Towards a Uniform Presentation of Logical Systems by Indexed Categories and Adjoint Situations

U. Wolter: Department of Informatics, University of Bergen - Norway
E-mail: Uwe.Wolter@ii.uib.no

A. Martini†: Faculdade de Informática, PUCRS - Brasil
E-mail: alfio.martini@pucrs.br

E. H. Häusler‡: Departamento de Ciência da Computação, PUC-Rio - Brasil
E-mail: hermann@inf.puc-rio

Abstract

Logical Systems are paramount to almost every subject in computer science. This vast number of application areas had a deep influence on us and thus on how we perceive what a formal specification of a logical system should be. Lawvere’s [29, 30] essential idea is that the fundamental relationship between syntax and semantics can be precisely formulated by adjoint functors. In this work, we show that Institutions from Goguen and Burstall [19] and Entailment Systems from Meseguer [36] are in its essence, a family of local adjoint situations between the syntactical and semantical aspects underlying these systems. These abstractions are named Indexed Frames. Also, from a categorical perspective, Tarski’s consequence operator [40] can be formalized as Indexed Closure Operators, a construction that maps each language to the corresponding co-monad. Finally, in the framework of preorder categories, both concepts, Indexed Frames and Indexed Closure Operators are equivalent.

Keywords: general logics, adjoint situations, indexed categories, galois connections, co-monads.

1 Introduction

There has been many influential works proposing a definition or presentation for what a logical system really is, like for instance, [37, 17, 19, 36, 1, 9, 16]. Especially, when considering logics for specification and programming, many of these approaches are either based or strongly related to the notion of an Institution [19]. This concept carries in its essence the linguistic aspects of a logic, namely, it considers a category of signatures and signature morphisms as a building block of the very concept of a logical system. Besides, it also draws attention to the fact that the language of indexed (and fibred) categories is a very powerful tool around which we can organize in a modular way many aspects of a logic [35, 41, 22, 9, 39, 8].

†Correspondent Author.
‡Partially supported by the Brazilian Research Council (CNPq) (grant No. 303427/2010-8)
On the other hand, it is well-known from category theory that adjoint situations arise everywhere [31, 3]. Especially, the seminal work of Lawvere [30, 29] stresses that adjoint functors capture in a precise and formal way the very nature of the Formal-Conceptual Duality that is ubiquitous in mathematics and logic. More precisely in [30], he states

"That pursuit of exact knowledge which we call mathematics seems to involve in an essential way two dual aspects, which we may call the Formal and the Conceptual. For example, we manipulate algebraically a polynomial equation and visualize geometrically the corresponding curve. Or we concentrate in one moment on the deduction of theorems from the axioms of group theory, and in the next consider the classes of actual groups to which the theorems refer. Thus the Conceptual is in a certain sense the subject matter of the Formal".

In the context of a logical system, the Formal is represented usually by a formal language Syn, i.e., a set of sentences, while the Conceptual (denotation) Den is often defined according to one of the following traditional views:

- a model theoretic approach, that takes the binary satisfaction relation between models and sentences as basic;
- a proof-theoretic approach, that requires a binary entailment relation between sets of sentences;
- a Tarskian-approach (if you like), that assumes a consequence or closure operator on a given set of sentences.

In the first case, one can immediately define a partial ordering on Syn and Den so that a Galois connection can be established between both structures and hence the mappings \( \text{sem} : \text{Syn} \rightarrow \text{Den} \) and \( \text{fml} : \text{Den} \rightarrow \text{Syn} \) become adjoint functors between the corresponding partial orders viewed as categories, where \( \text{fml} \) (formulas) is left-adjoint to \( \text{sem} \) (semantics), \( \text{fml} \dashv \text{sem} \):

\[
\begin{array}{ccc}
\text{Syn} & \xleftarrow{\text{fml}} & \text{Den} \\
& \text{sem} \downarrow &
\end{array}
\]

The arrows in Den are inclusion of subcategories of models, while in Syn, they are (opposite) inclusions of specifications (theory presentations). The mappings (functors) \( \text{sem} \) and \( \text{fml} \) are defined by assignment of models with respect to specifications, respectively, theories (closed set of sentences) induced by a certain class of models. These functions are actually compatible in a natural way with respect to change of languages (or syntactic vocabulary) so that both \( \text{sem} \) and \( \text{fml} \) become indexed functors. This crucial observation leads to the notion of Indexed Frames. By a chain of stepwise constructions and lemmas we will see that in the case of Institutions, the satisfaction condition is both necessary and sufficient to impose this naturality. Thus, at the level of specifications and corresponding subcategories of models, an Institution defines exactly an indexed frame, where each index refers to a partial order category. Hence they are called Indexed Partial Order Frames.

\[\text{This remark from Lawvere was also pointed out in [36].}\]

\[\text{The first two are already mentioned in [36].}\]
In the proof-theoretic approach, the arrows in $\text{Den}$ are entailment relations between specifications, (so that $\text{Den}$ is actually a preorder), while the arrow in $\text{Syn}$ are reverse inclusions of specifications. The axioms of $\text{Entailment Systems}$ described in [36] ensure again a naturality condition so that $\text{sem}$ and $\text{fml}$ are also indexed functors. Here $\text{sem}$ is just an inclusion, while $\text{fml}$ maps each specification to its corresponding closed set with respect to entailment. Thus, at the level of specifications and corresponding entailments between set of formulas, an Entailment System defines again an Indexed Frame, where each index refers to a preorder category. Thus here, they are called $\text{Indexed Preorder Frames}$.

In the Tarskian-style [40] we have the following situation: a closure (or consequent) operator from the point of view of category theory is simply a $\text{co-monad}$. Here closure operators essentially maps specifications to their corresponding closed set of sentences. Moreover, if we take into account change of languages, we are naturally led to the notion of $\text{Indexed Closure Operator}$, which associates each language (index) to the corresponding monad. However, there is a caveat: this closure is no longer compatible in a natural way with respect to change of syntax. Fortunately, the categorical notion of $\text{lax natural transformation}$ fixes in a precise way how closures and translation of languages behave with respect to each other.

Furthermore, both categorical structures, indexed frames and indexed closure operators are in some sense equivalent in the framework of preorder categories. Via canonical factorization of co-monads into adjoints one can build an Indexed Frame carrying the same information. Likewise, local adjointness entails lax naturality, and hence that each indexed frame defines a corresponding Indexed Closure Operator.

This paper is organized as follows: in section 3 we present the concept of Institutions [19] and show that at the level of specifications and corresponding (full) subcategories of model classes, the (invariant) institution condition has an equivalent description as a natural transformation. An abstraction of this observation is named an Indexed Frame, a categorical, more general, specification of $\text{Institution-like}$ logics, where each index is associated to a partial order category. In section 4 we present the concept of Entailment Systems [36], and again, at the level of entailment between specifications we observe that this concept is also an instance of the concept of an Indexed Frame, where each index is associated to a preorder category. In section 6 we provide a detailed presentation of linear logic (actually a sublogic of it) as an Indexed Frame. The example is important because it shows a situation where the categories induced by the indexes of a given category $\text{Ind}$ are neither pre- nor partial order categories. Moreover, in section 7, we discuss another important logical system as an Indexed Frame, namely that of generalized sketches. The special situation here is that this logical sytem is defined with fibred rather than indexed semantics, e.g., as in Institutions. Then we show that generalized sketches, even with fibred semantics, also give rise to Indexed Partial Order Frames. Finally, in section 8, we extend the notion of monads and introduce the idea of an Indexed Closure Operator. By exploiting the initial and final solutions to the factorization problem of monads by means of adjoint functors, we are also able to state that Indexed Closure Operators and Indexed Preorder Frames are equivalent categorical formulations and thus essentially the same concept.

2 Notations and Basics

Below we summarize notation and essential background for reading this paper. Most of it can be seen as folklore. It can be omitted (on a first reading) if you are familiar with these
topics.

Consider a set \( A \) and a binary relation \( \leq \) on \( A \). Then \( \leq \) is called a \textit{preorder} on \( A \), written \((A, \leq)\), if we have that, for all \( x, y, z \in A \) that

- \( x \leq x \) (\textit{reflexivity})
- \( x \leq y \) and \( y \leq z \) then \( x \leq z \) (\textit{transitivity})

If the relation \( \leq \) is in addition \textit{antisymmetric}, that is, for all \( x, y \in A \) we have that \( x \leq y \) and \( y \leq x \) implies \( x = y \) then \((A, \leq)\) is also called a \textit{partial order}. For instance, let \( P(A) = \{X \mid X \subseteq A\} \), i.e., the set of all subsets of \( A \), called the \textit{powerset} of \( A \). Then \((P(A), \subseteq)\) and \((P(A), \supseteq)\) are partially order sets (also known as posets) and thus also preorder sets. Given two partially order sets \((A, \leq_1)\) and \((B, \leq_2)\), a \textit{Galois connection} between these posets is comprised by two monotone functions \( F : A \rightarrow B \) and \( G : B \rightarrow A \), such that for all \( a \in A \) and \( b \in B \) we have that

\[
a \leq_1 G(a) \quad \text{if and only if} \quad F(a) \leq_2 b
\]

Most Galois connections arise in the following way. Let \( A \) and \( B \) be sets and \( R \subseteq A \times B \) a binary relation from \( A \) to \( B \). Then for any subset \( X \) on \( A \) we define a subset \( \text{fml}(X) \) of \( B \) by the rule

\[
\text{fml}(X) = \{y \in B \mid \forall x \in X. (x, y) \in R\}
\]

and similarly, for any subset \( Y \) of \( B \), we define a subset \( \text{sem}(Y) \) of \( A \) by

\[
\text{sem}(Y) = \{x \in A \mid \forall y \in Y. (x, y) \in R\}
\]

Thus we have two mappings

\[
\text{fml} : P(A) \rightarrow P(B) \quad \text{and} \quad \text{sem} : P(B) \rightarrow P(A)
\]

such that \( \text{fml} \) and \( \text{sem} \) form a Galois connection, once one of partial order sets the inclusion is interpreted in the opposite ordering, i.e, if the order \( \leq \) is interpreted as the reverse inclusion \( \supseteq \) between sets.

**Definition 2.1.** Given a preorder \((A, \leq)\), a closure operator on \( A \) is a function \( cl : A \rightarrow A \) with the following properties:

- \( x \leq cl(x) \) for all \( x \in A \), i.e. \( cl \) is extensive
- for all \( x, y \in A \), if \( x \leq y \), then \( cl(x) \leq cl(y) \), i.e. \( cl \) is monotonically increasing
- \( cl(cl(x)) \leq cl(x) \) for all \( x \in A \), i.e. \( cl \) is an idempotent function.

**Fact 2.2.** Let \( A \) and \( B \) be sets and \( K : P(A) \rightarrow P(B), L : P(B) \rightarrow P(A) \) be maps which form a Galois connection. Then:

- The map \( L \circ K \) is a closure operator on \( P(A) \).
- The map \( K \circ L \) is a closure operator on \( P(B) \).
Thus, given $\text{fml} : P(A) \to P(B)$ and $\text{sem} : P(B) \to P(A)$ as above, we have that $\text{fml} ; \text{sem}$ is a closure operator on $P(A)$ and $\text{sem} ; \text{fml}$ is a closure operator on $P(B)$.

Below, we assume the reader is familiar with basic notions of category theory (e.g., [3, 31]). Nevertheless, the essential background is presented below. The collection of objects of a category $C$ will be denoted $|C|$. Given objects $a, b \in |C|$, the set of arrows from $a$ to $b$ is denoted $C(a, b)$. The category $\text{Set}$ is the category of sets and total functions, while $\text{Cat}$ is the category of all categories and functors (lying in a higher set-theoretic universe).

Let $C$ be a category and $A, B$ arbitrary objects in $|C|$. Then $C$ is a preorder category if the cardinality of the collection $C(A, B)$ is at most equal to one. Additionally, $C$ is called a partial order category if for any objects $A$ and $B$ in $C$, the existence of an isomorphism $i : A \to B$ entails $A = B$ and thus $i = \text{id}_A = \text{id}_B$.

We also write $\text{PO}$ to denote the category of all partial orders. We consider here partial orders as special categories thus $\text{PO}$ is taken as a full subcategory of the category $\text{Cat}$ of categories. In the same way, we write $\text{PRE}$ to represent the category of all preorders. Thus $\text{PRE}$ is a full subcategory of $\text{Cat}$ and it follows that $\text{PO}$ is also a full subcategory of $\text{PRE}$.

We represent composition of maps (functors) in diagrammatic order. For instance if $F : A \to B$ and $G : B \to C$ are functors and $a \in |A|$, then $F; G : A \to C$ and $(F; G)(a) \stackrel{\text{def}}{=} G(F(a))$ is an object of $C$, or equivalently, $G(F(a)) \in |C|$.

Also, different institutions, frames, logics, are denoted with primed superscripts (e.g., $T, T', T''$, etc.), while different objects within an institution (frame, logic), as signatures, models, etc., are denoted with numbered subscripts (e.g., $\Sigma_1, \Sigma_2, M_1, M_2$, etc.).

Moreover, if $\alpha : F \Rightarrow G : A \to B$ and $\beta : G \Rightarrow H : A \to B$ are natural transformations, then the vertical composition of $\alpha$ and $\beta$ is denoted $\alpha; \beta : F \Rightarrow H : A \to B$ such that for each $a \in |A|$, $(\alpha; \beta)(a) \stackrel{\text{def}}{=} \alpha(a); \beta(a)$. Also, if $F : A \to B, G, G' : B \to C, H : C \to D$ are functors and $\alpha : G \Rightarrow G' : B \to C$ is a natural transformation, then the horizontal compositions of $F$ with $\alpha$, and $\alpha$ with $H$ are represented as $(F, \alpha) : F; G \Rightarrow F; G' : A \to C$ and $(\alpha, H) : G; H \Rightarrow G'; H : B \to D$ such that for each $b \in |B|, a \in |A|, (F, \alpha)(a) \stackrel{\text{def}}{=} \alpha(F(a))$ whereas $(\alpha, H)(b) \stackrel{\text{def}}{=} H(\alpha(b))$.

We have immediately according to the definition of composition

$$id_B \cdot \alpha = \alpha \quad \text{and} \quad \alpha \cdot id_C = \alpha$$

and since functors map identities to identities we have also

$$F \cdot id_G = id_{F;G} \quad \text{and} \quad id_G \cdot H = id_{G;H}$$

Moreover we have for functors $F : A \to B, G : B \to A, H, H', H'' : C \to D, I : D \to E$ and natural transformations $\gamma : H \Rightarrow H', \delta : H' \Rightarrow H''$ the following laws

$$A \xrightarrow{F} B \xrightarrow{G} C \xrightarrow{H} D \xrightarrow{I} E$$

$$\xrightarrow{\gamma} \quad \xrightarrow{\delta}$$

5
\[(F; G) \cdot \gamma = F \cdot (G \cdot \gamma) \quad \text{and} \quad G \cdot (\gamma; \delta) = (G \cdot \gamma); (G \cdot \delta) \quad (3)\]
\[(G \cdot \gamma) \cdot I = G \cdot (\gamma \cdot I) \quad \text{and} \quad (\gamma; \delta) \cdot I = (\gamma \cdot I); (\delta \cdot I) \quad (4)\]

Let A and B be categories. If \( F : A \to B \) and \( G : B \to A \), we say that \( F \) is left adjoint to \( G \) and \( G \) is right adjoint to \( F \) provided there is a natural transformation \( \eta : id_A \Rightarrow F; G \) such that for any two objects \( a \) of \( A \) and \( b \) of \( B \) and any arrow \( f : a \to G(b) \) there is a unique arrow \( g : F(a) \to b \) that solves the equation

\[f = \eta(a); G(?)\]

It is usual to write \( F \dashv G \) to denote the above situation. The data \((F, G, \eta)\) constitute an adjunction and the transformation \( \eta \) is called the unit of the adjunction. The definition above, implies the following proposition, establishing a crucial symmetry of the above situation:

**Fact 2.3.** Suppose that \( F : A \to B \) and \( G : B \to A \) are functors such that \( F \dashv G \). Then, there is a natural transformation \( \epsilon : G; F \Rightarrow id_B \) such that for any arrow \( g : F(a) \to b \), there is a unique arrow \( f : a \to G(b) \) that solves the equation

\[g = F(?); \epsilon(b)\]

### 3 Institutions and Indexed Frames

The concept of an *Institution* introduced by Goguen and Burstall [19] formally captures the notion of logical systems and allowed them to reformulate and to generalize the work they had done in the 70’s on structuring (equational) specifications (see [4, 5]) independently of the underlying logic. A similar proposal of an abstract concept of a logic had been given already by Barwise [1]. The following definition can be found in [19].

**Definition 3.1 (Institution).** An INSTITUTION \( \mathcal{I} = (\text{Sign}, \text{Sen}, \text{Mod}, \models) \) is given by the following data and operations:

- A category of abstract signatures \( \text{Sign} \);
- A syntax functor \( \text{Sen} : \text{Sign} \to \text{Set} \), defining for each signature its set of sentences;
- A model functor \( \text{Mod} : \text{Sign}^{op} \to \text{Cat} \), defining for each signature, its corresponding category of models;
- An indexed family of satisfaction relations \( \models_{\Sigma} \subseteq |\text{Mod}(\Sigma)| \times \text{Sen}(\Sigma), \Sigma \in |\text{Sign}| \) such that the following Institution condition holds for each \( \phi : \Sigma_1 \to \Sigma_2 \) in \( \text{Sign} \), \( M_2 \in |\text{Mod}(\Sigma_2)| \), and \( \varphi_1 \in \text{Sen}(\Sigma_1) \).

**Remark 3.2 (Component Implications).** It will be essential for us to have a deeper understanding of the above institution condition when we see it from the perspective of the two component implications. Therefore, we label them for future reference.

\[
\begin{align*}
\text{IC}_{\text{Sen} \Rightarrow \text{Mod}} & \quad \text{Mod}(\phi)(M_2) \models_{\Sigma_1} \varphi_1 \iff M_2 \models_{\Sigma_2} \text{Sen}(\phi)(\varphi_1) \\
\text{IC}_{\text{Mod} \Rightarrow \text{Sen}} & \quad \text{Mod}(\phi)(M_2) \models_{\Sigma_1} \varphi_1 \Rightarrow M_2 \models_{\Sigma_2} \text{Sen}(\phi)(\varphi_1)
\end{align*}
\]
Institutions are based on a pointwise assignment of signatures, sentences, and models. But in formal specifications, the relevant objects are not sentences, but sets of sentences (theories). The following investigations on this level of abstraction provide new insights into the conceptual nature of logical systems so that we can give a fully categorical account of the institution condition.

Assuming an institution \( I \), and a set of \( \Sigma \)-sentences \( \Gamma \subseteq \text{Sen}(\Sigma) \), we define the category \( \text{mod}(\Sigma)(\Gamma) \) as the full subcategory induced by those models in \( \text{Mod}(\Sigma) \) that satisfy \( \Gamma \), i.e.,

\[
\text{mod}(\Sigma)(\Gamma) \overset{def}{=} \{ M \in \text{Mod}(\Sigma) | \forall \varphi \in \Gamma : M \models \Sigma \varphi \}.
\]  

(5)

Analogously, we define for a given subcategory \( M \subseteq \text{Mod}(\Sigma) \) of \( \Sigma \)-models, the set of theorems \( \text{th}(\Sigma)(M) \subseteq \text{Sen}(\Sigma) \) given by those sentences \( \varphi \in \text{Sen}(\Sigma) \) which are satisfied by all models in \( M \), i.e., we have

\[
\text{th}(\Sigma)(M) \overset{def}{=} \{ \varphi \in \text{Sen}(\Sigma) | \forall M \in |M| : M \models \Sigma \varphi \}.
\]  

(6)

Obviously, \( \text{mod}(\Sigma) \) and \( \text{th}(\Sigma) \) induce mappings between \( \Sigma \)-specifications and subcategories of \( \Sigma \)-models and vice-versa.

**Definition 3.3** (Partial Order Categories). Given a signature \( \Sigma \in |\text{Sign}| \), we define the partial order category \( \text{Spec}(\Sigma) \) of strict specifications, the category where the objects are all sets \( \Gamma \subseteq \text{Sen}(\Sigma) \), and the arrows are all the inverse inclusions \( \Gamma_1 \supseteq \Gamma_2 \). Analogously, the partial order category \( \text{Sub}(\Sigma) \) has as objects all subcategories \( M \subseteq \text{Mod}(\Sigma) \) and as arrows all inclusion functors \( M_1 \subseteq M_2 \).

By definition of \( \text{Spec}(\Sigma) \) and \( \text{Sub}(\Sigma) \), we can formulate the usual categorical presentation of the Galois correspondence arising from any (satisfaction) relation as an adjunction \( \text{th}(\Sigma) \dashv \text{mod}(\Sigma) \).

**Proposition 3.4** (Galois Correspondence). Given a signature \( \Sigma \in |\text{Sign}| \), the equations (5) and (6) define functors \( \text{mod}(\Sigma) : \text{Spec}(\Sigma) \to \text{Sub}(\Sigma) \) and \( \text{th}(\Sigma) : \text{Sub}(\Sigma) \to \text{Spec}(\Sigma) \), such that \( \text{th}(\Sigma) \) is left-adjoint to \( \text{mod}(\Sigma) \), written \( \text{th}(\Sigma) \dashv \text{mod}(\Sigma) \), where for any \( M, M_1, M_2 \in |\text{Mod}(\Sigma)| \), and \( \Gamma, \Gamma_1, \Gamma_2 \in |\text{Spec}(\Sigma)| \), we have:

1. \( \text{th}(\Sigma) \) functor: \( M_1 \subseteq M_2 \) implies \( \text{th}(\Sigma)(M_1) \supseteq \text{th}(\Sigma)(M_2) \);
2. \( \text{mod}(\Sigma) \) functor: \( \Gamma_2 \supseteq \Gamma_1 \) implies \( \text{mod}(\Sigma)(\Gamma_2) \subseteq \text{mod}(\Sigma)(\Gamma_1) \);
3. unit: \( M \subseteq \text{mod}(\Sigma)(\text{th}(\Sigma)(M)) \);
4. counit: \( \text{th}(\Sigma)(\text{mod}(\Sigma)(\Gamma)) \supseteq \Gamma \);
5. adjointness: \( M \subseteq \text{mod}(\Sigma)(\Gamma) \) iff \( \text{th}(\Sigma)(M) \supseteq \Gamma \).

Proof.

- \( \text{th}(\Sigma) \) functor: Let \( \varphi \in \text{th}(\Sigma)(M_2) \). Then \( \forall M \in |M_2| : M \models \Sigma \varphi \). This implies that \( \forall M \in |M_1| : M \models \Sigma \varphi \), since by assumption \( M_1 \subseteq M_2 \). Thus \( \varphi \in \text{th}(\Sigma)(M_1) \).
• \( mod(\Sigma) \) functor: Let \( M \in |mod(\Sigma)(\Gamma_2)| \). Then \( \forall \varphi \in \Gamma_2 : M \models_\Sigma \varphi \). This implies \( \forall \varphi \in \Gamma_1 : M \models_\Sigma \varphi \), since by assumption \( \Gamma_1 \subseteq \Gamma_2 \). Thus \( M \in |mod(\Sigma)(\Gamma_1)| \). Thus \( |mod(\Sigma)(\Gamma_2)| \subseteq |mod(\Sigma)(\Gamma_1)| \).

• unit: It follows from the adjointness property below.

• counit: It follows from the adjointness property below. adjointness: Note that:

\[
\begin{align*}
M \subseteq mod(\Sigma)(\Gamma) \\
\text{iff} \forall M \in |M| : M \in |mod(\Sigma)(\Gamma)| \\
\text{iff} \forall M \in |M| : \forall \varphi \in \Gamma : M \models_\Sigma \varphi \\
\text{iff} \forall \varphi \in \Gamma : \forall M \in |M| : M \models_\Sigma \varphi \\
\text{iff} \forall \varphi \in \Gamma : \varphi \in th(\Sigma)(M) \\
\text{iff} \Gamma \subseteq th(\Sigma)(M)
\end{align*}
\]

\[\square\]

**Fact 3.5** (Power Functors). For a set \( A \) let \( \mathcal{P}(A) \) be the partial order \((\mathcal{P}(A), \supseteq)\) considered as a category, where \( \mathcal{P}(A) \) is the powerset of \( A \). And for a category \( C \) let \( \mathcal{P}(C) \) be the partial order category over \( C \), where the objects are all subcategories \( M \subseteq C \) and arrows all inclusion functors \( M_1 \subseteq M_2 \). Then we obtain

\( \star \) for any function \( f : A \to B \) the existential image powerset functor \( f : \mathcal{P}(A) \to \mathcal{P}(B) \), by setting \( f(A') = \{ f(a) | a \in A' \} \) for any \( A' \in \mathcal{P}(A) \), and

\( \star \) for any functor \( F : C \to D \) the inverse image power category functor \( F^{-1} : \mathcal{P}(D) \to \mathcal{P}(C) \), where for any \( M \subseteq D : C \in |F^{-1}(M)| \) iff \( F(C) \in M \) and \( g \) is an arrow in \( F^{-1}(M) \) iff \( F(g) \) is an arrow in \( M \).

**Remark 3.6.** Note that if \( A \supseteq A' \) then \( f(A) \supseteq f(A') \) by monotonicity of function application. Preservation of compositionality and identities follow from transitivity of subset inclusion and the fact that there is at most one arrow (inclusion) between any two subsets. Analogously for any two subcategories \( M, M' \) in \( |\mathcal{P}(D)| \), if \( M \subseteq M' \) then \( F^{-1}(M) \subseteq F^{-1}(M') \).

By Fact 3.5, we obtain for any signature morphism \( \phi : \Sigma_1 \to \Sigma_2 \) functors

\[
\begin{align*}
Sen(\phi) : \text{Spec}(\Sigma_1) &\to \text{Spec}(\Sigma_2) & \text{and} & \quad Mod(\phi)^{-1} : \text{Sub}(\Sigma_1) &\to \text{Sub}(\Sigma_2).
\end{align*}
\]

**Lemma 3.7.** A 4-tuple \( \mathcal{I} = (\text{Sign}, \text{Sen}, \text{Mod}, \models) \) satisfies the institution condition for a signature morphism \( \phi : \Sigma_1 \to \Sigma_2 \) in \( \text{Sign} \) iff

\[
|mod(\Sigma_2)(\text{Sen}(\phi)(\Gamma_1))| = |mod(\Sigma_1)(\Gamma_1)|, \quad \text{for all} \quad \Gamma_1 \in |\text{Spec}(\Sigma_1)|.
\]

\[
\begin{array}{ccc}
\Sigma_1 & \xrightarrow{\text{Spec}(\Sigma_1)} & \text{Sub}(\Sigma_1) \\
\phi & \xrightarrow{\text{Sen}(\phi)} & \xrightarrow{\text{Mod}(\phi)^{-1}} \\
\Sigma_2 & \xrightarrow{\text{Spec}(\Sigma_2)} & \text{Sub}(\Sigma_2)
\end{array}
\]

**Proof.** It suffices to show that the institution condition is equivalent to the following inequalities:
1. \( \text{mod}(\Sigma_2)(\text{Sen}(\phi)(\Gamma_1)) \subseteq \text{Mod}(\phi)^{-1}(\text{mod}(\Sigma_1)(\Gamma_1)) \) for all \( \Gamma_1 \in |\text{Spec}(\Sigma_1)| \).

2. \( \text{Mod}(\phi)^{-1}(\text{mod}(\Sigma_1)(\Gamma_1)) \subseteq \text{mod}(\Sigma_2)(\text{Sen}(\phi)(\Gamma_1)) \), for all \( \Gamma_1 \in |\text{Spec}(\Sigma_1)| \).

\( \text{IC}_{\text{Sen} \Rightarrow \text{Mod}} \) implies inequality 1: For any \( M_2 \in |\text{Mod}(\Sigma_2)| \) we obtain

\[
M_2 \in |\text{mod}(\Sigma_2)(\text{Sen}(\phi)(\Gamma_1))| \\
\Leftrightarrow M_2 \models_{\Sigma_2} \text{Sen}(\phi)(\varphi_1) \quad \text{for all} \quad \varphi_1 \in \Gamma_1 \quad \text{(def.} \text{mod}(\Sigma_2)) \\
\Rightarrow \text{Mod}(\phi)(M_2) \models_{\Sigma_1} \varphi_1 \quad \text{for all} \quad \varphi_1 \in \Gamma_1 \quad \text{(IC}_{\text{Sen} \Rightarrow \text{Mod}}) \\
\Leftrightarrow \text{Mod}(\phi)(M_2) \in |\text{mod}(\Sigma_1)(\Gamma_1)| \quad \text{(def.} \text{mod}(\Sigma_1)) \\
\Leftrightarrow M_2 \in |\text{Mod}(\phi)^{-1}(\text{mod}(\Sigma_1)(\Gamma_1))| \quad \text{(def.} \text{Mod}(\phi)^{-1})
\]

Note that, concerning Lemma 3.7.1 and 3.7.2, the reader has to bear in mind that the inverse image of a full subcategory becomes a full subcategory as well.

Inequality 1 implies \( \text{IC}_{\text{Sen} \Rightarrow \text{Mod}} \): For any \( M_2 \in |\text{Mod}(\Sigma_2)| \) we obtain

\[
M_2 \models_{\Sigma_2} \text{Sen}(\phi)(\varphi_1) \iff M_2 \in |\text{mod}(\Sigma_2)(\text{Sen}(\phi)(\varphi_1))| \quad \text{(def.} \text{mod}(\Sigma_2)) \\
\Rightarrow M_2 \in |\text{Mod}(\phi)^{-1}(\text{mod}(\Sigma_1)(\varphi_1))| \quad \text{(inequality 1)} \\
\Leftrightarrow \text{Mod}(\phi)(M_2) \in |\text{mod}(\Sigma_1)(\varphi_1)| \quad \text{(def.} \text{Mod}(\phi)^{-1}) \\
\Leftrightarrow \text{Mod}(\phi)(M_2) \models_{\Sigma_1} \varphi_1 \quad \text{(def.} \text{mod}(\Sigma_1))
\]

The equivalence of \( \text{IC}_{\text{Mod} \Rightarrow \text{Sen}} \) and inequality 2 is shown analogously.

Since the categories \( \text{Spec}(\Sigma) \) and \( \text{Sub}(\Sigma) \) are partial order categories, i.e., only the existence of arrows matters, lemma 3.7 shows that the institution condition (the two implications altogether) is equivalent to the following equation:

\[
\text{mod}(\Sigma_1); \text{Mod}(\phi)^{-1} = \text{Sen}(\phi); \text{mod}(\Sigma_2). \quad (7)
\]

\( \square \)

To summarize our analysis and to give an overall picture we have to remind:

**Fact 3.8** (Power Construction Functors). Let \( A, B \) be sets and \( C, D \) be categories. Then we have:

1. The assignments \( A \mapsto \mathcal{P}(A) \) and \( (f : A \to B) \mapsto (f : \mathcal{P}(A) \to \mathcal{P}(B)) \), due to Fact 3.5, define a functor \( \mathcal{P} : \text{Set} \to \text{Cat} \).

2. The assignments \( C \mapsto \mathcal{P}(C) \) and \( (F : C \to D) \mapsto (F^{-1} : \mathcal{P}(D) \to \mathcal{P}(C)) \), due to Fact 3.5, define a functor \( \mathcal{P}^{-} : \text{Cat}^{\text{op}} \to \text{Cat} \).

Now, our definitions can be compressed to

\[
\text{Spec} \overset{\text{def}}{=} \text{Sen}; \mathcal{P} : \text{Sign} \to \text{Cat} \quad \text{and} \quad \text{Sub} \overset{\text{def}}{=} \text{Mod}^{\text{op}}; \mathcal{P}^{-} : \text{Sign} \to \text{Cat} \quad (8)
\]

and equation 7 says that \( \text{mod}(\Sigma) \) is the component at \( \Sigma \) of an indexed functor \( \text{mod} : \text{Spec} \Rightarrow \text{Sub} : \text{Sign} \to \text{Cat} \), i.e., the institution condition is equivalent to a naturality requirement.

Together with proposition 3.4 this crucial observation leads to the following abstract concept of logic, which was introduced in [35].
Definition 3.9 (Indexed Frame). An Indexed Frame
\[ \mathcal{IF} = (\text{Ind}, \text{Syn}, \text{Den}, \text{sem}, \text{fml}) \]
is given by the following data and operations:

- A category \( \text{Ind} \);
- An indexed category (syntax functor) \( \text{Syn} : \text{Ind} \to \text{Cat} \);
- An indexed category (denotation functor) \( \text{Den} : \text{Ind} \to \text{Cat} \);
- A family \( \text{sem} \) of functors \( \text{sem}(i) : \text{Syn}(i) \to \text{Den}(i) \), \( i \in |\text{Ind}| \);
- A family \( \text{fml} \) of functors \( \text{fml}(i) : \text{Den}(i) \to \text{Syn}(i) \), \( i \in |\text{Ind}| \);
- A family \( (\text{fml}(i), \text{sem}(i), \eta(i), \epsilon(i)) \), \( i \in |\text{Ind}| \) of local adjunctions, i.e., \( \text{fml}(i) \) left-adjoint to \( \text{sem}(i) \), \( \text{fml}(i) \dashv \text{sem}(i) \) in symbols, with unit \( \eta(i) : \text{id}_{\text{Den}(i)} \Rightarrow \text{fml}(i) ; \text{sem}(i) \) and co-unit \( \epsilon(i) : \text{sem}(i) ; \text{fml}(i) \Rightarrow \text{id}_{\text{Syn}(i)} \);

such that the family of functors \( \text{sem}(i) \), \( i \in |\text{Ind}| \) constitute an indexed functor \( \text{sem} : \text{Syn} \Rightarrow \text{Den} : \text{Ind} \to \text{Cat} \), that is to say, the following indexed frame condition

\[ \text{sem}(i) ; \text{Den}(\sigma) = \text{Syn}(\sigma) ; \text{sem}(j) \]

holds for each \( \sigma : i \to j \) in \( \text{Ind} \).

Moreover, if the indexed functors \( \text{Syn} \) and \( \text{Den} \) are into the category \( \text{PO} \) of all partial order categories, we call the above structure an Indexed Partial Order Frame, written \( \mathcal{IF}_{\text{PO}} \). In case they are into the category \( \text{PRE} \) of all preorders, we name it an Indexed Preorder Frame, written \( \mathcal{IF}_{\text{PRE}} \).

Theorem 3.10 (Institutional Frame). A 4-tuple \( \mathcal{I} = (\text{Sign}, \text{Sen}, \text{Mod}, \models) \) is an institution iﬀ the 5-tuple

\[ \mathcal{IF} = (\text{Sign, Spec, Sub, mod, th}) \]
is an indexed partial order frame, called the institutional frame for \( \mathcal{I} \).

Remark 3.11. Note that the above theorem does not say that an indexed partial order frame is an Institution, but that the data and axioms of an Institution define a structure that at the level of specifications and corresponding categories of models match all the data and satisfy all constrains of an indexed frame. Hence, an indexed frame is a categorical structure which in some sense is more general that an Institution.

4 Entailment Systems

The concept of entailment system introduced in [36] reflects those properties of the entailment relation $\Gamma \vdash \varphi$ which are independent from the particular rules used to generate the relation $\vdash$. This section is devoted to validate the concept of indexed partial order frame by showing that also entailment systems give rise, quite naturally, to indexed partial order frames.

Definition 4.1 (Entailment System). An entailment system $E = (\text{Sign}, \text{Sen}, \vdash)$ with $\text{Sign}$ a category of signatures, $\text{Sen} : \text{Sign} \to \text{Set}$ a functor, and $\vdash$ a function associating to each signature $\Sigma$ a binary relation $\vdash_\Sigma \subseteq P(\text{Sen}(\Sigma)) \times \text{Sen}(\Sigma)$ called entailment relation such that the following properties are satisfied:

- reflexivity: for any $\varphi \in \text{Sen}(\Sigma)$, $\{\varphi\} \vdash \varphi$;
- monotonicity: if $\Gamma_1 \vdash_\Sigma \varphi$ and $\Gamma_2 \supseteq \Gamma_1$ then $\Gamma_2 \vdash_\Sigma \varphi$;
- transitivity: if $\Gamma \vdash_\Sigma \varphi_i$, for $i \in I$, and $\Gamma \cup \{\varphi_i \mid i \in I\} \vdash_\Sigma \psi$, then $\Gamma \vdash_\Sigma \psi$;
- $\vdash$-translation: if $\Gamma_1 \vdash_\Sigma_1 \varphi_1$, then for any $\phi : \Sigma_1 \to \Sigma_2$ in $\text{Sign}$, $\text{Sen}(\phi)(\Gamma_1) \vdash_\Sigma_2 \text{Sen}(\phi)(\varphi_1)$.

The entailment relation $\vdash_\Sigma$ between sets of $\Sigma$-sentences and single $\Sigma$-sentences can be straightforwardly extended to an equivalent entailment relation between $\Sigma$-specifications. For any signature $\Sigma$ in $\text{Sign}$ we define a corresponding extended entailment relation $\vdash^p_\Sigma \subseteq P(\text{Spec}(\Sigma)) \times P(\text{Spec}(\Sigma)) = |\text{Spec}(\Sigma)| \times |\text{Spec}(\Sigma)|$, where we set for any $\Sigma$-specifications $\Gamma_1, \Gamma_2 \in \text{Spec}(\Sigma)$:

$$\Gamma_1 \vdash^p_\Sigma \Gamma_2 \quad \text{iff} \quad \forall \varphi_2 \in \Gamma_2 : \Gamma_1 \vdash_\Sigma \varphi_2 \quad (9)$$

Lemma 4.2. Given a signature $\Sigma$ in $\text{Sign}$ the entailment relation $\vdash_\Sigma$ has the properties (1)-(4) of definition 4.1 if and only if the extended entailment relation $\vdash^p_\Sigma$ satisfies the following properties for any $\Gamma_1, \Gamma_2, \Gamma_3 \in |\text{Spec}(\Sigma)|$:

1. projection: $\Gamma_1 \cup \Gamma_2 \vdash^p_\Sigma \Gamma_2$, i.e., especially $\Gamma_1 \supseteq \Gamma_2$ implies $\Gamma_1 \vdash^p_\Sigma \Gamma_2$;
2. product property: if $\Gamma_1 \vdash^p_\Sigma \Gamma_2$ and $\Gamma_1 \vdash^p_\Sigma \Gamma_3$, then $\Gamma_1 \vdash^p_\Sigma \Gamma_2 \cup \Gamma_3$;
3. compositionality: if $\Gamma_1 \vdash^p_\Sigma \Gamma_2$ and $\Gamma_2 \vdash^p_\Sigma \Gamma_3$, then $\Gamma_1 \vdash^p_\Sigma \Gamma_3$;
4. functor property: if $\Gamma_1 \vdash^p_\Sigma \Gamma_2$, then for any $\phi : \Sigma_1 \to \Sigma_2$ in $\text{Sign}$, $\text{Sen}(\phi)(\Gamma_1) \vdash^p_\Sigma \text{Sen}(\phi)(\Gamma_2)$.

Proof. Necessity:
• **Reflexivity**: Special case of projection, taking $\Gamma_2 = \{ \phi \}$ and $\Gamma_1 \subseteq \Gamma_2$.

• **Monotonicity**: Special case of the projection $\Gamma_1 \cup \Gamma_2 \vdash^p_\Sigma \Gamma_1$, taking $\Gamma_1 = \{ \phi \}$, and assuming $\Gamma_1 \vdash^p_\Sigma \phi$ and $\Gamma_2 \supseteq \Gamma_1$.

• **Transitivity**: Assuming compositionality, it follows directly if we set $\Gamma_1 = \{ \phi_i \mid i \in I \}$, $\Gamma = \emptyset$ and $\Gamma_3 = \{ \psi \}$.

• **$\vdash$-translation**: Special case of the functor property, if $\Gamma_2 = \{ \psi \}$.

Note that for $\Gamma_1 = \emptyset$ the projection property provides the identity property, i.e., $\Gamma_1 \vdash^p_\Sigma \Gamma$ for any $\Gamma \in \text{Spec}(\Sigma)$. Lemma 4.2 makes clear that the extended entailment relation $\vdash^p_\Sigma$ determines a preorder category $\text{Ent}(\Sigma)$.

**Definition 4.3** (Extended Entailment). Given a signature $\Sigma \in |\text{Sign}|$ we define the preorder category $\text{Ent}(\Sigma)$ of all $\Sigma$-specifications with entailment as follows: objects: are all specifications $\Gamma \in |\text{Spec}(\Sigma)|$; morphisms: are all extended entailment relations $\Gamma_1 \vdash^p_\Sigma \Gamma_2$; identities: are all entailments $\Gamma \vdash^p_\Sigma \Gamma$, $\Gamma \in |\text{Spec}(\Sigma)|$; composition: is given by the compositionality property in lemma 4.2.

The projection property in lemma 4.2 entails that $\text{Ent}(\Sigma)$ is an extension of the category $\text{Spec}(\Sigma)$, i.e., for any signature $\Sigma$ we obtain an embedding functor

$$\text{em}(\Sigma) : \text{Spec}(\Sigma) \to \text{Ent}(\Sigma) \quad \text{with} \quad \text{em}(\Sigma)(\Gamma) = \Gamma. \quad (10)$$

If we consider the set of sentences provable from a specification $\Gamma$, i.e., if we set

$$\text{clo}(\Sigma)(\Gamma) = \{ \varphi \in \text{Sen}(\Sigma) \mid \Gamma \vdash^p_\Sigma \varphi \}, \quad (11)$$

we obtain a so-called closure functor from $\text{Ent}(\Sigma)$ to $\text{Spec}(\Sigma)$.

**Proposition 4.4** (Galois Correspondence). Given a signature $\Sigma \in |\text{Sign}|$, the equations (10) and (11) define functors $\text{em}(\Sigma) : \text{Spec}(\Sigma) \to \text{Ent}(\Sigma)$ and $\text{clo}(\Sigma) : \text{Ent}(\Sigma) \to \text{Spec}(\Sigma)$, such that $\text{clo}(\Sigma)$ is left-adjoint to $\text{em}(\Sigma)$, i.e., for any specifications $\Gamma, \Gamma_1, \Gamma_2 \in |\text{Spec}(\Sigma)| = |\text{Ent}(\Sigma)|$ we have:

1. **$\text{em}(\Sigma)$ functor**: $\Gamma_1 \supseteq \Gamma_2$ implies $\Gamma_1 \vdash^p_\Sigma \Gamma_2$;
2. **$\text{clo}(\Sigma)$ functor**: $\Gamma_1 \vdash^p_\Sigma \Gamma_2$ implies $\text{clo}(\Sigma)(\Gamma_1) \supseteq \text{clo}(\Sigma)(\Gamma_2)$;
3. **unit**: $\Gamma \vdash^p_\Sigma \text{clo}(\Sigma)(\Gamma)$ in $\text{Ent}(\Sigma)$;
4. **counit**: $\text{clo}(\Sigma)(\Gamma) \supseteq \Gamma$ in $\text{Spec}(\Sigma)$;
5. **adjointness**: $\Gamma_1 \vdash^p_\Sigma \Gamma_2$ iff $\text{clo}(\Sigma)(\Gamma_1) \supseteq \Gamma_2$.

\[
\begin{array}{ccc}
\text{Ent}(\Sigma) & \xrightarrow{\text{clo}(\Sigma)} & \text{Spec}(\Sigma) \\
\text{em}(\Sigma) & \xrightarrow{\vdash^p_\Sigma} & \text{Spec}(\Sigma)
\end{array}
\]

**Proof.**

• **$\text{em}(\Sigma)$ functor**: Directly from the projection property, assuming $\Gamma_1 \supseteq \Gamma_2$. 

12
4 ENTAILMENT SYSTEMS

- **clo(Σ) functor**: Assuming $Γ_1 ⊨_Σ Γ_2$, we must show that $clo(Σ)(Γ_1) ⊇ clo(Σ)(Γ_2)$. To see this, note that

$$
clo(Σ)(Γ_1) = \{ ϕ_1 ∈ Sen(Σ) | Γ_1 ⊨ ϕ_1 \} \quad \text{(by definition)}
$$

$$

⊇ \{ ϕ_1 ∈ Sen(Σ) | Γ_2 ⊨ ϕ_1 \} \quad (Γ_1 ⊨ Γ_2 \text{ and compositionality})
$$

$$

= clo(Σ)(Γ_2) \quad \text{(by definition)}
$$

- **unit**: It follows from the adjointness property below.

- **co-unit**: It follows from the adjointness property below.

- **adjointness**: Note that $Γ_1 ⊨ P_Σ Γ_2$ iff $∀ ϕ ∈ Γ_2 : Γ_1 ⊨ ϕ$ iff $\{ ψ ∈ Sen(Σ) | Γ_1 ⊨ ψ \} ⊇ Γ_2$ iff $clo(Σ)(Γ_1) ⊇ Γ_2$

In the next steps we exploit the functor property in lemma 4.2 to gain a fully categorical presentation of the concept “entailment system”.

**Proposition 4.5** (Entailment Functor). The mappings $Spec : |Sign| → |PO|$ and $Ent : |Sign| → |PRE|$ given by definition 3.3 and definition 4.3 respectively can be extended to functors $Spec : Sign → PRE$ and $Ent : Sign → PRE$ if we set for any $φ : Σ_1 → Σ_2$ in $Sign$ and for any $Γ_1 ∈ |Spec(Σ_1)|$: $Spec(φ)(Γ_1) = Ent(φ)(Γ_1) = Sen(φ)(Γ_1)$. In this way all the functors $em(Σ) : Spec(Σ) → Ent(Σ)$, $Σ ∈ |Sign|$ define an indexed functor $em : Spec ⇒ Ent : Sign → PRE$.

**Proof.** Let $φ_1 : Σ_1 → Σ_2$ and $φ_2 : Σ_2 → Σ_3$ be arrows in $Sign$. Then the equations $Spec(φ_1; φ_2) = Spec(φ_1); Spec(φ_2)$ and $Spec(id_Σ) = id_{Spec(Σ)}$ hold because of fact 3.5 and remark 3.6. On the other hand, the equations $Ent(φ_1; φ_2) = Ent(φ_1) ; Ent(φ_2)$ and $Ent(id_Σ) = id_{Ent(Σ)}$ follow from the fact that we are on top of (set-based) preorder categories, and the compositionality and functor properties of lemma 4.2. The family of functors $em(Σ), Σ ∈ |Sign|$ are natural with respect to the arrows in $Sign$ because for each $φ : Σ_1 → Σ_2$ and $Γ ∈ Spec(Σ_1)$, $[em(Σ_1); Ent(φ)](Γ) = Sen(φ)(Γ) = [Sen(φ); em(Σ_2)](Γ)$.

**Theorem 4.6** (Entailment Frames). A triple $Σ = (Sign, Sen, ⊨)$ is an entailment system iff the 5-tuple

$$
Σ_1 \xymatrix{ Spec(Σ_1) \ar[r]^{clo(Σ_1)} \ar[d]_{Spec(φ)} & Ent(Σ_1) \ar[d]^{Ent(φ)} \ar[d]_{em(Σ_2)} ^{clo(Σ_2)} \ar[l]_{em(Σ_1)} } \Sigma_2
$$

defines an indexed preorder frame, called the ENTAILMENT FRAME for $Σ$. 
Proof. It follows directly from propositions 4.5, 4.4 and lemma 4.2.

Note that the same comments made in remark 3.11 also apply here. Also, the concept of \( \pi \)-institutions as originally introduced by Fiadeiro and Sernadas are essentially the same as entailment systems. Only transitivity is formulated in a different but equivalent way (see [15, 16]). An abstract interpretation of \( \pi \)-institutions can be given in terms of closure operators, i.e., in terms of the functors \( \text{em}(\Sigma) ; \text{clo}(\Sigma) : \text{Spec}(\Sigma) \rightarrow \text{Spec}(\Sigma) \). The generalization of this abstraction step to arbitrary indexed frames based on the co-monads given by the adjunctions \( \text{fml}(i) \vdash \text{sem}(i) \) is approached in section 8.

5 General Logics

The concept of a general logic introduced in [36] combines the concepts of institution and entailment system. In this section we analyze and discuss this concept in view of indexed frames.

Definition 5.1 (General Logic). A **General Logic** is a 5-tuple

\[ \mathcal{L} = (\text{Sign}, \text{Sen}, \text{Mod}, \vdash, \models) \]

such that \((\text{Sign}, \text{Sen}, \vdash)\) is an entailment system; \((\text{Sign}, \text{Sen}, \text{Mod}, \models)\) is an institution, and the following soundness condition is satisfied for any \( \Sigma \in \text{Sign} \), \( \Gamma \subseteq \text{Sen}(\Sigma) \), \( \varphi \in \text{Sen}(\Sigma) \)

\[ \Gamma \vdash_\Sigma \varphi \implies \Gamma \models_\Sigma \varphi \]

where \( \Gamma \models_\Sigma \varphi \) iff \( \text{mod}(\Sigma)(\Gamma) \subseteq \text{mod}(\Sigma)(\varphi) \).

A logic is **complete** if, in addition, \( \Gamma \vdash_\Sigma \varphi \iff \Gamma \models_\Sigma \varphi \).

In view of the corresponding institutional frame \( \mathcal{IF} = (\text{Sign}, \text{Spec}, \text{Sub}, \text{mod}, \text{th}) \) and the corresponding entailment frame \( \mathcal{EF} = (\text{Sign}, \text{Spec}, \text{Ent}, \text{em}, \text{cl}) \) the soundness condition means that \( \Gamma_1 \vdash_\Sigma \Gamma_2 \) implies \( \text{mod}(\Sigma)(\Gamma_1) \subseteq \text{mod}(\Sigma)(\Gamma_2) \) for any \( \Sigma \in \text{Sign} \) and any \( \Gamma_1, \Gamma_2 \in \text{Spec}(\Sigma) \).

That is for any \( \Sigma \in \text{Sign} \) the assignments \( \Gamma \mapsto \text{mod}(\Sigma)(\Gamma) \) define a functor \( \text{mod}(\Sigma) : \text{Ent}(\Sigma) \rightarrow \text{Sub}(\Sigma) \) such that \( \text{mod}(\Sigma) = \text{em}(\Sigma) ; \text{mod}(\Sigma) \). Since all the functors \( \text{mod}^+(\Sigma) : \text{Ent}(\Sigma) \rightarrow \text{Sub}(\Sigma) \) define an indexed functor \( \text{mod}^+ : \text{Ent} \Rightarrow \text{Sub} : \text{Sign} \rightarrow \text{Cat} \) the soundness condition is equivalent to the existence of a factorization \( \text{mod} = \text{em} ; \text{mod}^+ \) and the logic is complete if all the functors \( \text{mod}^+(\Sigma) \) are full. To put things more precisely we introduce

Definition 5.2 (Indexed Logic). An **Indexed Logic** is a 8-tuple

\[ \mathcal{IL} = (\text{Ind}, \text{Syn}, \text{Den}, \text{Log}, \text{sem}, \text{fml}, \text{log}, \text{sub}) \]

such that:

- \( \text{Ind} \), \( \text{Syn} \), \( \text{Den} \), \( \text{sem} \), \( \text{fml} \) are indexed partial order frame frames;
- \( \text{Ind} \), \( \text{Syn} \), \( \text{Log} \), \( \text{log} \), \( \text{sub} \) are indexed preorder frames, and
- there exists an indexed functor \( \text{sem}^+ : \text{Log} \Rightarrow \text{Den} : \text{Ind} \rightarrow \text{Cat} \) such that \( \text{sem} = \text{log} ; \text{sem}^+ \)
An indexed logic is complete if, in addition, all the functors \( \text{sem}^i : \text{Log}(i) \to \text{Den}(i) \) \( i \in |\text{Ind}| \) are full.

**Theorem 5.3** (Indexed General Logics). A 5-tuple \( \mathcal{L} = (\text{Sign}, \text{Sen}, \text{Mod}, \vdash, \models) \) is a general logic iff the 8-tuple

\[
\mathcal{IL} = (\text{Sign}, \text{Spec}, \text{Sub}, \text{Ent}, \text{mod}, \text{th}, \text{em}, \text{cl})
\]

defines an indexed logic.

**Proof.** This follows from theorems 3.10 and 4.6. \( \square \)

### 6 Linear Logic as an Indexed Frame

Below we make a detailed presentation of linear logic (actually a sublogic of it) as an Indexed Frame. The example is important because it shows a situation where the categories induced by the indexes of a given category \( \text{Ind} \) are neither pre- or partial order categories.

**Example 6.1** (Linear Logic). In this example we consider a substructural logic. We remember that a sequent calculus presentation for the intuitionistic logic has some structural rules. They are the rules of weakening, contraction, permutation and cut. A substructural logic does not have all of these rules. Our example is the intuitionistic rudimentary linear logic \( \text{IRLL} \), which is a fragment of \( \text{MILL} \) (Multiplicative Intuitionistic Linear Logic), since we consider only the \( \{\otimes, \multimap\} \) fragment. In this logic we have a multiplicative conjunction \( \otimes \) and its adjoint, the linear implication \( \multimap \). The rules of its sequent calculus presentation are shown below.

\[
\begin{align*}
A \vdash A & \quad \text{identity} \\
\Gamma_1, A, B, \Gamma_2 \vdash C & \quad \Gamma_1, B, A, \Gamma_2 \vdash C \quad \text{permut-left} \\
\Gamma_1, A \otimes B, \Gamma_2 \vdash C & \quad \otimes\text{-left} \\
\Gamma_1 \vdash A & \quad \Gamma_2, B \vdash C \quad \multimap\text{-left} \\
\Gamma_2, A \multimap B, \Gamma_1 \vdash C & \quad \multimap\text{-right} \\
\Gamma, A \vdash B & \quad \Gamma, A \vdash B \quad \multimap\text{-right} \\
\Gamma, I \vdash A & \quad I\text{-left} \\
\Gamma, I \vdash A & \quad I\text{-right} \\
\Gamma_1 \vdash A & \quad \Gamma_2, A, \Gamma_3 \vdash B \quad \text{cut-rule} \\
\Gamma_2, \Gamma_1, \Gamma_3 \vdash B & \quad \text{cut-rule}
\end{align*}
\]

The cut rule is admissible. This means that from cut-free proofs of \( \Gamma_1 \vdash A \) and \( A, \Gamma_2 \vdash B \), we can obtain a cut-free proof of \( \Gamma_1, \Gamma_2 \vdash B \). This result is called Haupsatz (firstly proved by Gentzen in 1936) and we recommend [18] for a detailed reference on its relation to Linear Logic. The Haupsatz can be also viewed as a process that from any proof of a sequent produces a cut-free proof of this very sequent. For example, in the proofs below, cut-elimination applied to the first proof produces the second.
There is a categorical way of reading proof-theoretical reductions. Formulas are interpreted as objects, and logical connectives as functors. In \( IRLL \), a sequent of the form \( \Delta_1, \ldots, \Delta_k \vdash B \) is interpreted as a morphism \( f \) from \( [\Delta_1] \otimes \cdots \otimes [\Delta_k] \) into \([B] \) in a suitable category \( C \). The rules of the sequent calculus provide means to point out morphisms in any category \( C \), equipped with special properties, used as semantics for the logic. The axiom \( A \vdash A \) is interpreted as the identity on the object \([A] \). \( I \) is the terminal object of the category. The empty tensor product \((\otimes)\) is \( I \) itself. Thus, axiom \( I\)-right is related to the identity morphism on \( I \). Given a morphism such as \( f : A \otimes B \to C \), the rule \( \to\)-right points out the need of an adjoint morphism \( \hat{f} : A \to (B \to C) \). The existence of the morphism \( \hat{f} \) must be provided by a natural bijection between \( Hom_C(A \otimes B, C) \) and \( Hom_C(A, B \to C) \). The cut has to be associated to the composition of morphisms. In fact, from the point of view of morphisms in the internal language of \( C \), we would have the following proof term (tree)

\[
\begin{align*}
 f : [\Gamma_1] &\to [A] \\
g : [\Gamma_2, A, \Gamma_3] &\to [B] \\
(id_{[\Gamma_2]} \otimes f \otimes id_{[\Gamma_3]}); g : [\Gamma_2, \Gamma_1, \Gamma_3] &\to [B]
\end{align*}
\]  

Thus, the corresponding proof term in the internal logic of \( C \) associated to proof (12) is

\[
\begin{align*}
 id_A : A &\to A \\
id_B : B &\to B \\
id_A \otimes id_B : A \otimes B &\to A \otimes B \\
ev_{B,A \otimes B} : (B \to (A \otimes B)) &\otimes B \to A \otimes B
\end{align*}
\]

However, we also require that \( C \) is equipped with a kind of closedness property, thus satisfying \((id_A \otimes id_B) \otimes id_B; ev_{B,A \otimes B} \) = \( id_A \otimes B \). What the cut-elimination ensures is that this morphism is equal to \( id_{A \otimes B} : A \otimes B \to A \otimes B \), which is pointed out by proof (13) and the following correspondent proof term in the internal language of \( C \):

\[
\begin{align*}
id_A : A &\to A \\
id_B : B &\to B \\
id_A \otimes id_B : A \otimes B &\to A \otimes B
\end{align*}
\]  

The cut-elimination is a recursively defined procedure that replaces any cut by cuts of lower degree of complexity. For example, the reduction from (17) to (18) used in the first proof yielding the second one corresponds to a case when the cut formula is a linear implication.
The categorical interpretation of the logic $\text{IRLL}$ must be done in a category having what is required by the sequent calculus and the cut-elimination reductions. Of course the category $1$ (the category with one object and one morphism) can be used as semantics. Any formula is mapped to the unique object inhabiting $1$ and any morphism to the identity on this object. This is not a good semantics, since any sequent proved in the calculus is mapped to the same morphism between the same formulas. On the other extreme we can interpret $\text{IRLL}$ in $\text{Set}$, using the cartesian product to interpret $\otimes$ and the function space $[B][A]$ to interpret $A \to B$. This, however, is not a good semantics either, since the cartesian product has projections and we cannot derive in the sequent calculus any morphism from $A \otimes B$ in to $B$ (or $A$). $\text{Set}$ has more morphisms and objects than $\text{IRLL}$ is able to point out. On the other hand, the syntactical category $\text{Proofs}(\Sigma)$ made up of formulas built from a signature $\Sigma$ and sequents provable in $\text{IRLL}$ seems to have the intended semantics to interpret $\text{IRLL}$. In fact, the more adequate semantics should be provided by the skeletal category of $\text{Proofs}(\Sigma)^3$. In the case that there are other mathematical structures adequate to provide semantics to $\text{IRLL}$ we need a way to express this. Categorically we state that $\text{Proofs}(\Sigma)$ is the free category in the subcategory of $\text{Cat}$ satisfying the restriction imposed by $\text{IRLL}$. This also means that there is an adjunction between any category serving as semantics for $\text{IRLL}$ and the category $\text{Proofs}(\Sigma)$. When $\Sigma$ is derived from a category $\mathcal{C}$, we call $\text{Proofs}(\Sigma)$ the internal language logic of $\mathcal{C}$.

Definition 6.2. A monoidal category is a category which has the structure of a monoid, that is, among the objects there is a binary operation which is associative and has an unique neutral or unit element. Specifically, a category $\mathcal{C}$ is monoidal if

1. there is a (tensor product) bifunctor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, where the images of object $(A, B)$ and arrow $(f, g)$ are written $A \otimes B$ and $f \otimes g$, respectively.

2. there is an isomorphism $\alpha_{ABC} : (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$ for arbitrary objects $A, B, C$ in $\mathcal{C}$, such that $\alpha_{ABC}$ is natural in $A, B, C$. In other words:
   - $\alpha_{\otimes BC} : (\otimes B) \otimes C \Rightarrow - \otimes (B \otimes C)$ is a natural transformation for arbitrary objects $B, C$ in $\mathcal{C}$.
   - $\alpha_{\otimes A-C} : (A \otimes -) \otimes C \Rightarrow A \otimes (- \otimes C)$ is a natural transformation for arbitrary objects $A, C$ in $\mathcal{C}$.
   - $\alpha_{\otimes AB} : (A \otimes B) \otimes - \Rightarrow A \otimes (B \otimes -)$ is a natural transformation for arbitrary objects $A, B$ in $\mathcal{C}$.

3. there is an object $I$ in $\mathcal{C}$ called the unit object (or simply the unit).

4. for any object $A$ in $\mathcal{C}$, there are isomorphisms
   $$\lambda_A : I \otimes A \cong A \quad \text{and} \quad \rho : A \otimes I \cong A,$$

3We remind the reader that a category is called skeletal if isomorphic objects are necessarily identical.
such that \( \lambda_A \) and \( \rho_A \) are natural in \( A \), i.e., both \( \lambda : I \otimes - \Rightarrow - \) and \( \rho : - \otimes I \Rightarrow - \) are natural transformations.

satisfying the following commutative diagrams:

- **UNIT COHERENCE LAW**

\[
\begin{array}{ccc}
(A \otimes I) \otimes B & \xrightarrow{\alpha_{AB}} & A \otimes (I \otimes B) \\
\rho_A \otimes \text{id}_B & \downarrow & \downarrow \text{id}_A \otimes \lambda_B \\
A \otimes B & & A \otimes B
\end{array}
\]

- **ASSOCIATIVE COHERENCE LAW**

\[
\begin{array}{ccc}
((A \otimes B) \otimes C) \otimes D & \xrightarrow{\alpha_3} & (A \otimes (B \otimes C)) \otimes D & \xrightarrow{\alpha_2} & A \otimes ((B \otimes C) \otimes D) \\
\alpha_1 & \downarrow & \downarrow & \alpha_4 & \\
(A \otimes B) \otimes (C \otimes D) & \xrightarrow{\alpha_5} & A \otimes (B \otimes (C \otimes D))
\end{array}
\]

where we have

\[
\begin{align*}
\alpha_1 & \overset{\text{def}}{=} \alpha_{ABC} \otimes \text{id}_D \\
\alpha_2 & \overset{\text{def}}{=} \text{id}_A \otimes \alpha_{(B \otimes C)D} \\
\alpha_3 & \overset{\text{def}}{=} \alpha_{(A \otimes B)CD} \\
\alpha_4 & \overset{\text{def}}{=} \text{id}_A \otimes \alpha_{BCD} \\
\alpha_5 & \overset{\text{def}}{=} \alpha_{AB(C \otimes D)}
\end{align*}
\]

A monoidal category \( \mathcal{C} \) with tensor product \( \otimes \) is said to be symmetric if for every pair of objects \( A, B \), there is an isomorphism \( \xi_{AB} : A \otimes B \cong B \times A \) that is natural in both \( A \) and \( B \) such that the following diagrams are commutative:

- **UNIT COHERENCE FOR \( \xi \)**

\[
\begin{array}{ccc}
(A \otimes I) & \xrightarrow{\xi_A} & (I \otimes A) \\
\rho_A & \downarrow & \downarrow \lambda_A \\
A & & A
\end{array}
\]

- **ASSOCIATIVE COHERENCE FOR \( \xi \)**

\[
\begin{array}{ccc}
(A \otimes B) \otimes C & \xrightarrow{\alpha_{ABC}} & A \otimes (B \otimes C) & \xrightarrow{\xi_{A(B \otimes C)}} & (B \otimes C) \otimes A \\
\xi_{AB} \otimes \text{id} & \downarrow & \downarrow & \alpha_{BCA} & \\
(B \otimes A) \otimes C & \xrightarrow{\alpha_{BAC}} & B \otimes (A \otimes C) & \xrightarrow{\text{id} \otimes \xi_C} & B \otimes (C \otimes A)
\end{array}
\]

A symmetric monoidal category is closed if for any object \( B \), the functor \( (- \otimes B) \) has a right adjoint \( (B \otimes -) \). On a global level, this means that we have a natural bijection from

\[
\text{Hom}_\mathcal{C}(- \otimes B, -) : \mathcal{C}^{op} \times \mathcal{C} \to \text{Set} \quad \text{to} \quad \text{Hom}_\mathcal{C}(-, B \otimes -) : \mathcal{C}^{op} \times \mathcal{C} \to \text{Set}
\]
with components 
\[ \theta_{AC} : \text{Hom}_C(A \otimes B, C) \to \text{Hom}_C(A, B \rightarrow C) \]

Symmetric closed monoidal categories have all the categorical structure required to model the multiplicative fragment of intuitionistic linear logic.

A monoidal closed functor between monoidal closed categories is a triple 
\[ \langle F, \mu \text{unit}, \mu \otimes \rangle, \]
where \( F : C \to D \) is a functor, together with two natural transformations \( \mu^I : I \Rightarrow F(I) \) and \( \mu_{AB} : F(A) \otimes F(B) \Rightarrow F(A \otimes B) \) satisfying some coherence diagrams (which we omit). \( F \) is strict if \( \mu^I \) and \( \mu_{AB} \) are identities, for each \( A \) and \( B \). A monoidal functor is symmetric if \( \mu \) commutes with the symmetries \( \xi_{F(A)F(B)}; \mu_{BA} = \mu_{AB} ; F(\xi_{AB}) \) for all \( A \) and \( B \).

A strong symmetric monoidal (closed) functor \( F \) between symmetric monoidal (closed) categories is a functor that strictly preserves the monoidal structure of the categories, that is, \( F(A \otimes B) = F(A) \otimes' F(B), F(I) = I' \), and hence \( F(A \rightarrow B) = F(A) \rightarrow' F(B) \).

We call \( SMC \) the category whose objects are symmetric monoidal closed (SMC) categories and whose morphisms are symmetric monoidal closed functors.

Given any symmetric monoidal closed category \( C \) and any object \( A \) belonging to it, we can interpret the proofs 12 and 13 above as morphism from \( A \) into \( A \). The proof 13 is interpreted as the identity on \( A \).

Rudimentary Linear Logic (\( RLL \)) is a propositional logic with binary connectives \( \otimes \) and \( \rightarrow \) and a logical constant \( I \). \( RLL \) provides an internal logic for \( SMC \) (see [32]). This means that there are two (endo) functors \( L \) and \( K \) in \( SMC \), such that, for each monoidal category \( M \), \( M \simeq K(L(M)) \). Moreover, for each \( RLL \) \( Th(l) \) based on the language \( l \), we have \( l \simeq L(K(l)) \).

Since these functors provide an equivalence of categories, we can see \( RLL \) as an Indexed Logic. The following Indexed Frame is used to define \( RLL \) as an Indexed Logic.

**Definition 6.3.**

\[ RLL = \langle \text{Ind}, \text{Proofs}, \text{Den}, \text{sem}, \text{fml} \rangle \]

is given by:

- A category of signatures \( \text{Ind} \);
- \( \text{Proofs}(i) \) is the category that has formulas of \( IRLL(i) \) as objects and proofs as morphisms. A proof of \( B \) from assumptions \( \Gamma_0, \ldots, \Gamma_k \) in \( IRLL(i) \) is a morphism from \( [\Gamma_0] \otimes \ldots \otimes [\Gamma_k] \) to \( |B| \). Equality of morphism is provided by proof-theoretical reductions (cut-elimination) in the sequent calculus of \( IRLL \).
- \( \text{Den}(i) \) is the (sub) category of symmetric monoidal categories having at least \( |i| \) distinct (possibly isomorphic) objects.
- \( \text{sem}(i) \) is the functor that maps a symmetric monoidal closed category into its skeletal category.
- \( \text{fml}(i) \) is the functor that maps any symmetric monoidal closed category into its internal language category.

The above example of an Indexed Logic does not consist of Indexed Ordered Frames. Neither \( A, B \not\vdash A \) nor \( A, B \not\vdash B \) hold. The syntax of the logic cannot be interpreted in a preordered setting. From \( \alpha \vdash \beta \) does not follow, in general, that \( \alpha, \gamma \vdash \beta \). That is, the monoidal structure cannot be, in general, described by a pre- or partial order structure. It
is known that in IRLL there are tautologies that have infinitely many non-equivalent proofs. Thus, for an arbitrary \( i \), \( \text{Proofs}(i) \) cannot be a pre-ordered category.

This view of a deduction pointing out a morphism in a category is deeply discussed in [14] and [23]. Lambek, see [25, 26, 27], seems to be the first to develop this idea following MacLane’s coherence theorems (see [28]).

7 Generalized Sketches as Indexed Frames

Usually, only a logical system with “indexed semantics” gives rise to an Institution. Indexed semantics separates strictly syntax from semantics and models are given by “interpretations” of the syntactic entities in a fixed “semantic universe” - the category \( \text{Set} \) in most cases. Model reduction is realized by simple composition of syntactic translations with semantic interpretations and, since composition is associative, we get, on the global level, a model functor.

In a logical system with “fibred semantics”, in contrast, syntactic and semantic entities live in the same universe/category and models are given by “instances” of syntactic entities, i.e., by morphisms in the common category with syntactic entities as targets. Those “fibrational approaches” are quite common in software engineering (see, for example [13, 38]).

Model reduction in fibred semantics is a bit more complicated and not that nicely behaving. Model reduction is realized by pullback construction and we have to rely on an arbitrary but fixed choice of pullbacks for the common category to define reduction functors. However, what ever choice we make, the composition of chosen pullbacks will be only associative up to isomorphism thus we get, on the global level, only a model pseudo functor for a fixed choice of pullbacks. For other fixed choices of pullbacks we will get other equivalent model pseudo functors.

Indexed frames abstract from single sentences and single models and allow to describe transformations of specifications or model classes, respectively, which may be not based on point-to-point transformations of single sentences or single models, respectively.

As a paradigmatic example for logical systems with fibred semantics we discuss in this section generalized sketches [6, 10, 33] with fibred semantics as they have been presented in [13]\(^4\). We show that generalized sketches give rise to Indexed Partial Order Frames. The example exceeds the corresponding result for institutions in two aspects. First, we do have syntax functors with target \( \text{Cat} \) instead of \( \text{Set} \). Second, in contrast to institutions, the idea to describe model extension simply as the inverse of model reduction doesn’t provide functoriality on the level of indexed frames. To overcome this obstacle we have to give up the choice of pullbacks (see Remark 7.7).

Applications of generalized sketches in software engineering are usually based on the category \( \text{Graph} \) of directed (multi)graphs [10, 12, 11, 38]. Generalized sketches, however, are not restricted to graphs only. In fact, they are generic in the sense that we can define for any base category \( \text{Base} \) with pullbacks a variety of generalized sketch systems of different types, as we will outline in the rest of this section.

In category theory we have sketches of different types like linear sketches, finite product sketches and finite limit sketches, for example [3]. And in software engineering we have different types of diagrammatic specification techniques like ER diagrams, class diagrams,

\(^4\)The interested reader should consult [13] for a more detailed software engineering oriented motivation and for a short history of generalized sketches.
The α dependencies correspond to implications stating that the concept "pullback" is based on the concept "commutative square". Logically, concept/property Q predicate symbols and dependency arrows (Meta-Signature)

Definition 7.1 (Meta-Signature). A meta-signature over Base is given by a category II of predicate symbols and dependency arrows and a functor α : II → Base. For an object P ∈ |II|, the Base-object α(P) is called the arity (shape) of P, and for a dependency arrow r : Q ↭ P in II, α(r) : α(Q) → α(P) is called an arity substitution.

A dependency r : Q ↭ P states that the concept/property P depends on (is based on) the concept/property Q. As an example we can consider a dependency r : \[\text{commsqu} \vdash \alpha(r)\) stating that the concept "pullback" is based on the concept "commutative square". Logically, dependencies correspond to implications α(P) ⇒ α(Q) with single predicative expressions α(P) and α(Q) as premise and conclusion, respectively (compare Definition 7.8).

The category Base is our common category for syntax and semantics. Considering an object in Base as a "signature" we can define a corresponding category of atomic constraints.

Definition 7.2 (Category of Atomic Constraints). For any object G ∈ |Base| we define the category \(\text{Fm}(G)\) of all atomic constraints on G as follows:

**Objects:** all pairs (P, d) with P ∈ II and a morphism d : α(P) → G in Base;

**Morphisms:** all labeled sequent’s r : (Q, e) ↭ (P, d) with r : Q ↭ P a dependency arrow in II and e = α(r); d;

**Identities:** \(\text{id}_{(P,d)} \overset{def}{=} (\text{id}_{P} : (P, d) \vdash (P, d))\) for all atomic constraints (P, d);

**Composition:** the composition of any two labeled sequent’s r : (Q, e) ↭ (P, d) and p : (P, d) ↭ (R, c) is given by the sequent r;p : (Q, e) ↭ (R, c).

Associativity of composition in Base ensures that composition of labeled sequent’s is indeed defined. Moreover, the identity and associativity law is ensured by the identity and associativity laws in Base and II, respectively, and by the functor property of α, thus we get indeed a category \(\text{Fm}(G)\).

Note, that we could say that the labeled sequent r : (Q, e) ↭ (P, d) is produced by applying the dependency/rule r : Q ↭ P to (P, d). In this sense, predicate dependencies serve as inference rules (see also Remark 7.11).

Now we are going to define a syntax functor on the abstraction level of "sentences". For a morphism \(\phi : G \rightarrow H\) in Base the corresponding translation of atomic constraints is simply given by post-composition: For any atomic constraint (P, d) in \(\text{Fm}(G)\) we define \(\text{Fm}(\phi)(P, d) = (P, d; \phi)\) and for any labeled sequent r : (Q, e) ↭ (P, d) in \(\text{Fm}(G)\) we get a labeled sequent \(\text{Fm}(\phi)(r) = r : (Q, e; \phi) ↭ (P, d; \phi)\) in \(\text{Fm}(H)\) since d trivially implies e; \(\phi = \alpha(r); (d; \phi)\). In this way a signature morphism \(\phi : G \rightarrow H\) gives rise to a functor \(\text{Fm}(\phi) : \text{Fm}(G) \rightarrow \text{Fm}(H)\) between the corresponding categories of atomic constraints.

Due to associativity of composition in Base, we have for any morphisms \(\phi : G \rightarrow H\) and \(\psi : H \rightarrow K\) in Base that \(\text{Fm}(\phi; \psi) = \text{Fm}(\phi); \text{Fm}(\psi)\) such that the assignments \(G \mapsto \text{Fm}(G)\) and \(\phi \mapsto \text{Fm}(\phi)\) define indeed a syntax functor \(\text{Fm} : \text{Base} \rightarrow \text{Cat}\) which has, however, the category Cat as target and not the category Set as in institutions.

Now we take the abstraction step from "sentences" to "specifications".

Definition 7.3 (Partial Order Category of Sketches). A (generalized) sketch S with base \(\phi \in |\text{Base}|\) is a subcategory \(S \subseteq \text{Fm}(G)\).

By \(\text{Sk}(G)\) we denote the partial order category

\[\text{Sk}(G) = \{S \subseteq \text{Fm}(G) : \text{S is a partial order category}\}\]

Note, that [13] works actually only with sketches closed under dependencies (compare Example 7.11).
where the objects are all sketches on $G$, and the arrows are all the inverse category inclusions $S_1 \supseteq S_2$.

For a signature morphism $\phi : G \to H$ the translation of sketches is defined as follows: For any sketch $S$ with base $G$ we denote by $\text{Sk}(\phi)(S)$ the smallest subcategory of $\text{Fm}(H)$ containing the image $\text{Fm}(\phi)(S)$ of $S$ w.r.t. the functor $\text{Fm}(\phi)$. Obviously, the assignments $S \mapsto \text{Sk}(\phi)(S)$ define a functor $\text{Sk}(\phi) : \text{Sk}(G) \to \text{Sk}(H)$ between the partial order categories $\text{Sk}(G)$ and $\text{Sk}(H)$.

For any morphisms $\phi : G \to H$ and $\psi : H \to K$ in $\text{Base} \text{Fm}(\phi; \psi) = \text{Fm}(\phi); \text{Fm}(\psi)$ entails $\text{Sk}(\phi; \psi) = \text{Sk}(\phi); \text{Sk}(\psi)$ thus we get finally the intended syntax functor on the abstraction level of specifications.

**Corollary 7.4 (Sketch Functor).** The assignments $G \mapsto \text{Sk}(G)$ and $\phi \mapsto \text{Sk}(\phi)$ define a functor $\text{Sk} : \text{Base} \to \text{Cat}$.

An instance of a signature $G \in |\text{Base}|$ is given by a "semantic" object $A \in |\text{Base}|$ and a morphism $a : A \to G$. That is, the category of instances of $G$ is a slice category $\text{Inst}(G) = \text{Base}\downarrow G$. For any fixed choice of pullbacks in $\text{Base}$ the assignments $G \mapsto \text{Inst}(G)$ can be extended to a pseudo functor $\text{Inst} : \text{Base}^{\text{op}} \to \text{Cat}$ (compare [13]). Since those pullback based model pseudo functors don’t give rise straightforwardly to denotational functors on the level of indexed frames (see Remark 7.7) we skip them here and continue to define directly denotational functors.

**Definition 7.5 (Partial Order Category of Subcategories of Instances).** For any base object $G \in \text{Base}$ the partial order category $\text{Sub}(G)$ has as objects all full subcategories $M \subseteq \text{Base}\downarrow G$ and as arrows all inclusion functors $M_1 \subseteq M_2$.

For any morphism $\phi : G \to H$ in $\text{Base}$ we can define a "pullback complement" functor $\text{PC}(\phi) : \text{Sub}(G) \to \text{Sub}(H)$ as follows: For any full subcategory $M \subseteq \text{Base}\downarrow G$ the category $\text{PC}(\phi)(M)$ is the full subcategory of $\text{Base}\downarrow H$ given by all pullback complements of instances in $M$: $(B, b) \in |\text{PC}(\phi)(M)|$ if there exists an instance $(A, a) \in |M|$ and a morphism $f : A \to B$ such that $(a, f)$ is a pullback of $(\phi, b)$ in $\text{Base}$ (see the diagram below). We call the pair $((A, a), f)$ a witness for $(B, b) \in |\text{PC}(\phi)(M)|$.

Obviously, $M_1 \subseteq M_2$ implies $\text{PC}(\phi)(M_1) \subseteq \text{PC}(\phi)(M_2)$ thus we obtain indeed a functor $\text{PC}(\phi) : \text{Sub}(G) \to \text{Sub}(H)$.

For any morphisms $\phi : G \to H$ and $\psi : H \to K$ in $\text{Base}$ we obtain $\text{PC}(\phi; \psi) = \text{PC}(\phi); \text{PC}(\psi)$: For any $M \in \text{Sub}(G)$ it holds that

$$(\text{PC}(\phi); \text{PC}(\psi))(M) = \text{PC}(\psi)(\text{PC}(\phi)(M)) \subseteq \text{PC}(\phi; \psi)(M)$$

since the composition of pullbacks is a pullback again.

---

\[\text{PC}(\phi) : \text{Sub}(G) \to \text{Sub}(H)\]

\[\begin{array}{ccc}
A & \xrightarrow{f} & B \\
\downarrow{a} & & \downarrow{g} \\
G & \xrightarrow{\phi} & H \\
\end{array}\]

\[\begin{array}{ccc}
 & & C \\
\downarrow{b} & & \downarrow{c} \\
 & & \psi \\
\end{array}\]

---

\[\text{We remind that slice category } \text{Base}\downarrow G \text{ has pairs } (A, a) \text{ with } a : A \to G \text{ as objects, and arrows } f : A \to B \text{ in } \text{Base} \text{ such that } f; b = a \text{ as morphisms.}\]
The inverse inclusion \( PC(\phi; \psi)(M) \subset (PC(\phi); PC(\psi))(M) = PC(\psi)(PC(\phi)(M)) \) is ensured by pullback decomposition: Assume \((C, c) \in |PC(\phi; \psi)(M)|\) with witness \((\tau, d)\). We consider an arbitrary pullback \((b, g)\) of \((\psi, c)\). For the unique \(f : A \to B\) with \(f; g = h\) and \(f; b = a; \phi\) we know that \((a, f)\) is a pullback of \((\phi, b)\) thus we get \((B, b) \in |PC(\phi)(M)|\) with witness \((\alpha, f)\) and thus, finally, \((C, c) \in |PC(\psi)(PC(\phi)(M))|\) with witness \((\tau, d)\).

**Corollary 7.6** (Denotational Functor). The assignments \(G \mapsto \text{Sub}(G)\) and \(\phi \mapsto PC(\phi)\) define a functor \(\text{Sub} : \text{Base} \to \text{Cat}\).

**Remark 7.7** (Functoriality of Pullback Complements). Functoriality is based on two arguments. First, that the composition of pullbacks is a pullback again. Second, that the resulting pullback \((a, f)\) of \((\phi, b)\), in the proof above, entails \((B, b) \in |PC(\phi)(M)|\). Both arguments fail for a fixed choice of pullbacks. First, the composition of chosen pullbacks isn’t necessarily a chosen one. Second, the pullback \((a, f)\) of \((\phi, b)\) may be not a chosen one, i.e., \((B, b) \notin \text{Inst}(\phi)^{-1}(A, a)\) for the corresponding pseudo functor \(\text{Inst} : \text{Base} \to \text{Cat}\).

To simulate the effect of \(PC(\phi)\) by means of \(\text{Inst}(\phi)\), i.e., to abstract from the choice of pullbacks, we have to consider the closure \(\overline{M}\) of \(M\) under isomorphisms. We get then \(PC(\phi)(M) = \text{Inst}(\phi)^{-1}(\overline{M})\).

Before defining satisfaction we have to fix a semantics for the predicate symbols in our meta-signature. Dependencies represent requirements for the relations between predicates which should be met by any semantics.

**Definition 7.8** (Semantics of Meta-signature). A semantic interpretation of a meta-signature \(\alpha : \Pi \to \text{Base}\) is a mapping \([\cdot]\), which assigns to each predicate symbol \(P\) a set \([P] \subset |\text{Base}| \cdot \alpha(p)|\) of valid instances, where \([P]\) is assumed to be closed under isomorphisms.

The semantic interpretation is consistent if for any dependency arrow \(r : Q \vdash P\) in \(\Pi\), any instance \(\tau : O \to \alpha(P)\) of \(P\), and any pullback \((\tau^*, \mu)\) of \((\alpha(r), \tau)\) we have that \((O, \tau) \in [P]\) implies \((O^*, \tau^*) \in [Q]\). Below we will assume consistency by default.

\[
\begin{array}{ccc}
O^* & \xrightarrow{\mu} & O \\
\downarrow{\tau^*} & & \downarrow{\tau} \\
\alpha(Q) & \xrightarrow{\alpha(r)} & \alpha(P) \\
\end{array}
\xymatrix{ & O' \ar[d]^{d'} & A \ar[l]_{\tau^*} \ar[d]^{a} \\
\alpha(Q) \ar[r]_{\alpha(r)} & \alpha(P) \ar[l]_{\mu} & \tau \ar[u]^{\tau^*} \\
\end{array}
\]

Now we can define satisfaction of atomic constraints in semantic instances.

**Definition 7.9** (Satisfaction relation). Let \(a : A \to G\) be an instance of an object \(G\) in \(\text{Base}\). We say that this instance satisfies an atomic constraint \((P, d)\) over \(G\) and write \((A, a) \models_G (P, d)\), iff \((O, \tau) \in [P]\) for a pullback \((\tau, d')\) of \((d, a)\) (see the diagram in Definition 7.8).

Note, that the definition of satisfaction is independent of the representative \((\tau, d')\) since \([P]\) is assumed to be closed under isomorphisms.

Since pullbacks compose, the consistency assumption ensures soundness w.r.t. inference (see the diagram in Definition 7.8): If \((A, a) \models_G (P, d)\) and \(r : (Q, e) \vdash (P, d)\) is a labeled sequent, then \((A, a) \models_G (Q, e)\).

Composition and decomposition of pullbacks provide also a satisfaction condition for fibred semantics.
Corollary 7.10 (Satisfaction Condition). For any morphism \( \phi : G \to H \) in \textbf{Base}, any instance \( b : B \to H \) of \( H \), any pullback \( (a : A \to G, f : A \to B) \) of \((\phi, b)\), and any atomic constraint \((P, d)\) on \( G \) we have

\[
(A, a) \models_G (P, d) \iff (B, b) \models_H (P, d; \phi).
\]

The implication from left to right is ensured by pullback composition and the implication from right to left by pullback decomposition, respectively.

Satisfaction gives rise to adjoint functors similar to institutional frames: For any base object \( G \) in \textbf{Base} we obtain a functor \( \text{inst}(G) : \text{Sk}(G) \to \text{Sub}(G) \) where for any sketch \( S \) with base \( G \) the category \( \text{inst}(G)(S) \) is the full subcategory induced by those instances in \([\text{Base}] \downarrow G\) that satisfy all the atomic constraints in \( S \), i.e.,

\[
|\text{inst}(G)(S)| \stackrel{\text{def}}{=} \{(A, a) \in [\text{Base}] \downarrow G | \forall (P, d) \in |S| : (A, a) \models_G (P, d)\}.
\]

Analogously, we obtain a functor \( \text{sk}(G) : \text{Sub}(G) \to \text{Sk}(G) \) where for any full subcategory \( M \subseteq [\text{Base}] \downarrow G \) of instances, the category \( \text{sk}(G)(M) \) is the full subcategory of \( \text{Fm}(G) \) given by those atomic constraints which are satisfied by all instances in \( M \), i.e., we have

\[
|\text{sk}(G)(M)| \stackrel{\text{def}}{=} \{(P, d) \in |\text{Fm}(G)| | \forall (A, a) \in |M| : (A, a) \models_G (P, d)\}.
\]

Remark 7.11 (Closed Sketches). Note, that due to soundness, all the sketches \( \text{sk}(G)(M) \) are closed under dependencies.

A sketch \( S \) on \( G \) is said to be \textit{closed under dependencies} if \((P, d)\) in \( S \) entails also \((Q, \alpha(r); d)\) and \( r : (Q, \alpha(r); d) \vdash (P, d) \) in \( S \) for all dependency arrows \( r : Q \dashv P \) in \( \Pi \). More abstractly, this means that the projection functor \( \text{dom} : S \to \Pi \) with \( \text{dom}(P, d) = P \) and \( \text{dom}(r : (Q, \alpha(r); d) \vdash (P, d)) = (r : Q \dashv P) \) is a split op-fibration - an observation communicated to the first author by Zinovy Diskin in 2006.

The same argumentation as in Proposition 3.4 shows that the functor \( \text{sk}(G) \) is left-adjoint to the functor \( \text{inst}(G) \), written \( \text{sk}(G) \dashv \text{inst}(G) \), for any \( G \in \textbf{Base} \).

Finally, a similar argumentation as in Lemma 3.7 shows that our definition of pullback complement functors and the satisfaction condition in Corollary 7.10 ensure that the \textit{indexed frame condition}

\[
\text{inst}(G); PC(\phi) = \text{Sk}(\phi); \text{inst}(H) \quad \text{holds for all } \phi : G \to H \text{ in } \textbf{Base}
\]

Summarizing, we can present generalized sketches in a new perspective.

Theorem 7.12 (Sketch Frame). For any category \textbf{Base} with pullbacks, any meta-signature \( \alpha : \Pi \to \textbf{Base} \), and any consistent semantic interpretation \([..]\) of \( \alpha : \Pi \to \textbf{Base} \) the 5-tuple \((\text{Base}, \text{Sk}, \text{Sub}, \text{inst}, \text{sk})\) is an indexed partial order frame, called the \textbf{SKETCH FRAME}
Besides shedding new light on generalized sketches this result gives another evidence that
the concept Indexed Frame is an appropriate abstraction to present logical systems in a
uniform way. Indexed Frames comprise not only institutions and entailment systems but
also pseudo institutions with a model pseudo functor defined by chosen pullbacks. The last
observation in Remark 7.7 may even open a way to prove a corresponding result for arbitrary,
and not only pullback based, model pseudo functors.

8 Indexed Closure Operators

In the preceding sections of this article we have seen that adjunctions can be taken as the
fundamental concept in defining a Logical Framework. Indexed Partial Order and Indexed
Preorder Frames have been shown to be general enough to express uniformly the most repre-
sentative approaches to General Logic definitions.

This is one side of the history. Tightly related to adjunctions are co-monads, also called
coi-triples. In fact, every adjunction gives rise to two co-monads, while there are at least two
ways of obtaining a pair of adjoint functors from a co-monad.

On the other hand, co-monads are naturally connected to central algebraic and logical
concepts. Co-monads appear, for example, as the categorical counterpart of logical closure
operators, as discussed below. Based on this observation we investigate in this section how
comonads relate to the main logical concepts in our approach.

Categorically, definition 2.1 can be reformulated as the very definition of a co-monad (see

Definition 8.1. A co-monad \( G = (G, \epsilon, \delta) \) in \( B \) is an endofunctor \( G : B \to B \) together with
two natural transformations \( \epsilon : G \Rightarrow id_B \) and \( \delta : G \Rightarrow G^2 \) making the following diagrams
commute:

\[
\begin{array}{ccc}
G & \xrightarrow{\delta} & G^2 \\
\downarrow{\delta} & & \downarrow{G \delta} \\
G^2 & \xrightarrow{\delta \epsilon} & G^3
\end{array}
\quad \quad \quad
\begin{array}{ccc}
G & \xrightarrow{\delta} & G^2 \\
\downarrow{G \epsilon} & & \downarrow{\epsilon G} \\
G^2 & \xrightarrow{\epsilon G} & G
\end{array}
\]

In the diagrams above, \( G^n \) means \( G \) composed with \( G \) itself \( n \) times. The component of
\( G \cdot \epsilon \) at \( b \in B \) is \( \epsilon(G(b)) \), whereas \( G(\epsilon(b)) \) is the component of \( \epsilon \cdot G \) at \( b \). Similarly for \( \delta \).

If \( B \) is a preorder category then both diagrams trivially commute and we obtain the
following categorical reformulation of closure operators as specializations of the categorical
concept of co-monads to preorder categories.

Definition 8.2. A CO-CLOSURE OPERATOR on a preorder category \( P \) is a co-monad \( (cl, \epsilon, \delta) \),
where \( cl : P \to P \) is a functor, and \( \epsilon : cl \Rightarrow id_P \) and \( \delta : cl \Rightarrow cl^2 \) natural transformations.
Note that the above definition is actually a closure operator on the reverse preorder in the sense of definition 2.1. Thus, from now on, whenever we mention the expression closure operator, we ask the reader to reverse the preorder (if necessary).

Due to the co-monad conditions we have for each \( b \in |P| \) that \( \delta(b); \epsilon(\text{cl}(b)) = \text{id}_G(b) \). The assumption that \( P \) is a preorder ensures, in addition, \( \epsilon(\text{cl}(b)); \delta(b) = \text{id}_{\text{cl}^2}(b) \) so that \( \delta(b) \) is an isomorphism with inverse \( \epsilon(\text{cl}(b)) \).

\[
\text{cl} = \begin{array}{ccc}
\text{cl} & \text{cl}^2 & \text{cl} \\
\downarrow & \downarrow & \downarrow \\
\delta & \epsilon & \delta
\end{array}
\]

**Corollary 8.3.** For any closure operator \((\text{cl}, \epsilon, \delta)\), \( \delta : G \Rightarrow G^2 \) is a natural isomorphism. In case \( P \) is a partial order category we have, moreover, that \( \text{cl} = \text{cl}^2 \) and \( \delta \) is the identical natural transformation on \( \text{cl}^2 \).

The following fact will be helpful as we proceed.

**Fact 8.4.** Let \( F : C \rightarrow B \) and \( U : B \rightarrow C \) be functors. Then \( F \dashv U \) with unit \( \eta : \text{id}_C \Rightarrow F; U \) and co-unit \( \epsilon : U; F \Rightarrow \text{id}_B \) if and only if

\[
(U \cdot \eta); (\epsilon \cdot U) = \text{id}_U \tag{21}
\]

\[
(\eta \cdot F); (F \cdot \epsilon) = \text{id}_F \tag{22}
\]

One of the most fundamental facts about co-monads is provided by the following fact discovered by P. Huber [21].

**Fact 8.5.** Let \( U : B \rightarrow C \) have a left adjoint \( F : C \rightarrow B \) with unit \( \eta : \text{id}_C \Rightarrow F; U \) and co-unit \( \epsilon : U; F \Rightarrow \text{id}_B \). Then \((U; F, \epsilon, U \cdot \eta, F)\) is a co-monad on \( B \).

The fact above specializes for preorder categories to the well-known fact that any Galois correspondence provides a closure operator and this specialization can be directly applied to indexed preorder frames.

**Proposition 8.6.** Given an Indexed Frame \( \mathcal{IF} = (\text{Ind}, \text{Syn}, \text{Den}, \text{sem}, \text{fml}) \) with \( \text{Syn} : \text{Ind} \rightarrow \text{PRE} \), we have for each \( i \in |\text{Ind}| \), a closure operator \((\text{cl}(i), \epsilon(i), \delta(i))\) on \( \text{Syn}(i) \) given by the functor \( \text{cl}(i) = \text{sem}(i) ; \text{fml}(i) : \text{Syn}(i) \rightarrow \text{Syn}(i) \) together with natural transformations \( \epsilon(i) : \text{cl}(i) \Rightarrow \text{id}_{\text{Syn}(i)} \) and \( \delta(i) \overset{\text{def}}{=} \text{sem}(i) \cdot \eta(i) \cdot \text{fml}(i) : \text{cl}(i) \Rightarrow \text{cl}(i)^2 \). The functor \( \text{cl}(i) \) is called the “logical closure operator” at index \( i \).

**Corollary 8.7.** Let \( \mathcal{IF} = (\text{Ind}, \text{Spec}, \text{Sub}, \text{mod}, \text{th}) \) and \( \mathcal{EF} = (\text{Sign}, \text{Spec}, \text{Ent}, \text{em}, \text{clo}) \) be respectively institutional and entailment frames. Then for each \( \Sigma \in |\text{Sign}| \),

\[
(\text{cl}_\mathcal{IF}(\Sigma), \epsilon_\mathcal{IF}(\Sigma), \delta_\mathcal{IF}(\Sigma)) \quad \text{and} \quad (\text{cl}_\mathcal{EF}(\Sigma), \epsilon_\mathcal{EF}(\Sigma), \delta_\mathcal{EF}(\Sigma))
\]

are respectively closure operators, where:
\( \text{for each } i \in |\text{Ind}| \) a functor \( \zeta(i) : C(i) \to D(i) \);

- \( \text{for each } \sigma : i \to j \in \text{Ind} \) and the corresponding two functors \( C(\sigma) : C(i) \to C(j) \) and \( D(\sigma) : D(i) \to D(j) \), a comparison cell (a natural transformation) \( \zeta(\sigma) : C(\sigma) \Rightarrow \zeta(i) ; D(\sigma) : C(i) \to D(j) \).

\[ \begin{array}{ccc}
\sigma & C(\sigma) & \zeta(\sigma) \\
\downarrow & \downarrow & \downarrow \\
C(j) & \zeta(j) & D(j)
\end{array} \]

Remark 8.9 (Uniqueness of Comparison Cells). Note, that there can be at most one natural transformation \( \zeta(\sigma) \) since \( D(j) \) is a preorder category. In the same way, the usual coherence condition for lax indexed functors is also trivially satisfied. That is, for any \( \sigma : i \to j \) and \( \tau : j \to k \) in \( \text{Ind} \) we have \( \zeta(\sigma ; \tau) = (C(\sigma) \cdot \zeta(\tau)) ; (\zeta(\sigma) \cdot D(\tau)) \) since \( D(k) \) is preorder category.

\[ \begin{array}{ccc}
i & C(i) & \zeta(i) \\
\downarrow & \downarrow & \downarrow \\
\sigma & C(\sigma) & \zeta(\sigma) \\
\downarrow & \downarrow & \downarrow \\
C(j) & \zeta(j) & D(j) \\
\downarrow & \downarrow & \downarrow \\
\tau & C(\tau) & \zeta(\tau) \\
\downarrow & \downarrow & \downarrow \\
C(k) & \zeta(k) & D(k)
\end{array} \]

\[ \begin{array}{ccc}
C(i) & \zeta(i) & D(i) \\
\downarrow & \downarrow & \downarrow \\
C(\sigma ; \tau) & \zeta(\sigma ; \tau) & D(\sigma ; \tau)
\end{array} \]
**Lemma 8.10** (Local Adjointness and Lax). Let

\[ I\mathcal{F} = (\text{Ind}, \text{Syn}, \text{Den}, \text{sem}, \text{fml}) \]

be an indexed frame with \( \text{Syn} : \text{Ind} \to \text{PRE} \). Then, the family of functors \( \text{fml}(i), i \in |\text{Ind}| \) can be extended to a lax indexed functor \( \text{fml} : \text{Den} \xrightarrow{\ll} \text{Syn} \).

**Proof.** First we have to show that for each \( \sigma : i \to j \) there is a comparison cell \( \text{fml}(\sigma) : \text{Den}(\sigma) ; \text{fml}(j) \Rightarrow \text{fml}(i) ; \text{Syn}(\sigma) : \text{Den}(i) \to \text{Syn}(j) \).

Now, note that, since \( \text{fml}(i) \dashv \text{sem}(i), \text{fml}(j) \dashv \text{sem}(j) \) by assumption, we have, for instance, the unit \( \eta : \text{id}_{\text{Den}(i)} \Rightarrow \text{fml}(i) ; \text{sem}(i) \) and co-unit \( \epsilon : \text{sem}(j) ; \text{fml}(j) \Rightarrow \text{id}_{\text{Syn}(j)} \), as depicted by the diagrams below, where \( f : a \to b \) is an arrow in \( \text{Den}(i) \) and \( g : c \to d \) in \( \text{Syn}(j) \).

Thus, we define \( \text{fml}(\sigma) \) as

\[ \text{fml}(\sigma) \overset{\text{def}}{=} (\eta, \text{Den}(\sigma) \cdot \text{fml}(j)) : (\text{fml}(i) \cdot \text{Syn}(\sigma) \cdot \epsilon) \]

which is easily seen to be well-defined by pasting together the following two-cells:

\[ \text{Den}(i) \xrightarrow{\text{id}} \text{Den}(i) \xrightarrow{\text{fml}(i)} \text{Den}(j) \xrightarrow{\text{fml}(j)} \text{Syn}(j) \]

and

\[ \text{Den}(i) \xrightarrow{\text{fml}(i)} \text{Syn}(i) \xrightarrow{\text{Syn}(\sigma)} \text{Syn}(j) \xrightarrow{\epsilon} \text{Syn}(j) \xrightarrow{\text{id}} \text{Syn}(j) \]

Note that this pasting is well-defined because, by assumption, we have \( \text{Syn}(\sigma) ; \text{sem}(j) = \text{sem}(i) ; \text{Den}(\sigma) \).
We can define now an (abstract) logic also by means of indexed closure operators.

**Definition 8.11.** An **Indexed Closure Operator** is a structure of the form \( \mathcal{ICO} = (\text{Ind}, \text{Syn}, \text{cl}, \epsilon, \delta) \), with \( \text{Syn} : \text{Ind} \rightarrow \text{PRE} \) an Indexed Category, \( \text{cl} : \text{Syn} \Rightarrow \text{Syn} \) a Lax Indexed Functor, and for each \( i \in |\text{Ind}| \) natural transformations \( \epsilon(i) : \text{cl}(i) \Rightarrow \text{id}_{\text{Syn}(i)} \), \( \delta(i) : \text{cl}(i) \Rightarrow \text{cl}(i)^2 \) constituting a closure operator on \( \text{Syn}(i) \):

\[
\begin{array}{ccc}
\text{Syn}(i) & \xrightarrow{\text{cl}(i)} & \text{Syn}(i) \\
\text{Syn}(\sigma) & \xrightarrow{\epsilon(i)} & \text{Syn}(\sigma) \\
\text{Syn}(j) & \xrightarrow{\delta(i)} & \text{Syn}(j)
\end{array}
\]

Since the composition of an indexed functor and a lax indexed functor results again in a lax indexed functor, we obtain by proposition 8.6 and lemma 8.10 the main result of this section.

**Theorem 8.12.** Any indexed frame \( \mathcal{IF} = (\text{Ind}, \text{Syn}, \text{Den}, \text{sem}, \text{fml}) \) with \( \text{Syn} : \text{Ind} \rightarrow \text{PRE} \) defines an indexed closure Operator \( (\text{Ind}, \text{Syn}, \text{cl}, \epsilon, \delta) \).

**Proof.** According to proposition 8.6 we have for each \( i \in |\text{Ind}| \) a closure operator \( (\text{cl}(i), \epsilon(i), \delta(i)) \) on \( \text{Syn}(i) \) with a functor \( \text{cl}(i) = \text{sem}(i); \text{fml}(i) : \text{Syn}(i) \rightarrow \text{Syn}(i) \) and natural transformations \( \epsilon(i) : \text{cl}(i) \Rightarrow \text{id}_{\text{Syn}(i)} \) and \( \delta(i) \overset{\text{def}}{=} \text{sem}(i) \cdot \eta(i) \cdot \text{fml}(i) : \text{cl}(i) \Rightarrow \text{cl}(i)^2 \). Moreover, we have, according to lemma 8.10 for each \( \sigma : i \rightarrow j \) in \( \text{Ind} \) a natural transformation \( \text{fml}(\sigma) : \text{Den}(\sigma); \text{fml}(j) \Rightarrow \text{fml}(i); \text{Den}(\sigma) \). This allows us to define the required natural transformation \( \text{cl}(\sigma) : \text{Syn}(\sigma); \text{cl}(j) \Rightarrow \text{cl}(i) ; \text{Syn}(\sigma) \) by

\[
\text{cl}(\sigma) \overset{\text{def}}{=} \text{sem}(i) \cdot \text{fml}(\sigma)
\]

\[
\begin{array}{ccc}
i & \xrightarrow{\text{sem}(i)} & \text{Den}(i) \\
\text{Syn}(i) & \xrightarrow{\text{fml}(i)} & \text{Syn}(i) \\
\text{Den}(\sigma) & \xrightarrow{\text{sem}(\sigma)} & \text{Syn}(\sigma) \\
\text{Den}(j) & \xrightarrow{\text{sem}(j)} & \text{Syn}(j)
\end{array}
\]

Note, that \( \text{cl}(\sigma) \) becomes indeed a natural transformation form \( \text{Syn}(\sigma); \text{cl}(j) \) to \( \text{cl}(i); \text{Syn}(\sigma) \) since we have, by assumption, \( \text{Syn}(\sigma); \text{sem}(j) = \text{sem}(i); \text{Den}(\sigma) \).

Theorem 8.12 can be applied to Institutional and Entailment Frames providing the well-known semantic or syntactic logical consequence operator, respectively (see corollary 8.7).

### 9 From Indexed Closure Operators to Indexed Frames

Given a co-monad there are (possibly) many ways of factoring it as adjoint pairs. Two canonical constructions arise from a co-monad as adjoint pairs — the Kleisli and the Eilenberg-Moore constructions. The former is initial and the latter is final in the category of all such factorizations. For quasi-idempotent co-monads \( (G^2 \cong G) \), as in case of indexed preorder frames, both constructions are equivalent. Let us briefly describe these constructions focusing on the logical meaning they have in the context of Indexed Frames.

The following definition can be found, for instance, in [2, 24, 3].

---

29
Definition 9.1 (Kleisli Category). Given a co-monad $G = (G, \epsilon, \delta)$ on $B$, the Kleisli category is the category $K(G)$, having the same objects of $B$ as its objects. Morphisms from $a$ to $b$ are exactly all the morphisms from $G(a)$ to $b$ in $B$. For a morphism $f : G(a) \to b$ in $B$ we will indicate by $f^K : a \to b$ its interpretation as a ”Kleisli morphism”, i.e., as a morphism in $K(G)$. The identical morphism from $a$ to $a$ in $K(G)$ is given by the morphism $\epsilon(a) : G(a) \to a$, i.e., $id_a \overset{def}{=} \epsilon(a)^K$. The composition of Kleisli morphisms $f^K : a \to b$ and $g^K : b \to c$ is defined by
$$f^K; g^K \overset{def}{=} (\delta(a); G(f)^K; g^K) : a \to c.$$

Remark 9.2. Note, that $K(G)$ becomes a preorder category if $B$ is a preorder category. If $B$ is a partial order category, however, $K(G)$ will be, in general, only a preorder category.

Example 9.3 (Theory Presentations). In an Institutional Frame $IF$, the Kleisli category $K(cl_{\Sigma}(\Sigma))$ for each $\Sigma \in |Sign|$ is the category of Theory Presentations $\Gamma \in |Spec(\Sigma)|$, and there is a morphism from $\Gamma \in |Spec(\Sigma)|$ into $\Delta \in |Spec(\Sigma)|$ iff $cl_{\Sigma}(\Sigma)(\Gamma) \supseteq \Delta$. In other words, $\Gamma \models_{\Sigma} \Delta$. This category is well-known for the algebraic logicians. It is worth noticing that, since there is at most one morphism from $\Gamma$ into $\Delta$, they are isomorphic as objects in $K(cl_{\Sigma}(\Sigma))$, whenever they are logically equivalent, that is $\Gamma \models_{\Sigma} \Delta$ and $\Delta \models_{\Sigma} \Gamma$.

On the other hand, theories are captured by the following construction [2, 24, 3].

Definition 9.4 (Eilenberg-Moore Category). Also known as the category of co-algebras for a co-monad, the Eilenberg-Moore construction $B^G$ for a co-monad $G = (G, \epsilon, \delta)$ on $B$ has $G$-algebras as objects, which are pairs $(a, \alpha)$, with $\alpha : a \to G(a)$ a morphism in $B$, as objects satisfying the following axioms, for each $a \in |B|$:

$$a \xrightarrow{\alpha} Ga \xrightarrow{\epsilon(a)} a \quad \text{and} \quad a \xrightarrow{\alpha} Ga \xrightarrow{\delta(a)} GGa.$$

Note that if $B$ is a preorder category, then the axioms above are trivially satisfied. Now a morphism between $G$-algebras $(a, \alpha)$ and $(b, \beta)$ is any morphism $f : a \to b$ in $B$ for which, the following diagram commutes:

$$a \xrightarrow{\alpha} Ga \xrightarrow{f} Gb \xrightarrow{G\beta} Gb.$$

The composition in $B^G$ is provided by the composition on $B$ itself.

Example 9.5 (Theories). In an Institutional Frame $IF$, the Eilenberg-Moore category $Spec(\Sigma)^cl_{\Sigma}(\Sigma)$ for each $\Sigma \in |Sign|$, is the category of Theories, i.e., of sets of sentences satisfying the “closedness condition” $\Delta \supseteq cl_{\Sigma}(\Sigma)(\Delta)$, i.e., satisfying the condition $\Delta = cl_{\Sigma}(\Sigma)(\Delta)$, since we have $cl_{\Sigma}(\Sigma)(\Gamma) \supseteq \Gamma$ for all sets of sentences. In such a way, $\Delta \supseteq \Gamma$ means also $cl(\Sigma)(\Delta) \supseteq cl(\Sigma)(\Gamma)$ for theories $\Delta$ and $\Gamma$ thus $Spec(\Sigma)^cl_{\Sigma}(\Sigma)$ is indeed the full subcategory of $Spec(\Sigma)$ given by all theories.

30
The above pair of examples on Kleisli and Eilenberg-Moore constructions show equivalent categories. For closure operators, in general, both constructions provide equivalent categories thus we can concentrate on one of these constructions. We will look now at the adjunction provided by the Kleisli construction.

**Proposition 9.6** (Kleisli Functors). For any co-monad \( G = (G, \epsilon, \delta) \) on \( \mathcal{B} \) the assignments \( a \mapsto a \) and \( (f : a \to b) \mapsto ((\epsilon(a); f^K : a \to b) \) define a functor \( U_G : \mathcal{B} \to K(G) \). Moreover, the assignments \( a \mapsto G(a) \) and \( (g^K : a \to b) \mapsto (\delta(a); G(g) : G(a) \to G(b)) \) define a functor \( F_G : K(G) \to \mathcal{B} \)

\[
\begin{array}{ccc}
G a & \xrightarrow{\epsilon(a)} & a \\
\downarrow f & & \downarrow f \\
b & \xleftarrow{\delta(a)} & Ga
\end{array}
\]

left-adjoint to \( U_G \), \( F_G \dashv U_G \), and such that \( U_G; F_G = G \).

\[
\begin{array}{ccc}
\mathcal{B} & \xleftarrow{F_G} & K(G) \\
\downarrow U_G & & \downarrow \epsilon \\
\end{array}
\]

\( \epsilon : U_G; F_G \Rightarrow id_\mathcal{B} \) is the co-unit and the unit \( \eta_G : id_{K(G)} \Rightarrow F_G; U_G \) is given by \( \eta_G(a) = (id^K_{G(a)} : a \to G(a)) \) for each \( a \) in \( K(G) \).

Finally, the co-monad that we construct due to fact 8.5 out of the adjunction \( (F_G, U_G, \eta_G, \epsilon) \) is exactly the original co-monad \( (G, \epsilon, \delta) \), i.e., in addition to \( G = U_G; F_G \) we have also \( \delta = U_G \cdot \eta_G \cdot F_G \).

**Example 9.7** (Semantical Entailment). For an Institutional Frame \( \mathcal{IF} \), the functor \( U_{cl_\Pi(\Sigma)} : Spec(\Sigma) \to K(cl_\Pi(\Sigma)) \) is the identity on objects and assigns to each inverse inclusion \( \Gamma \supseteq \Delta \) the semantical entailment \( \Gamma \models_\Sigma \Delta \). And, the functor \( F_{cl_\Pi(\Sigma)} : K(cl_\Pi(\Sigma)) \to Spec(\Sigma) \) translates a semantical entailment \( \Gamma \models_\Sigma \Delta \) into an inverse inclusion \( cl_\Pi(\Sigma)(\Gamma) \supseteq cl_\Pi(\Sigma)(\Delta) \) between the corresponding theories.

For preorder categories proposition 9.6 allows to construct for any closure operator a Galois correspondence that defines exactly this closure operator. And this specialization can be directly applied to Indexed Closure Operators.

**Proposition 9.8.** Given an indexed closure operator \( ICO = (\text{Ind}, \text{Syn}, cl, \epsilon, \delta) \) we have for each \( i \in \text{Ind} \), a Galois correspondence (adjunction) between preorder categories

\( (F_{cl}(i), U_{cl}(i), \eta_{cl}(i), \epsilon(i)) \)

with functors \( F_{cl}(i) \overset{def}{=} F_{cl(i)} : K(cl(i)) \to \text{Syn}(i) \), \( U_{cl}(i) \overset{def}{=} U_{cl(i)} : \text{Syn}(i) \to K(cl(i)) \) such that \( F_{cl}(i) \dashv U_{cl}(i) \), and \( U_{cl}(i); F_{cl}(i) = cl(i) \) together with natural transformations \( \epsilon(i) : cl(i) \Rightarrow id_{\text{Syn}(i)} \) and \( \eta_{cl}(i) \overset{def}{=} \eta_{cl(i)} : id_{K(cl(i))} \Rightarrow F_{cl}(i); U_{cl}(i) \).
Proof. It follows directly from definition 8.11, remark 9.2 and proposition 9.6.

In the next step we can show that the "local" Galois correspondences give rise to a global natural transformation.

**Proposition 9.9.** Given an indexed closure operator \( ICO = (\text{Ind}, \text{Syn}, cl, \epsilon, \delta) \) the assignments \( i \mapsto K(cl(i)) \) can be extended to a functor \( K(cl) : \text{Ind} \rightarrow \text{PRE} \) such that the functors \( U_{cl}(i) : \text{Syn}(i) \rightarrow K(cl(i)) \) constitute a natural transformation \( U_{cl} : \text{Syn} \Rightarrow K(cl) \).

Proof. First, we show that for each \( \sigma : i \rightarrow j \) in \( \text{Ind} \) the functor \( \text{Syn}(\sigma) : \text{Syn}(i) \rightarrow \text{Syn}(j) \) can be extended to a functor \( K(cl(\sigma)) : K(cl(i)) \rightarrow K(cl(j)) \) such that \( U_{cl}(i); K(cl(\sigma)) = \text{Syn}(\sigma); U_{cl}(j) \):

\[
\begin{array}{ccc}
\text{Syn}(i) & \xrightarrow{U_{cl}(i)} & K(cl(i)) \\
\downarrow \text{Syn}(\sigma) & & \downarrow \text{K}(cl(\sigma)) \\
\text{Syn}(j) & \xrightarrow{U_{cl}(j)} & K(cl(j)) \\
\end{array}
\]

For each \( i \in \text{Ind} \) we have \( |\text{Syn}(i)| = |K(cl(i))| \) and \( U_{cl}(i) \) is the identity on objects thus we can set \( K(cl(\sigma))(a) \overset{\text{def}}{=} \text{Syn}(\sigma)(a) \) for all \( a \in |K(cl(i))| \) and obtain commutativity for all objects \( a \in |\text{Syn}(i)| \):

\[
K(cl(\sigma))(U_{cl}(i)(a)) = K(cl(\sigma))(a) = \text{Syn}(\sigma)(a) = U_{cl}(j)(\text{Syn}(\sigma)(a))
\]

Further the natural transformation \( cl(\sigma) \) allows us to assign to any Kleisli morphism \( f^K : a \rightarrow b \), i.e., any morphism \( f : cl(i)(a) \rightarrow b \) in \( \text{Syn}(i) \) the morphism

\[
K(cl(\sigma))(f^K) \overset{\text{def}}{=} (cl(\sigma)(a); \text{Syn}(\sigma)(f))^K \quad \text{in} \quad K(cl(j)).
\]

Since \( K(cl(j)) \) is a preorder category these assignments are trivially compatible with identities and composition thus we get a functor \( K(cl(\sigma)) : K(cl(i)) \rightarrow K(cl(j)) \).

It remains to show commutativity for arbitrary morphisms \( g : a \rightarrow b \) in \( \text{Syn}(i) \): Due to the definition of \( U_{cl}(i) \) and \( K(cl(\sigma)) \) we have

\[
K(cl(\sigma))(U_{cl}(i)(g)) = (cl(\sigma)(a); \text{Syn}(\sigma)(\epsilon(i)(a)); \text{Syn}(\sigma)(g))^K
\]

Since \( \text{Syn}(j) \) is a preorder category we can assume

\[
cl(\sigma)(a); \text{Syn}(\sigma)(\epsilon(i)(a)) = \epsilon(j)(\text{Syn}(\sigma)(a))
\]

thus we get finally the required equality due to the definition of \( U_{cl}(j) \):

\[
K(cl(\sigma))(U_{cl}(j)(g)) = (\epsilon(\text{Syn}(\sigma)(a)); \text{Syn}(\sigma)(g))^K = U_{cl}(j)(\text{Syn}(\sigma)(g)).
\]
10 SEMANTICAL ENTAILMENT FOR INDEXED PREORDER FRAMES

Syn : Ind \rightarrow PRE is a functor and the functors K(cl(\sigma)) coincide with the functors Syn(\sigma) on objects thus the assignments \sigma \mapsto K(cl(\sigma)) are compatible w.r.t. identities and composition since all the categories K(cl(i)) are preorder categories.

Now we can formulate the main result of this section that can be seen as an inverse to theorem 8.12.

**Theorem 9.10.** Any Indexed Closure Operator ICO = (Ind, Syn, cl, \epsilon, \delta) defines an indexed preorder frame (Ind, Syn, K(cl), Ucl, Fcl).

**Proof.** Follows immediately from proposition 9.8 and proposition 9.9.

In the discussion above, the Eilenberg-Moore construction associated to the presentation of the logic L in terms of its Indexed Closure Operator cl resembles the syntactical model built up from maximal consistent set of sentences, largely used in proof of completeness, while the Kleisli construction resembles the algebraically defined Lindenbaum models for a logic L based on equivalence classes of equiprovable sentences. This connection between proofs of completeness and solutions to factoring of co-monads is, apart from worth of mentioning, out of scope of the present article.

10 Semantical Entailment for Indexed Preorder Frames

The concept of General Logic [36] is based on semantical entailment in institutions (see definition 5.1). In the last section we have seen, in the examples 9.3 and 9.7, that the corresponding semantical entailment in institutional frames can be described, on an abstract categorical level, by the Kleisli construction.

Following this observation, we want to generalize in this section semantical entailment to arbitrary indexed preorder frames. The key for this generalization is the well-known fact that the Kleisli adjunction is initial in the category of all adjoint factorizations of a given co-monad. By lifting up this result to indexed preorder categories we can define semantical entailment for arbitrary indexed preorder frames and we can show, finally, that semantical entailment provides for any indexed preorder frame a complete Indexed Logic in the sense of definition 5.2.

Due to fact 8.5 any adjunction \((F, U, \eta, \epsilon)\) between categories B and C gives rise to a co-monad \((G, \epsilon, \delta) = (U; F, \epsilon, U \cdot \eta, F)\) on B and, due to proposition 9.6, the Kleisli construction provides for this co-monad another adjunction \((F_G, U_G, \eta_G, \epsilon)\) with \(U_G; F_G = G(U; F)\).

A complete proof of the initiality of the Kleisli category can be found in [24]. Here we need only the initiality w.r.t. the original adjunction.

**Fact 10.1.** For any adjunction \((F, U, \eta, \epsilon)\) between categories B and C there exists a unique functor \(E_G : K(G) \rightarrow C\) such that \(U_G; E_G = U, E_G; F = F_G\) and \(\epsilon \cdot E_G = E_G \cdot \eta.\)

33
$E_G$ is given by $E_G(a) = U(a)$ for all objects $a$ in $K(G)$ and $E_G(f^K) = \eta(U(a)); U(f)$ for all morphisms $f^K : a \to b$ in $K(G)$.

Since any adjunction $(F,U,\eta,\epsilon)$ between categories $B$ and $C$ provides a bijection between the hom-sets $K(G)(a,b) = B(FUa,b)$ and $C(Ua,Ub)$ for any objects $a$ and $b$ in $B$ we have immediately:

**Corollary 10.2.** For any adjunction $(F,U,\eta,\epsilon)$ between categories $B$ and $C$ the functor $E_G : K(G) \to C$ is full.

Now, we lift up the initiality of Kleisli categories to indexed preorder frames.

**Proposition 10.3.** Given an indexed preorder frame $IF = (\text{Ind}, \text{Syn}, \text{Den}, \text{sem}, \text{fml})$ we consider the corresponding indexed closure operator $(\text{Ind}, \text{Syn}, cl, \epsilon, \delta)$, due to theorem 8.12. Then the functors $E_{cl}(i) \overset{\text{def}}{=} E_{cl}(i) : K(cl(i)) \to \text{Den}(i)$, which can be construct due to fact 10.1, constitute a natural transformation $E_{cl} : K(cl) \Rightarrow \text{Den}$ such that $U_{cl}; E_{cl} = \text{sem}$.

**Proof.** For any $\sigma : i \to j$ in $\text{Ind}$ we have to show that $E_{cl}(i); \text{Den}(\sigma)) = K(cl(\sigma)); E_{cl}(j)$:

\[
\begin{array}{ccc}
\text{Syn}(i) & \xrightarrow{U_{cl}(i)} & K(cl(i)) & \xrightarrow{E_{cl}(i)} & \text{Den}(i) \\
\text{Syn}(j) & \xrightarrow{U_{cl}(j)} & K(cl(j)) & \xrightarrow{E_{cl}(j)} & \text{Den}(j)
\end{array}
\]

For any $a \in |K(cl(i))|$ we have, due to the definition of $E_{cl}(i)$ and $K(cl(\sigma))$:

\[
\begin{align*}
\text{Den}(\sigma)(E_{cl}(i)(a)) &= \text{Den}(\sigma)(\text{sem}(i)(a)) \\
E_{cl}(j)(K(cl(\sigma))(a)) &= \text{sem}(j)(\text{Syn}(\sigma)(a))
\end{align*}
\]

thus commutativity for objects is ensured by the indexed frame condition

$\text{sem}(i); \text{Den}(\sigma) = \text{Syn}(\sigma); \text{sem}(j)$.

The commutativity for morphisms follows trivially since $\text{Den}(j)$ is a preorder category. Finally, $U_{cl}; E_{cl} = \text{sem}$ follows immediately from fact 10.1.

Finally, we can formulate our last theorem summarizing the properties of semantical entailment.

**Theorem 10.4.** Any indexed preorder frame $IF = (\text{Ind}, \text{Syn}, \text{Den}, \text{sem}, \text{fml})$ defines a complete indexed logic

\[(\text{Ind}, \text{Syn}, \text{Den}, \text{K(cl)}, \text{sem}, \text{fml}, U_{cl}, F_{cl})\]

with the indexed functor $E_{cl} : K(cl) \Rightarrow \text{Den}$ such that $sem = U_{cl}; E_{cl}$.

**Proof.** Follows immediately from definition 5.2, theorem 8.12, theorem 9.10, and proposition 10.3 where completeness is ensured by corollary 10.2.
11 Concluding Remarks

Logical Systems are paramount to almost every subject in computer science. This vast number of application areas have had a deep influence on us and thus on how we perceive what a formal specification of a logical system should be. *Institution-like* presentation of logics have shown to be of great use for computer scientists interested in the development of meta-theories for specification, programming, and also in the reuse and borrowing of logical tools. The latter is especially based on appropriate notions of maps between logical systems [7, 20, 34], which we have not considered in this paper. On the other hand, axiomatizations of logics using simple concepts like *satisfaction relations, entailment relations* and *closure operators* are universal tools which are familiar to anyone with the slightest interest in logic itself.

Using Lawvere’s essential idea that the fundamental relationship between *syntax* and *semantics* can be precisely formulated by *Galois connections* or *adjoint functors*, we have shown that even complex formalizations of logics like Institutions of Goguen and Burstall and Entailment Systems of Meseguer are in their essence, a family of local adjoint situations between the syntactical and semantical aspects (of each index) satisfying some kind of natural coherence condition with respect to change of syntax. Indexed categories here are used essentially to fix changes of languages, which is essential for computer science applications. But it is by using adjoint functors that we establish a uniform language to describe the essential ideas behind satisfaction relations, entailment relations, and closure operators as well.

One should not perceive the tools introduced here as yet another formal notion of logic. They should not be understood in any way as substitutes for any candidate of what a logical system really is. In fact, these notions were brought to the surface by a careful examination of the concepts exemplified in this paper. They serve here primarily as a tool to present these complex systems in a simple, elegant and uniform way. Yet, they become also witness to the fact that adjoint situations arise everywhere, especially when one is looking for the essential relationship between syntax and semantics in any given logical system.

References


