' \approx m$ and $n' \approx n$.

Proof It is trivial to check that each $\approx$ is an equivalence relation. ✓ ■

The intended characterization of isomorphism in Pref follows.

Lemma 151 Let $\sigma : \mathsf{sg}(S_1^\star) \to \mathsf{sg}(S_2^\star)$ be a signature isomorphism. Then the Pref-morphism $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$ is a Pref-isomorphism iff $\breve{\sigma}(S_2^\star) \approx S_1^\star$.

Proof

- Assume that $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$ is a Pref-isomorphism and let $\overleftarrow{\bar{\sigma}}\colon S_1^\star \to S_2^\star$ be its inverse isomorphism. The morphism condition for $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$ is $\breve{\sigma}(S_2^\star) \Subset S_1^\star$. Therefore $\breve{\sigma}(S_2^\star) \mathrel{\tilde{\Subset}} S_1^\star$. We have to check that $S_1^\star \mathrel{\tilde{\Subset}} \breve{\sigma}(S_2^\star)$.
  - Let $m \in |S_1^\star|$. We see that $m \approx \breve{\sigma}(\breve{\bar{\sigma}}(m))$. Moreover $\breve{\sigma}(\breve{\bar{\sigma}}(m)) \in |\breve{\sigma}(S_2^\star)|$ since $\breve{\bar{\sigma}}(m) \in |S_2^\star|$: this is the morphism condition for $\overleftarrow{\bar{\sigma}}\colon S_1^\star \to S_2^\star$.
    To see that $m \approx \breve{\sigma}(\breve{\bar{\sigma}}(m))$ recall that $\hat{\sigma}$ is a Set isomorphism with inverse $\hat{\bar{\sigma}} = \hat{\sigma}^{-1}$. Therefore, given a Σ_1-formula f we have $f = \hat{\sigma}^{-1}(\hat{\sigma}(f))$. From the satisfaction condition it follows that $m \vDash_{\Sigma_1} f$ iff $\breve{\bar{\sigma}}(m) \vDash_{\Sigma_2} \hat{\sigma}(f)$ iff $\breve{\sigma}(\breve{\bar{\sigma}}(m)) \vDash_{\Sigma_1} f$. In this way $m \approx \breve{\sigma}(\breve{\bar{\sigma}}(m))$.

- Let now $m \sqsubseteq_1 n$ (where $\sqsubseteq_1$ is $\mathsf{rl}(S_1^\star)$). We see that $\breve{\sigma}(\breve{\overline{\sigma}}(m)) \sqsubseteq_1 \breve{\sigma}(\breve{\overline{\sigma}}(n))$. This is obviously the case since $\breve{\sigma}(\breve{\overline{\sigma}}(m))$ satisfies precisely the same formulas as m and, therefore, is equivalent according to $S_1^\star$ to m (it satisfies precisely the same defaults as m). The same holds for n and $\breve{\sigma}(\breve{\overline{\sigma}}(n))$. In this way $m \equiv_1 \breve{\sigma}(\breve{\overline{\sigma}}(m)) \sqsubseteq_1 n \equiv_1 \breve{\sigma}(\breve{\overline{\sigma}}(n))$.

- Assume that $S_1^\star \approx \breve{\sigma}(S_2^\star)$. We only have to show that $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$ and $\overleftarrow{\overline{\sigma}}\colon S_1^\star \to S_2^\star$ are morphisms since the isomorphism property follows from the corresponding property for σ and $\overline{\sigma}$ in Sign. This amounts to show that $\breve{\sigma}(S_2^\star) \Subset S_1^\star$ and $\breve{\overline{\sigma}}(S_1^\star) \Subset S_2^\star$.
  - From $\breve{\sigma}(S_2^\star) \tilde{\Subset} S_1^\star$ we conclude $\breve{\sigma}(S_2^\star) \Subset S_1^\star$: given $m \in |\breve{\sigma}(S_2^\star)|$ there is $m' \approx m$ and $m' \in |S_1^\star|$. But since $m' \approx m$ then m' satisfies the axioms from S_1 iff m does. From this it follows that if $m' \in |S_1^\star|$ then also $m \in |S_1^\star|$. This shows $|\breve{\sigma}(S_2^\star)| \subseteq |S_1^\star|$. The proof for the relation pairs is similar.
  - From $S_1^\star \tilde{\Subset} \breve{\sigma}(S_2^\star)$ we conclude (we omit trivial details) that $\breve{\overline{\sigma}}(S_1^\star) \tilde{\Subset} \breve{\overline{\sigma}}(\breve{\sigma}(S_2^\star))$. It is also trivial to establish that $\breve{\overline{\sigma}}(\breve{\sigma}(S_2^\star)) \tilde{\Subset} S_2^\star$. Therefore $\breve{\overline{\sigma}}(S_1^\star) \tilde{\Subset} S_2^\star$. From this it follows $\breve{\overline{\sigma}}(S_1^\star) \Subset S_2^\star$ as before. ✓

3.1.3 Syntax and Semantics

The relationship between syntax and semantics is made explicit by the functors $\mathsf{Sem} : \mathsf{Spec} \to \mathsf{Pref}^{\mathsf{op}}$ that to a specification associates its preference relation and $\mathsf{Syn} : \mathsf{Pref}^{\mathsf{op}} \to \mathsf{Spec}$ that to a preference relation associates the corresponding theory. Canonical properties of these functors allow to conclude that the colimits in the category of specifications correspond to limits in the category of preference relations and that the limits in Pref correspond to colimits in Spec.

The property of functor for Syn and Sem relies in the correspondence between the morphism condition for Spec and PreOrder (and also Pref) that we make explicit in the following lemma.

Lemma 152 There is a Spec morphism $\sigma : S_1 \to S_2$ iff there is a PreOrder morphism $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$.

Proof We have to show that $\hat{\sigma}(S_1^{\star\star}) \Subset S_2^{\star\star}$ iff $\breve{\sigma}(S_2^\star) \Subset S_1^\star$.

- Assume that $\hat{\sigma}(S_1^{\star\star}) \Subset S_2^{\star\star}$. This is equivalent (lemma 145 above) to $\hat{\sigma}(S_1) \Subset S_2^{\star\star}$. From the Galois connection in theorem 30 we have $\hat{\sigma}(S_1)^\star \Supset S_2^{\star\star\star} = S_2^\star$. From $\hat{\sigma}(S_1)^\star \Supset S_2^\star$ we conclude $\breve{\sigma}(\hat{\sigma}(S_1)^\star) \Supset \breve{\sigma}(S_2^\star)$. To conclude $S_1^\star \Supset \breve{\sigma}(S_2^\star)$ we have to show that $S_1^\star \Supset \breve{\sigma}(\hat{\sigma}(S_1)^\star)$.
  - Consider an interpretation structure m' participating in $\breve{\sigma}(\hat{\sigma}(S_1)^\star)$. Then $m' = \breve{\sigma}(m)$ is the reduct of some interpretation structure m participating

in $\hat{\sigma}(S_1)^\star$. This, in turn, is a model of the axioms in $\hat{\sigma}(S_1)$. Therefore $m \vDash \hat{\sigma}(\mathsf{ax}(S_1))$. From the satisfaction condition we conclude that $m' \vDash \mathsf{ax}(S_1)$ ($m' = \check{\sigma}(m)$), i.e. $m' \in |S_1^\star|$. This shows that $|\check{\sigma}(\hat{\sigma}(S_1)^\star)| \subseteq |S_1^\star|$.

- Assume now that $m' \sqsubseteq' n'$ according to $\check{\sigma}(\hat{\sigma}(S_1)^\star)$. We have to show that m' and n' are also related according to $S_1^\star$. Since $m' \sqsubseteq' n'$ according to $\check{\sigma}(\hat{\sigma}(S_1)^\star)$ then $m' = \check{\sigma}(m)$ and $n' = \check{\sigma}(n)$ and either $m \sqsubseteq n$ according to $\hat{\sigma}(S_1)^\star$ or $m' \sqsubseteq' n'$ results from the transitive closure of such pairs (see definition 147 of reduct pre-order).
  * Assume that $m' = \check{\sigma}(m)$ and $n' = \check{\sigma}(n)$ and $m \sqsubseteq n$ according to $\hat{\sigma}(S_1)^\star$. That $m \sqsubseteq n$ means that the defaults from $\hat{\sigma}(S_1)$ satisfied by m are also satisfied by n: given $\hat{\sigma}(d) \in \hat{\sigma}(\mathsf{df}(S_1))$ if $m \vDash \hat{\sigma}(d)$ then also $n \vDash \hat{\sigma}(d)$. Again from the satisfaction condition this implies that the defaults from S_1 satisfied by $m' = \check{\sigma}(m)$ are also satisfied by $n' = \check{\sigma}(n)$. Therefore m' and n' are related according to $S_1^\star$.
  * The pairs $m' \sqsubseteq' n'$ that result from the transitive closure of the ones above must also be in $S_1^\star$, since $S_1^\star$ is a pre-order and contains the transitive closure of any of its subparts.

  This shows that $\mathsf{rl}(\check{\sigma}(\hat{\sigma}(S_1)^\star)) \subseteq \mathsf{rl}(S_1^\star)$. Therefore $\check{\sigma}(\hat{\sigma}(S_1)^\star) \Subset S_1^\star$ as intended.

- Assume now that $\check{\sigma}(S_2^\star) \Subset S_1^\star$. Then $\check{\sigma}(S_2^\star)^\star \Supset S_1^{\star\star}$ and $\hat{\sigma}(\check{\sigma}(S_2^\star)^\star) \Supset \hat{\sigma}(S_1^{\star\star})$. We have to show that $S_2^{\star\star} \Supset \hat{\sigma}(\check{\sigma}(S_2^\star)^\star)$ to conclude $S_2^{\star\star} \Supset \hat{\sigma}(S_1^{\star\star})$.

  To show $S_2^{\star\star} \Supset \hat{\sigma}(\check{\sigma}(S_2{}^\star)^\star)$ we see that every axiom (resp. default) from $\hat{\sigma}(\check{\sigma}(S_2^\star)^\star)$ is an axiom (resp. default) from $S_2^{\star\star}$.

  - Let $\hat{\sigma}(a)$ be an axiom from $\hat{\sigma}(\check{\sigma}(S_2^\star)^\star)$. This means that a is an axiom from $\check{\sigma}(S_2^\star)^\star$, i.e. a holds in all interpretation structures participating in $\check{\sigma}(S_2^\star)$. These are the reducts of the interpretation structures from $S_2^\star$. From the satisfaction condition $\hat{\sigma}(a)$ holds in all interpretation structures from $S_2^\star$. Therefore $\hat{\sigma}(a)$ is an axiom from the theory of S_2, $S_2^{\star\star}$.
  - Let now $\hat{\sigma}(d)$ be a default from $\hat{\sigma}(\check{\sigma}(S_2^\star)^\star)$. Then d is a default from $\check{\sigma}(S_2^\star)^\star$. This means that given interpretation structures m', n' from $|\check{\sigma}(S_2^\star)|$ with $m' \sqsubseteq' n'$ according to $\check{\sigma}(S_2^\star)$, if $m' \vDash d$ then $n' \vDash d$. Consider now interpretation structures m, n from $S_2^\star$ and assume that $m \sqsubseteq n$ according to $S_2^\star$ and that $m \vDash \hat{\sigma}(d)$. Then, from the satisfaction condition, $\check{\sigma}(m) \vDash d$ and from the definition of pre-order reduct $\check{\sigma}(m) \sqsubseteq' \check{\sigma}(n)$. Therefore $\check{\sigma}(n) \vDash d$. Again from the satisfaction condition n satisfies $\hat{\sigma}(d)$. This shows that $\hat{\sigma}(d)$ is an implicit default from $S_2^\star$, i.e. $\hat{\sigma}(d)$ is a default from the theory of S_2, $S_2^{\star\star}$. ✓

The definition of the functors $\mathsf{Sem} : \mathsf{Spec} \to \mathsf{Pref}^{\mathsf{op}}$ and $\mathsf{Syn} : \mathsf{Pref}^{\mathsf{op}} \to \mathsf{Spec}$ follows.

Definition 153

- The functor $\mathsf{Sem} : \mathsf{Spec} \to \mathsf{Pref}^{\mathsf{op}}$ associates to each specification S its preference relation $S^\star$ and to each Spec morphism $\sigma : S_1 \to S_2$ the Pref morphism $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$.
- The functor $\mathsf{Syn} : \mathsf{Pref}^{\mathsf{op}} \to \mathsf{Spec}$ associates to each preference relation $S^\star$ its theory $S^{\star\star}$ and to each Pref morphism $\overleftarrow{\sigma}\colon S_2^\star \to S_1^\star$ the Spec morphism $\sigma : S_1^{\star\star} \to S_2^{\star\star}$.

Proof That Sem and Syn are functors is a trivial consequence of lemma 152 above. ✓ ■

Composition of specifications is formalized by colimits of diagrams in the category Spec. The following figure 3.1 illustrates such a diagram, relating two specifications, BATS and HUMANS. Its colimit is the union of these specifications, as it is explained in example 162 below.

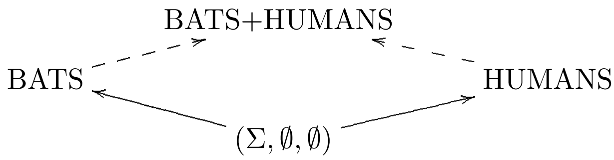

Figure 3.1: Composition of BATS and HUMANS

Each such construction has a semantic counterpart in a limit in the category Pref. The functor Sem sends each colimit in Spec to its corresponding limit in Pref. The functor Syn sends each limit in Pref to a colimit in Spec, thus associating to a composition of preference relations its corresponding composition of specifications. Clearly the existence of syntactical constructions is equivalent to the existence of semantical constructions. Before stating formally this correspondence we need to recall the associated definitions of diagram, colimits, limits, cocompleteness and completeness.

Definition 154 The notions herein defined can be found in any book on category theory. We use [1] and [51].

- A *diagram* in a category $\mathcal{C}$ is a graph homomorphism $\mathcal{D} : \mathcal{S} \to \mathcal{C}$. The graph $\mathcal{S}$ is the "shape" of the diagram. The diagram $\mathcal{D}$ assigns to each node in $|\mathcal{S}|$ an object from $|\mathcal{C}|$ and to each arrow in $\mathsf{Arr}(\mathcal{S})$ a morphism from $\mathsf{Mor}(\mathcal{C})$.

  When $|\mathcal{S}|$ and $\mathsf{Arr}(\mathcal{S})$ are *sets* the diagram is said *small*. We will be concerned only with small diagrams.

- A *cocone* for a diagram $\mathcal{D} : \mathcal{S} \to \mathcal{C}$ is an object $v \in |\mathcal{C}|$ and morphisms from each object in the diagram to v. Denoting the object assigned by $\mathcal{D}$ to the node $i \in |\mathcal{S}|$ by o_i then a cocone for $\mathcal{D}$ is the family $\{v, m_i : o_i \to v, i \in |\mathcal{S}|\}$.

- The cocone $\{v, m_i : o_i \to v, i \in |\mathcal{S}|\}$ is said *commutative* when is compatible with the morphisms in the diagram: given any morphism $m : o_i \to o_j$ in $\mathcal{D}$ then $m_i = m_j \circ m$ ($m : o_i \to o_j$ is the morphism assigned by $\mathcal{D}$ to the arrow $a : i \to j \in \mathsf{Arr}(\mathcal{S})$ from $\mathcal{S}$).

- A *colimit* of $\mathcal{D}$ is a commutative cocone $\{v, m_i : o_i \to v, i \in |\mathcal{S}|\}$ for $\mathcal{D}$ that is canonical among the commutative cocones for $\mathcal{D}$: given another commutative cocone $\{v', m'_i : o_i \to v', i \in |\mathcal{S}|\}$ for $\mathcal{D}$ then there exists a unique morphism $u : v \to v'$ such that $m'_i = u \circ m_i$ for all $i \in |\mathcal{S}|$.

- The notions of *cone* and *limit* are dual and omitted.

- A category having all colimits (resp. limits) of *small* diagrams is said *cocomplete* (resp. *complete*). ■

The following figure shows a cocone for a diagram $\mathcal{D}$.

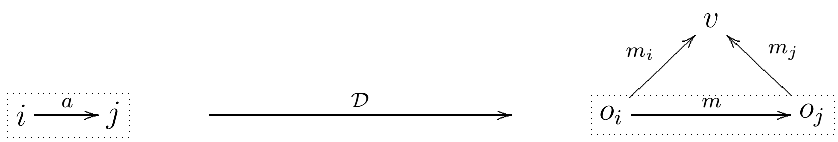

Figure 3.2: Cocone for $\mathcal{D}$

And the next one illustrates the definition of colimit of the previous diagram.

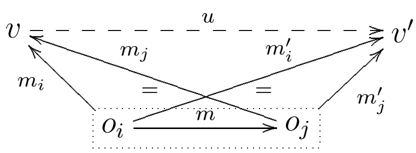

Figure 3.3: The commutative cocone on the left is a colimit for $\mathcal{D}$

Theorem 155 The image by Syn of a limit in Pref is a colimit in Spec and the image by Sem of a colimit in Spec is a limit in Pref.

Moreover the category Spec is cocomplete iff Pref is complete.

Proof Preservation of constructions follows trivially from lemma 152 above. That if Spec is cocomplete then Pref is complete is also trivial: one only has to note that the image of a diagram $\mathcal{D}$ in Pref by Sem $\circ$ Syn is again $\mathcal{D}$ (i.e. Sem $\circ$ Syn $\circ\, \mathcal{D} = \mathcal{D}$). Therefore the limit of $\mathcal{D}$ is the image by Sem of the existing colimit of the diagram Syn $\circ\, \mathcal{D}$. The other direction is similar with the difference that the image by Syn $\circ$ Sem of a diagram $\mathcal{D}$ in Spec is not $\mathcal{D}$ itself, but an isomorphic copy (since S and $S^{\star\star}$ are isomorphic in Spec). ✓

Remark 156 A trivial corollary of the theorem above is that two specifications are isomorphic iff their preferences are isomorphic. Note that S_2 is isomorphic to S_1 iff it is the colimit of the diagram having S_1 as only object. Also $S_2^\star$ is isomorphic to $S_1^\star$ iff it is the limit of the diagram containing only $S_1^\star$.

3.1.4 Isomorphism and Extensions

Two isomorphic specifications S_1 and S_2 are indistinguishable in the category Spec. This means that these specifications have, up to a change of vocabulary, the same information. In fact they have isomorphic preferences, i.e. they have the same meaning. A consequence of this fact is that two isomorphic specifications have, up to a change of vocabulary, the same extensions, and in particular the same credulous or skeptical consequences. We formalize these considerations in the following theorem.

Theorem 157 Let $\sigma : S_1 \to S_2$ be an isomorphism in the category Spec. Then $\mathcal{E}_1$ is an extension of S_1 iff $\hat{\sigma}(\mathcal{E}_1)$ is an extension of S_2.

Proof Let $\overline{\sigma} : S_2 \to S_1$ be the inverse isomorphism of $\sigma : S_1 \to S_2$. Recall that $\hat{\sigma}$ is a Set isomorphism with inverse $\hat{\overline{\sigma}} = \hat{\sigma}^{-1}$.

We see below that given an isomorphism $\sigma : S_1 \to S_2$ and an extension $\mathcal{E}_1$ of S_1 there is an extension $\mathcal{E}_2$ of S_2 such that $\hat{\sigma}(\mathcal{E}_1) \subseteq \mathcal{E}_2$. Also, since $\overline{\sigma} : S_2 \to S_1$ is again an isomorphism, given an extension $\mathcal{E}_2$ of S_2 there is an extension $\mathcal{E}_1'$ of S_1 with $\hat{\overline{\sigma}}(\mathcal{E}_2) = \hat{\sigma}^{-1}(\mathcal{E}_2) \subseteq \mathcal{E}_1'$.

- From $\hat{\sigma}^{-1}(\mathcal{E}_2) \subseteq \mathcal{E}_1'$ we conclude $\hat{\sigma}(\hat{\sigma}^{-1}(\mathcal{E}_2)) = \mathcal{E}_2 \subseteq \hat{\sigma}(\mathcal{E}_1')$. And $\hat{\sigma}(\mathcal{E}_1') \subseteq \mathcal{E}_2'$ for some extension $\mathcal{E}_2'$ of S_2. In this way $\mathcal{E}_2 \subseteq \hat{\sigma}(\mathcal{E}_1') \subseteq \mathcal{E}_2'$. From theorem 55, the two extensions $\mathcal{E}_2$ and $\mathcal{E}_2'$ must be the same. We conclude that $\mathcal{E}_2 = \hat{\sigma}(\mathcal{E}_1') = \mathcal{E}_2'$.
- We have shown that $\hat{\sigma}(\mathcal{E}_1')$ is an extension of S_2 when $\mathcal{E}_1'$ is an extension of S_1. Let now $\mathcal{E}_2$ be an extension of S_2. Using this same result for $\overline{\sigma} : S_2 \to S_1$, $\hat{\sigma}^{-1}(\mathcal{E}_2)$ is an extension of S_1. Call it $\mathcal{E}_1$ and note that $\mathcal{E}_2 = \hat{\sigma}(\mathcal{E}_1)$. This ends the proof.

We have finally to show that given an isomorphism $\sigma : S_1 \to S_2$ and an extension $\mathcal{E}_1$ of S_1 there is an extension $\mathcal{E}_2$ of S_2 such that $\hat{\sigma}(\mathcal{E}_1) \subseteq \mathcal{E}_2$. Note that since S_1 and S_2 are isomorphic also $S_1^\star$ and $S_2^\star$ are so. Therefore $\check{\sigma}(S_2^\star) \Subset S_1^\star$ and $\check{\overline{\sigma}}(S_1^\star) \Subset S_2^\star$.

- If S_1 is inconsistent then $S_1^\star$ is the empty preference relation. Since $\check{\sigma}(S_1^\star) \Supset S_2^\star$ we conclude that $S_2^\star$ must be the empty preference relation, i.e. S_2 is also inconsistent. The only extension $\mathcal{E}_2$ of S_2 is the whole language of the signature of S_2. Trivially $\hat{\sigma}(\mathcal{E}_1) \subseteq \mathcal{E}_2$.
- Assume that S_1 is consistent and let m_1 be a model of the consistent extension $\mathcal{E}_1$. Then m_1 is maximal in $S_1^\star$ and $\mathcal{E}_1$ is the theory of the equivalence class $[m_1]_1$ (theorem 62).
  - We see firstly that $\breve{\overline{\sigma}}(m_1)$ is maximal in $S_2^\star$. Consider any m_2 with $\breve{\overline{\sigma}}(m_1) \sqsubseteq_2 m_2$. We want to conclude that $\breve{\overline{\sigma}}(m_1) \equiv_2 m_2$. In fact, from $\check{\sigma}(S_2^\star) \Subset S_1^\star$, we have $\check{\sigma}(\breve{\overline{\sigma}}(m_1)) \sqsubseteq_1 \check{\sigma}(m_2)$. We have seen in lemma 151 that $\check{\sigma}(\breve{\overline{\sigma}}(m_1))$ satisfies precisely the same Σ_1-formulas as m_1. Therefore m_1 is equivalent to $\check{\sigma}(\breve{\overline{\sigma}}(m_1))$. Moreover $m_1 \equiv_1 \check{\sigma}(\breve{\overline{\sigma}}(m_1)) \sqsubseteq_1 \check{\sigma}(m_2)$ and m_1 is maximal. This implies that $\check{\sigma}(m_2)$ is also equivalent to m_1. From $\check{\sigma}(m_2) \equiv_1 m_1$ and $\breve{\overline{\sigma}}(S_1^\star) \Subset S_2^\star$ we conclude $\breve{\overline{\sigma}}(\check{\sigma}(m_2)) \equiv_2 \breve{\overline{\sigma}}(m_1)$. As before m_2 satisfies precisely the same Σ_2-formulas as $\breve{\overline{\sigma}}(\check{\sigma}(m_2))$ which implies $m_2 \equiv_2 \breve{\overline{\sigma}}(\check{\sigma}(m_2)) \equiv_2 \breve{\overline{\sigma}}(m_1)$ as intended.
  - The formulas holding in the equivalence class $[\breve{\overline{\sigma}}(m_1)]_2$ constitute an extension $\mathcal{E}_2 = [\breve{\overline{\sigma}}(m)]_2^\bullet$ of S_2 (theorem 62). We now see that $\hat{\sigma}(\mathcal{E}_1) \subseteq \mathcal{E}_2$. This amounts to show that any $m_2' \equiv_2 \breve{\overline{\sigma}}(m_1)$ is a model of $\hat{\sigma}(\mathcal{E}_1)$. From $m_2' \equiv_2 \breve{\overline{\sigma}}(m_1)$ and $\check{\sigma}(S_2^\star) \subseteq S_1^\star$ we have $\check{\sigma}(m_2') \equiv_1 \check{\sigma}(\breve{\overline{\sigma}}(m_1)) \equiv_1 m_1$. In this way $\check{\sigma}(m_2')$ belongs to the equivalence class $[m_1]_1$ and models $\mathcal{E}_1$. From the satisfaction condition we conclude that $m_2' \vDash \hat{\sigma}(\mathcal{E}_1)$ as intended. ✓

We illustrate the concepts presented in the previous sections by means of a simple example.

Example 158 Consider a specification NIXON of the well known *Nixon Diamond* example (see, for example, [63]). Nixon is known to be a Republican and a Quaker. This is modeled by the axiom $\texttt{Rep(Nx)} \wedge \texttt{Quak(Nx)}$, where `Nx` is the constant symbol identifying Nixon. Republicans are by default non-pacifists whereas Quakers are by default pacifists. This corresponds to the defaults $\texttt{Quak(Nx)} \Rightarrow \texttt{Pax(Nx)}$ and $\texttt{Rep(Nx)} \Rightarrow \neg\texttt{Pax(Nx)}$.

In this way our choice of symbols is the first order logic signature $\texttt{sg}$(NIXON) with the set of constant symbols $\{\texttt{Nx}\}$ and $\{\texttt{Quak}, \texttt{Rep}, \texttt{Pax}\}$ as set of predicate symbols.

Compare this specification with a simpler version of BATMAN (example 21) having as axioms $\texttt{Hum(bm)} \wedge \texttt{Bat(bm)}$ (`bm` (Batman) is known to be both a bat and a human) and the defaults $\texttt{Bat(bm)} \Rightarrow \texttt{Fl(bm)}$, $\texttt{Hum(bm)} \Rightarrow \neg\texttt{Fl(bm)}$.

The two specifications have the same form. One expects the conclusions of one of them to be translated to the other. We now see how this impression can be formalized with the concepts presented above.

Firstly we can establish a signature morphism $\sigma : \mathtt{sg}(\text{NIXON}) \to \mathtt{sg}(\text{BATMAN})$ from $\mathtt{sg}(\text{NIXON})$ to $\mathtt{sg}(\text{BATMAN})$ by assigning to the constant symbol $\mathtt{Nx}$ the constant symbol $\mathtt{bm}$ and to the predicate symbols $\mathtt{Quak}, \mathtt{Rep}, \mathtt{Pax}$ the predicate symbols $\mathtt{Bat}, \mathtt{Hum}$ and $\mathtt{Fl}$ respectively. This signature morphism is clearly a bijection. Moreover this signature morphism translates the axiom $\mathtt{Rep}(\mathtt{Nx}) \wedge \mathtt{Quak}(\mathtt{Nx})$ to the formula $\mathtt{Hum}(\mathtt{bm}) \wedge \mathtt{Bat}(\mathtt{bm})$ and the defaults $\mathtt{Quak}(\mathtt{Nx}) \Rightarrow \mathtt{Pax}(\mathtt{Nx})$ and $\mathtt{Rep}(\mathtt{Nx}) \Rightarrow \neg\mathtt{Pax}(\mathtt{Nx})$ to $\mathtt{Bat}(\mathtt{bm}) \Rightarrow \mathtt{Fl}(\mathtt{bm})$ and $\mathtt{Hum}(\mathtt{bm}) \Rightarrow \neg\mathtt{Fl}(\mathtt{bm})$ respectively. Clearly this signature morphism is a morphism between both specifications. Moreover the inverse signature morphism is also a specification morphism. In this way the two specifications are isomorphic in the category of first order specifications.

We are now concerned in displaying the impact of this fact on the conclusions of both specifications. We expect that we can translate the consequences of one into consequences of the other since they only differ in the choice of vocabulary, but have the same logical content. For this purpose we turn to the semantics of both specifications. The corresponding preference relations are also isomorphic in the category of first order preference relations. The preference relation of BATMAN compares the models of $\mathtt{Hum}(\mathtt{bm}) \wedge \mathtt{Bat}(\mathtt{bm})$ according to its defaults. The reducts of these models are interpretation structures participating in the preference relation of NIXON. These are models of $\mathtt{Rep}(\mathtt{Nx}) \wedge \mathtt{Quak}(\mathtt{Nx})$.

It is not difficult to see that the preference of NIXON coincides with the reduct of the preference of BATMAN. The later consists of two unrelated equivalence classes: that of the models of $\mathtt{Fl}(\mathtt{bm})$ and that of the models of $\neg\mathtt{Fl}(\mathtt{bm})$. The reducts of the models of $\mathtt{Fl}(\mathtt{bm})$ are the models of $\mathtt{Pax}(\mathtt{Nx})$. Since the models of $\mathtt{Fl}(\mathtt{bm})$ are equivalent so are the models of $\mathtt{Pax}(\mathtt{Nx})$. Similarly with the models of $\neg\mathtt{Fl}(\mathtt{bm})$ and $\neg\mathtt{Pax}(\mathtt{Nx})$. In this way, as expected, the preference of NIXON consist of the two equivalence classes of the models of $\mathtt{Pax}(\mathtt{Nx})$ and $\neg\mathtt{Pax}(\mathtt{Nx})$.

From lemma 157 we know that the extensions of BATMAN are obtained from those of NIXON by changing the signature symbols (and vice versa). Since Batman credulously flies we conclude that Nixon is credulously a pacifist. Skeptically we can not conclude neither that Nixon is a pacifist nor that Batman flies. $\triangle$

3.1.5 Existence of Constructions

Colimits of diagrams in the category of specifications express combinations of the parameter specifications, related to each other as described by the corresponding diagram. The existence of a colimit states that there exists a specification which expresses the result of combining the parameter specifications in this particular way.

We now see that the existence of colimits in Spec depends only on the existence of a colimit signature. In this way any form of combination of specifications has a result, as long as there is a signature rich enough to express the result specification. The resulting specification is the smallest one preserving the axioms and defaults of the parameter specifications.

Theorem 159 The category Spec is cocomplete if the underlying category Sign of signatures is so.

Proof Recall the definition of diagram, cocone and colimit in 154. The colimit of a (small) diagram $\mathcal{D} : \mathcal{S} \to \mathsf{Spec}$ is obtained by lifting the colimit in Sign of the corresponding diagram $\mathcal{F} \circ \mathcal{D} : \mathcal{S} \to \mathsf{Sign}$ where $\mathcal{F}$ is the forgetful functor sending Spec to the underlying category of signatures. The functor $\mathcal{F}$ sends each specification S to its signature $\mathsf{sg}(S)$ and each morphism $\sigma : S \to S'$ to the signature morphism $\sigma : \mathsf{sg}(S) \to \mathsf{sg}(S')$.

We will denote the specification assigned by $\mathcal{D}$ to the node $i \in |\mathcal{S}|$ by S_i. Its signature will be Σ_i. In this way Σ_i is the signature assigned by the diagram $\mathcal{F} \circ \mathcal{D}$ to the node $i \in |\mathcal{S}|$.

Let $\mathcal{C}_{\mathsf{Sign}} = \{\sigma_i : \Sigma_i \to \Sigma^{\oplus}, i \text{ in } |\mathcal{S}|\}$ be the cocone colimit of $\mathcal{F} \circ \mathcal{D}$ (in Sign). We have to provide the specification $S^{\oplus}$ of the colimit signature $\Sigma^{\oplus}$ such that $\mathcal{C}_{\mathsf{Spec}} = \{\sigma_i : S_i \to S^{\oplus}, i \text{ in } |\mathcal{S}|\}$ is a colimit in Spec.

Choose for $S^{\oplus}$ the union $\Cup_{i\in|\mathcal{S}|}\hat{\sigma}_i(S_i)$, i.e. $\mathsf{ax}(S^{\oplus})$ is the union $\bigcup_{i\in|\mathcal{S}|} \hat{\sigma}_i(\mathsf{ax}(S_i))$ of the σ_i-translations of the axioms of the parameter specifications and $\mathsf{df}(S^{\oplus})$ is the union $\bigcup_{i\in|\mathcal{S}|} \hat{\sigma}_i(\mathsf{df}(S_i))$ of the translations of their defaults.

In this way $S^{\oplus}$ is the smallest (w.r.t. $\Subset$) specification such that each $\sigma_i : S_i \to S^{\oplus}$ is a Spec-morphism. Moreover $\mathcal{C}_{\mathsf{Spec}}$ is a commutative cocone (for $\mathcal{D}$) since $\mathcal{C}_{\mathsf{Sign}}$ is a commutative cocone (for $\mathcal{F} \circ \mathcal{D}$).

The property of colimit for $\mathcal{C}_{\mathsf{Spec}}$ follows from the corresponding property for $\mathcal{C}_{\mathsf{Sign}}$. We have to check that, given another commutative cocone $\mathcal{C}'_{\mathsf{Spec}} = \{\nu_i : S_i \to S', i \text{ in } |\mathcal{S}|\}$ for $\mathcal{D}$ there is a unique morphism $\mu : S^{\oplus} \to S'$ such that $\nu_i = \mu \cdot \sigma_i$, for all i in $|\mathcal{S}|$.

From the commutative cocone $\mathcal{C}'_{\mathsf{Spec}}$ we obtain the commutative cocone $\mathcal{C}'_{\mathsf{Sign}} = \{\nu_i : \Sigma_i \to \mathsf{sg}(S'), i \text{ in } |\mathcal{S}|\}$ for $\mathcal{F} \circ \mathcal{D}$. Since Sign is cocomplete there exists a unique signature morphism $\mu : \Sigma^{\oplus} \to \Sigma'$ such that $\nu_i = \mu \cdot \sigma_i$, for all i in $|\mathcal{S}|$. We only have to show that this signature morphism is also a Spec morphism, i.e. that $\hat{\mu}(S^{\oplus}) \Subset S'^{\star\star}$. Since $\nu_i = \mu \cdot \sigma_i$ we have $\hat{\nu}_i(S_i) = \hat{\mu}(\hat{\sigma}_i(S_i))$. The morphism condition for each S_i is $\hat{\nu}_i(S_i) = \hat{\mu}(\hat{\sigma}_i(S_i)) \Subset S'^{\star\star}$. This clearly implies $\hat{\mu}(S^{\oplus}) \Subset S'^{\star\star}$ since $S^{\oplus} = \Cup_{i\in|\mathcal{S}|}\hat{\sigma}_i(S_i)$.
✓

As seen above colimits in Spec correspond to “unions” of axioms and defaults. Their semantic counterpart, limits in Pref correspond to “intersections”, both of the classes of models and the preference relations. Existence of constructions in PreOrder (and Pref) is assured from theorems 155 and 159 and is stated in the next theorem. Its (extra) proof displays the limits in PreOrder (and Pref).

Theorem 160 The categories PreOrder and Pref are complete if the underlying category Sign of signatures is cocomplete.

Proof The limit of a (small) diagram $\mathcal{D} : \mathcal{S} \to \mathsf{PreOrder}$ is obtained by lifting the colimit of the diagram $\mathcal{G} \circ \mathcal{D} : \mathcal{S} \to \mathsf{Sign}$ where $\mathcal{G}$ is the forgetful functor sending PreOrder to the underlying category of signatures. The functor $\mathcal{G}$ sends each pre-order R to its signature $\Sigma = \mathsf{sg}(R)$ and each PreOrder-morphism $\overleftarrow{\sigma}: R \to R'$ to the signature morphism $\sigma : \mathsf{sg}(R') \to \mathsf{sg}(R)$ (note the contravariance).

We will denote the pre-order assigned by $\mathcal{D}$ to the node $i \in |\mathcal{S}|$ by R_i. Its signature will be Σ_i. In this way Σ_i is the signature assigned by the diagram $\mathcal{G} \circ \mathcal{D}$ to the node $i \in |\mathcal{S}|$.

Let $\mathcal{C}_{\mathsf{Sign}} = \{\sigma_i : \Sigma_i \to \Sigma^\oplus, i \in |\mathcal{S}|\}$ be the cocone colimit of $\mathcal{G} \circ \mathcal{D}$ (in Sign). We have to provide the pre-order $R^\otimes$ among the interpretation structures of the colimit signature $\Sigma^\oplus$ such that $\mathcal{L}_{\mathsf{PreOrder}} = \{\overleftarrow{\sigma}_i: R^\otimes \to R_i, i \in |\mathcal{S}|\}$ is a limit in PreOrder.

The interpretation structures participating in $R^\otimes$ are the $\Sigma^\oplus$-interpretation structures whose reducts w.r.t. each σ_i participate in the parameter pre-order R_i. That is $|R^\otimes| = \{m \in |\mathsf{Mod}(\Sigma^\oplus)| : \check{\sigma}_i(m) \in |R_i|,$ for all $i \in |\mathcal{S}|\}$. Two interpretation structures are related by $R^\otimes$ iff their reducts w.r.t. σ_i are related in each parameter pre-order R_i. That is $m \sqsubseteq^\otimes n$ iff $\check{\sigma}(m) \sqsubseteq_i \check{\sigma}(n)$ for all $i \in |\mathcal{S}|$ ($\sqsubseteq^\otimes$ is $\mathsf{rl}(R^\otimes)$ and $\sqsubseteq_i$ is $\mathsf{rl}(R_i)$). It is trivial to check that $R^\otimes$ is a pre-order. It is also the biggest $\Sigma^\oplus$-pre-order such that each $\overleftarrow{\sigma}_i: R^\otimes \to R_i$ is a PreOrder morphism. Moreover $\mathcal{L}_{\mathsf{PreOrder}}$ is a commutative cone (for $\mathcal{D}$) since $\mathcal{C}_{\mathsf{Sign}}$ is a commutative cocone for $\mathcal{G} \circ \mathcal{D}$.

The property of limit for $\mathcal{L}_{\mathsf{PreOrder}}$ follows from the corresponding property of colimit for $\mathcal{C}_{\mathsf{Sign}}$. Let $\mathcal{L}'_{\mathsf{PreOrder}} = \{\overleftarrow{\nu}_i: R' \to R_i, i \in |\mathcal{S}|\}$ be a commutative cone for $\mathcal{D}$. We need to provide a unique $\overleftarrow{\mu}: R' \to R^\otimes$ such that $\overleftarrow{\nu_i}=\overleftarrow{\sigma_i} \cdot \overleftarrow{\mu}$, for all i in $|\mathcal{S}|$. From the commutative cone $\mathcal{L}'_{\mathsf{PreOrder}}$ for $\mathcal{D}$ we obtain the commutative cocone $\mathcal{C}'_{\mathsf{Sign}} = \{\nu_i : \Sigma_i \to \mathsf{sg}(R'), i \in |\mathcal{S}|\}$ for $\mathcal{G} \circ \mathcal{D}$. Since Sign is cocomplete there is a unique signature morphism $\mu : \Sigma^\oplus \to \mathsf{sg}(R')$ such that $\nu_i = \mu \cdot \sigma_i$, for all i in $|\mathcal{S}|$. We only have to show that $\overleftarrow{\mu}: R' \to R^\otimes$ is a PreOrder-morphism. This amounts to show $\check{\mu}(R') \Subset R^\otimes$.

- Let m' be an interpretation structure from $|R'|$. We want to check that $\check{\mu}(m') \in |R^\otimes|$, i.e. that $\check{\sigma}_i(\check{\mu}(m')) \in |R_i|$, for all $i \in |\mathcal{S}|$. Now $\check{\sigma}_i(\check{\mu}(m')) = \check{\nu}_i(m')$ since $\nu_i = \mu \cdot \sigma_i$. And $\check{\nu}_i(m') \in |R_i|$, since each $\overleftarrow{\nu}_i: R' \to R_i$ is a PreOrder-morphism. In this way $\check{\mu}(|R'|) \subseteq |R^\otimes|$.

- Let now $m' \sqsubseteq' n'$ where $\sqsubseteq'$ is $\mathsf{rl}(R')$. We want to show that $\check{\mu}(m') \sqsubseteq^\otimes \check{\mu}(m')$, i.e. that, for all $i \in |\mathcal{S}|$, $\check{\sigma}(\check{\mu}(m')) \sqsubseteq_i \check{\sigma}(\check{\mu}(n'))$. This is equivalent to $\check{\nu}_i(m') \sqsubseteq_i \check{\nu}_i(m)$ and holds since $\check{\nu}_i(\mathsf{rl}(R')) \subseteq \mathsf{rl}(R_i)$, $i \in |\mathcal{S}|$ (each $\overleftarrow{\nu}_i: R' \to R_i$ is a PreOrder-morphism). Therefore $\check{\mu}(\mathsf{rl}(R')) \subseteq \mathsf{rl}(R^\otimes)$. This ends the proof of completeness of PreOrder.

Limits in Pref are calculated in the same way. We have only to check that $R^\otimes$ is the preference of some specification $S^\oplus$ whenever each R_i is the preference of some S_i. This is established as in the proof of theorem 155: limits in Pref are the image by Sem of colimits in Spec. ✓

We now illustrate the constructions above for the case when all specifications share the same signature. In this case colimits are unions of axioms and defaults.

Example 161 Let $\emptyset$ denote the Σ-specification with an empty set of axioms and defaults. Let S_1 and S_2 be Σ-specifications and let $\emptyset \to S_1$ and $\emptyset \to S_2$ denote the morphisms corresponding to the identity signature morphism from Σ to Σ. Then the (vertex of the) pushout of the diagram $S_1 \leftarrow \emptyset \to S_2$ is the specification $S_1 \Cup S_2$ having as axioms the union of the axioms from S_1 and S_2 and as defaults the union of the defaults from S_1 and S_2. This is illustrated by the following figure.

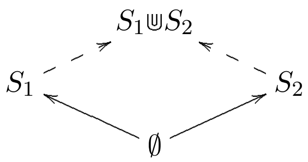

Figure 3.4: Union as a Colimit

Note that, from theorem 155 the previous diagram can be redrawn at the semantic level by changing the specifications to their preference relations and reverting the morphisms. The preference relation associated with $S_1 \Cup S_2$ will be the limit of the new diagram. This preference relation is $S_1^\star \Cap S_2^\star$, the intersection, both on the classes of models and the classes of relation pairs of the preference relations associated with and S_1 and S_2. △

More generally, when a default theory presentation is the domain of several morphisms it states what symbols, axioms and defaults are shared by the codomain specifications. We use this fact to get the properties of `bm` (Batman) from the properties of bats and humans in the following example.

Example 162 Consider the specifications BATS and HUMANS. The specification BATS consists of the axioms `Bat(b)` and the defaults `Bat(b)` $\Rightarrow$ `Fl(b)` for b in a designated set of constant symbols identifying bats. The specification HUMANS is similar and ascertains that humans usually dream and do not fly, where humans are identified in a set of constant symbols. We now compose these two specifications by stating that the predicate `Fl` is the same in both specifications and that there is a particular being that is a bat and a human. The result is the colimit of the diagram BATS $\leftarrow (\Sigma, \emptyset, \emptyset) \to$ HUMANS where the signature Σ consists of the predicate symbol `Fl` and the constant symbol `bm`. The signature morphism $(\Sigma, \emptyset, \emptyset) \to$ HUMANS assigns to the predicate symbol `Fl` the predicate symbol `Fl` from the specification HUMANS and to the constant symbol `bm` its identifying symbol as an human. The signature morphism $(\Sigma, \emptyset, \emptyset) \to$ BATS is defined in a similar way. The following diagram illustrates this composition.

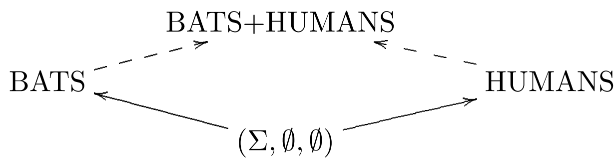

Figure 3.5: Composition of BATS and HUMANS

Note that in BATS+HUMANS there will be a special constant symbol, call it `bm` that is a bat (`Bat(bm)`) and a human (`Hum(bm)`)), and that, by default, flies and does not fly and dreams. △

3.1.6 Canonicity of the Semantics

In the previous sections we have displayed a formalization of composition of specifications and the corresponding semantical account: to each syntactic construction given by a colimit in Spec (a composition of the argument specifications) there corresponds a semantic construction given by a limit in Pref.

We see in this section that (under some conditions) the preference semantics is the *minimal* one that interprets composition of specifications.

To prove this we proceed by identifying two requirements on semantics of specifications. The first is *logical compatibility*: specifications having different consequences should have different semantics. The second is the ability of *interpreting composition* of specifications, in particular the operation of *union*.

Finally we see that the preference semantics is *minimal* among the semantics satisfying these two criteria. This result is only proved for semantics of specifications from *compact* institutions having *negation*.

We begin with the (rather general) definition of semantics of Σ-specifications.

Definition 163 A *semantics* of Σ-specifications is a function $[\![]\!]$ with the set of all Σ-specifications as domain. ▪

Clearly some of these "semantics" have to be rejected. For example the function that assigns to all specifications the same semantics cannot be accepted as a proper semantics. The requirement of logical compatibility, that we now present, rejects such unreasonable cases.

Logical compatibility is a consequence of imposing that the consequences of a specification may be derived from the corresponding semantics. In particular this implies that specifications having different consequences must have different semantics.

Since there are (at least) two common types of consequence of a specification, the skeptical and the credulous consequences, there are also two different notions of logical compatibility: logical compatibility w.r.t. the skeptical consequences and logical compatibility w.r.t. the credulous consequences.

Definition 164 Let $[\![]\!]$ be a *semantics* of Σ-specifications. The semantics $[\![]\!]$ is said to be

- *logically compatible with the skeptical consequences* iff whenever the skeptical consequences of two Σ-specifications S_1 and S_2 differ then also $[\![S_1]\!] \neq [\![S_2]\!]$ and
- *logically compatible with the credulous consequences* iff whenever the credulous consequences of two Σ-specifications S_1 and S_2 differ then also $[\![S_1]\!] \neq [\![S_2]\!]$. ∎

The preference semantics is logically compatible with both the skeptical and the credulous consequences. Other (reasonable) examples of semantics are logically compatible with at least one of these consequences: the credulous consequences of S, seen as the semantics of S, is trivially compatible with the credulous consequences. Similarly with the skeptical consequences (or the class of maximal models) of S. The semantics obtained by assigning to S the set of extensions of S (or the set of maximal equivalence classes of S) is compatible with both consequence types.

Logical compatibility is an expected property of any semantics of specifications. The second requirement, that we now formalize, is stronger. We require from such a semantics that it interprets the operation $\Cup$. This means that the semantics of $S_1 \Cup S_2$ should be given by an operation having as arguments the semantics of S_1 and S_2 only[6]. More formally:

Definition 165 Let $[\![]\!]$ be a *semantics* of Σ-specifications with codomain $\mathcal{C}$. The semantics $[\![]\!]$ is said to *interpret* $\Cup$ if there exists an operation $\mathcal{O} : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ such that

$$[\![S_1 \Cup S_2]\!] = \mathcal{O}([\![S_1]\!], [\![S_2]\!]),$$

for any Σ-specifications S_1 and S_2. ∎

[6]In other words: the semantics of specifications is *denotational* with respect to $\Cup$.

The preference semantics interprets the operation $\Cup$ since the preference semantics of $S_1 \Cup S_2$ is $S_1^\star \Cap S_2^\star$ (see the Galois connection in theorem 30). None of the alternative semantics presented so far has this property.

Example 166 Take for $[\![S]\!]$ the class of maximal models of S (or equivalently the skeptical consequences of S). We show that this semantics does not interpret $\Cup$. Consider the first order logic specifications $S_1 = (\emptyset, \emptyset)$, $S_1' = (\emptyset, \{f, \neg f\})$ and $S_2 = (\emptyset, \{f\})$, where f is not a tautology. Note that the maximal models of S_1 and S_1' coincide. Therefore $[\![S_1]\!] = [\![S_1']\!]$. Assume, by absurd, that this semantics interprets $\Cup$. Then $[\![S_1 \Cup S_2]\!] = \mathcal{O}([\![S_1]\!], [\![S_2]\!]) = \mathcal{O}([\![S_1']\!], [\![S_2]\!]) = [\![S_1' \Cup S_2]\!]$. Since $S_2 = S_1 \Cup S_2$ and $S_1' = S_1' \Cup S_2$ we conclude $[\![S_1']\!] = [\![S_2]\!]$. But this is not true since the maximal models of S_2 (the models of f) are not the maximal models of S_1' (these are all Σ-interpretation structures).

This example also shows that the alternative semantics corresponding to the credulous consequences, or to the set of extensions do not, in the same way, interpret $\Cup$. △

We know that the preference semantics interprets $\Cup$ and is logically compatible with both types of consequence. We see now that it is the minimal semantics with these properties. In fact we show more: any semantics that is logically compatible with *at least one of the types of consequence* and interprets $\Cup$ has more structure than the preference semantics.

The following definition is of convenience.

Definition 167 A semantics $[\![]\!]$ of Σ-specifications is said a *proper compositional semantics* iff it interprets $\Cup$ and is compatible with the skeptical *or* with the credulous consequences. ▪

Finally we have to formalize the relation "$[\![]\!]$ has more structure than $[\![]\!]'$". This is motivated as follows. Take the set of extensions of S as its semantics. This semantics has more structure than the semantics corresponding to assigning to S its credulous or skeptical consequences. In fact the credulous and the skeptical consequences can be derived from the set of extensions of S (and not the other way around). Therefore the set of extensions has more structure (more information) than these consequences. Similarly the preference of S has more structure than the set of extensions of S (and again more structure that the skeptical or credulous consequences.) Note that two specifications having the same preference have the same set of extensions. And two specifications having the same sets of extensions have the same credulous and skeptical consequence. This is the property we choose for comparing semantics.

Definition 168 Let $[\![]\!]$ and $[\![]\!]'$ be two *semantics* of Σ-specifications. The relation $[\![]\!] \succcurlyeq [\![]\!]'$, read "$[\![]\!]$ has more structure than $[\![]\!]'$" is defined as follows: $[\![]\!] \succcurlyeq [\![]\!]'$ iff, given any Σ-specifications S_1 and S_2, whenever $[\![S_1]\!] = [\![S_2]\!]$ then also $[\![S_1]\!]' = [\![S_2]\!]'$. ∎

Any semantics logically compatible with the skeptical consequences has more structure than these consequences (seen as semantics). In fact the two propositions are equivalent (see definition 164). The same holds for the credulous consequences. Therefore a proper compositional semantics has more structure than the skeptical or than the credulous consequences.

We now show that the preference semantics is the one having less structure from among the proper compositional semantics. Note again that this result is only proved for semantics of specifications from *compact* institutions *having negation* (see definitions 65 and 118 and also remark 119).

Theorem 169 Let Σ be a signature of a compact institution $(\mathcal{I}, \mathsf{neg})$ *having negation*. The preference semantics is the *least* (w.r.t. $\succcurlyeq$) proper compositional semantics of these Σ-specifications.

Proof We have to show that, given a proper compositional semantics $[\![]\!]$ of Σ-specifications and Σ-specifications S_1 and S_2 if $[\![S_1]\!] = [\![S_2]\!]$ then $S_1^\star = S_2^\star$. The structure of the proof is the following: Assume that $S_1^\star \neq S_2^\star$ and add ($\Cup$) to both S_1 and S_2 the same (well chosen) specification S_3, thus obtaining specifications S_1' and S_2'. Since $[\![]\!]$ interprets $\Cup$ we have that $[\![S_1]\!] = [\![S_2]\!]$ implies $[\![S_1']\!] = [\![S_2']\!]$. Therefore, either the credulous or skeptical consequences of S_1' and S_2' must coincide. We conclude that this is not the case.

- We begin by showing that $|S_1^\star| = |S_2^\star|$ if $[\![S_1]\!] = [\![S_2]\!]$. Assume $|S_1^\star| \neq |S_2^\star|$. Then there is an interpretation structure m such that $m \in |S_1^\star|$ and $m \notin |S_2^\star|$ (or vice versa). This means that m is a model of the axioms from S_1 and it is not a model of the axioms from S_2. Let $a_2 \in \mathsf{ax}(S_2)$ be an axiom from S_2 not satisfied by m. Then m satisfies $\neg a_2$ ($\neg a_2$ abbreviates $\mathsf{neg}_\Sigma(a_2)$).

  Consider the specifications $S_1' = S_1 \Cup (\{\neg a_2\}, \emptyset)$ and $S_2' = S_2 \Cup (\{\neg a_2\}, \emptyset)$. We now see that S_1' and S_2' have different credulous and different skeptical consequences. Note that S_2' is inconsistent (both a_2 and $\neg a_2$ are axioms) whereas S_1' is not (m is a model of the axioms from S_1 and also of $\neg a_2$). Since S_2' is inconsistent it only has one extension and its skeptical and credulous consequences are the whole Σ-language. In particular both a_2 and $\neg a_2$ are credulous and skeptical consequences of S_2'. On the other hand S_1' is consistent and so are its extensions. Moreover each such extension contains the axioms, in particular $\neg a_2$. Therefore no extension contains a_2. In this way a_2 is neither a credulous nor a skeptical consequence of S_1'.

- Assume now that $[\![S_1]\!] = [\![S_2]\!]$ and $|S_1^\star| = |S_2^\star|$ but $\sqsubseteq_1 \neq \sqsubseteq_2$ (i.e. $\mathsf{rl}(S_1^\star) \neq \mathsf{rl}(S_2^\star)$). This means that there exist $m, n \in |S_1^\star|$ with $m \sqsubseteq_1 n$ but $m \not\sqsubseteq_2 n$ (or vice versa). The interpretation structure n satisfies all defaults from S_1 that m satisfies and there is a default d_2 from S_2 satisfied by m and not by n. Note also that both specifications are consistent since $|S_1^\star| = |S_2^\star| \neq \emptyset$.

  Consider now the sets $D_1(m) = \mathsf{df}(S_1)(m)$ of the defaults from S_1 satisfied by m and $\overline{D}_1(n) = \{\neg d_1 : n \nvDash d_1 \text{ and } d_1 \in \mathsf{df}(S_1)\}$ of the negations of the defaults from S_1 not satisfied by n. Let $S_3 = (D_1(m) \cup \overline{D}_1(n), \{\neg d_2\})$ be the specification having the union of $D_1(m)$ and $\overline{D}_1(n)$ as axioms and $\neg d_2$ as single default. Let $S_1' = S_1 \uplus S_3$ and $S_2' = S_2 \uplus S_3$. We now proceed to show that, although $[\![S_1']\!] = [\![S_2']\!]$, their skeptical and credulous consequences differ.

  Note firstly that both m and n belong to $|S_1'^\star| = |S_2'^\star|$. In fact both are models of $\mathsf{ax}(S_1)$ and $\mathsf{ax}(S_2)$ (since $m, n \in |S_1^\star| = |S_2^\star|$) and moreover they are also models of $D_1(m)$ and $\overline{D}_1(n)$. In this way both S_1' and S_2' are consistent and so are their extensions.

  We now see that d_2 is a credulous consequence of S_2' and also that $\neg d_2$ is not a skeptical consequence of S_2'. Since $m \vDash d_2$ and m is a model of the axioms of S_2' we conclude that $\mathsf{ax}(S_2') \cup \{d_2\}$ is consistent. Since $d_2 \in \mathsf{df}(S_2) \subseteq \mathsf{df}(S_2')$ we know from *coverage* in theorem 67 that there is a consistent extension of S_2' containing d_2. This implies that d_2 is a credulous consequence of S_2' and also that $\neg d_2$ is not a skeptical consequence of S_2'. We see below that $\neg d_2$ is a skeptical (and credulous) consequence of S_1' and that d_2 is not a credulous consequence of S_1'. In this way S_1' and S_2' have different credulous consequences and different skeptical consequences, contradicting the hypothesis.

  We now have to see that $\neg d_2$ is a skeptical consequence of S_1' and that d_2 is not a credulous consequence of S_1'.

  The preference relation of S_1' is obtained from that of S_1 by further comparing the interpretation structures according to $\neg d_2$ and restricting the result to the models of the axioms of S_1 that are also models of $D_1(m, n)$. Recall that both m and n participate in $S_1'^\star$. Moreover $m \sqsubseteq_1' n$ since the new default $\neg d_2$ is satisfied by n (and not by m). Any other interpretation structure m' from $S_1'^\star$ is such that $m \sqsubseteq_1' m' \sqsubseteq_1' n$. In fact m' satisfies at least the same defaults from S_1' as m since m' is a model of $D_1(m)$ and m does not satisfy $\neg d_2$. And m' does not satisfy more defaults than n since m' is a model of $\overline{D}_1(n)$ and n satisfies the new default $\neg d_2$. This implies that n is maximal in $S_1'^\star$ and its equivalence class is the only maximal equivalence class of $S_1'^\star$. In this way $S_1'^\star$ has only one extension, namely $\mathcal{E}_1' = ([n]_1')^\bullet$. Both the credulous and skeptical consequences of S_1' coincide with $\mathcal{E}_1'$. Note that $\neg d_2$ is in $\mathcal{E}_1'$ since the interpretation structures equivalent to n must satisfy precisely the same defaults from S_1' as n, in particular $\neg d_2$. Therefore $\neg d_2$ is both a skeptical and credulous consequence of S_1'. And d_2 is neither a credulous or skeptical consequence of S_1'. ✓

3.2 Hierarchic Specifications

The properties of the composition of hierarchic specifications generalize corresponding properties of composition of specifications. The added expressiveness is given by composition of the partial orders of priority. This is presented in the next section 3.2.1. Other concepts and properties are presented in a form structurally similar to section 3.1. Composition of specifications is formalized on the syntactical side by colimits in the category hieSpec of hierarchic specifications (section 3.2.2). This category is mirrored on the semantic side by the category hiePref of hierarchies of differential preferences (section 3.2.3). The relation between both formalizations is presented in section 3.2.4. We see in section 3.2.5 that isomorphic hierarchic specifications have the same logical meaning. In section 3.2.6 we study the existence of constructions that formalize composition of hierarchic specifications. In section 3.2.7 we show that the hierarchy of differential preferences is the least semantics of hierarchic specifications that supports composition of hierarchic specifications.

3.2.1 Category of Partial Orders

The additional compositional expressiveness available for hierarchic specifications results from composition of the respective partial orders of priority. Composition of partial orders is formalized by colimits in the category 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 (since this identification is represented by the union of their defaults).

Definition 170 The category 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 StPart is *not* cocomplete as opposed to the category Part of partial orders and partial order morphisms. It is however easy to check that the colimits in StPart, when they exist, coincide with the corresponding colimits in Part.

The colimits ruled out in StPart correspond to combinations of partial orders whose result is obtained by identifying levels related in the argument partial orders (see below for an example of this situation). These combinations of partial orders are not considered meaningful in the formalism presented in the next sections.

Example 171 Consider the priority orderings sketched in the following picture and the (trivially strict) morphisms i sending $\bullet$ to $\bullet$ and j sending $\bullet$ to $\circ$.

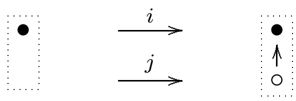

Figure 3.6: Forbidden Composition

The colimit of this diagram in Part (the co-equalizer of the two morphisms) identifies $\circ$ and $\bullet$. Therefore it cannot be a colimit in StPart. △

In the following two examples (see also example 191) we assume that the relations to be composed have a minimum and a maximum (priority level). These examples are inspired in the formalism presented in [2]. There, the authors define and algebra based on two operations on relations: "But" and "On The Other Hand". The terms build with these operations denote preference relations. "R_1 But R_2"

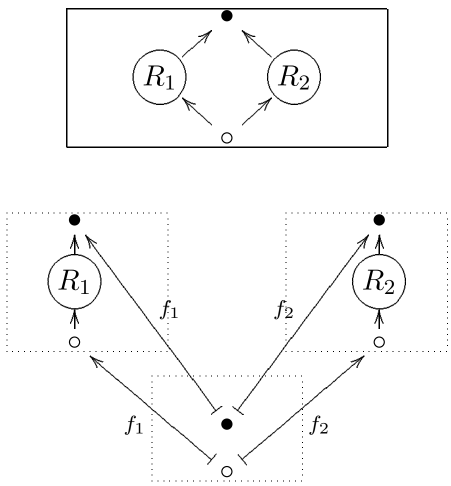

Figure 3.7: R_1 On The Other Hand R_2

is understood as "R_1 is less important than R_2" and denotes the lexicographic combination of both, assigning more importance to R_2. "R_1 On The Other Hand

R_2" is read "R_1 is as important as R_2", and denotes again the lexicographic combination of both, seen as equally important (in this case the lexicographic combination coincides with the intersection).

As the authors show, it is not difficult to express a hierarchy of pre-orders in terms of these operations. The corresponding term denotes the lexicographic combination of all those pre-orders.

These operations are represented in our formalism by explicitly putting one hierarchic specification above or to the side of the other. The same operations can be defined on the semantic side (by combining their hierarchies of differential preferences in the same way).

For that purpose it is only necessary to exhibit the corresponding operations on the priority structure. This is shown in the next two examples.

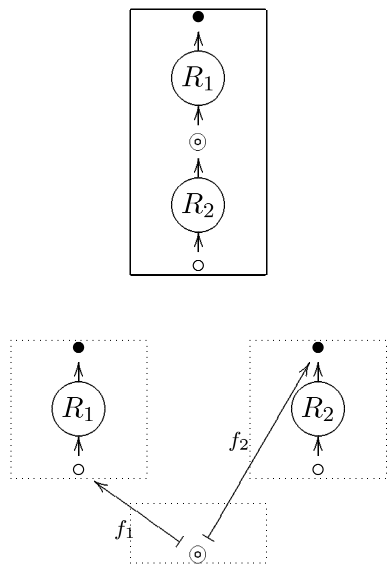

Figure 3.8: R_1 But R_2

Example 172 The operation "On The Other Hand" corresponds to combine the partial orders R_1 and R_2 in "parallel", i.e. each is equally important. This is simply the "coproduct" of R_1 and R_2 (with the identification of the minima and maxima). The diagram and the vertex of its colimit are shown in figure 3.7. △

Example 173 In this example the minimum of R_1 is identified with the maximum of R_2. In this way the resulting partial order will have the levels R_1 above those of R_2 ("R_1 But R_2"). The diagram and the vertex of its colimit are shown in figure 3.8. △

3.2.2 Category of Hierarchic Specifications

In this section we introduce the category hieSpec with hierarchic specifications as objects. Hierarchic specifications are compared via the respective theories. Theories of different signatures and different partial orders of priority are related by signature and partial order morphisms that preserve the axioms and the defaults from corresponding priority levels. The category hieSpec extends the possible constructions of Spec by adding to specifications a priority structure[7].

We firstly need to translate a hierarchic specification to another with a different signature and a different partial order of priority. Such translation is induced both by a signature and a partial order morphism. The formulas in the original hierarchic specification are translated by the signature morphism. The translations of the axioms will be the new axioms. The translations of the defaults at each level h will be the defaults at the new priority level image of h. When different levels have the same image the union of the translations of their defaults is taken. This also covers the case of those new h' which are not the image of some h. They are assigned the empty set of defaults.

Definition 174 Let S_1 be a Σ_1-hierarchic specification and $(H_1, \preceq_1) = \mathsf{po}(S_1)$ its partial order of priority. Let $\phi : (H_1, \preceq_1) \to (H_2, \preceq_2)$ be a partial order morphism and $\sigma : \Sigma_1 \to \Sigma_2$ a signature morphism. Recall that $\hat{\sigma} : \mathsf{Sen}(\Sigma_1) \to \mathsf{Sen}(\Sigma_2)$ assigns to each Σ_1-formula a Σ_2-formula. In definition 143 $\hat{\sigma}$ is extended to sets of formulas (and specifications). The *translation* $\hat{\tau}^{\sigma}_{\phi}(S_1)$ of S_1 with respect to both ϕ and σ is the Σ_2-hierarchic specification with:

- $\mathsf{ax}(\hat{\tau}^{\sigma}_{\phi}(S_1)) = \hat{\sigma}(\mathsf{ax}(S_1))$, the image by $\hat{\sigma}$ of the axioms of S_1 as axioms,
- the partial order $(H_2, \preceq_2)$ codomain of ϕ as partial order of priority and
- for each $h_2 \in H_2$ the set $\mathsf{df}(\hat{\tau}^{\sigma}_{\phi}(S_1), h_2) = \bigcup_{\{h_1 : \phi(h_1) = h_2\}} \hat{\sigma}(\mathsf{df}(S_1, h_1))$, the union of the images by $\hat{\sigma}$ of the defaults from S_1 at the levels h_1 such that $\phi(h_1) = h_2$ (note that $\mathsf{df}(\hat{\tau}^{\sigma}_{\phi}(S_1), h_2)$ is empty if h_2 there is no h_1 such that $\phi(h_1) = h_2$).

It will also be technically convenient to see the translation $\hat{\tau}^{\sigma}_{\phi}$ as consisting of the composition of two operations, the signature translation of axioms and defaults and the priority translation:

- the operation $\hat{\sigma}$ assigns to a hierarchic specification S_1 the hierarchic specification $\hat{\sigma}(S_1)$ with axioms $\hat{\sigma}(\mathsf{ax}(S_1))$, the same partial order of priority as S_1

[7] The category Spec is the special case of hieSpec obtained by restricting the hierarchic specifications to those with one only priority level (i.e. Spec is isomorphic to the full subcategory of hieSpec obtained by considering only those hierarchic specifications).

and, for each level $h \in \mathsf{po}(\hat{\sigma}(S_1)) = \mathsf{po}(S_1)$ the set of defaults $\hat{\sigma}(\mathsf{df}(S_1, h))$ and

- the operation ϕ that assigns to S_1 the hierarchic specification $\phi(S_1)$ with the same axioms as S_1, the partial order $(H_2, \preceq_2)$ codomain of ϕ as partial order of priority and for each $h_2 \in H_2$ the union $\bigcup_{\{h_1:\phi(h_1)=h_2\}} \mathsf{df}(S_1, h_1)$ of the defaults from S_1 at the levels $h_1 \in \mathsf{po}(S_1)$ such that $\phi(h_1) = h_2$.

With these definitions we have $\hat{\tau}^{\sigma}_{\phi}(S_1) = \phi(\hat{\sigma}(S_1))$. ■

The definition of the category hieSpec follows.

Definition 175 The category hieSpec of hierarchic specifications consists of:

- **Objects:** All hierarchic specifications,
- **Morphisms:** A hierarchic specification morphism $(\sigma, \phi) : S_1 \to S_2$ from the Σ_1-hierarchic specification S_1 to the Σ_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

  - $\hat{\sigma}(\mathsf{ax}(S_1^{\ominus\ominus})) \subseteq \mathsf{ax}(S_2^{\ominus\ominus})$ and
  - $\hat{\sigma}(\mathsf{df}(S_1^{\ominus\ominus}, h)) \subseteq \mathsf{df}(S_2^{\ominus\ominus}, \phi(h))$ for each level $h \in \mathsf{po}(S_1)$.

  The morphism condition can be equivalently written $\hat{\tau}^{\sigma}_{\phi}(S_1^{\ominus\ominus}) \Subset S_2^{\ominus\ominus}$.

Proof It is straightforward to check that hieSpec is indeed a category. Composition results from the identity $\hat{\tau}^{\sigma_2 \cdot \sigma_1}_{\phi_2 \circ \phi_1}(S) = \hat{\tau}^{\sigma_2}_{\phi_2}(\hat{\tau}^{\sigma_1}_{\phi_1}(S))$ for any Σ_1-specification S with appropriate partial order of priority. ✓ ■

We show in the following the *Presentation Lemma* for hierarchic specifications. The condition $\hat{\tau}^{\sigma}_{\phi}(S_1^{\ominus\ominus}) \Subset S_2^{\ominus\ominus}$ can be replaced by $\hat{\tau}^{\sigma}_{\phi}(S_1) \Subset S_2^{\ominus\ominus}$. Therefore to check the morphism condition we have only to check whether the translations of the axioms from S_1 are semantically entailed by the axioms of S_2 and whether the translations of the defaults from each level h_1 in S_1 are implicit defaults from S_2 in the level $\phi(h_1)$.

Lemma 176 Let S_1 be a Σ_1-hierarchic specification, S_2 a Σ_2-hierarchic specification $\sigma : \Sigma_1 \rightarrow \Sigma_2$ a signature morphism and $\phi : \mathsf{po}(S_1) \rightarrow \mathsf{po}(S_2)$ a *strict* partial order morphism. Then the $\mathsf{hieSpec}$-morphism condition $\hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus}) \Subset S_2^{\ominus\ominus}$ is equivalent to $\hat{\tau}^\sigma_\phi(S_1) \Subset S_2^{\ominus\ominus}$.

Proof

- The if part is trivial: since $S_1 \Subset S_1^{\ominus\ominus}$ then $\hat{\tau}^\sigma_\phi(S_1) \Subset \hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus}) \Subset S_2^{\ominus\ominus}$.
- The other direction is similar to the corresponding proof for specifications (lemma 145). We see below that $\hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus})^{\ominus\ominus} = \hat{\tau}^\sigma_\phi(S_1)^{\ominus\ominus}$. This equality implies $\hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus}) \Subset \hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus})^{\ominus\ominus} = \hat{\tau}^\sigma_\phi(S_1)^{\ominus\ominus} \Subset S_2^{\ominus\ominus}$, since $\hat{\tau}^\sigma_\phi(S_1) \Subset S_2^{\ominus\ominus}$.
  - To see that $\hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus})^{\ominus\ominus} = \hat{\tau}^\sigma_\phi(S_1)^{\ominus\ominus}$ we show that if two hierarchic specifications have the same semantics so have their translations: if $S^\ominus = S'^\ominus$ then $\hat{\tau}^\sigma_\phi(S)^\ominus = \hat{\tau}^\sigma_\phi(S')^\ominus$. Since $S_1^{\ominus\ominus}$ and S_1 have the same semantics, we conclude that $\hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus})$ has the same semantics, and therefore the same theory as $\hat{\tau}^\sigma_\phi(S_1)$. I.e. $\hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus})^{\ominus\ominus} = \hat{\tau}^\sigma_\phi(S_1)^{\ominus\ominus}$ as wanted.
  - We only have to show that if $S^\ominus = S'^\ominus$ then $\hat{\tau}^\sigma_\phi(S)^\ominus = \hat{\tau}^\sigma_\phi(S')^\ominus$. Recall from definition 174 that $\hat{\tau}^\sigma_\phi(S) = \phi(\hat{\sigma}(S))$. We see that both operations preserve the semantics: If $S^\ominus = S'^\ominus$ then $\hat{\sigma}(S)^\ominus = \hat{\sigma}(S')^\ominus$ and also $\phi(\hat{\sigma}(S))^\ominus = \phi(\hat{\sigma}(S'))^\ominus$.
    * It is straightforward to see that $\hat{\sigma}(S)^\ominus = \hat{\sigma}(S')^\ominus$ if $S^\ominus = S'^\ominus$. The proof is by induction on the partial order of priority of S (or S') and formally similar to the proof of the presentation lemma 145 for specifications.
    * To show that $\phi(S_1)^\ominus = \phi(S_2)^\ominus$ if $S_1^\ominus = S_2^\ominus$ we firstly note that $|\phi(S_1)^\ominus| = |S_1^\ominus| = |S_2^\ominus| = |\phi(S_2)^\ominus|$. It is not difficult to show that $\phi(S_1)^\ominus$ has in the level h_2 the intersection of the preferences from $S_1^\ominus$ from the levels $h_1 \in \phi^{-1}(h_2)$. Recalling that $S_1^\ominus = S_2^\ominus$ this coincides with the intersection of the preferences from $S_2^\ominus$ from the same levels $h_1 \in \phi^{-1}(h_2)$. This intersection is the preference assigned by $\phi(S_2)^\ominus$ to the level h_2. We conclude that $\phi(S_2)^\ominus$ coincides with $\phi(S_1)^\ominus$ in each level h_2. ✓

Finally we see that isomorphic hierarchic specifications are obtained by renaming both the signature symbols and the priority levels. (We state only the non-trivial implication).

Lemma 177 Let $\sigma : \mathsf{sg}(S_1) \rightarrow \mathsf{sg}(S_2)$ be a signature isomorphism and $\phi : \mathsf{po}(S_1) \rightarrow \mathsf{po}(S_2)$ a partial order isomorphism. Then $(\sigma, \phi) : S_1 \rightarrow S_2$ is a $\mathsf{hieSpec}$-isomorphism iff $\hat{\tau}^\sigma_\phi(S_1^{\ominus\ominus}) = S_2^{\ominus\ominus}$.

Proof Let $\bar{\sigma} : \mathsf{sg}(S_2) \to \mathsf{sg}(S_1)$ be the inverse isomorphism of $\sigma : \mathsf{sg}(S_1) \to \mathsf{sg}(S_2)$ and $\bar{\phi} : \mathsf{po}(S_1) \to \mathsf{po}(S_2)$ be the inverse isomorphism of $\phi : \mathsf{po}(S_2) \to \mathsf{po}(S_1)$. Recall (proof of lemma 146) that $\hat{\sigma}$ and $\hat{\bar{\sigma}}$ are inverse functions (on formulas). Also ϕ and $\bar{\phi}$ are inverse functions (on priority levels). This implies that $\hat{\tau}^{\sigma}_{\phi}$ and $\hat{\tau}^{\bar{\sigma}}_{\bar{\phi}}$ are also inverse functions (on hierarchic specifications). Using this fact the rest of the proof is trivial (formally like the proof of lemma 146). ✓

3.2.3 Category of Hierarchies of Pre-orders

The category of hierarchies of pre-orders hiePre is introduced in the following. Also the category $\mathsf{hiePref}$ whose objects are the differential semantics of some hierarchic specification. In the same way as with specifications the category $\mathsf{hiePref}$ mirrors the category $\mathsf{hieSpec}$ of hierarchic specifications. Its morphisms are the semantical counterpart of the hierarchic specification morphisms and correspond to the inclusion of the classes of interpretation structures and of the (differential) relations in each priority level. The category hiePre (resp. $\mathsf{hiePref}$) extends the constructions of $\mathsf{PreOrder}$ (resp. Pref) with the additional priority structure[8].

The first concept to introduce is the translation of a hierarchy of pre-orders to another signature and to another priority structure. Such translation is induced by corresponding signature and partial order morphisms and goes in the opposite direction.

As with specifications the interpretation structures are mapped to their reducts. Relation are mapped to their reduct relations. The priority structure is deal with in the following way: each level h is assigned the reduct of the pre-order at $\phi(h)$.

Definition 178 Let $\mathcal{H}_2$ be a Σ_2-hierarchy of pre-orders and $(H_2, \preceq_2) = \mathsf{po}(\mathcal{H}_2)$ its partial order of priority. Let $\phi : (H_1, \preceq_1) \to (H_2, \preceq_2)$ be a partial order morphism and $\sigma : \Sigma_1 \to \Sigma_2$ a signature morphism. Recall that $\check{\sigma} : \mathsf{Mod}(\Sigma_2) \to \mathsf{Mod}(\Sigma_1)$ assigns to each Σ_2-interpretation structure a Σ_1-interpretation structure (its reduct). In definition 147 $\check{\sigma}$ is extended to classes of interpretation structures and to pre-orders.

The *reduct hierarchy of pre-orders* $\check{\tau}^{\sigma}_{\phi}(\mathcal{H}_2)$ of the hierarchy of pre-orders $\mathcal{H}_2$ with respect to both ϕ and σ is the Σ_1-hierarchy of pre-orders with:

- $|\check{\tau}^{\sigma}_{\phi}(\mathcal{H}_2)| = \check{\sigma}(|\mathcal{H}_2|)$, the class of reducts of the interpretation structures from $\mathcal{H}_2$ as class of interpretation structures,
- the partial order $(H_1, \preceq_1)$ domain of ϕ as partial order of priority and

[8]As with specifications and hierarchic specifications $\mathsf{PreOrder}$ is (isomorphic to) the full subcategory of $\mathsf{hiePref}$ of the hierarchies of pre-orders with one only priority level. The same holds for Pref and $\mathsf{hiePref}$.

- for each $h_1 \in H_1$ the pre-order $\mathsf{rl}(\check{\tau}^{\sigma}_{\phi}(\mathcal{H}_2), h_1) = \check{\sigma}(\mathsf{rl}(\mathcal{H}_2, \phi(h_1)))$, the reduct of the pre-order from $\mathcal{H}_2$ at the level $\phi(h_1)$. ■

The definition of the category hiePre of hierarchies of pre-orders follows.

Definition 179 The category hiePre of Σ-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 Σ_2-hierarchy of pre-orders $\mathcal{H}_2$ to the Σ_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

  - $\check{\sigma}(|\mathcal{H}_2|) \subseteq |\mathcal{H}_1|$, i.e. the reducts of the interpretation structures participating in $\mathcal{H}_2$ are interpretation structures participating in $\mathcal{H}_1$ and
  - $\check{\sigma}(\mathsf{rl}(\mathcal{H}_2, \phi(h_1))) \subseteq \mathsf{rl}(\mathcal{H}_1, h_1)$ 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 morphism condition can equivalently be written $\check{\tau}^{\sigma}_{\phi}(\mathcal{H}_2) \Subset \mathcal{H}_1$.

Proof Composition results from the identity $\check{\tau}^{\sigma_2 \cdot \sigma_1}_{\phi_2 \circ \phi_1}(\mathcal{H}_3) = \check{\tau}^{\sigma_1}_{\phi_1}(\check{\tau}^{\sigma_2}_{\phi_2}(\mathcal{H}_3))$ for any hierarchy of pre-orders $\mathcal{H}_3$ in the domain of $\check{\tau}^{\sigma_2 \cdot \sigma_1}_{\phi_2 \circ \phi_1}$. ✓ ■

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

Definition 180 The category hiePref is the full subcategory of hiePre with objects the Σ-hierarchies of pre-orders $\mathcal{H}$ such that $\mathcal{H} = \mathcal{H}^{\ominus\ominus}$. ■

We see in the following that isomorphic hierarchies of differential preferences are "pointwise equivalent" in the sense that the differential preferences at the same level h are equivalent ($\approx$, recall definition 150).

Definition 181 Let $\mathcal{H}$ and $\mathcal{H}'$ be Σ-hierarchies of pre-orders with the same partial order of priority $(H, \preceq) = \mathsf{po}(\mathcal{H}) = \mathsf{po}(\mathcal{H}')$. The hierarchies of pre-orders $\mathcal{H}$ and $\mathcal{H}'$ are said *equivalent*, written $\mathcal{H} \approx \mathcal{H}'$ iff

- $|\mathcal{H}| \approx |\mathcal{H}'|$ and
- $\sqsubseteq_h \approx \sqsubseteq'_h$ for each $h \in H$ where $\sqsubseteq_h$ and $\sqsubseteq'_h$ are the pre-orders $\mathsf{rl}(\mathcal{H}, h)$ and $\mathsf{rl}(\mathcal{H}', h)$ assigned by $\mathcal{H}$ and $\mathcal{H}'$ to the level h. ▪

The intended characterization of isomorphism in hiePref follows.

Lemma 182 Let $\sigma : \mathsf{sg}(S_1^\ominus) \to \mathsf{sg}(S_2^\ominus)$ be a signature isomorphism and let $\phi : \mathsf{po}(S_1^\ominus) \to \mathsf{po}(S_2^\ominus)$ be a strict partial order isomorphism. Then $\overleftarrow{(\sigma, \phi)} : S_2^\ominus \to S_1^\ominus$ is a hiePref-isomorphism iff $\check{\tau}_\phi^\sigma(S_2^\ominus) \approx S_1^\ominus$.

Proof By induction on the partial order of priority $\mathsf{po}(S_1^\ominus) = \mathsf{po}(\check{\tau}_\phi^\sigma(S_2^\ominus))$ and formally similar to the proof of the corresponding property for preference relations (in lemma 151). ✓

3.2.4 Syntax and Semantics

The categories hieSpec of hierarchic specifications and hiePref of their hierarchies of differential preferences formalize the syntactical and semantical expression of the same constructions. We see that syntactical morphisms and (co)constructions can be translated to semantical morphisms and constructions. The reverse is also true. This translation is given by the functors $\mathsf{hieSem} : \mathsf{hieSpec} \to \mathsf{hiePref}^{\mathsf{op}}$ and $\mathsf{hieSyn} : \mathsf{hiePref}^{\mathsf{op}} \to \mathsf{hieSpec}$.

Lemma 183 There is a hieSpec morphism $(\sigma, \phi) : S_1 \to S_2$ iff there is a hiePre (or equivalently a hiePref) morphism $\overleftarrow{(\sigma, \phi)} : S_2^\ominus \to S_1^\ominus$.

Proof We must show that $\hat{\tau}_\phi^\sigma(S_1^{\ominus\ominus}) \Subset S_2^{\ominus\ominus}$ iff $\check{\tau}_\phi^\sigma(S_2^\ominus) \Subset S_1^\ominus$. From the presentation lemma in 176 we can equivalently show that $\hat{\tau}_\phi^\sigma(S_1) \Subset S_2^{\ominus\ominus}$ iff $\check{\tau}_\phi^\sigma(S_2^\ominus) \Subset S_1^\ominus$.

The condition on the axiom-interpretation structure part is the classical one and has been shown in lemma 152 (for specifications). Therefore $\mathsf{ax}(\hat{\tau}_\phi^\sigma(S_1^{\ominus\ominus})) \subseteq \mathsf{ax}(S_2^{\ominus\ominus})$ iff $\mathsf{ax}(\hat{\tau}_\phi^\sigma(S_1)) \subseteq \mathsf{ax}(S_2^{\ominus\ominus})$ iff $|\check{\tau}_\phi^\sigma(S_2^\ominus)| \subseteq |S_1^\ominus|$.

Let $(H_1, \preceq_1) = \mathsf{po}(S_1) = \mathsf{po}(S_1^{\ominus\ominus})$ be the partial order of priority of S_1 and $S_1^{\ominus\ominus}$ and $(H_2, \preceq_2) = \mathsf{po}(S_2) = \mathsf{po}(S_2^{\ominus\ominus})$ be the partial order of priority of S_2 and $S_2^{\ominus\ominus}$.

- Assume that $\hat{\tau}_\phi^\sigma(S_1^{\ominus\ominus}) \Subset S_2^{\ominus\ominus}$. This is equivalent to $\hat{\tau}_\phi^\sigma(S_1) \Subset S_2^{\ominus\ominus}$. Recall that the differential preference $\sqsubseteq_{S_1,h_1}^\ominus$ of S_1 at level $h_1 \in H_1$ is the intersection of the equivalence $\equiv^{\ominus}_{S_1,h_1^-}$ of the levels under h_1 with the preference $\sqsubseteq^{\odot}_{S_1,h_1}$ induced by the defaults in h_1. Let $\sqsubseteq^{\ominus}_{S_2,\phi(h_1)}$ be the differential semantics from S_2 at level $\phi(h_1)$. We must show $\sqsubseteq^{\ominus}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$.

We see firstly that $\sqsubseteq^{\odot}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$. This is proved as follows: $\sqsubseteq^{\odot}_{S_1,h_1}$ is the preference of $s_{h_1} = (\mathsf{ax}(S_1), \mathsf{df}(S_1, h_1))$. The defaults from the theory of S_2 at $\phi(h_1)$ are the defaults implicit in $\sqsubseteq^{\ominus}_{S_2,\phi(h_1)}$. Therefore $\sqsubseteq^{\ominus}_{S_2,\phi(h_1)}$ is contained (Galois connection 30) in the preference $\sqsubseteq_{s'_{\phi(h_1)}}$ of $s'_{\phi(h_1)} = (\mathsf{ax}(S_2^{\ominus\ominus}), \mathsf{df}(S_2^{\ominus\ominus}, \phi(h_1)))$. From $\hat{\tau}^{\sigma}_{\phi}(S_1) \Subset S_2^{\ominus\ominus}$ we have $\hat{\sigma}(s_{h_1}) \Subset s'_{\phi(h_1)}$ and from lemma 152 this implies $s_{h_1} \Supset \check{\sigma}(s'_{\phi(h_1)})$. On the relation part this is $\sqsubseteq^{\odot}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq_{s'_{\phi(h_1)}})$. Finally $\sqsubseteq^{\odot}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq_{s'_{\phi(h_1)}}) \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$.

We now proceed by induction in the priority structure of S_1.

- Assume h_1 is minimal. Then $\sqsubseteq^{\ominus}_{S_1,h_1}$ coincides with $\sqsubseteq^{\odot}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$.
- Assume now that h_1 is not minimal. Then $\sqsubseteq^{\ominus}_{S_1,h_1}$ is the intersection of $\equiv^{\ominus}_{S_1,h_1^-}$ and $\sqsubseteq^{\odot}_{S_1,h_1}$. We already know $\sqsubseteq^{\odot}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$ so we have to check that $\equiv^{\ominus}_{S_1,h_1^-} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$. This is straightforward from the induction hypothesis for $h_1' \prec_1 h_1$. Firstly note that $\phi(h_1') \prec_2 \phi(h_1)$ and therefore $\phi(h_1^-) \subseteq \phi(h_1)^-$ (the images of the levels under h_1 are levels under $\phi(h_1)$). Since $\sqsubseteq^{\ominus}_{S_1,h_1'} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1')})$ we obtain $\equiv^{\ominus}_{S_1,h_1^-} \supseteq \check{\sigma}(\equiv^{\ominus}_{S_2,\phi(h_1^-)}) \supseteq \check{\sigma}(\equiv^{\ominus}_{S_2,\phi(h_1)^-}) \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$.

* Assume that $\check{\tau}^{\sigma}_{\phi}(S_2^{\ominus}) \Subset S_1^{\ominus}$. This means that for each level $h_1 \in H_1$ the inclusion $\sqsubseteq^{\ominus}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$ holds. From definition 105 the defaults $\mathsf{df}(S_1^{\ominus}, h_1)$ are the defaults implicit in the pre-order $R^1_{h_1} = (|S_1^{\ominus}|, \sqsubseteq^{\ominus}_{S_1,h_1})$. The defaults $\mathsf{df}(S_2^{\ominus}, \phi(h_1))$ are the defaults implicit in $R^2_{\phi(h_1)} = (|S_2^{\ominus}|, \sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$. We now see that given a default d_1 implicit in $R^1_{h_1}$ then $\hat{\sigma}(d_1)$ is a default implicit in $R^2_{\phi(h_1)}$. Assume it is not. Then there are interpretation structures $m_2 \sqsubseteq^{\ominus}_{S_2,\phi(h_1)} n_2$ with $m_2 \vDash \hat{\sigma}(d_1)$ and $n_2 \nvDash \hat{\sigma}(d_1)$. From the satisfaction condition the reduct $\check{\sigma}(m_2)$ of m_2 satisfies d_1 and the reduct $\check{\sigma}(n_2)$ of n_2 does not. From $\sqsubseteq^{\ominus}_{S_1,h_1} \supseteq \check{\sigma}(\sqsubseteq^{\ominus}_{S_2,\phi(h_1)})$ we have $\check{\sigma}(m_2) \sqsubseteq^{\ominus}_{S_1,h_1} \check{\sigma}(n_2)$ contradicting the hypothesis of d_1 being an implicit default in $R^1_{h_1}$.

  We have seen that $\mathsf{df}(S_1^{\ominus}, h_1) \subseteq \hat{\sigma}(\mathsf{df}(S_2^{\ominus}, \phi(h_1)))$ and therefore $\hat{\tau}^{\sigma}_{\phi}(S_1^{\ominus\ominus}) \Subset S_2^{\ominus\ominus}$.

✓

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 184

- 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 $(\sigma, \phi) : S_1^{\ominus\ominus} \to {S'}_2^{\ominus\ominus}$.

Proof That hieSem and hieSyn are functors is a trivial consequence of lemma 183 above. ✓ ■

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

Proof Preservation of constructions follows trivially from lemma 183 above. The proof is formally identical with the proof of theorem 155. ✓

3.2.5 Isomorphism and Extensions

Isomorphic hierarchic specifications have theories that are related by a renaming of both the signature and priority symbols. In this way isomorphic specifications have, up to such a renaming, the same meaning. This implies, in particular, that their credulous and skeptical consequences are related by such a renaming (of signature symbols).

Theorem 186 Let $(\sigma, \phi) : S_1 \to S_2$ be an isomorphism in the category $\mathsf{hieSpec}$. Then $\mathcal{E}_1$ is an extension of S_1 iff $\hat{\sigma}(\mathcal{E}_1)$ is an extension of S_2.

Proof Omitted. We note only that $\overleftarrow{(\sigma, \phi)} : S_2^{\ominus} \to S_1^{\ominus}$ is a $\mathsf{hiePref}$-isomorphism and therefore $\check{\tau}_{\phi}^{\sigma}(S_2^{\ominus}) \approx S_1^{\ominus}$. This implies that $S_2^{\ominus}$ and $S_1^{\ominus}$ have isomorphic (in $\mathsf{PreOrder}$) lexicographic combinations: $\check{\sigma}(\mathsf{lex}^{\Box}(S_2^{\ominus})) \approx \mathsf{lex}^{\Box}(S_1^{\ominus})$. These lexicographic combinations are the lexicographic preferences of the original specifications. ✓

3.2.6 Existence of Constructions

The category StPart of partial orders is not cocomplete. Therefore, hieSpec does not have all colimits and hiePref does not have all limits. However, when the underlying category 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 hieSpec provided that the corresponding combination of partial orders exists. The same is true for the limits in hiePref.

The following definitions of the forgetful functors sending a hierarchic specification or a hierarchy of pre-orders to their partial order of priority follows are of convenience.

Definition 187

- The forgetful functor $\mathcal{P}$: hieSpec $\rightarrow$ StPart sends each hierarchic specification S to its partial order of priority po(S) and each hieSpec morphism $(\sigma, \phi) : S \rightarrow S'$ to the strict partial order morphism ϕ : po(S) $\rightarrow$ po(S').
- The forgetful functor $\mathcal{Q}$: hiePre $\rightarrow$ StPart sends each hierarchy of pre-orders $\mathcal{H}$ to its partial order of priority po($\mathcal{H}$) and each hiePre morphism $\overleftarrow{(\sigma, \phi)} : \mathcal{H}' \rightarrow \mathcal{H}$ to the strict partial order morphism ϕ : po($\mathcal{H}$) $\rightarrow$ po($\mathcal{H}'$). ∎

Existence of colimits in hieSpec is presented in the following theorem.

Theorem 188 Let $\mathcal{D}$ be a diagram in hieSpec. Then $\mathcal{D}$ has a colimit in hieSpec if the category Sign of signatures is cocomplete and the diagram $\mathcal{P}(\mathcal{D})$ has a colimit in StPart (i.e. the composition of the corresponding partial orders of priority is defined).

Proof Similar to the proof of theorem 159. The colimit of a (small) diagram $\mathcal{D} : \mathcal{S} \rightarrow$ hieSpec is obtained by lifting two colimits: the colimit in Sign of the corresponding diagram of signatures $\mathcal{F} \circ \mathcal{D} : \mathcal{S} \rightarrow$ Sign where $\mathcal{F}$ is the forgetful functor sending Spec to the underlying category of signatures and the colimit of the corresponding diagram of partial orders $\mathcal{P} \circ \mathcal{D} : \mathcal{S} \rightarrow$ StPart.

Let S_i be the specification assigned by $\mathcal{D}$ to the node $i \in |\mathcal{S}|$. Let Σ_i be its signature and R_i its partial order. Then the diagram $\mathcal{F} \circ \mathcal{D}$ assigns to the node $i \in |\mathcal{S}|$ the signature Σ_i. And the diagram $\mathcal{P} \circ \mathcal{D}$ assigns R_i to the node $i \in |\mathcal{S}|$.

Let $\mathcal{C}_{\mathsf{Sign}} = \{\sigma_i : \Sigma_i \rightarrow \Sigma^{\oplus}, i \text{ in } |\mathcal{S}|\}$ be the cocone colimit of $\mathcal{F} \circ \mathcal{D}$ (in Sign) and let $\mathcal{C}_{\mathsf{StPart}} = \{\phi_i : R_i \rightarrow R^{\oplus}, i \text{ in } |\mathcal{S}|\}$ be the cocone colimit of $\mathcal{P} \circ \mathcal{D}$ (in StPart).

We have to provide the hierarchic specification $S^{\oplus}$ of the colimit signature $\Sigma^{\oplus}$ and with partial order of priority $R^{\oplus}$ such that $\mathcal{C}_{\mathsf{hieSpec}} = \{(\sigma_i, \phi_i) : S_i \to S^{\oplus}, i \text{ in } |\mathcal{S}|\}$ is a colimit in hieSpec.

Choose (as for specifications) for $\mathsf{ax}(S^{\oplus})$ the union $\bigcup_{i \in |\mathcal{S}|} \hat{\sigma}_i(\mathsf{ax}(S_i))$ of the σ_i-translations of the axioms of the parameter specifications. For defaults consider firstly a priority level $h^{\oplus}$ from $R^{\oplus}$. This level represents all those levels from each hierarchic specification S_i having $h^{\oplus}$ as image under the corresponding ϕ_i. Therefore the defaults assigned to $h^{\oplus}$ are $\mathsf{df}(S^{\oplus}, h^{\oplus}) = \bigcup_{i \in |\mathcal{S}|} \hat{\sigma}_i \left(\cup_{h_i \in \phi_i^{-1}(h^{\oplus})} \mathsf{df}(S_i, h_i) \right)$.

An equivalent (but shorter) definition of $S^{\oplus}$ is $S^{\oplus} = \Cup_{i \in |\mathcal{S}|} \hat{\tau}^{\sigma_i}_{\phi_i}(S_i)$. In this way $S^{\oplus}$ is the smallest (w.r.t. $\Subset$) hierarchic specification such that each $(\sigma_i, \phi_i) : S_i \to S^{\oplus}$ is a hieSpec-morphism.

Moreover $\mathcal{C}_{\mathsf{hieSpec}}$ is a commutative cocone (for $\mathcal{D}$) since $\mathcal{C}_{\mathsf{Sign}}$ is a commutative cocone for $\mathcal{F} \circ \mathcal{D}$ and $\mathcal{C}_{\mathsf{StPart}}$ is a commutative cocone for $\mathcal{P} \circ \mathcal{D}$.

The property of colimit for $\mathcal{C}_{\mathsf{hieSpec}}$ follows from the corresponding properties for $\mathcal{C}_{\mathsf{Sign}}$ and $\mathcal{C}_{\mathsf{StPart}}$. We have to check that, given another commutative cocone $\mathcal{C}'_{\mathsf{hieSpec}} = \{(\nu_i, \psi_i) : S_i \to S', i \text{ in } |\mathcal{S}|\}$ for $\mathcal{D}$ there is a unique morphism $(\mu, \chi) : S^{\oplus} \to S'$ such that $(\nu_i, \psi_i) = (\mu, \chi) \circledcirc (\sigma_i, \phi_i)$ for all i in $|\mathcal{S}|$ (composition $\circledcirc$ in hieSpec is pairwise composition).

From the commutative cocone $\mathcal{C}'_{\mathsf{Spec}}$ we obtain the commutative cocones $\mathcal{C}'_{\mathsf{Sign}} = \{\nu_i : \Sigma_i \to \mathsf{sg}(S'), i \text{ in } |\mathcal{S}|\}$ for $\mathcal{F} \circ \mathcal{D}$ and $\mathcal{C}'_{\mathsf{StPart}} = \{\psi_i : R_i \to \mathsf{po}(S'), i \text{ in } |\mathcal{S}|\}$ for $\mathcal{P} \circ \mathcal{D}$.

Since Sign is cocomplete there exists a unique signature morphism $\mu : \Sigma^{\oplus} \to \Sigma'$ such that $\nu_i = \mu \circ \sigma_i$, for all i in $|\mathcal{S}|$.

Since we assume that there is a colimit in StPart of $\mathcal{P} \circ \mathcal{D}$ then there exists a unique partial order morphism $\chi : R^{\oplus} \to R'$ such that $\psi_i = \chi \cdot \phi_i$, for all i in $|\mathcal{S}|$.

We only have to show that (μ, χ) is also a hieSpec morphism, i.e. that $\hat{\tau}^{\mu}_{\chi}(S^{\oplus}) \Subset S'^{\ominus\ominus}$. Since $\nu_i = \mu \circ \sigma_i$ and $\psi_i = \chi \cdot \phi_i$ we have $\hat{\tau}^{\nu_i}_{\psi_i}(S_i) = \hat{\tau}^{\mu}_{\chi}(\hat{\tau}^{\sigma_i}_{\phi_i}(S_i))$. The morphism condition for each S_i is $\hat{\tau}^{\nu_i}_{\psi_i}(S_i) = \hat{\tau}^{\mu}_{\chi}(\hat{\tau}^{\sigma_i}_{\phi_i}(S_i)) \Subset S'^{\ominus\ominus}$. This clearly implies $\hat{\tau}^{\mu}_{\chi}(S^{\oplus}) \Subset S'^{\ominus\ominus}$ since $S^{\oplus} = \Cup_{i \in |\mathcal{S}|} \hat{\tau}^{\sigma_i}_{\phi_i}(S_i)$. ✓

Existence of limits in hiePre and hiePref is assured by theorems 185 and 188. This property is presented formally in the next theorem. Its (extra) proof displays the construction of limits in hiePre and hiePref.

Theorem 189 Let $\mathcal{D}$ be a diagram in hiePre or hiePref. Then $\mathcal{D}$ has a limit in hiePre (resp. hiePref) if the category Sign of signatures is cocomplete and the diagram $\mathcal{Q}(\mathcal{D})$ has a colimit in StPart (i.e. the composition of the corresponding partial orders of priority is defined).

Proof Similar to the proof of theorem 160. The limit of a (small) diagram $\mathcal{D} : \mathcal{S} \to$ hiePre is obtained by lifting two colimits: that of the diagram $\mathcal{G} \circ \mathcal{D} : \mathcal{S} \to$ Sign where

$\mathcal{G}$ is the forgetful functor sending hiePre to the underlying category of signatures and the colimit of the corresponding diagram of partial orders $\mathcal{Q} \circ \mathcal{D} : \mathcal{S} \to$ StPart.

Let $\mathcal{H}_i$ be the hierarchy of pre-orders assigned by $\mathcal{D}$ to the node $i \in |\mathcal{S}|$. Let Σ_i be its signature and R_i its partial order of priority. In this way Σ_i is the signature assigned by the diagram $\mathcal{G} \circ \mathcal{D}$ to the node $i \in |\mathcal{S}|$ and R_i is the partial order assigned by $\mathcal{Q} \circ \mathcal{D} : \mathcal{S} \to$ StPart to i.

Let $\mathcal{C}_{\mathsf{Sign}} = \{\sigma_i : \Sigma_i \to \Sigma^{\oplus}, i \text{ in } |\mathcal{S}|\}$ be the cocone colimit of $\mathcal{G} \circ \mathcal{D}$ (in Sign) and let $\mathcal{C}_{\mathsf{StPart}} = \{\phi_i : R_i \to R^{\oplus}, i \text{ in } |\mathcal{S}|\}$ be the cocone colimit of $\mathcal{Q} \circ \mathcal{D}$ (in StPart).

We have to provide the hierarchy of pre-orders $\mathcal{H}^{\otimes}$ among the interpretation structures of the colimit signature $\Sigma^{\oplus}$ and with partial order of priority $R^{\oplus}$ such that $\mathcal{L}_{\mathsf{hiePre}} = \{ \overleftarrow{(\sigma_i, \phi_i)} : \mathcal{H}^{\otimes} \to \mathcal{H}_i, i \in |\mathcal{S}|\}$ is a limit in hiePre.

The interpretation structures participating in $\mathcal{H}^{\otimes}$ are the $\Sigma^{\oplus}$-interpretation structures whose reducts w.r.t. each σ_i participate in $\mathcal{H}_i$. That is $|\mathcal{H}^{\otimes}| = \{m \in |\mathsf{Mod}(\Sigma^{\oplus})| : \breve{\sigma}_i(m) \in |\mathcal{H}_i|, \text{ for all } i \in |\mathcal{S}|\}$. Let now $h^{\oplus}$ be a level from $R^{\oplus}$. Two interpretation structures are related by $\mathcal{H}^{\otimes}$ in level $h^{\oplus}$ iff their reducts w.r.t. σ_i are related in the pre-orders from $\mathcal{H}_i$ at levels h_i such that $\phi_i(h_i) = h^{\oplus}$.

That is $m \sqsubseteq_{\otimes}^{h^{\oplus}} n$ iff $\breve{\sigma}(m) \sqsubseteq_i^{h_i} \breve{\sigma}(n)$ for all h_i such that $\phi_i(h_i) = h^{\oplus}$ and for all $i \in |\mathcal{S}|$ ($\sqsubseteq_{\otimes}^{h^{\oplus}}$ is $\mathsf{rl}(\mathcal{H}^{\otimes}, h^{\oplus})$ and $\sqsubseteq_i$ is $\mathsf{rl}(\mathcal{H}_i, h_i)$).

It is trivial to check that each $\sqsubseteq_{\otimes}^{h^{\oplus}}$ is a pre-order. The hierarchy of pre-orders $\mathcal{H}^{\otimes}$ is also the biggest $\Sigma^{\oplus}$-hierarchy of pre-orders such that each $\overleftarrow{(\sigma_i, \phi_i)} : \mathcal{H}^{\otimes} \to \mathcal{H}_i$ is a hiePre morphism. The property of limit for $\mathcal{L}_{\mathsf{PreOrder}}$ follows from the corresponding property of colimit for $\mathcal{C}_{\mathsf{Sign}}$ and $\mathcal{C}_{\mathsf{StPart}}$. We omit this proof (it is similar to the proof of the corresponding theorem 160 for specifications).

Limits in hiePref are calculated in the same way. We have only to check that $\mathcal{H}^{\otimes}$ is the hierarchy of differential preferences of some hierarchic specification $S^{\oplus}$ whenever each $\mathcal{H}_i$ is the hierarchy of differential preferences of some S_i. This holds since limits in hiePref are the image by hieSem of colimits in hieSpec (theorem 185). ✓

We end this section with two examples of combination of hierarchic specifications. Firstly we see that the union of hierarchic specifications (with same partial order of priority) is expressed by a canonical construction (a pushout).

Example 190 Let $R_{\emptyset}$ denote the Σ-hierarchic specification with R as priority structure, an empty set of axioms and an empty set of defaults in each level $h \in |R|$. Let S_1 and S_2 be Σ-hierarchic specifications both with R as partial order of priority. Let $(i_{\Sigma}, i_R) : R_{\emptyset} \to S_1$ and $(i_{\Sigma}, i_R) : R_{\emptyset} \to S_2$ denote the hieSpec morphisms consisting of the identity signature morphism from Σ to Σ and the identity partial order morphism from R to R. Then the (vertex of the) pushout of the diagram

$$S_1 \overset{(i_{\Sigma}, i_R)}{\longleftarrow} R_{\emptyset} \overset{(i_{\Sigma}, i_R)}{\longrightarrow} S_2$$

is the specification $S_1 \Cup S_2$ having as axioms the union of the axioms from S_1 and S_2 and in each level h the union of from S_1 and S_2 at that level. This is illustrated by the following figure.

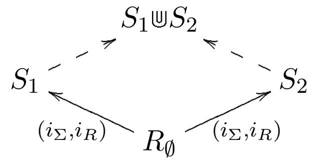

Figure 3.9: Union as a Colimit

Note finally that, from theorem 185, the previous diagram can be redrawn at the semantic level by changing the specifications to their hierarchies of differential preferences and reverting the morphisms. The hierarchy of differential preferences associated with $S_1 \Cup S_2$ will be the (vertex of the) limit of the new diagram. It is $S_1^{\ominus} \Cap S_2^{\ominus}$, the pointwise intersection of the hierarchies of differential preferences associated with S_1 and S_2. △

We have already seen in examples 173 and 172 two types of combinations of partial orders. In the next example we illustrate how a new priority level can be added in between existing priority levels. (Since the combination of signatures has been illustrated for specifications we concentrate in the combination of priority levels).

Example 191 Recall the hierarchic specification MAMMALS from example 75 and consider now that we have constructed only the portion of this specification concerned with mammals, bats and Batman (see figure 3.10). The diagram stat-

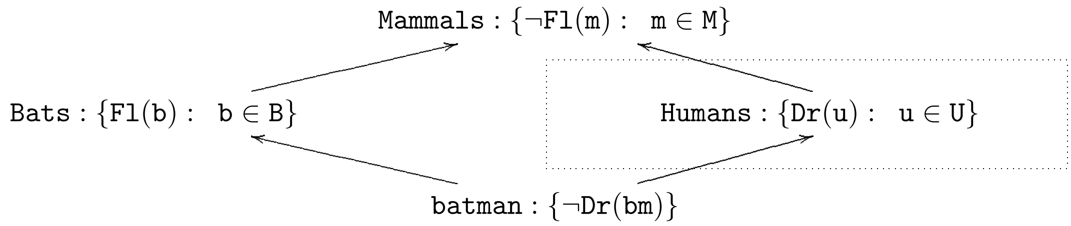

Figure 3.10: MAMMALS

ing how to construct the original specification, i.e. stating that the level `Humans` and its defaults should be added in between `batman` and `Mammals` is displayed in the following figure 3.11.

The colimit of this diagram is the specification MAMMALS already displayed. Note that the hierarchic specifications within boxes state that the level μ is

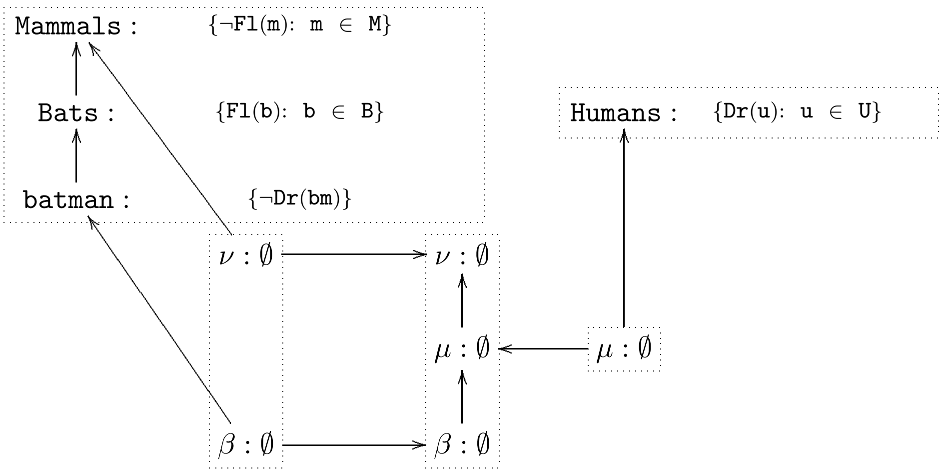

Figure 3.11: Diagram of MAMMALS

between ν and β. And μ should be identified with `Humans` and β and ν with `batman` and `Mammals` respectively. △

3.2.7 Canonicity of the Semantics

In this section we see that the semantics of hierarchic specifications is minimal among other alternative semantics that can express composition of such specifications. We mean now the addition of syntactical entities such as axioms and defaults and also of priority levels and new relations between them.

The priority structure introduces further restrictions on the allowed semantics of hierarchic specifications. Firstly we impose that hierarchic specifications having the same semantics must have the same partial order of priority. And secondly (see below in definition 193) such semantics must interpret the addition of further relations between existing priority levels. Other concepts are similar to those introduced in section 3.1.6. Note that the operation $\uplus$ is now (definition 108) pointwise union, i.e. union of axioms and, for each level h, union of defaults at that level.

Definition 192 A *semantics* of Σ-hierarchic specifications is a function $[\![]\!]$ with the set of all Σ-hierarchic specifications as domain, such that if $[\![S_1]\!] = [\![S_2]\!]$ then $\mathsf{po}(S_1) = \mathsf{po}(S_2)$ for any two Σ-hierarchic specifications S_1 and S_2. ∎

We also demand that this semantics is well defined for operations on the priority structure. In particular we are concerned with the identification of further relations of priority between existing unrelated levels. This corresponds to the operations that, given a partial order of priority R result in a partial order of priority R' with more relations among the same levels: $R \Subset R'$ and $|R| = |R'|$. The effect of such an operation on a hierarchic specification S with R as partial order of priority is the specification S' having R' as partial order of priority.

Definition 193 Let S be a Σ-hierarchic specification with $R = \mathsf{po}(S)$ as partial order of priority. Let R' be a partial order with $R \Subset R'$ and $|R| = |R'|$. Let $i : |R| \to |R'|$ be the identity function (from $|R|$ to $|R'| = |R|$) and note that $i : R \to R'$ is a strict partial order morphism. Recall from definition 174 that $i_{R'}(S)$ is the Σ-hierarchic specification with R' as partial order of priority, the same axioms as S and, at each level h, the same defaults as S at that level (i.e. only the priority structure changes).

The semantics $[\![]\!]$ is said to *be well defined under addition of priority relations* iff $[\![i_{R'}(S_1)]\!] = [\![i_{R'}(S_2)]\!]$ whenever $[\![S_1]\!] = [\![S_2]\!]$, for every Σ-hierarchic specifications S_1 and S_2 with R as partial order of priority and any partial order R' such that $R \Subset R'$ and $|R| = |R'|$. ∎

The motivation for the definition above is that it is possible to derive the semantics of $i_{R'}(S)$ from the semantics of S. In other words the semantics of S alone has enough information to account for the addition of further priority relations among the levels present in S. A special case of such operations, that is of technical importance is the *minimization* of a level h from S. This minimization corresponds to make h as important as possible. The level h cannot be made more important than another level h' under h, since such levels have been specified as being more important than h itself. But it can be made more important than the levels previously unrelated to h. Minimization w.r.t. h is the addition of relations of priority stating that h is better (under) the levels h'' previously unrelated to h. The levels h' under h will also be better than such h''. In figure 3.12 we display a priority structure (left) and its minimization w.r.t. the level 1 (right). The level 1 remains less important than 0 but becomes more important than $2'$.

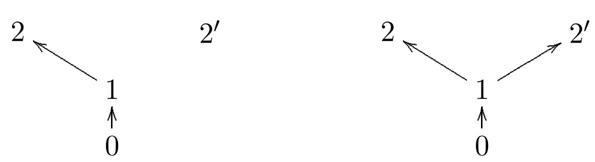

Figure 3.12: Minimization

Definition 194 Let S be a Σ-hierarchic specification with $(H, \preceq) = \mathsf{po}(S)$ as partial order of priority. Let $h \in H$ and $R' = (H', \preceq')$ be the partial order with $H = H'$ and $h_1 \preceq' h_2$ iff $h_1 \preceq h_2$ or $h_1 \preceq h$ and $h_2 \not\preceq h$. The minimization of S

w.r.t. h is the Σ-hierarchic specification $\mathsf{min}(S)(h) = i_{R'}(S)$ having R' as partial order of priority.

Proof We omit the proof that R' is indeed a partial order. ✓ ∎

Finally the semantics of hierarchic specifications satisfying the appropriate conditions presented above are, in this context, also named proper compositional semantics.

Definition 195 A semantics $[\![]\!]$ of Σ-hierarchic specifications is said a *proper compositional semantics* iff

- it is compatible either with the skeptical *or* with the credulous consequences,
- interprets $\Cup$ and
- is well defined under addition of priority relations. ∎

The hierarchy of differential preferences is a *proper compositional semantics*. In fact it supports much more operations than those referred.

The reason for choosing only these operations and not more is the following: we see that any semantics providing them must have more structure than the hierarchy of differential preferences. Any semantics providing an interesting theory of composition should provide these operations. Therefore it has more structure than the chosen semantics. This justifies the choice of the hierarchy of differential preferences as the semantics for composition.

That the hierarchy of differential preferences is the least proper compositional semantics of hierarchic specifications is presented in the following theorem.

As for specifications this property is established only for semantics of hierarchic specifications from *compact* institutions *having negation* (see definitions 65 and 118 and also remark 119).

Theorem 196 Let Σ be a signature of a compact institution $(\mathcal{I}, \mathsf{neg})$ *having negation*. The hierarchy of differential preferences is the *least* (w.r.t. $\succcurlyeq$) proper compositional semantics of the Σ-hierarchic specifications.

Proof

- Firstly we must check that the hierarchy of differential preferences is a proper compositional semantics. Clearly it is compatible with either the skeptical or credulous consequences. In fact it is compatible with both since these consequences are derived from the lexicographic preference of a specification S (see definition 134 and lemma 131). The lexicographic preference can be derived from $S^{\ominus}$: it is the lexicographic combination of $S^{\ominus}$ (see theorem 99).

  Moreover it interprets $\uplus$. See the Galois connection presented in 111. The operation interpreting $\uplus$ is the intersection $\Cap$ of the hierarchies of differential preferences.

  And it is well defined for addition of priority levels. This amounts to show that $i_{R'}(S_1)^{\ominus} = i_{R'}(S_2)^{\ominus}$ whenever $S_1^{\ominus} = S_2^{\ominus}$. We have seen in the proof of the presentation lemma 176 that $\phi(S_1)^{\ominus} = \phi(S_2)^{\ominus}$ whenever $S_1^{\ominus} = S_2^{\ominus}$ for any strict partial order morphism ϕ.

- Furthermore we must show that given a proper compositional semantics $[\![]\!]$ of Σ-hierarchic specifications and Σ-hierarchic specifications S_1 and S_2 if $[\![S_1]\!] = [\![S_2]\!]$ then $S_1^{\ominus} = S_2^{\ominus}$. This is equivalent to show the equality $S_1^{\oplus} = S_2^{\oplus}$ of the hierarchies of lexicographic preferences due to the equivalence between both semantics presented in theorem 94.

  An important preliminary result is that if $[\![S_1]\!] = [\![S_2]\!]$ then S_1 and S_2 must have the same lexicographic preference $\mathsf{lex}^{\circ}(S_1) = \mathsf{lex}^{\circ}(S_2)$. We omit this proof that is similar to the proof of theorem 169 (canonicity of the preference semantics) and uses addition ($\uplus$) of axioms and defaults.

  Using this fact the structure of the proof is as follows. We assume that $S_1^{\oplus} \neq S_2^{\oplus}$ and construct from S_1 and S_2, using $\uplus$ and *minimization*, the hierarchic specifications S_1'' and S_2''. Since $[\![S_1]\!] = [\![S_2]\!]$ then also $[\![S_1'']\!] = [\![S_2'']\!]$ which implies $\mathsf{lex}^{\circ}(S_1'') = \mathsf{lex}^{\circ}(S_2'')$. However, we see that S_1'' and S_2'' have different lexicographic preferences. This implies $S_1^{\oplus} = S_2^{\oplus}$ as intended.

  Let h be a priority level where $S_1^{\oplus}$ and $S_2^{\oplus}$ differ. This means that $\sqsubseteq^{\oplus}_{S_1,h} \neq \sqsubseteq^{\oplus}_{S_2,h}$ where $\sqsubseteq^{\oplus}_{S_1,h}$ and $\sqsubseteq^{\oplus}_{S_2,h}$ are the lexicographic preferences from $S_1^{\oplus}$ and $S_2^{\oplus}$ at that level. Therefore there are $m \sqsubseteq^{\oplus}_{S_1,h} n$ and $m \not\sqsubseteq^{\oplus}_{S_2,h} n$ (or vice versa). It is important for the rest of the proof that the relation $m \sqsubseteq^{\oplus}_{S_1,h} n$ is strict. If it is let $S_1' = S_1$ and $S_2' = S_2$. If not then we construct S_1' and S_2' as follows: since $m \not\sqsubseteq^{\oplus}_{S_2,h} n$ there is a default d_2 from a level $h' \preceq h$ in S_2 with $m \vDash d_2$ and $n \nvDash d_2$. Add this default d_2 to the level h in both hierarchic specifications thus obtaining S_1' and S_2' (using $\uplus$ with the hierarchic specification having d_2 as only default in that level and empty sets of defaults in other levels). These hierarchic specifications still differ at level h. We have now $m \not\sqsubseteq^{\oplus}_{S_2',h} n$ as before and $m \sqsubset^{\oplus}_{S_1',h} n$ (strict) as wanted. We want now that these relations are carried to the overall lexicographic preference.

  From S_1' and S_2' obtain the hierarchic specifications $S_1'' = \mathsf{min}(S_1')(h)$ and $S_2'' = \mathsf{min}(S_2')(h)$ by *minimizing* h.

We now see that the overall lexicographic preferences of S_1'' and S_2'' differ. In fact the lexicographic preference is obtained by intersecting the local lexicographic preferences. Since S_1'' and S_2'' are obtained by minimization w.r.t. h this corresponds to consider the level h, the levels under h and those above h.

– It is easy to check that the lexicographic preference at level h does not change by minimization since the priority structure of the levels under h is not modified. In this way $\sqsubseteq^{\oplus}_{S_1',h} = \sqsubseteq^{\oplus}_{S_1'',h}$ and $\sqsubseteq^{\oplus}_{S_2',h} = \sqsubseteq^{\oplus}_{S_2'',h}$.
– The levels under h can be omitted in the intersection since their lexicographic preferences contain the lexicographic preference at h.
– In the lexicographic preferences at the levels k above h we have $m \sqsubset^{\oplus}_{S_1'',k} n$ and $m \not\sqsubseteq^{\oplus}_{S_2'',k} n$ since interpretation structures that are unrelated (resp. strictly related) at a certain level remain unrelated (resp. strictly related) at levels above.

It is straightforward to conclude that the lexicographic preference $\sqsubseteq^{\oplus}_{S_1''}$ of S_1'' will have $m \sqsubset^{\oplus}_{S_1''} n$ whereas that from S_1'' will have $m \not\sqsubseteq^{\oplus}_{S_2''} n$. We conclude $\mathsf{lex}^{\circ}(S_1'') \neq \mathsf{lex}^{\circ}(S_2'')$ as wanted. ✓

3.2.8 Other Composition Forms

Composition of (hierarchic) specifications is understood in the previous formalizations as the addition of syntactical entities such as axioms, defaults, priority levels and relations between them.

This provides the most general framework for the identification of composition operations. Indeed the specification result must be constructed by adding such syntactical entities to the parameter specifications.

This does not mean, however, that the composition forms between specifications must correspond to colimits in the complete categories Spec or hieSpec. In the following we refer some examples of composition forms that are formalized by colimits in subcategories of either Spec or hieSpec.

A first motivation is the following. The formalism adopted is defeasible in the sense that previous conclusions may be rejected when new information is added. This is important when modeling certain patterns of reasoning (see [63]) or, closer to specification issues, when putting together data bases with inconsistent data, in modeling the frame rule or the closed world assumption ([8]), the specificity principle ([74, 76]) or simply inheritance with overriding ([9, 12]). However it is also important that the specifier has the means to state that certain parts of the specification (possibly build with constructions allowing for defeasibility) are *stable*, in the sense that their conclusions can no longer be contradicted. One possibility to achieve this is to identify such parts of the specification with their skeptical consequences (see [10]). The same goal can be achieved by stating that, when composing such parts of the specification, the skeptical consequences should be preserved. In this way we are looking for the smallest specification build from the parameter specifications that furthermore preserves the skeptical consequences of such parameter specifications.

The formalization of such an operation is simple. Instead of taking Spec or hieSpec one needs to consider the categories Spec + Skept or hieSpec + Skept obtained by restricting the morphisms in Spec or hieSpec to those that preserve the skeptical consequences. Colimits in these categories are now the smallest specifications that include the axioms and defaults (and possibly priority structure) of the parameter specifications and that, furthermore, respect the new morphism condition, i.e. preserve the skeptical consequences of the parameter specifications. The semantic counterpart of the new categories is also easy to obtain (inclusion of maximal models).

This idea can easily be generalized. The formalization of a new general form of composition corresponds to taking colimits in appropriate subcategories of Spec or hieSpec. These subcategories are obtaining by further restricting the morphism condition with the intended preservation property.

Another example relevant for specification is the following. It is often the case that, although using a defeasible formalism, one would like to be as close to the classical case as possible. This is the case, for example when the defaults are of a very restricted nature, and are used for imposing some global properties such as the frame rule. It is known that the composition of specifications having only one extension does not yield necessarily specifications with that property. But we may propose a new form of composition by imposing this property. The first solution is to restrict the category of specifications to those that only have one extension. Another possibility is to impose the additional morphism condition that the number of extensions should not augment (if the arguments have one only extension so will the result). The corresponding semantic condition is again easy to find (on the number of maximal equivalence classes). Composition in this case corresponds to the smallest specification having the axioms, defaults (and possibly priority structure) of the parameters and, furthermore, only one extension.

The general rule for the definition of new forms of composition has to be checked in each case. It is not necessarily the case that the corresponding colimits exist. Usually it is simpler to check the existence of such constructions at the semantic level. This corresponds to finding the biggest pre-order (or hierarchy of pre-orders) satisfying the new conditions. After a semantic solution is found (if at all) one must find a specification having this solution as semantics. The best candidate is the theory of this pre-order, since it is the specification whose semantics is closer to the given pre-order. Finally and also important is to check whether there is a finite way of expressing such specification (see [25] for a treatment of the condition of preservation of skeptical consequences).

3.3 Final Remarks

This chapter has generalized the classical theory of composition of presentations to specifications and hierarchic specifications. Composition is formalized both on the syntactic side and on the semantic side and consists of the addition of syntactical entities (axioms, defaults, priority levels and relations between these). The semantical characterization of such constructions assures that they are abstract, i.e. that they depend only on the meaning of the specifications involved, and not on the particular way these are written. Further independence of representation is obtained by the notion of isomorphism. Composition does not depend on the choice of signature symbols and priority level names (since categorial constructions are defined up to isomorphism). Moreover, it is often the case that general forms of composition are defined on the semantic level. The framework provides the means to exhibit the corresponding syntactical expression (see section 3.2.8).

We saw also that the semantics adopted have precisely the structure needed to formalize these forms of composition. The proof of this fact (and the techniques used to prove it) are to our best knowledge, new.

This general framework can be extended to account for still other forms of composition as sketched in section 3.2.8. The concepts here presented are inspired in the composition for the classical case (see [46]) and driven by [12] where the concepts of syntactical composition are put forward. The semantic account is, to our best knowledge, new.

Chapter 4

Use of Defaults in Specification

This chapter illustrates the use of hierarchic specifications and their composition operations in specification. The concepts presented in this chapter have been introduced elsewhere, and correspond to the application of the "abstract specification theory" to object oriented systems modeled with temporal logic and defaults. The references that most directly influenced this chapter are [31, 11, 84, 61].

Firstly an institution, inspired in the Object Specification Logic from [85] is defined (in section 4.2). This logic serves as the base formalism for the specification of object oriented systems. Questions specific to the use of axioms and defaults (with priorities) to specify classes (of objects) are discussed in section 4.3. Their application in a simple example is presented in section 4.4.

Operations combining specifications are divided in two types: those enlarging the structure of the specification that are formalized with the concepts introduced in chapter 3 and those simplifying it. In this way the specification process evolves by composition of specification parts and is punctuated by simplifications of structure such as *priority abstraction*, and *cristalization*. This later operation assigns to a specification a set of axioms that has the original intended models as only models. Its importance results from the recognition that many of the properties specified with the use of defaults represent absence of information at *specification time*. This information will be available at the end of the specification process. At that point the specification can be cristalized. The use and definition of these operations is presented in section 4.1. They are illustrated in section 4.5. That defaults are also needed at *run time* is illustrated in section 4.6. In section 4.7 we conclude the chapter.

4.1 Operations on Specifications

The basic parts of a specification are defined by the use of axioms and prioritized defaults of a convenient institution, like the one presented in section 4.2. These basic parts are them combined in order to define larger and more complex specifications. In this section we concentrate in these institution independent operations. A first set of operations has been formalized in chapter 3. These are the colimits in the category Spec or hieSpec and build specifications from smaller parts by addition of formulas, either axioms or defaults, possibly adding or organizing priority levels.

This set of operations that augment the structure of the specification is complemented with operations that simplify that structure, still keeping their logical meaning. These operations are *priority abstraction* and *cristalization*. Priority abstraction should be used when further operations do not refer the internal structure of the abstracted specification, but may still defeat it. Cristalization, on the contrary, transforms a specification in a classical presentation, keeping the skeptical consequences. This operation should be used at a later stage in the specification.

The use of composition operations in specification is briefly referred in section 4.1.1. Priority abstraction and cristalization are referred in section 4.1.2. Cristalization is defined in this section.

4.1.1 Composition

The composition operations have been used in the classical context to formalize concepts such as communication and aggregation ([31]), inheritance ([9, 21, 22, 61]), parametric specifications ([16]) and reification ([30]), among other.

Colimits in Spec or hieSpec generalize the classical composition with the addition of defaults and priorities among them. This means that the classical operations may be generalized to this more expressive framework. For example the formalization of communication presented in section 4.5.4 is a trivial adaptation of [31].

Inheritance corresponds to the construction of a new specification (usually of a class, see section 4.5.1) by the (re)use of some previously defined specification. The classical composition operations can only formalize a monotonic form of inheritance, where more properties are added to the original specification. However, the possibility of redefining parts of the original specification improves modularity and reusability ([11, 61]) and is formalized by overriding of properties in [74, 12, 75, 11, 61]. In this way some of the properties of a class may no longer hold for a subclass.

The framework presented in chapter 3 allows for the definition of a form of inheritance where some parts of the original specification are kept and other overridden. This possibility is named *inheritance with selective overriding* and corresponds to a hierarchic specification morphism between the original and the resulting specifications. An example of this construction is presented in section 4.5.1.

4.1.2 Simplification of Structure

The operations of priority abstraction and cristalization are defined in the following. Priority abstraction assigns to a hierarchic specification an equivalent one with only axioms and a level of defaults. Cristalization assigns to a specification the set of axioms that have the original skeptical consequences as consequences. This later operation is defined below.

Of less importance, but still useful, are the simplifications that consist in removing from the specification defaults that are redundant (i.e. implicit). Care has to be taken with this process since redundant defaults may also be useful, in the sense that they correspond to lemmas, i.e. they may fasten the derivation process. These simplifications use the characterization of implicit defaults presented in theorems 51 and 123.

Priority Abstraction

Priority abstraction consists in forgetting the structure of a specification and providing an equivalent one with only one level of defaults. This is formalized by the operation $\mathsf{flat}(S)$ presented in section 2.2.2. Recall that S must be finite and the underlying institution must have disjunctions and conjunctions.

Priority abstraction is important when combining large specifications, still obtaining a manageable specification. The abstractions of the argument specifications can be taken as the basis for composition, instead of their detailed descriptions. Inheritance with selective overriding may again be used at this stage until further priority abstraction is needed. The use of priority abstraction is illustrated in section 4.5.2.

On the Need of Cristalization

A specification is a rigorous description of the potential behavior of a system. Such a system may use some defeasible mechanism and the corresponding specification must describe it. For example databases can be seen as theories that evolve by revising previous beliefs (imposed by the closed world assumption).

However, the use of defaults at specification time is not a consequence of the need of some defeasible mechanism in the final system. In particular the specification of systems that do not use any defeasibility mechanism may itself use defaults. These formalize absence of specification information. They express properties that may be overridden at other points in the specification.

Such properties, however, can be expressed as axioms when all specification information is available. This corresponds to apply to some part of the specification a *cristalization* operation. This operation yields a specification where the interactions between defaults have been computed and the global effects expressed as axioms. Operations of cristalization are illustrated in sections 4.5.3 and 4.5.7.

Definition of Cristalization

Cristalization consists in assigning to a specification a corresponding presentation (axioms) having the same skeptical consequences.

We define this operation for flat finite specifications of compact institutions having disjunctions. The corresponding operation for a hierarchic specification is obtained by the cristalization of $\mathsf{flat}(S)$.

Firstly we define the operation that assigns to a flat specification S another specification `OneExt(S)` having the same axioms as S but only one extension. The union of the axioms and defaults from `OneExt(S)` is the intended cristalization of S. The operation `OneExt(S)` is interesting in itself since it approximates the classical case with the advantage of further overriding of defaults being still possible.

The specification `OneExt(S)` is defined only for finite specifications of compact institutions having disjunction as follows.

Definition 197 Let S be a *finite* Σ-specification of a compact institution $(\mathcal{I}, \mathsf{dj})$ having disjunctions. Let $\check{\Delta}$ denote the disjunction $\mathsf{dj}_\Sigma(\Delta)$ of the formulas in the set Δ. Then `OneExt(S)` is the specification with

- $\mathsf{ax}(\texttt{OneExt(S)}) = \mathsf{ax}(S)$, the same axioms as S and
- $\mathsf{df}(\texttt{OneExt(S)}) = \{\check{\Delta} : \Delta \subseteq \mathsf{df}(S)$ and $\check{\Delta}$ is a skeptical consequence of $S\}$, the disjunctions of the original defaults from S, that are skeptical consequences of S as defaults. ∎

Note that the defaults of `OneExt(S)` are obtained by constructing all subsets of $\mathsf{df}(S)$ and then checking skeptical consequence of S. Alternatively, if the set of extension presentations of S is available each set Δ is obtained by picking one default from each such extension presentation.

The intended properties of `OneExt(S)` are stated in the following lemma.

Lemma 198 Let S be a *finite* Σ-specification of a compact institution $(\mathcal{I}, \mathsf{dj})$ having disjunctions. Then $\mathtt{OneExt(S)}$ has only one extension and the skeptical consequences of S coincide with those of $\mathtt{OneExt(S)}$.

Proof If S is inconsistent so is $\mathtt{OneExt(S)}$. Both have one only extension and both have the whole Σ-language as skeptical consequences.

If S is consistent so are its skeptical consequences. In particular the union $E = \mathsf{ax}(\mathtt{OneExt(S)}) \cup \mathsf{df}(\mathtt{OneExt(S)})$ is consistent. It is obvious that E is the only extension presentation of $\mathtt{OneExt(S)}$.

We have to show that the skeptical consequences of S coincide with those of $\mathtt{OneExt(S)}$. The skeptical consequences of $\mathtt{OneExt(S)}$, $\mathtt{skept(OneExt(S))}$ are the consequences of its only extension presentation E. Since $E \subseteq \mathtt{skept(S)}$ it follows trivially that

$$\mathtt{skept(OneExt(S))} = \mathtt{E}^{\bullet\bullet} \subseteq \mathtt{skept(S)}^{\bullet\bullet} = \mathtt{skept(S)}.$$

The other inclusion is proved as follows. We see that any model m of E is maximal according to S. Therefore $E^{\bullet} \subseteq \mathsf{max}(S)$ and $E^{\bullet\bullet} \supseteq \mathsf{max}(S)^{\bullet} = \mathtt{skept(S)}$, which ends the proof.

Consider a model m of E. In the preference of S there is a maximal model m' with $m \sqsubseteq m'$ (lemma 66). Let $d \in \mathsf{df}(S)$ and assume that $m' \vDash d$. We see that m also satisfies d and is, therefore equivalent to m' and also maximal. Let $[m']$ be the (maximal) equivalence class of m' and choose for each maximal equivalence class $[m'_1], ..., [m'_n]$ different from $[m']$ a default d'_i that is satisfied by m'_i and not by m' (this is possible since m' and m'_i are unrelated and therefore must satisfy different defaults). Note that since S is finite there is also a finite number of such maximal equivalence classes (as many as extension presentations). The disjunction $d \vee {'_1} \vee ... \vee d'_n$ of the previous formulas is a skeptical consequence of S since it holds in all its maximal equivalence classes. Therefore $d \vee d'_1 \vee ... \vee d'_n \in E$ and $m \vDash d \vee d'_1 \vee ... \vee d'_n$. Since m' does not satisfy any of the d'_i and $m \sqsubseteq m'$ then also m does not satisfy any of the d'_i . We conclude from $m \vDash d \vee d'_1 \vee ... \vee d'_n$ that $m \vDash d$. In this way m is equivalent to m'. ✓

The cristalization of S is defined simply as the union of the axioms and defaults from $\mathtt{OneExt(S)}$.

Definition 199 Let S be a *finite* Σ-specification of a compact institution $(\mathcal{I}, \mathsf{dj})$ having disjunctions. The cristalization of S is the presentation $\mathtt{Axiomatize(S)} = \mathsf{ax}(\mathtt{OneExt(S)}) \cup \mathsf{df}(\mathtt{OneExt(S)})$ having as (classical) consequences the skeptical consequences of S.

Proof That the consequences of $\mathtt{Axiomatize(S)}$ are the skeptical consequences of S has been shown in the proof of the lemma 198 above. ✓ ■

An alternative definition of $\mathtt{Axiomatize(S)}$ may be found in [10].

4.2 The Underlying Logic

The underlying logic is linear temporal logic (propositional) with future and past operators ([85]). Each trajectory begins with a start state. This means that states are named by the natural numbers ($\mathbb{N}_0$) and not by the integers (no infinite past). As in [31, 85] propositional symbols are of two kinds. Those representing observations of the behavior of an object, the attributes and those representing occurrence or enabling of actions. Occurrence of actions is separated from their enabling for methodological reasons. This procedure separates the specification of an action in the specification of the conditions in which it may occur (enabling) from the effects that its occurrence may have (see section 4.4 for examples). Enabling is a non trivial concept in temporal logic with multilinear time ([33]).

The institution corresponding to this logic is briefly described below by defining the signatures and language and the interpretation structures and satisfaction.

4.2.1 Syntax

Signatures and their language are defined as follows.

Definition 200 A *signature* Σ is a pair $(\Sigma_{act}, \Sigma_{obs})$ where Σ_{act} is the set of *action symbols* and Σ_{obs} is the set of *observation symbols* (or *attributes*).

The set of propositional symbols P_Σ from the signature Σ is the set $\Sigma_{obs} \cup \{\Diamond a, \triangledown a; a \in \Sigma_{act}\}$ of the observation symbols and the enabling and occurrence of the action symbols.

The set L_Σ of Σ-formulas is inductively generated as expected. The formula $*$ represents (holds only in) the initial state that corresponds to the creation of the whole system. The connective U is *until* and S is *since*.

- $P_\Sigma \subseteq L_\Sigma$,
- $* \in L_\Sigma$,
- $\neg f \in L_\Sigma$ if $f \in L_\Sigma$,
- $f \Rightarrow f'$, $f \mathsf{U} f'$ and $f \mathsf{S} f'$ belong to L_Σ if f and f' do.

Other connectives such as $\wedge$, $\vee$, true and false are defined as the usual abbreviations. Also $\mathsf{F}f$ abbreviates $\mathsf{true}\ \mathsf{U}f$ (sometime in the future), $\mathsf{P}f$ abbreviates $\mathsf{true}\ \mathsf{S}f$ (sometime in the past), $\mathsf{G}f$ abbreviates $\neg\mathsf{F}\neg f$ (always in the future), $\mathsf{H}f$ abbreviates $\neg\mathsf{P}\neg f$ (always in the past), $\mathbf{X}f$ abbreviates $\mathsf{false}\mathsf{U}f$ (next) and $\mathbf{Y}f$ abbreviates $\mathsf{false}\mathsf{S}f$ (previous). ▪

Attributes (and also actions) will often be represented with "parameters" as in $o(u)$. In fact these are here "user friendly" propositional symbols.

4.2.2 Semantics

The semantics of the linear temporal logic consists of anchored models that are defined as follows.

Definition 201 A (linear) frame is the set $\mathbb{N}_0$ of natural numbers (with the $<$ ordering as visibility relation). A P_Σ-model (or life-cycle) $\mathcal{M}$ is a function $\mathcal{M} : \mathbb{N}_0 \to 2^{P_\Sigma}$ assigning to each time point $n \in \mathbb{N}_0$ a set of propositional symbols (that hold in n). An *anchored* P_Σ-model is a pair $(\mathcal{M}, n)$ where $\mathcal{M}$ is a P_Σ-model and $n \in \mathbb{N}_0$. ▪

Truth of a formula at point n in a model $\mathcal{M}$ is defined in the construction of formulas as expected:

- $\mathcal{M} \vDash_n p$ iff $p \in \mathcal{M}(n)$ for $p \in P_\Sigma$,
- $\mathcal{M} \vDash_n *$ iff $n = 0$,
- $\mathcal{M} \vDash_n \neg f$ iff $\mathcal{M} \nvDash_n f$,
- $\mathcal{M} \vDash_n f \Rightarrow f'$ iff either $\mathcal{M} \nvDash_n f$ or $\mathcal{M} \vDash_n f'$,
- $\mathcal{M} \vDash_n f \mathsf{U} f'$ iff there exists $t > n$ such that $\mathcal{M} \vDash_t f'$ and $\mathcal{M} \vDash_s f$ for all s such that $n < s < t$,
- $\mathcal{M} \vDash_n f \mathsf{S} f'$ iff there exists $t < n$ such that $\mathcal{M} \vDash_t f'$ and $\mathcal{M} \vDash_s f$ for all s such that $t < s < n$.

Satisfaction of a formula in the anchored model (life-cycle) $(\mathcal{M}, n)$ represented by $(\mathcal{M}, n) \vDash f$ is truth of f in $n \in \mathbb{N}_0$.

Remark 202 As noted in remark 119 this institution has negation and the negation of f is $\neg f$. It also has the obvious disjunctions and conjunctions.

4.2.3 Built in Axioms

Similar versions of this logic presented in [31, 85] restrict the life cycles to those satisfying certain conditions related with the paradigm of object orientation. These conditions are imposed by appropriate axioms in sections 4.3 and 4.4.

4.2.4 Decidability

The linear temporal logic presented is decidable (see [28]).

4.3 Classes

Objects are grouped into classes when they have the same archetypical behavior. The specification of a class is the specification of that archetypical behavior, i.e. of the *template* of the class.

The objects populating a class are distinguished by some identification mechanism. This is usually a *data type*. We will simply assume that an appropriate set of object names is available (see also section 4.5.5).

The specification of the template of the class consists in the specification of its behavior, i.e. the conditions regulating the occurrence of local *actions* (or local methods), plus the effects of the occurrence of those actions in the local *attributes*.

In this section we concentrate in the conditions that are general to the specification of (the template of) any class. These are imposed by adding corresponding axioms or defaults to the specification of each class. Differences to the standard use of occurrence of actions are referred in section 4.3.1. The relation between enabling and occurrence of actions is displayed in section 4.3.2. In section 4.3.3 the (technically useful) *parameterless* actions are introduced. In section 4.3.4 the conditions regulating creation and destruction of objects are presented. The locality axioms (that local attributes may only be changed by local actions) and their relation to the frame rule are referred in section 4.3.5. The specification of sequential (non-concurrent) objects is presented in section 4.3.7. In section 4.3.8 "(re)active" objects are introduced. Some methodological considerations on the specification of enabling of actions are presented in section 4.3.6.

4.3.1 Past, Present and Future

Contrary to usual specification style ([31, 85]) we prefer to interpret the change in the value of an attributes as simultaneous with the occurrence of the action that changes it. This means that to specify that the occurrence of action a causes attribute o to become true the formula $\triangledown a \Rightarrow o$ is taken, instead of $\triangledown a \Rightarrow \mathbf{X}o$. Our motivation for this "time shift" and the alternative specification style is the following. The effect of an action happens somewhere between the beginning of the occurrence of an action and its conclusion. When actions are modeled as timeless their effects are simultaneous with their occurrence (somewhere between the coinciding beginning and conclusion). There is no technical reason for using this style.

4.3.2 Occurrence and Enabling

Actions can only occur if they were enabled. This is formalized by an axiom of the form $\triangledown a \Rightarrow \mathbf{Y}\Diamond a$, for each action a.

4.3.3 Parameterless Actions

An action $a(x)$ codes a potentially infinite number of actions. For example the action $\mathsf{file}\circ\mathsf{birth}(u)$ (specified below in section 4.4.4) corresponds to the creation of a file by an user an u. This action will be "copied" resulting in several actions $\mathsf{file}\circ\mathsf{birth}(u_1), ..., \mathsf{file}\circ\mathsf{birth}(u_n)$, one for each user (section 4.5.5). It is convenient to add to each action with parameters a corresponding parameterless action $a()$ with the meaning that an occurrence of $a()$ corresponds to an occurrence of at least one of the $a(x)$.

This meaning is imposed as follows: the axioms $\Diamond a(x) \Rightarrow \Diamond a()$ and $\triangledown a(x) \Rightarrow \triangledown a()$ impose that enabling (resp. occurrence) of $a(x)$ implies enabling (resp. occurrence) of $a()$. In order to impose that these are the only circumstances where $a()$ should be enabled or occur the defaults $\neg\Diamond a()$ and $\neg\triangledown a()$ are added.

4.3.4 Creation and Destruction

The rules regulating creation and destruction of objects are the following: objects can only be created if they do not exist. And can only be destructed if they exist. No method (except creation) of an object can be invoked if the object does not exist. Axioms of the form $\Diamond\mathsf{birth}() \Rightarrow \neg\mathsf{exists}$, $\Diamond\mathsf{death}() \Rightarrow \mathsf{exists}$ and $\Diamond a() \Rightarrow \mathsf{exists}$ formalize these restrictions. Each object has a $\mathsf{birth}()$ and a $\mathsf{death}()$ actions and an attribute exists.

4.3.5 Locality and the Frame Rule

Common to all objects (from all classes) is the requirement that they behave according to the paradigm of object orientation. In particular we assume that the properties of individual objects can only be changed by local methods (i.e. local actions). This requirement can be coded by the axioms $(\mathbf{Y}o \not\Leftrightarrow o) \Rightarrow (\triangledown a_1() \vee ... \vee \triangledown a_n())$, for each attribute o, where $a_1(), ..., a_n()$ are all local actions without parameters. In this way we impose that o changes because one of these actions occurred. (Recall that if the action that occurred is $a_1()$ this means that some of the $a_1(x)$ occurred.)

Although these axioms impose the locality principle they still allow unwanted changes of the values of the attributes. In fact not all of those actions act upon all attributes. We would like to add an axiom stating that the value of each attribute can only change by the occurrence of those local actions that explicitly change that attribute. A convenient way of specifying this condition is to add a priority level "frame" with the defaults expressing the frame rule: the values of the attributes persist unless explicitly stated otherwise. These defaults are of the form $\mathbf{Y}o \Leftrightarrow o$. The specification of the actions that actually change the attribute will override this default.

4.3.6 Default Enabling Conditions

Default enabling conditions are the conditions determining whether the action may or may not occur in those situations that are not covered by the rest of the specification (i.e. unthought of). The enabling conditions of a potentially destructive action (such as `death`) should be specified by declaring that, by default, the action is not enabled. And, at a second stage, explicitly declare in which situations it is enabled.

The opposite style, of declaring an action as enabled by default and then explicitly stating in which situations is not enabled, is also possible but should be reserved for actions whose occurrence is not destructive.

It is worthwhile noting that both styles may be used in the *description* of an action, organized in levels of less and less generality, independently of the actual specification. For example the description of the UNIX"rm" command (remove) is that "rm" removes files. The description is complemented by the conditions in which that is possible. A safe implementation of the same command should be contrary to its description.

4.3.7 Sequentiality

Actions (possibly from different objects) may occur in concurrency and their occurrence may be simultaneous. It is often considered that "atomic" objects behave sequentially and that concurrency is the result of their composite behavior ([27]). Sequentiality of atomic objects may be imposed by stating that their actions do not occur simultaneously. This corresponds to add the axioms $\triangledown a_1() \Rightarrow \neg\triangledown a_2()$ for each pair of different local actions without parameters $a_1()$ and $a_2()$. These axioms, however, still allow the simultaneous occurrence of the same action with different parameters. I.e. $a_1(x)$ may occur simultaneously with $a_1(y)$. This may be intended: for example a file may be read by different users at the same time. But the same file cannot be written by different users at the same time. In the last case a semaphore mechanism must be implemented. The abstract conditions imposing the intended sequentiality must be stated when composing the atomic object (classes) involved.

One interesting possibility is to state that, when composing classes, the result specification should be as sequential as possible. This means that only the actions that must occur simultaneously do occur simultaneously. This maximal sequentiality is imposed by the defaults $\triangledown a_1 \Rightarrow \neg\triangledown a_2$, for any pair of actions from different object classes.

4.3.8 Liveness

Some objects have *initiative* in the sense that under some circumstances they are the cause of the occurrence of some of their actions. Objects with initiative ([20]) may correspond to real objects (outside the system) that interact with it. Initiative is also expected from the system for example when *responding* to a request ([69]). This behavior can be modeled by formulas of the form $c \wedge \Diamond a \Rightarrow \mathsf{F}\triangledown a$ meaning that a will occur if its is enabled and some extra condition c holds. The use of these formulas as axioms may easily lead to inconsistency in the presence of communication. Even if locally $c \wedge \Diamond a \Rightarrow \mathsf{F}\triangledown a$ holds it may cease to hold when the action a is involved in some form of communication with another action a' from a different object. If some condition prevents a' to occur also a will not occur contradicting $c \wedge \Diamond a \Rightarrow \mathsf{F}\triangledown a$.

Inconsistency is prevented by the use of the formulas $c \wedge \Diamond a \Rightarrow \mathsf{F}\triangledown a$ as defaults.

4.4 Files and Users

The example specification concerns the interaction between users and files (of the operating system UNIX). The class File (and the class User) are specified by stating the template of each class, i.e. the behavior of an arbitrary member of the class. The signatures of File and User are presented in section 4.4.1. The specification of File includes the rules governing access of local actions to local attributes. These are presented in section 4.4.2. The specification of some actions of File is presented in sections 4.4.4, 4.4.5 and 4.4.6. These are divided in the specification of conditions common to all actions, the enabling conditions and the effects of actions upon attributes.

4.4.1 Signatures

We proceed by stating the signatures both of the template of File and of User. In this example we are mainly concerned with creation and destruction of files by users, taking the permissions of the file into account. Each file acts as a "file controller" and controls this interaction.

Files are acted upon by users. This interaction corresponds to sharing of corresponding actions. For this reason the template File refers to an arbitrary user and the template of User refers to an arbitrary file. For example the action file∘birth(u) of File refers to the creation of *this* file by user u. When composing the specification of a file with the specifications of several users there will be an action file∘birth(u) for each such user. The notation f.file∘birth(u) refers to the action file∘birth(u) of the file f and it will be used when several files are available.

Since actions with parameters such as file∘birth(u) code a potentially infinite number of actions it is convenient to introduce a parameterless action representing the occurrence of any of those actions. For example file∘birth() is the parameterless action corresponding to file∘birth(u). The axiom $\triangledown$file∘birth(u) $\Rightarrow$ $\triangledown$file∘birth() imposes the intended meaning of file∘birth(). There are several situations where this device is needed, for example in writing the locality rule. In the following signatures we assume that to each action with parameters is associated a parameterless one. The later are not explicit referred.

File

The signature stating the symbols relevant for the template of File is the following. We consider the following set of actions:

- file∘birth(u), user u creating this file,
- file∘death(u), user u deleting this file,
- chowner(u, v_1), ..., chowner(u, v_n), user u setting the owner of this file to v_i,
- user u changing the owner permissions of this file:
  - chmod∘R∘owner(u),
  - chmod∘not∘R∘owner(u),
  - chmod∘W∘owner(u),
  - chmod∘not∘W∘owner(u),
- user u changing the all (world) permissions of this file:
  - chmod∘R∘all(u),
  - chmod∘not∘R∘all(u),
  - chmod∘W∘all(u),
  - chmod∘not∘W∘all(u).

And the following set of properties (attributes):

- exists, a predicate stating whether the file exists or has not been created yet,
- read∘all, stating that the file is readable by any user,
- write∘all, stating that the file is writable by any user,
- read∘owner, stating that the file is readable by its owner,
- write∘owner, stating that the file is writable by its owner,
- owner(u), true iff u is the owner of the file.

User

Users are created and destructed (by the operating system). They interact with files by creating them, deleting them, and changing their permissions. The actions of User are:

- user∘birth, creates this user,
- user∘death, deletes this user,
- file∘birth(f), this user creates file f,
- exec∘file∘birth(f), this user creates the *executable* file f (see section 4.5.1),
- file∘death(f), this user deletes file f,
- chowner(f, v_1), . . . , chowner(f, v_n), this user changes the owner of the file f to v_i,
- this user changes the owner permissions of f:
  - chmod∘R∘owner(f),
  - chmod∘W∘owner(f),
  - chmod∘W∘owner(f),
  - chmod∘not∘W∘owner(f),
- this user changes the all (world) permissions of f:
  - chmod∘R∘all(f),
  - chmod∘not∘R∘all(f),
  - chmod∘W∘all(f),
  - chmod∘not∘W∘all(f).

The only attribute of User is:

- exists, a predicate stating whether the user exists or has not been created yet.

4.4.2 Locality

The axioms and defaults imposing the locality rule for the class File are now presented (other restrictions are presented when specifying each action). The axioms state that each attribute may only change value upon occurrence of one of the local actions. The frame rule, that the values of the attributes persist (unless explicitly stated otherwise later in the specification) is expressed by defaults. The corresponding axioms and defaults for the class User are formally similar and omitted.

Axioms

The locality axiom for read∘all is the following.

$$\begin{aligned}(\mathbf{Y}\mathsf{read{\circ}all} \not\Leftrightarrow \mathsf{read{\circ}all}) \quad\Rightarrow\quad & (\triangledown\mathsf{file{\circ}birth}() \vee \\ & \triangledown\mathsf{file{\circ}death}() \vee \\ & \triangledown\mathsf{chmod{\circ}R{\circ}owner}() \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}R{\circ}owner}() \vee \\ & \triangledown\mathsf{chmod{\circ}W{\circ}owner}() \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}W{\circ}owner}() \vee \\ & \triangledown\mathsf{chmod{\circ}R{\circ}all}() \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}R{\circ}all}() \vee \\ & \triangledown\mathsf{chmod{\circ}W{\circ}all}() \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}W{\circ}all}()).\end{aligned}$$

For the other attributes ($\mathsf{write{\circ}all}, \mathsf{read{\circ}owner}$ and $\mathsf{owner}(u)$) the axioms are similar. The attribute exists is an exception. The creation and destruction rules (see section 4.3.4) allow this attribute to be changed only by the creation and destruction actions: $(\mathbf{Y}\mathsf{exists} \not\Leftrightarrow \mathsf{exists}) \Rightarrow (\triangledown\mathsf{file{\circ}birth}() \vee \triangledown\mathsf{file{\circ}death}())$.

Defaults

The defaults expressing the frame rule are the following:

- $\mathbf{Y}\mathsf{exists} \Leftrightarrow \mathsf{exists}$,
- $\mathbf{Y}\mathsf{read{\circ}all} \Leftrightarrow \mathsf{read{\circ}all}$,
- $\mathbf{Y}\mathsf{write{\circ}all} \Leftrightarrow \mathsf{write{\circ}all}$,
- $\mathbf{Y}\mathsf{read{\circ}owner} \Leftrightarrow \mathsf{read{\circ}owner}$ and
- $\mathbf{Y}\mathsf{owner}(u) \Leftrightarrow \mathsf{owner}(u)$.

The following hierarchic specification represents the restrictions imposed by the locality principle and the frame rule.

Axioms are presented in boxes and the priority structure with corresponding defaults in dashed boxes.

$$\boxed{\text{frame}} : \{\mathbf{Y}\mathsf{read{\circ}all} \Leftrightarrow \mathsf{read{\circ}all}, \mathbf{Y}\mathsf{write{\circ}all} \Leftrightarrow \mathsf{write{\circ}all}, ...\}$$

$$\{(\mathbf{Y}\mathsf{read{\circ}all} \not\Leftrightarrow \mathsf{read{\circ}all}) \Rightarrow (\triangledown\mathsf{file{\circ}birth()} \vee ... \vee \triangledown\mathsf{file{\circ}death()}),$$

$$(\mathbf{Y}\mathsf{write{\circ}all} \not\Leftrightarrow \mathsf{write{\circ}all}) \Rightarrow (\triangledown\mathsf{file{\circ}birth()} \vee ... \vee \triangledown\mathsf{file{\circ}death()}), ...\}$$

Figure 4.1: Locality and the Frame Rule

4.4.3 Sequentiality

The axioms imposing sequentiality of an arbitrary file state that the occurrence of an action implies that none of the other occur. For the occurrence of $\mathsf{file{\circ}birth()}$ we must specify

$$\begin{aligned} \triangledown\mathsf{file{\circ}birth()} \quad \Rightarrow \quad & \neg(\triangledown\mathsf{file{\circ}death()} \vee \\ & \triangledown\mathsf{chmod{\circ}R{\circ}owner()} \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}R{\circ}owner()} \vee \\ & \triangledown\mathsf{chmod{\circ}W{\circ}owner()} \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}W{\circ}owner()} \vee \\ & \triangledown\mathsf{chmod{\circ}R{\circ}all()} \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}R{\circ}all()} \vee \\ & \triangledown\mathsf{chmod{\circ}W{\circ}all()} \vee \\ & \triangledown\mathsf{chmod{\circ}not{\circ}W{\circ}all()}). \end{aligned}$$

Similar axioms respecting the occurrence of the other actions must be added.

4.4.4 Creation

Creation corresponds to the specification of the actions $\mathsf{file}\circ\mathsf{birth}(u)$ and $\mathsf{file}\circ\mathsf{birth}()$ of File.

Enabling and Occurrence

The axioms imposing the relation between enabling and occurrence:

$$\{\triangledown\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \mathbf{Y}\Diamond\mathsf{file}\circ\mathsf{birth}(u),$$
$$\triangledown\mathsf{file}\circ\mathsf{birth}() \Rightarrow \mathbf{Y}\Diamond\mathsf{file}\circ\mathsf{birth}()\}$$

Figure 4.2: Enabling and Occurrence

Parameterless actions

The next axiom regulates the relation between enabling of $\mathsf{file}\circ\mathsf{birth}(u)$ and $\mathsf{file}\circ\mathsf{birth}()$:

- $\Diamond\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \Diamond\mathsf{file}\circ\mathsf{birth}()$.

In this way $\mathsf{file}\circ\mathsf{birth}()$ is enabled when any of the $\mathsf{file}\circ\mathsf{birth}(u)$ are enabled.

Further enabling conditions for $\mathsf{file}\circ\mathsf{birth}()$ are derived from those of $\mathsf{file}\circ\mathsf{birth}(u)$. The action $\mathsf{file}\circ\mathsf{birth}()$ should be enabled whenever any $\mathsf{file}\circ\mathsf{birth}(u)$ is. But this is the only situation where it should be enabled. This can be achieved by adding a further default $\neg\Diamond\mathsf{file}\circ\mathsf{birth}()$ of less priority than any occurring in the conditions defining the enabling of $\mathsf{file}\circ\mathsf{birth}(u)$. In this way $\mathsf{file}\circ\mathsf{birth}()$ will not be enabled unless explicitly stated otherwise. And it is explicitly stated that it is enabled whenever $\mathsf{file}\circ\mathsf{birth}(u)$ is. The formulas related with enabling of $\mathsf{file}\circ\mathsf{birth}()$ are summarized in the following hierarchic specification (figure 4.3). Note that this specification will be composed with the one corresponding to the enabling conditions of $\mathsf{file}\circ\mathsf{birth}(u)$ (see below).

Occurrence of $\mathsf{file}\circ\mathsf{birth}()$ is formally similar to its enabling. The action $\mathsf{file}\circ\mathsf{birth}()$ should occur iff any $\mathsf{file}\circ\mathsf{birth}(u)$ occurs. The if condition is imposed by the axiom

- $\triangledown\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \triangledown\mathsf{file}\circ\mathsf{birth}()$.

The only if condition is imposed by the default $\neg\triangledown\mathsf{file}\circ\mathsf{birth}()$. These considerations are displayed in the hierarchic specification in figure 4.4. The relation between other parameterless actions and the corresponding actions is similar to the previous considerations. It will be omitted in the following.

$$\boxed{\Diamond'} : \{\neg\Diamond\mathsf{file}\circ\mathsf{birth}()\}$$

$$\{\Diamond\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \Diamond\mathsf{file}\circ\mathsf{birth}()\}$$

Figure 4.3: Enabling of $\mathsf{file}\circ\mathsf{birth}()$

$$\boxed{\triangledown'} : \{\neg\triangledown\mathsf{file}\circ\mathsf{birth}()\}$$

$$\{\triangledown\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \triangledown\mathsf{file}\circ\mathsf{birth}()\}$$

Figure 4.4: Occurrence of $\mathsf{file}\circ\mathsf{birth}()$

Enabling

The next step is the specification of the enabling conditions for the actions $\mathsf{file}\circ\mathsf{birth}(u)$ and $\mathsf{file}\circ\mathsf{birth}()$. A restriction in the enabling conditions for $\mathsf{file}\circ\mathsf{birth}(u)$ has already been referred in section 4.3.4. This action may only happen if the file it refers does not exist. This is formalized by the axiom $\Diamond\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \neg\mathsf{exists}$. Further restrictions should be added at this point. We consider this action benignous (see the discussion in 4.3.6) and therefore we assume that there are no further restrictions. This action is enabled in all situations not covered by $\Diamond\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \neg\mathsf{exists}$. For this reason we add the default $\Diamond\mathsf{file}\circ\mathsf{birth}(u)$. The enabling conditions for $\mathsf{file}\circ\mathsf{birth}()$ are derived from those of $\mathsf{file}\circ\mathsf{birth}(u)$ and have already been presented.

$$\boxed{\Diamond} : \{\Diamond\mathsf{file}\circ\mathsf{birth}(u)\}$$

$$\{\Diamond\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \neg\mathsf{exists}\}$$

Figure 4.5: Enabling of $\mathsf{file}\circ\mathsf{birth}()$

Effects

The effect of the $\mathsf{file}\circ\mathsf{birth}(u)$ and $\mathsf{file}\circ\mathsf{birth}()$ action is the initialization of the attributes of the file. Initialization of each object corresponds to state the values of its attributes at the time the object is created. This initialization, together with the locality restrictions guarantees that the attributes, after the creation of the object, will have a value that depends only in the past occurrence of actions. The first attribute to be set is exists: $\triangledown\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \mathsf{exists}$. We begin with the initialization of the attributes of File by considering the predicate $\mathsf{owner}(u)$. If u has created the file then it is its owner. Otherwise is not. This can be formalized by stating that by default u is not the owner of the file: $\triangledown\mathsf{file}\circ\mathsf{birth}() \Rightarrow \neg\mathsf{owner}(u)$. But the more important default $\triangledown\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \mathsf{owner}(u)$ overrides the previous one if u has created the file.

The other predicates are initialized as follows: the file is readable and writable for its owner, readable by all and not writable by all. This corresponds to the defaults $\triangledown\mathsf{file}\circ\mathsf{birth}() \Rightarrow \mathsf{read}\circ\mathsf{all}$, $\triangledown\mathsf{file}\circ\mathsf{birth}() \Rightarrow \neg\mathsf{write}\circ\mathsf{all}$, $\triangledown\mathsf{file}\circ\mathsf{birth}() \Rightarrow \mathsf{read}\circ\mathsf{owner}$, $\triangledown\mathsf{file}\circ\mathsf{birth}() \Rightarrow \mathsf{write}\circ\mathsf{owner}$.

It is important, at this stage, that these are defaults since special files (exceptions) may be created with different permissions. This will be stated in the specification of such subclasses of File (see section 4.5.1). The previous initialization holds unless explicitly stated otherwise, somewhere else in the specification.

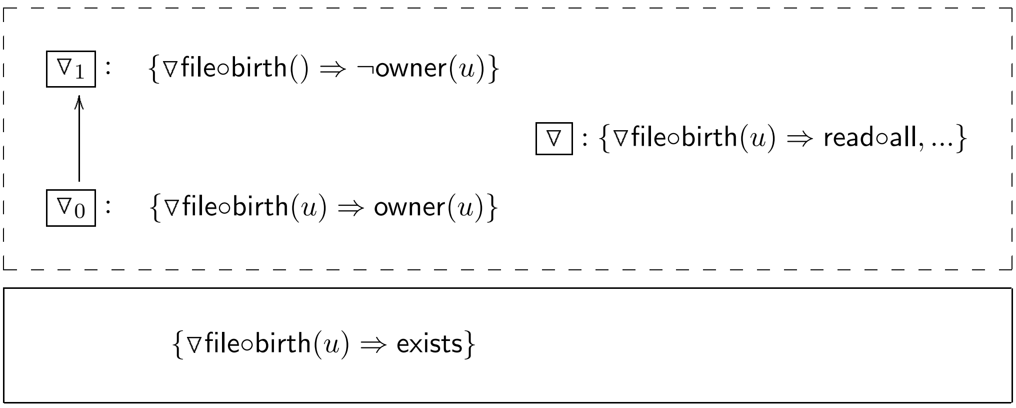

Figure 4.6: Occurrence of $\mathsf{file}\circ\mathsf{birth}(u)$

Composition

The specification of the actions $\mathsf{file}\circ\mathsf{birth}()$ and $\mathsf{file}\circ\mathsf{birth}(u)$ is obtained by composing the previous parts of the specification. The diagram corresponding to this composition is displayed in figure 4.7. There are two restrictions to be respected. The first respects the relation between the enabling conditions of $\mathsf{file}\circ\mathsf{birth}()$ and $\mathsf{file}\circ\mathsf{birth}(u)$. In the final specification the default $\neg\Diamond\mathsf{file}\circ\mathsf{birth}()$ must be above (less important) than those defining the enabling of $\mathsf{file}\circ\mathsf{birth}(u)$.

The second is global to all actions: in the final specification of the File the defaults imposing the frame rule must be above (less important) than those defining the effects of each action. For this purpose we introduce a priority level frame, without defaults, that will be used in the final composition to receive the frame defaults.

The structure of the specification of the file∘birth(u) and file∘birth() actions is the following. The ⇒ arrows represent priority relations imposed by this composition

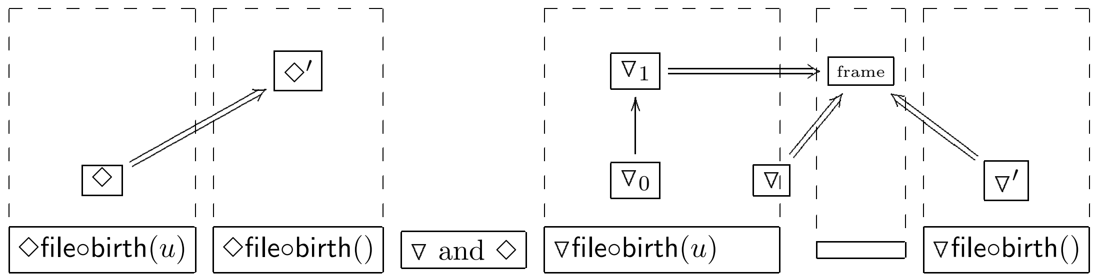

Figure 4.7: file∘birth(u) and file∘birth()

(recall chapter 3, namely section 3.2.1). The → arrows were present already.

4.4.5 Destruction

Corresponds to specify the action file∘death(u) of File.

Enabling and Occurrence

The meaning of enabling is stated by the axiom:

- $\triangledown$file∘death(u) ⇒ **Y**◇file∘death(u) .

Enabling

Destruction actions are only enabled if the object they refer to is alive. This is formalized by the axiom ◇file∘death(u) ⇒ exists. Further enabling conditions are specified as follows. Since this action is pernicious it will not be enabled by default. This is expressed by the formula ¬◇file∘death(u). At a more important level, we specify that it is enabled if the attribute write∘all, that allows the file to be written by any user is true: write∘all ⇒ ◇file∘death(u). Still at a more important level we specify that the owner can delete the file when the attribute write∘owner is true: owner(u) ∧ write∘owner ⇒ ◇file∘death(u). Finally the superuser can delete the file: ◇file∘death($root$). These conditions are represented in the following hierarchic specification displayed in figure 4.8.

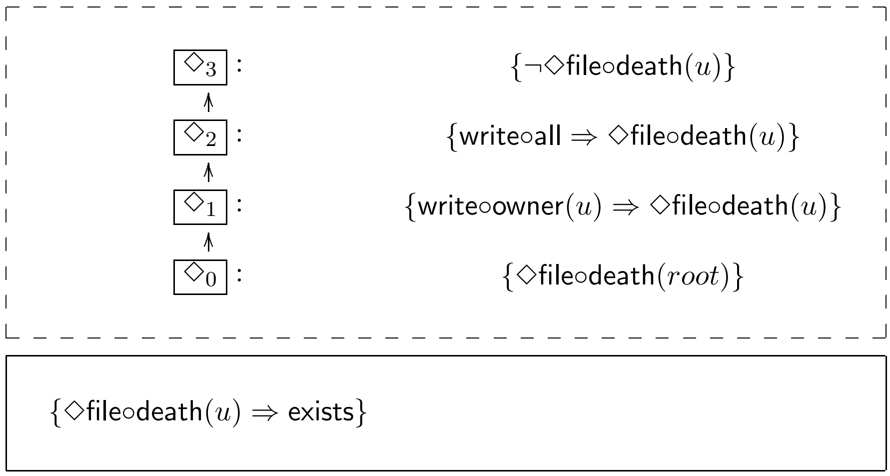

Figure 4.8: Enabling of $\mathsf{file{\circ}death}(u)$

Effects

The destruction actions have only one effect: the destruction of the object. The enabling conditions have been carefully written so that this action can only occur in the previously referred circumstances. In these circumstances the occurrence of the action should delete it. This is formalized by the axiom $\triangledown\mathsf{file{\circ}death}(u) \Rightarrow \neg\mathsf{exists}$. (There may be files that can never be deleted. To cover such cases we may either specify the formula $\triangledown\mathsf{file{\circ}death}(u) \Rightarrow \neg\mathsf{exists}$ as a default (to be overridden) or override the specification of the enabling conditions of the $\mathsf{file{\circ}death}(u)$ action and not allow this action ever to be enabled).

Composition

The overall specification of the $\mathsf{file{\circ}death}(u)$ and $\mathsf{file{\circ}death}()$ actions is obtained by composing their parts in a way similar to the $\mathsf{file{\circ}birth}(u)$ and $\mathsf{file{\circ}birth}()$ actions.

4.4.6 More Actions

The specification of other actions is similar to those already presented. We consider still another example, namely the actions $\mathsf{chowner}(u, v)$, for v is the set of user names. All these actions have the same form. We define $\mathsf{chowner}(u, v)$ for arbitrary v.

Enabling and Occurrence

The specification includes the following axiom due to the meaning of enabling:

- $\triangledown\mathsf{chowner}(u, v) \Rightarrow \mathbf{Y}\Diamond\mathsf{chowner}(u, v)$.

Enabling

The first condition is global to all actions (except creation actions): no action is enabled when the corresponding object does not exist. For each v this is formalized by the axiom $\Diamond\mathsf{chowner}(u,v) \Rightarrow \mathsf{exists}$. Other conditions are the following: only the owner of the file or the superuser (root) are allowed to change the owner of the file. This is achieved by stating that, by default this action is not enabled: $\neg\Diamond\mathsf{chowner}(u,v)$. At the owner level we state that the owner may $\mathsf{chowner}(u,v)$: $\mathsf{owner}(u) \Rightarrow \Diamond\mathsf{chowner}(u,v)$. And at the root level we state the corresponding property of root: $\Diamond\mathsf{chowner}(root,v)$.

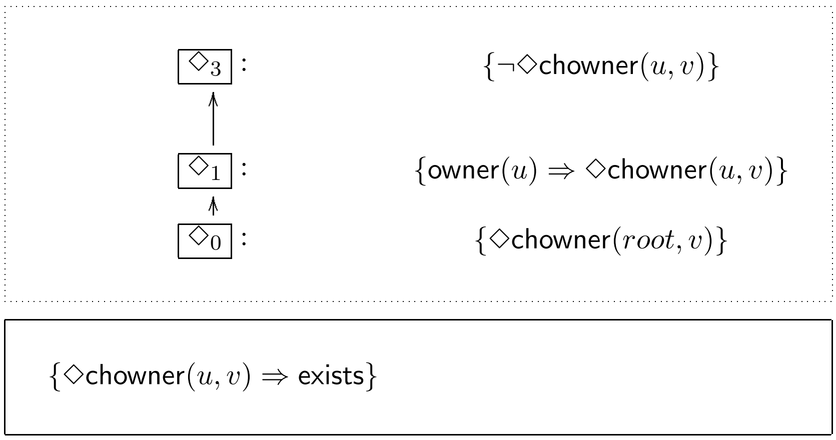

Figure 4.9: Enabling of $\mathsf{chowner}(u,v)$

Effects

The occurrence of $\mathsf{chowner}(u,v)$ changes the owner of the file to v. This corresponds to set the former $\mathsf{owner}(u')$ to false and to set $\mathsf{owner}(v)$ to true. The first effect is achieved by the default $\triangledown\mathsf{chowner}(v) \Rightarrow \neg\mathsf{owner}(u)$ where $\mathsf{chowner}(v)$ is the parameterless action corresponding to $\mathsf{chowner}(u,v)$. The second by the more important default $\triangledown\mathsf{chowner}(u,v) \Rightarrow \mathsf{owner}(v)$.

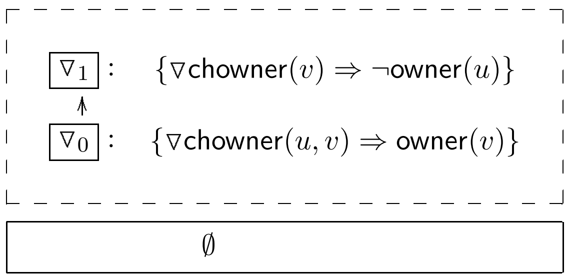

Figure 4.10: Effects of $\mathsf{chowner}(u,v)$

Composition

We omit the composition yielding the specification of the chowner(u, v) and chowner(v) actions.

4.4.7 Overall Specification

The specification of an arbitrary file is obtained by putting together the smaller specifications of its actions, possibly identifying priority levels. The locality and sequentiality axioms and the defaults of the frame rule must also be included. The following diagram represents the parts to be composed. The diagram on the signature part is omitted (we are assuming that all actions are written in the same signature, that of File). Moreover we identify only the level of the frame rule, although other priority levels could be identified. For example all enabling conditions (resp. all effects) could be identified.

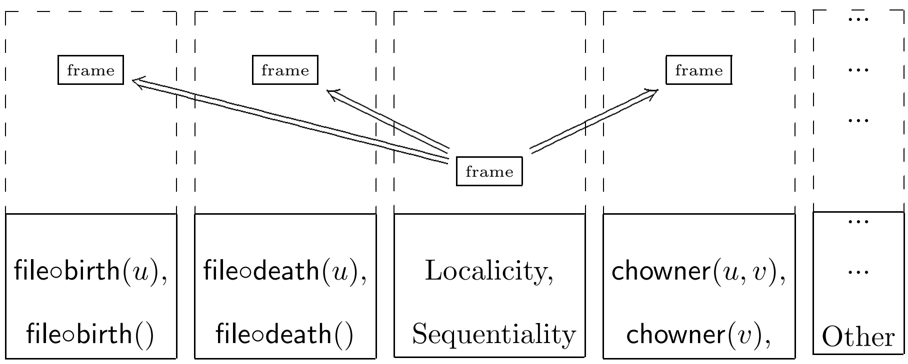

Figure 4.11: Template of File

4.5 More Files and Users

The class File is in this section (re)used for further specification. This includes the definition of a subclass in section 4.5.1 and a redefinition of the class File taking the new subclass into account. Communication of files and users by action sharing is illustrated in sections 4.5.4 and 4.5.5. Restrictions to concurrency are added in section 4.5.6. The operation of cristalization is used in sections 4.5.3 and 4.5.7.

4.5.1 Executable Files

Consider the subclass Exec∘File of file constituted by those files that are executable. These files are created by compilation processes, for example. We consider that an Exec∘File has an additional action exec(u) representing the execution of the file by user

u (the specification of this action is omitted). Furthermore these files have another predicate stating that they are executable. They may be executable by the owner of the file or by an arbitrary user. The specification of the create command is now different. At creation time the files should be executable by the owner and by all. Furthermore it should not be writable by the owner (thus overriding the specification of File). This corresponds to change the specification of the effects of the create action by adding the new defaults at the right levels. The new defaults (only) are shown in the next figure.

$$\boxed{\triangledown'} : \{\triangledown\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \mathsf{exec}\circ\mathsf{all}, \mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \mathsf{exec}\circ\mathsf{owner}\}$$

$$\boxed{\triangledown''} : \{\triangledown\mathsf{file}\circ\mathsf{birth}(u) \Rightarrow \neg\mathsf{write}\circ\mathsf{owner}\}$$

$$\emptyset$$

Figure 4.12: Defaults to be added to $\mathsf{file}\circ\mathsf{birth}(u)$

Note that two new attributes are referred: $\mathsf{exec}\circ\mathsf{all}$ and $\mathsf{exec}\circ\mathsf{owner}$. The new specification of the effects of create command is obtained by composition of the specification above with the original specification of $\triangledown\mathsf{file}\circ\mathsf{birth}(u)$. This corresponds to augment the signature with the new symbols $\mathsf{exec}(u)$ (and $\mathsf{exec}()$), $\mathsf{exec}\circ\mathsf{all}$ and $\mathsf{exec}\circ\mathsf{owner}$ and to identify the corresponding priority levels. This means that $\boxed{\triangledown'}$ is identified with $\boxed{\triangledown}$ and $\boxed{\triangledown''}$ is a new level of higher importance than $\boxed{\triangledown}$ (since the initialization of the $\mathsf{write}\circ\mathsf{owner}$ attribute must be overridden). The following figure depicts the diagram imposing these relations (the operation on the signatures is omitted).

The final specification of $\mathsf{Exec}\circ\mathsf{File}$ is obtained as in section 4.4.7, including the specification of the action $\mathsf{exec}(u)$.

4.5.2 Classes and Subclasses

The specification of $\mathsf{Exec}\circ\mathsf{File}$ above raises the question of redefining the class File, since it should represent an arbitrary file, whether executable or not. A possible solution is the following. The new specification is obtained by composing the old specification of File with that of $\mathsf{Exec}\circ\mathsf{File}$, and identifying the logical symbols that are common to both. Since the $\mathsf{file}\circ\mathsf{birth}(u)$ action has been redefined this symbol should not be identified. In this way we obtain two actions $\mathsf{file}\circ\mathsf{birth}(u)$ (and two actions $\mathsf{file}\circ\mathsf{birth}()$) one applying to executable files and the other to other files. In general each redefined action should have two versions. The new specification will have the attributes from File plus $\mathsf{exec}\circ\mathsf{all}$ and $\mathsf{exec}\circ\mathsf{owner}$. And the actions from File plus $\mathsf{exec}(u)$ and $\mathsf{exec}\circ\mathsf{file}\circ\mathsf{birth}(u)$

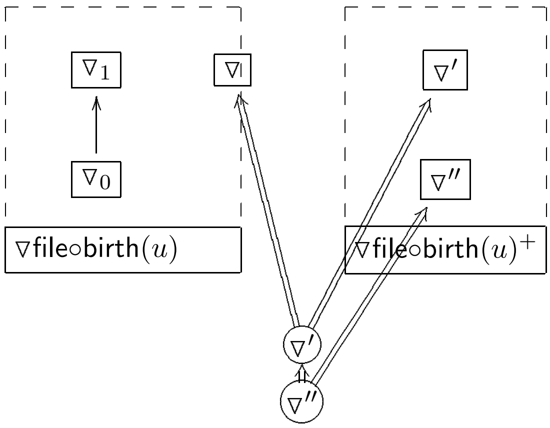

Figure 4.13: Action file∘birth(u) of Exec∘File

(the file∘birth(u) action of executable files). In order to distinguish between files and executable files a new attribute executable is added. The creation action file∘birth(u) of Exec∘File sets this attribute to true and the action file∘birth(u) of File sets it to false. The action exec(u) that is particular to an Exec∘File can only be enabled if executable holds ($\Diamond$exec(u) $\Rightarrow$ executable). There is no need to impose any condition on attributes specific to executable files. They can be accessed only by specific actions that are enabled only if the file is executable.

When combining the specifications of File and Exec∘File their internal priority structure is no longer relevant. This was not the case when producing the specification Exec∘File from File. But, assuming that no further subclasses are needed, the internal structure can be forgotten since future operations will not deal with the internal priority levels. Therefore we may transform the specifications of File and Exec∘File to flat equivalent ones. These have the same logical meaning as the original specifications (see sections 4.1.2 and 2.2.2). Moreover, the defaults of each specification have equal importance and no strict priority relation between them should be imposed. The final specification should consist of the union of these flat specifications. This union is obtained by combining the flat specifications in the category Spec of (flat) specifications. (Alternatively the flat specifications may be seen as hierarchic specifications with only one priority level. This particular composition corresponds to the identification of all single priority levels from different specifications.)

Since we compose the flat specifications in Spec only the signature morphisms must be provided. To represent this composition as the colimit of an appropriate diagram the following specifications must be introduced. Firstly a specification $S_\cap$ stating which logical symbols are common to the flat versions of File and Exec∘File. This specification is only the common signature and has no axioms and no defaults. Its signature is

$$\begin{aligned}\Sigma_\cap &= (\{\mathsf{file{\circ}death}(u), \mathsf{file{\circ}death}(), \mathsf{chowner}(u,v), \mathsf{chowner}(v), ...\},\\ &\quad \{\mathsf{exists}, \mathsf{read{\circ}all}, ...\}).\end{aligned}$$

The new attribute executable is introduced by another trivial specification having only this symbol and no axioms and defaults. The effects of the $\mathsf{file{\circ}birth}(u)$ (from File) command on $\mathsf{executable}$ are written in a third specification S' with signature

$$\Sigma_{S'} = (\{\mathsf{executable}\}, \{\mathsf{file{\circ}birth}(u)\}).$$

This specification includes the axiom $\triangledown\mathsf{file{\circ}birth}(u) \Rightarrow \neg\mathsf{executable}$. And the effects of the action $\mathsf{file{\circ}birth}(u)$ (from $\mathsf{Exec{\circ}File}$) on $\mathsf{executable}$ plus the enabling conditions for $\mathsf{exec}(u)$ are again written on a separate specification S'', with signature

$$\Sigma_{S''} = (\{\mathsf{executable}\}, \{\mathsf{file{\circ}birth}(u), \mathsf{exec}(u)\}).$$

The axioms $\triangledown\mathsf{file{\circ}birth}(u) \Rightarrow \mathsf{executable}$ and $\Diamond\mathsf{exec}(u) \Rightarrow \mathsf{executable}$ impose the intended conditions. The following diagram represents the inclusions between the signatures. Its colimit is the signature of the new version of File.

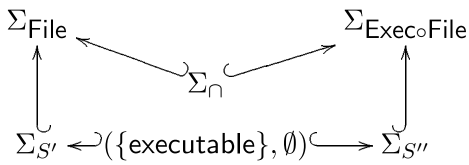

Figure 4.14: Composition of Signatures

The diagram on specifications is obtained by substituting the signatures by their corresponding specifications and noting that the inclusions are Spec morphisms.

4.5.3 Cristalization

In the previous construction of specifications several examples of properties have been expressed as defaults since they might be overridden at other points in the specification. This overriding may be introduced by different specification operations. However, when all of those possible conflicts are explicitly or implicitly written down it may be convenient to compute them into axioms.

An example of this situation is the specification of the locality condition for actions. At some point of the specification process the actions that actually change each attribute will be apparent and we will want to automatically produce the corresponding axiom.

For example the attribute $\mathsf{read{\circ}all}$ can only be changed by the actions from File. But in fact the only that change it are $\mathsf{chmod{\circ}R{\circ}all}(u)$, $\mathsf{chmod{\circ}not{\circ}R{\circ}all}(u)$ and the creation of either of a file or of an executable file. In this way the formula $\mathbf{Y}\mathsf{read{\circ}all} \not\Leftrightarrow \mathsf{read{\circ}all} \Rightarrow (\triangledown\mathsf{exec{\circ}file{\circ}birth}(u) \vee \triangledown\mathsf{file{\circ}birth}(u) \vee \triangledown\mathsf{chmod{\circ}R{\circ}all}(u) \vee \triangledown\mathsf{chmod{\circ}not{\circ}R{\circ}all}(u))$ is a skeptical consequence of the new specification of File and may be added as an axiom to it. Similar axioms may, at this stage, be introduced for the other attributes.

4.5.4 Communication by Action Sharing

The class User corresponds to the interface actions available to an arbitrary user. The specification of this class consists of the locality and sequentiality axioms, plus the global conditions regulating creation and destruction and the relations between the parameterless actions with their respective actions with parameters. The intended behavior of an arbitrary user is obtained by action sharing. This means that the actions of an user are identified with corresponding actions of an arbitrary file. The enabling conditions have been specified in the template of File. In this way each file controls the interaction with an arbitrary user and no further conditions need to be specified.

The interaction of an arbitrary user with an arbitrary file corresponds to the identification of corresponding actions from User and File. This communication mechanism is known as *(action) sharing*. Its categorial expression is simply the (signature) identification of the actions participating in the action sharing. I.e. the actions that participate in sharing are simply the same action in the result specification. The sharing of the actions $\mathsf{chmod{\circ}R{\circ}all}(f)$ from User and $\mathsf{chmod{\circ}R{\circ}all}(u)$ from File is depicted in the following figure.

$$\mathsf{chmod{\circ}R{\circ}all}(u) \in \Sigma_{\mathsf{File}} \;\longleftarrow\!\!\dashv\; \mathsf{chmod{\circ}R{\circ}all} \;\vdash\!\!\longrightarrow\; \mathsf{chmod{\circ}R{\circ}all}(f) \in \Sigma_{\mathsf{User}}$$

Figure 4.15: Action Sharing

The parameterless actions are not identified. The user action $\mathsf{chmod{\circ}R{\circ}all}()$ corresponds to this user changing the permissions of several files and the $\mathsf{chmod{\circ}R{\circ}all}()$ action from a file corresponds to its permissions being changed by several users.

The specification User||File of an arbitrary user communicating with an arbitrary file is obtained by the colimit of the following diagram. We are again assuming that these

$$\mathsf{File} \hookleftarrow \Sigma_{\mathsf{share}} \hookrightarrow \mathsf{User}$$

Figure 4.16: User||File

specifications are now flat. The specification Σ_{share} has no axioms and no defaults and consists of the actions that must be shared in the specifications User and File. The resulting specification will have, *inter alia*:

- the attributes $f.\mathsf{exists}$ and $u.\mathsf{exists}$, two copies of the predicate exists, one of the arbitrary user and the other of the arbitrary file;
- the shared actions $(f||u).\mathsf{file{\circ}birth}$ corresponding to $f.\mathsf{file{\circ}birth}(u)||u.\mathsf{file{\circ}birth}(f)$ (user u creating file f) and $(f||u).\mathsf{exec{\circ}file{\circ}birth}$ corresponding to $f.\mathsf{exec{\circ}file{\circ}birth}(u)||u.\mathsf{exec{\circ}file{\circ}birth}(f)$ (user u creating executable file f);
- and the non shared actions $u.\mathsf{user{\circ}birth}$, $u.\mathsf{user{\circ}death}$, the creation and destruction of an user and the parameterless actions from both class. For example

f.file∘birth() that corresponds (see below in section 4.5.7) to several users creating the file.

The axioms and defaults from User and File belong to the specification User||File. The enabling conditions and effects of the shared actions are now the union of the corresponding conditions for the actions participating in the sharing.

4.5.5 Multiplying Files and Users

The behavior of an actual file will be the result of its interaction with all users. The diagram displaying the signature relations is the following:

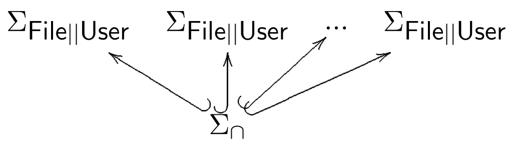

Figure 4.17: Signature of several users interacting with the same file

The signature $\Sigma_\cap$ consists of the symbols of the File||User that are common to the interaction of this file with other users. These are all attributes from File except owner(u) and all parameterless actions from File. In this way we have one f.exists attribute from File and several copies of f.owner(u). The later can be referred to as f.owner(u_1), ..., f.owner(u_n), one for each User in the diagram. The symbols from User have not been identified. In particular there will be several copies of the user exists predicate. These may be referred to by u_1.exists, ..., u_n.exists. The action $(f||u)$.file∘birth is the sharing of f.file∘birth(u) from File with the action u.file∘birth(f) from User. In the resulting specification there will be several of these actions, namely $(f||u_1)$.file∘birth, ..., $(f||u_n)$.file∘birth, representing the different users creating the same file. Finally there is one only action f.file∘birth() (due to identification). This corresponds to several users creating the file. The diagram yielding the intended composition (of a file with several users) is the following. The specification $\Sigma_\cap$ consists only of the signature and an

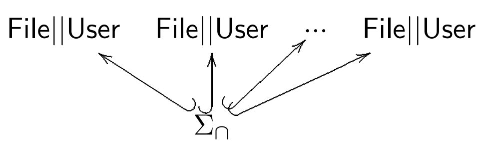

Figure 4.18: Several users interacting with the same file

empty set of axioms and defaults. The axioms and defaults from each User||File belong to the final specification.

4.5.6 Concurrency

The conditions regulating the concurrent behavior of users can only be included at this stage. Although the interaction of an user with a file is sequential (see section 4.4.3) this is no longer the case when several users interact with the same file. In particular there is, up to now, no restriction on the simultaneous occurrence of any of the actions from the file. For example, different users may create the same file simultaneously. This leads to several alternative possibilities of the values of the attributes from the file, after creation since each such creation imposes different values to some of the attributes. If the effects of the $\mathsf{file{\circ}birth}(u)$ action had been stated as axioms this would be an inconsistent situation. But still we prefer to impose the condition that only one of such actions may occur at the same time. This corresponds to add the axiom $\triangledown(f||u).\mathsf{file{\circ}birth} \Leftrightarrow \neg\triangledown(f||u').\mathsf{file{\circ}birth}$, for each two different actions $(f||u).\mathsf{file{\circ}birth}$ and $(f||u').\mathsf{file{\circ}birth}$. Similar conditions for other actions should be also stated.

4.5.7 Cristalization Again

The properties that may be overridden by composition may only be transformed into axioms at this point in the specification. We illustrate the enabling conditions of $f.\mathsf{file{\circ}birth}()$, the parameterless action whose occurrence represents the occurrence of some of the actions $f.\mathsf{file{\circ}birth}(u)$. These enabling conditions have been defined in section 4.4.4 in terms of the enabling conditions of $f.\mathsf{file{\circ}birth}(u)$. The specification File is taken here in its flat version. The axioms and (flattened) defaults from File corresponding to enabling of $f.\mathsf{file{\circ}birth}()$ and $f.\mathsf{file{\circ}birth}(u)$ are the following:

$$\begin{array}{rl} \text{Defaults:} & \{\Diamond f.\mathsf{file{\circ}birth}(u), \\ & \Diamond f.\mathsf{file{\circ}birth}() \Rightarrow \Diamond f.\mathsf{file{\circ}birth}(u)\}; \\ \text{Axioms:} & \{\Diamond f.\mathsf{file{\circ}birth}(u) \Rightarrow \Diamond f.\mathsf{file{\circ}birth}(), \\ & \Diamond f.\mathsf{file{\circ}birth}(u) \Rightarrow \neg f.\mathsf{exists}\}. \end{array}$$

In User||File the action $f.\mathsf{file{\circ}birth}(u)$ is identified with $u.\mathsf{file{\circ}birth}(f)$. The previous conditions are rewritten by changing $f.\mathsf{file{\circ}birth}(u)$ to $(f||u).\mathsf{file{\circ}birth}(u)$. Finally, when several users are introduced the previous axioms and defaults are copied into the final specification. The relevant axioms and defaults become:

Defaults:
$\{\Diamond(f||u_1).\mathsf{file{\circ}birth}, ..., \Diamond(f||u_n).\mathsf{file{\circ}birth},$
$\Diamond f.\mathsf{file{\circ}birth}() \Rightarrow \Diamond(f||u_1).\mathsf{file{\circ}birth}, ..., \Diamond f.\mathsf{file{\circ}birth}() \Rightarrow \Diamond(f||u_n)\mathsf{file{\circ}birth}\};$

Axioms:
$\{\Diamond(f||u_1).\mathsf{file{\circ}birth} \Rightarrow \Diamond f.\mathsf{file{\circ}birth}(), ..., \Diamond(f||u_n)\mathsf{file{\circ}birth} \Rightarrow \Diamond f.\mathsf{file{\circ}birth}(),$
$\Diamond(f||u_1).\mathsf{file{\circ}birth} \Rightarrow \neg f.\mathsf{exists}, ..., \Diamond(f||u_n).\mathsf{file{\circ}birth} \Rightarrow \neg f.\mathsf{exists}\}.$

Since no more conditions on the enabling of these actions will be introduced these may be rewritten with axioms. That is the "best" models of the previous part of the specification will be the only models. The corresponding formulas may be automatically generated (see [10] and section 4.1.2). We note that each action $(f||u).\mathsf{file{\circ}birth}$ is enabled unless explicitly stated otherwise (by the axiom $\Diamond(f||u).\mathsf{file{\circ}birth} \Rightarrow \neg f.\mathsf{exists}$). Therefore $(f||u).\mathsf{file{\circ}birth}$ is enabled iff $\neg f.\mathsf{exists}$. And $\Diamond(f||u).\mathsf{file{\circ}birth} \Leftrightarrow \neg f.\mathsf{exists}$ is a skeptical consequence of the specification. Moreover also $\Diamond f.\mathsf{file{\circ}birth}() \Leftrightarrow \neg f.\mathsf{exists}$is a skeptical consequence of the specification. The previous axioms and defaults can be substituted by the axioms:

$$\{\Diamond(f||u_1).\mathsf{file{\circ}birth} \Leftrightarrow \neg f.\mathsf{exists},$$
$$\ldots$$
$$\Diamond(f||u_n).\mathsf{file{\circ}birth} \Leftrightarrow \neg f.\mathsf{exists},$$
$$\Diamond f.\mathsf{file{\circ}birth}() \Leftrightarrow \neg f.\mathsf{exists}\}$$

4.6 Animating the Specification

The previous (cristalized) specification has as models all such (anchored) life-cycles that respect the properties imposed. These life-cycles correspond to all potential behavior of a system having object oriented files and users. The specification itself may be *animated* in order to simulate such a system. The system evolves by the occurrence of (enabled) actions that cause global time to increase. The animator includes a global clock and consists of adding to the specification the formulas stating which actions just occurred (if enabled). This is formalized by axioms of the form $\mathbf{Y}^n* \Rightarrow (\Diamond a \Rightarrow \nabla a)$ (note that several actions may occur simultaneously). The formula $\mathbf{Y}^n*$ holds at time n after the birth of the system.

The models of the (evolving) specification include the life-cycles satisfying the occurrence of the explicitly stated actions. Among these life-cycles there will be some where more than the explicitly referred actions have occurred. The occurrence of extra actions may either be a consequence of the occurrence of the explicitly stated actions (if involved in communication with the former) or not. This last situation is unintended: the animator should represent the system resulting from the occurrence of the explicitly referred actions and no other.

This is a form of the closed world assumption for occurrence of actions since it states that no action occurs unless explicitly stated otherwise. As expected this situation may be formalized by additional inclusion of the *defaults*:

$$\mathbf{Y}^n* \Rightarrow (\neg\nabla a'),$$

for all actions a'.

The animator just outlined corresponds to a system having a well defined past, characterized by the actions that have occurred. The future, however, remains open.

4.7 Final Remarks

The specification of files and users presented in this chapter illustrates the use of hierarchic specifications and their operations in specification.

The underlying logic used to describe properties of object oriented systems has been a linear temporal logic ("a la" OSL [85]). Some such properties have been expressed with the help of prioritized defaults.

Specification constructs such as inheritance, aggregation and communication have been described by canonical operations on hierarchic specifications.

The need to simplify structure has been noted and two operations provided that are useful for that purpose: priority abstraction and cristalization.

It is important to note that an actual specification language should allow the specifier to concentrate in the definition of the problem at hand. For that purpose the mechanisms identified as global, such as the locality principle and the frame rule should be built in the (denotation of) the specification language. Also the fundamental composition operations should be identified.

The previous example serves the purpose of illustrating some such constructions. It does not identify all standard features and operations of such a specification language. More research is needed on this subject (see [11, 61] for additional contributions).

Chapter 5

Final Remarks

The main contribution of this work has been the establishment of an institution independent theory of composition of specifications and hierarchic specifications. Composition is formalized on the syntactic and semantic levels by canonical operations on appropriate categories.

The semantical characterization of composition guarantees that it depends only on the meaning of the specifications involved, and not on the particular way these are written. Further independence of representation is obtained by the notion of isomorphism. Composition does not depend on the choice of signature symbols and priority level names (since categorial constructions are defined up to isomorphism).

There are other advantages of the semantical characterization. Firstly it is often simpler to define constructions on the semantic level. These can be translated to their syntactic expression due to the connection between the semantic and syntactic categories involved.

Another, and perhaps more important advantage is to use such semantics as the basis of verification tools, proving important properties of specifications, such as the different types of consequence. The priority structure provides extra information. For instance one may be interested in knowing in which level a property is overridden. A brief exposition of our work in this direction is presented in section 5.2.

The theory of composition is summarized in section 5.1. Future research directions, either identifying new constructions and properties of the formalism presented, or extending it are presented in section 5.2.

5.1 Summary

Many different logics are relevant for knowledge representation, depending on the different properties to be modeled. The most important are certainly first order logic and propositional logic, used to formalize databases and knowledge bases. Evolving systems require dynamic, temporal and action logics ([65, 57, 35, 66, 32, 80, 85]). Distributed systems and systems with autonomous agents may require multi-modal deontic logics and multi-modal logics of belief and knowledge ([55, 68]).

The need to override or revise properties expressed in such logics ([74, 12, 9, 11, 77, 61]) requires that a defeasibility mechanism is added to them.

The first concern of this thesis has been, following [12, 9], to provide to an arbitrary (classical) logic such a defeasibility mechanism. This is formalized on the syntactic level by assigning to the formulas of the underlying institution different levels of reliability (different priority levels). On the semantic side the models of the underlying institution are organized in preference relations or hierarchies of preference relations.

Languages for the specification of such systems (using the added expressive power) should provide compositional constructs for supporting the modular construction of specifications from smaller specifications ([16, 45, 34, 37, 87, 26, 31, 36, 21, 82]).

This has been the second (and more important) concern of this thesis. An institution independent theory of composition of specifications and hierarchic specifications has been defined. It generalizes the classical theory of composition of presentations, put forward in [45] and gives a semantic account of the composition of hierarchic specifications from [12]. Composition of specifications is understood as building specifications by putting together axioms, defaults and priority levels from the parameter specifications. Composition of specifications is formalized by appropriate categories both on the syntactic and on the semantic levels. In this way composition is not sensitive to the particular way specifications are written, only to their meaning. The main concepts and results associated with this theory are presented below.

Composition operations augment the complexity of the priority structure of the specification. The opposite direction, that of simplifying priority structure has also been considered and a corresponding operation defined. A further operation, associating with a specification a classical presentation having the same skeptical consequences has been defined. It is important at later stages in the specification process, when all overriding interactions due to incomplete specification information can be computed.

The appropriateness of this formalism has been tested in chapter 4. In the same line of [87, 12, 9, 31, 21, 11, 61, 82] the constructions of inheritance, aggregation and communication of objects are formalized by colimits in the categories of specifications and hierarchic specifications (from linear temporal logic).

Program

The steps taken to establish the institution independent theory of composition are summarized in the following. These correspond to the definition of a semantics that interprets composition; the characterization of the syntactical counterpart of the semantics, the theories; the correspondence between the syntactical relation of inclusion with its semantic counterpart; and finally the establishment of the categories of (hierarchic) specifications and of their semantics, possessing appropriate constructions.

Semantics

Crucial to the theory of composition is the notion of semantics of a specification written in an arbitrary institution. This notion abstracts from the particular way specifications are written: specifications with the same semantics have the same logical meaning. Moreover, specifications with the same semantics behave equivalently w.r.t. composition.

Theorems 169 and 196 show that the adopted semantics of specifications and hierarchic specifications satisfy these requirements *minimally*.

The preference semantics of specifications is standard. But composition of hierarchic specifications requires more structure. This extra structure is expressed in the hierarchy of differential preferences (definition 85) or, equivalently, in the hierarchy of lexicographic preferences (definition 83).

Theories

The theory of a (hierarchic) specification S is a canonical (hierarchic) specification with the same semantics as S. It is the (hierarchic) specification with the same semantics as S with most axioms and defaults: see lemmas 33 for specifications and 114 for hierarchic specifications.

The theory of a specifications S has as axioms the consequences of the axioms of S and as defaults the defaults implicit in S. The situation for hierarchic specifications is similar. In this case defaults implicit in each priority levels must be considered. The characterization of implicit defaults has been presented in theorems 51 and 123.

Galois Connection

Inclusion of (hierarchic) specifications has an abstract expression in reversed inclusion of their semantics, or equivalently in inclusion of their theories. This property is presented in theorem 6 for specifications and in theorem 111 for hierarchic specifications. It implies that the union of (hierarchic) specifications is mirrored at the semantic level by intersection of their semantics. These are the basic requirements of the theory of composition.

Compositionality

Composition of specifications and hierarchic specifications is formalized by colimits of appropriate diagrams in the categories Spec and $\mathsf{hieSpec}$ of specifications and hierarchic specifications. Existence of such constructions is displayed in theorems 159 and 188.

The composition operations are interpreted as semantic operations in the mirror categories Pref and $\mathsf{hiePref}$ (by corresponding limits). This correspondence between syntactic and semantic operations is stated in theorems 155 and 185.

5.2 Further Work

In this section some important topics of future research are presented. These are: the generalization of the framework to Default Logic; the generalization of composition to specifications using *default rules* and the study and definition of further composition constructions. Furthermore, we refer an uniform language for describing and *verifying* properties of specifications and discuss the design of specification languages.

Other Default Formalisms

Composition of specifications should be generalized to Default Logic (and possibly to similar formalisms such as Cumulative Default Logic ([13]) and Łukaszewicz's version of Default Logic ([62])).

The research direction leading to such formalisms should follow the steps referred above. The starting point, that of finding a semantics that explains composition, is the most important. The semantics of Default Logic presented in [29] has formal similarities with the preference semantics of specifications. It compares classes of models, (instead of models) of the axioms of the specification. This formal similarity should be helpful in generalizing this framework to Default Logic.

Instantiation Mechanisms

The framework presented can be extended by including default rules (also called default schemas) as specification instruments. Default schemas correspond to sets of defaults. For example the open default $\mathtt{Fl}(\mathtt{x})$ should be seen as the set of its instances $\{\mathtt{Fl}(\mathtt{b}), \mathtt{b} \in \mathtt{B}\}$. Similar machinery is needed for other formalisms ([9, 12, 75, 76, 11]).

In [24] we formalize default rules by means of *instantiation mechanisms* that assign to each formula the corresponding set of its instances. Composition of specifications using default rules can be reduced to composition of their instantiated versions. The generalization to hierarchic specifications is expected to be straightforward.

More Canonical Operations

The categories hieSpec and hiePref formalize the composition operations that add more priority structure to the parameter specifications. We have seen in chapter 4 that the opposite direction, that of simplifying the structure of a specification is also important. It is interesting to seek a formalism that is able to explain such operations in a canonical way.

A promising proposal is to relate hierarchic specifications via their hierarchies of *lexicographic* preferences (not differential) in a way formally identical with the category hiePref. Since the notion of semantics is the same (up to equivalence) there will be no difference in the notions of isomorphism and theory. The difference will be in the way hierarchic specifications become related: by inclusion of the hierarchies of lexicographic preferences. In particular, in this setting, the operations flat(S) and `Axiomatize(S)`, abstracting a hierarchic specification S to a specification or to a presentation make "canonical" sense. There is a morphism from S to either flat(S) or `Axiomatize(S)`.

Composition and Monotonicity

Excluding the classical projection of the formalism presented no logical property is, in general, preserved by composition. In fact any skeptical or credulous consequence of a (hierarchic) specification can be defeated by adding to it an appropriate axiom[1].

However, preservation of logical properties is important since it allows to conclude a property of a complex system from a property of a simpler part of it.

We have discussed in section 3.2.8 the possibility of imposing some of these properties by appropriate (new) composition forms. An extreme example of this is to identify the parameter specifications with their skeptical consequences (see the operation `Axiomatize(S)` in section 199) and compose them classically. This corresponds to decide that these specifications are "stable" and cannot be subject to further overriding. Another form of composition, preserving the skeptical consequences but still allowing for future overriding is defined in [25]. Further identification of such forms of composition and their impact on specification must be proceeded.

A second direction is the identification of conditions guaranteeing preservation of (some) properties. For example it is expected that the overriding of properties in a localized part of the specification cannot affect the properties of unrelated parts of it.

Moreover, the addition of new information to a specification does not necessarily imply overriding of logical properties. Cumulativity (theorems 67 and 140) and Semimonotonicity (theorems 67 and 142) are examples of this fact (for compact institutions). But more research is needed to recognize other important cases.

Parametric Specifications

Feature orientation emerged recently as an alternative specification paradigm ([54]). It consists in centering specification in the features the entities to be specified must make available. The specification of new entities may reuse other by adding to it new features or revising other.

Features are formalized by parametric specifications (in the classical context: see [53]). The expressive power added by the use of parametric specifications with defaults and its appropriateness to model feature orientation is an important direction of future research.

[1]If the logic has some way of representing the negation of the original consequence.

Meta Language and Verification

In [23] we presented a meta-language that uniformly describes properties of a specification (written in propositional logic). These include the properties of d being a implicit default of S and classical, skeptical and credulous consequence of S. The meta-language is the language of modal logic describing the properties of the preference relation of S, seen as a modal ($S4$) frame. The previous properties hold for a formula f iff the corresponding meta-formula holds in the preference relation of S.

A procedure to determine truth of such formulas in the preference relation of S in terms of consistency and entailment of the axioms of S has also been presented. (Decidability of deciding truth of this formulas can only be guaranteed if the underlying logic is decidable.)

The generalization of the meta-language and of this procedure to arbitrary institutions is expected to be straightforward.

The extension of the meta-language to describe properties of hierarchic specifications should use a multi-modal logic, where the modal formulas are interpreted in the hierarchy of lexicographic preferences. Each lexicographic preference from each level h should have a corresponding modal operator $\Box_h$. In this way the properties of implicit default and credulous and skeptical consequence in a level h from S would be expressed by truth of meta-formulas in the hierarchy of lexicographic preferences of S. The meta-language and corresponding procedure for truth of meta-formulas for the hierarchic case are topics of future research.

Specification Languages

The framework presented provides the formalization of specification languages (with composition constructs) that use defaults. One long term goal of this work is to design a specification language, for communities of concurrent objects, in the style of Gnome ([71]). There are two main concerns when defining the primitives of such a language.

- **Identification and definition of built-in logical constraints**. These are conditions, such as locality and sequentiality of atomic objects that every specification must satisfy. The specifier(s) should concentrate on the particular Universe of Discourse he/she is describing: the global conditions and general specification patterns should be provided by the specification language. These are formalized by axioms and prioritized defaults.

- **Identification and definition of composition operations.** These are operations such as aggregation, inheritance with selective overriding, communication and parameterization that build specifications from other specification modules. These are formalized by colimits either in the category Spec or hieSpec. Additionally the abstraction operations of hierarchic specifications to specifications or to classical presentations should be provided.

Moreover, the specification facilities described should be organized in a workbench, providing the verification of properties of the specification. This corresponds to the computation of the truth of meta-formulas (referred above).

The automatic generation of the relevant specification constructs should also be provided. This corresponds to the computation of colimits and may be implemented with techniques similar to those presented in [79].

Bibliography

[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] J. Barwise and S. Feferman, editors. *Model Theoretic Logics.* Springer, New York, 1985.

[4] J. L. Bell and M. Machover. *A Course in Mathematical Logic.* North-Holland, 1977.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] S. Braß and U. W. Lipeck. Bottom-up query evaluation with partially ordered defaults. In Stefano Ceri, Katsumi Tanaka, and Shalom Tsur, editors, *Deductive and Object-Oriented Databases, Third Int. Conf., (DOOD'93)*, number 760 in LNCS, pages 253–266. Springer, 1993.

[11] 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.

[12] 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.

[13] G. Brewka. Cumulative default logic: In defense of nonmonotonic inference rules. *Artificial Intelligence*, 50:183–205, 1991.

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

[15] R. Bull and K. Segerberg. Basic modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, chapter 1, pages 1–88. Kluver Academic Publishers, 1984.

[16] 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.

[17] 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.

[18] R. Carnap. *Meaning and Necessity.* University of Chicago Press, Chicago, Illinois, 1956.

[19] B. F. Chellas. *Modal logic, An Introduction.* Cambridge University Press, 1980.

[20] J. F. Costa and A. Sernadas. Progress assumption in concurrent systems. *Formal Aspects of Computing*, 7(1):18–36, 1995. Available as Research Report since 1993.

[21] J. F. Costa, A. Sernadas, and C. Sernadas. Data encapsulation and modularity: Three views of inheritance. In A. Borzyszkowski and S. Sokolowski, editors, *Mathematical Foundations of Computer Science 93*, pages 382–391. Springer, 1993.

[22] J. F. Costa, A. Sernadas, and C. Sernadas. Object inheritance beyond subtyping. *Acta Informatica*, 31:5–26, 1994.

[23] F. M. Dionísio, S. Braß, M. Ryan, and U. W. Lipeck. Hypothetical reasoning with defaults. Research report, Institut fur Informatik, University of Hannover, Lange Laube 22, Hannover, Germany, 1994.

[24] F. M. Dionísio, U. W. Lipeck, and S. Braß. Composition and instantiation mechanisms for default theories. Research report, Institut fur Informatik, University of Hannover, Lange Laube 22, Hannover, Germany, 1995. Presented at the 2nd Dutch-German Workshop on Non-monotonic Reasoning Techniques and their Applications.

[25] 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.

[26] H.-D. Ehrich, J. A. Goguen, and A. Sernadas. A categorial theory of objects as observed processes. In J. de Bakker, W.-P de Roever, and G. Rozenberg, editors, *Foundations of Object-Oriented Languages*, pages 203–228. Springer, 1991.

[27] H.-D. Ehrich and A. Sernadas. Local specification of distributed families of sequential objects. In E. Astesiano, G. Reggio, and A. Tarlecki, editors, *Recent Trends in Data Type Specification*, pages 219–235. Springer, 1995. LNCS 906.

[28] E. Allen Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 997–1072. Elsevier Science Publishers B. V. 1990.

[29] D. Etherington. A Semantics for Default Logic. In *Proceedings IJCAJ*, 1987.

[30] 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.

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

[32] J. Fiadeiro and T. Maibaum. Sometimes 'tomorrow' is 'sometime': action refinement in a temporal logic of objects. In D. Gabbay and H. Ohlbach, editors, *Temporal Logic*, pages 48–66. LNAI 827, Springer Verlag, 1994.

[33] J. Fiadeiro and A. Sernadas. Specification and verification of database dynamics. *Acta Informatica*, 25:625–661, 1988.

[34] J. Fiadeiro and A. Sernadas. Structuring theories on consequence. In D. Sannella and A. Tarlecki, editors, *Recent Trends in Data Type Specification: 5th Workshop on Specification of Abstract Data Types - Selected Papers*, pages 44–72. Springer, 1988.

[35] J. Fiadeiro and A. Sernadas. Logics of modal terms for systems specification. *Journal of Logic and Computation*, 1(2):187–227, 1990.

[36] J. Fiadeiro and A. Sernadas. Structuring theories for knowledge representation. In J. Falcão e Cunha, editor, *Terceira Escola Avançada em Inteligência Artificial*, pages 21–25. APPIA, 1992.

[37] J. Fiadeiro, A. Sernadas, and C. Sernadas. Knowledgebases as structured theories. In K. Nori, editor, *Foundations of Software Technology and Theoretical Computer Science*, pages 469–486. Springer, 1988.

[38] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Synthese Library 169. Reidel, Dordrecht, Netherlands, 1983.

[39] K. Futatsugui, J. A. Goguen, J-P. Jouannaud, and J. Meseguer. Principles of obj2. In Brian Reid, editor, *Proceedings of the 12th ACM Symposium on Principles of Programming Languages*, pages 52–66, New York, 1985. ACM.

[40] D. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In K. Apt, editor, *Logic and Models of Concurrent Systems*. Springer, 1985.

[41] Gallois. *Logic for Computer Science. Foundations of Automatic Theorem Proving.* John Wiley & Sons, 1987.

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

[43] 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.

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

[45] J. A. Goguen and R. M. Burstall. Introducing institutions - abstract model theory for computer science. In E. Clarke and D. Kozen, editors, *Logics of Programs — Workshop Carnegie Mellon University Pittsburgh, 1983*, number 164 in LNCS, pages 221–256, Berlin, 1984. Springer.

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

[47] J. A. Goguen, C. Kirchner, H. Kirchner, A. Mégrelis, and J. Meseguer. An introduction to obj3. In Jean-Pierre Jouannaud and Stephane Kaplan, editors, *Proceedings of the 9th International Conference on Software Engineering*, number 308 in LNCS, pages 258–263, Berlin, 1988. Springer.

[48] J. A. Goguen and J. Meseguer. Eqlog: Equality, types and generic modules for logic programming. In Douglas DeGroot and Gary Lindstrom, editors, *Logic Programming: Functions, Relations, and Equations*, pages 295–363. Prentice Hall, Englewood Cliffs, N.J., 1986.

[49] J. A. Goguen and J. Meseguer. Unifying functional, object-oriented and relational programming, with logical semantics. In Bruce Schriver and Peter Wegner, editors, *Research Directions in Object-Oriented Programming*, pages 417–477. MIT Press, Cambridge, 1987.

[50] R. Goldblatt. *Logics of Time and Computation.* CSLI Lecture Notes Number 7. Center for Studies in Language and Information, Stanford University, 2nd edition, 1992.

[51] R. Goldblatt. *TOPOI - The Categorial Analysis of Logic.* Studies in Logic and the Fondations of Mathematics. North-Holland, 2nd edition, 1992.

[52] B. N. Grosof. Generalizing prioritization. In J. Allen, R. Fikes, and E. Sandwall, editors, *Proc. Second International Conference on Principles of Knowlage Representation and Reasoning (KR'91)*, pages 289–300. Morgan Kaufmann, 1991.

[53] S. Guerra and M. Ryan. Feature-oriented specifications. Research report, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1096 Lisboa, Portugal, 1995.

[54] S. Guerra and M. Ryan. Towards feature-oriented specifications. Research report, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1096 Lisboa, Portugal, 1995.

[55] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[56] A. G. Hamilton. *Logic for Mathematicians.* Cambridge University Press, 1978.

[57] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Phylosophical Logic, Vol II*, page 715 ff. Reidel Publ. Co., Dordrecht, Holland, 1987.

[58] T. Hartmann, J. Kusch, G. Saake, and P. Hartel. Revised Version of the Conceptual Modeling and Design Language TROLL. In R. J. Wieringa and R. Feenstra, editors, *Working papers of the International Workshop on Information Systems - Correctness and Reusability*, pages 89–103. Vrije Universiteit Amsterdam, 1994.

[59] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic.* Methuen, London, 1984.

[60] V. Lifschitz. Some results on circumscription. Technical Report 1019 1019, Stanford University, Dept. of Computer Science, 1984.

[61] U. W. Lipeck and S. Braß. Object-oriented system specification using defaults. In Kai von Luck and Heinz Marburger, editors, *Management and Processing of Complex Data Structures, Third Workshop on Information Systems and Artificial Intelligence*, number 777 in LNCS, pages 22–43. Springer, 1994.

[62] W. Łukaszewicz. Considerations on default logic : an alternative approach. *Computational Intelligence*, 4:1–16, 1988.

[63] W. Łukaszewicz. *Non-monotonic reasoning: formalization of commonsense reasoning.* Ellis Horwood, 1990.

[64] D. Makinson. General theory of cumulative inference. In M. Reinfrank, J. de Kleer, M. L. Ginsberg, and E. Sandewall, editors, *Proceedings of the Second International Workshop on Non-Monotonic Reasoning*, number 346 in LNAI, pages 1–18, Grassau, Germany, 1988. Springer Verlag.

[65] Z. Manna and A. Pnueli. Verification of temporal programs: the temporal framework. In R. S. Boyer and J. S. Moore, editors, *The Correctness Problem in Computer Science*. Academic Press, New York, 1981.

[66] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer Verlag, New York, 1991.

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

[68] J.-J. Ch. Meyer and R. J. Wieringa, editors. *Deontic Logic In Computer Science: Normative System Specification*. Wiley Professional Computing, 1993.

[69] A. Pnueli. The temporal logic of programs. In *Proc. 18th Annual Symposium on Foundations of Computer Science*, pages 45–57. IEEE, 1977.

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

[71] J. Ramos and A. Sernadas. A brief introduction to GNOME. Research report, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1096 Lisboa, Portugal, 1995.

[72] 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.

[73] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[74] 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.

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

[76] 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.

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

[78] M. Ryan and P.-Y. Schobbens. The lexicographic combination of preferences. In *Proc. Dutch/German Workshop on Non-monotonic Reasoning and its Application*, Aachen, Germany, 1993.

[79] D. Rydeheard and R. M. Burstall. *Computational Category Theory.* Prentice Hall, 1988.

[80] 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.

[81] P.-Y. Schobbens. *Exceptions in Algebraic Specifications.* PhD thesis, Université Catholique de Louvain, Faculté des Sciences Appliquées, 1992.

[82] 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.

[83] 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.

[84] 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.

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

[86] A. Sernadas, C. Sernadas, and H.-D. Ehrich. Object-oriented specification of databases: An algebraic approach. In P. Hammerslay, editor, *Proc. 13th International Conference on Very Large Databases (VLDB'87)*, pages 107–116, Palo Alto, 1987. Morgan-Kaufmann.

[87] C. Sernadas, J. Fiadeiro, and A. Sernadas. Modular construction of logic knowledge bases: An algebraic approach. *Information Systems*, 15(1):37–59, 1990.

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

[89] Y. Shoham. Chronological ignorance: experiments in nonmonotonic temporal reasoning. *Artificial Intelligence*, 36:271–331, 1988.

[90] A. Tarski. *Logic, Semantics, Metamathematics.* Oxford University Press, Oxford, 1956.

[91] G. Winskel. *The Formal Semantics of Programming Languages. An Introduction.* The MIT Press, 1993.

Lebenslauf

Name:	Francisco Miguel Alves Campos de Sousa Dionísio
Geburtsdatum:	8.4.1961
Geburtsort:	Beja, Portugal
Eltern:	Francisco de Sousa Dionísio, Maria Dolores Alves Campos de Sousa Dionísio

Ausbildungsgang:

1967-1976	Schulbesuch
1976-1978	Curso Complementar do Liceu in Liceu Rainha Dona Leonor, Lissabon
1978-1979	Ano Propedêutico in Liceu Gil Vicente, Lissabon (Erwerb der Hochschulzugangsberechtigung)
1979-1985	Licenciatura em Engenharia Electrotécnica in Universidade Técnica de Lisboa (Ingenieur-Studium der Elektrotechnik an der Technischen Universität Lissabon)
1986-1987	Grundwehrdienst
1987-1991	Mestrado em Matemática Aplicada in Universidade Técnica de Lisboa (Master-Studium der Elektrotechnik an der Technischen Universität Lissabon)
seit 1992	Promotion an der Universität Hannover

Berufstätigkeit: wissenschaftlicher Mitarbeiter an der Technischen Universität Lissabon

Lehrtätigkeit an der Abteilung Mathematik der Technischen Universität Lissabon[2]

Von 1987 bis 1989:

"Mathematische Analysis III" (3Ü) und "Mathematische Analysis III" (2V+3Ü) in der Sektion Algebra und Analysis.

Von 1989 bis 1991:

"Programmierung", "Syntax und Semantik von Sprachen", "Formale Spezifikationen" und "Algebraische Elemente der Programmierung" in der Sektion Informatik.

Seit 1995:

"Programmierung", "Algebraische Elemente der Programmierung" und "Übersetzerbau" in der Sektion Informatik.

Andere

Juli 1989:

"Syntax und Semantik von Sprachen" und "Algebraische und Logische Grundlagen der Informatik" (Intensivkurse) an der Universität Agostinho Neto, in Luanda, Angola.

[2] Universidade Técnica de Lisboa

Keywords

- - Defaults
  - Specification
  - Composition
- - Defaults
  - Spezifikation
  - Komposition