The IFF Institution (meta) Ontology (IFF-INS)

The Context of Institutions

`INS`

Possibly every researcher in the theory of logic has used institutions either directly or indirectly, whether consciously or not. The author was no exception*. The first person to use institutions was Tarski. However, he did not view these abstractly and did not use morphisms either within or without. The discoverers of institutions were Goguen and Burstall. Institutions are very powerful and useful concepts. However, several features of institutions are rather anomalous. These features, which are discussed below, could lead the theoretical researcher to question the authenticity or correctness of the institution concept. However, since institutions are very natural concepts, these anomalous features need to be enrolled into the theoretical developments.

The first anomalous feature: the placement of their axiomatization (as in this document) is definitely not straightforward. More precisely, the question is whether to place them in the upper or top metalevel of the IFF. Normally, a module is place in the metalevel corresponding to the "size" of its main collection, which is determined by the size of their component collections. Modules with "small" component collections and hence "large" main collections are place in the 1st or lower metalevel, modules with "large" component collections and hence "very large" main collections are place in the 2nd or upper metalevel, etc. For one example, first order logic languages are place in the lower metalevel, since all of their components are small sets and their main collection is a class (large collection). For another example, the original category theory meta-ontology was place in the upper metalevel, since the component object and morphism collections are "large". From this reasoning, institutions should be place in the 3rd or top metalevel, since their model components are "very large" collections, even though their sentence components are just classes. However, other than institutions, all the IFF modules yet axiomatized are definable within the metalanguage at their metalevel. This means that they use terminology only from that metalevel upwards. In fact, an even stronger statement can be made – except for the core exponent module, most IFF modules use terminology from only their own metalevel and the one directly above. However, just look (below) at the terminology used (indicated by the colors) in the basic axiomatization for an institution. You will see black, red, blue and green – colors for all four metalevels. From this reasoning, institutions should be place in the upper metalevel (the green is only used for core terminology, and the new view of metalanguages allows them to reference all the core). Another justification for the placement of a module is the location of its illustrative examples – it should be place just above their level. In fact, this criterion can be used to decide,

* The author (Robert E. Kent) did not read about institutions until early 2003. However, in his development and championship of the Information Flow Framework, he was effectively using institutions as the "truth structures" introduced in the theory of Information Flow (Barwise and Seligman). One might say that he has been doing institutions all his adult theoretical life, but just didn't know it.

when the previous two criteria are conflicting. For example, the axiomatization for 2-categories is correctly place in the top metalevel since its most illustrative example is CAT, the category of all categories, whose assertion is correctly place within IFF-CAT at the upper metalevel. The examples of institutions occur at the lower metalevel, implying that the institution meta-ontology (IFF-INS) should be place in the upper metalevel. Hence, this is where the IFF-INS will be situated. In summary, it will solve the problem with the metalanguage, but will not conform to the usual criteria using the size of the component structures.

The second anomalous feature: institutions are defined with model categories, but these are not used in the basic satisfaction condition. In fact, these are not compatible with the satisfaction relation in the manner of bimodules. As will be developed below, this fact has several ramifications. Each institution has an associated institution that is used, either implicitly or explicitly, in the development of the formal conceptual structures of institutions known as specifications and theories. In addition, this fact is useful for the development of the concrete structure of the signature aspect of institutions.

The terminology listed in Table 1 is for the IFF Institution (meta) Ontology. This is part of the **IFF-Upper** metalanguage. The 21 terms listed here represent only the barest beginning. We anticipate that there will eventually be 100+ terms in the final version of the IFF-INS.

Table 1: The Institutions part of the Upper Metalanguage

	UR$object	UR$morphism	UR$relation
`INS`			
`INS.OBJ`	`institution`	`signature sentence model`	`satisfaction`
		`classification concept-lattice`	
`INS.MOR`	`institution-morphism`	`source target` `signature sentence model`	
		`classification concept-lattice`	
	`composable-pair`	`composition identity` `morphism0 morphism1`	`composable`

Unless the generic set of class-indexed collections forms another collection, the category of institutions is larger than very large, since every institution has a very large functor as a component.

Preliminaries

Basics

`set`
`SET`
`CAT`

Let Set denote the (large) category of small sets and functions between them, let SET denote the (very large) category of classes (large sets) and class functions, and let |CAT| denote the (very large) category underlying the 2-category CAT of large categories, functors and natural transformations. For convenience, we redefine these here. There is an object functor $|\text{-}| : |\mathsf{CAT}| \rightarrow \mathsf{SET}$, that maps a (large) category to its class of objects and maps a functor to its object class function.

```
(1) (CAT$category set$Set)
    (= (CAT$object set$Set) set$set)
    (= (CAT$morphism set$Set) set.ftn$function)
    (= (CAT$source set$Set) set.ftn$source)
    (= (CAT$target set$Set) set.ftn$target)
    (= (CAT$composition set$Set) set.ftn$composition)
    (= (CAT$identity set$Set) set.ftn$identity)

(2) (vlrg.cat$category SET$SET)
    (= (vlrg.cat$object SET$SET) SET$class)
    (= (vlrg.cat$morphism SET$SET) SET.FTN$function)
    (= (vlrg.cat$source SET$SET) SET.FTN$source)
    (= (vlrg.cat$target SET$SET) SET.FTN$target)
```

```
     (= (vlrg.cat$composition SET$SET) SET.FTN$composition)
     (= (vlrg.cat$identity SET$SET) SET.FTN$identity)

 (3) (2-cat$2-category CAT$CAT)
     (= (2-cat$object CAT$CAT) CAT$category)
     (= (2-cat$arrow CAT$CAT) FUNC$functor)
     (= (2-cat$2-cell CAT$CAT) NAT$natural-transformation)
     (= (2-cat$source CAT$CAT) FUNC$source)
     (= (2-cat$target CAT$CAT) FUNC$target)
     (= (2-cat$composition CAT$CAT) FUNC$composition)
     (= (2-cat$identity CAT$CAT) FUNC$identity)
     (= (2-cat$source-arrow CAT$CAT) NAT$source-functor)
     (= (2-cat$target-arrow CAT$CAT) NAT$target-functor)
     (= (2-cat$source-object CAT$CAT) NAT$source-category)
     (= (2-cat$target-object CAT$CAT) NAT$target-category)
     (= (2-cat$horizontal-composition CAT$CAT) NAT$horizontal-composition)
     (= (2-cat$horizontal-identity CAT$CAT) NAT$horizontal-identity)
     (= (2-cat$vertical-composition CAT$CAT) NAT$vertical-composition)
     (= (2-cat$vertical-identity CAT$CAT) NAT$vertical-identity)

 (4) (vlrg.func$functor CAT$object)
     (= (vlrg.func$source CAT$object) (2-cat$category CAT$CAT))
     (= (vlrg.func$target CAT$object) SET$SET)
     (= (vlrg.func$object CAT$object) CAT$object)
     (= (vlrg.func$morphism CAT$object) FUNC$object)
```

Conceptual Structures

CLS
CL

The following axiomatization should be added to the IFF Classification (meta) Ontology (IFF-CLSN). By the categorical version of the FCA basic theorem (Kent, 2002), the (large) category of classifications and infomorphisms CLSN is categorically equivalent to the (large) category of concept lattices and concept morphisms CNLAT. The categorical equivalence

$$\mathsf{CNLAT} \equiv \mathsf{CLSN}$$

is mediated by the two functors

- a concept lattice to classification forgetful functor $C : \mathsf{CNLAT} \to \mathsf{CLSN}$, and
- a classification to concept lattice construction functor $L : \mathsf{CLSN} \to \mathsf{CNLAT}$.

Let CLsn and CNLat denote the subcategories whose classifications and concept lattices have collections of types that are small sets. The category CLSN has two projection functors (fibrations), the instance functor $inst : \mathsf{CLSN}^{op} \to \mathsf{SET}$ and the type functor $typ : \mathsf{CLSN} \to \mathsf{SET}$. The subcategory CLsn has two projection restrictions, the instance functor $inst : \mathsf{CLsn}^{op} \to \mathsf{SET}$ and the type functor $typ : \mathsf{CLsn} \to \mathsf{Set}$. CLsn is categorically equivalent to CNLat. The categorical equivalence

$$\mathsf{CNLat} \equiv \mathsf{CLsn}$$

is mediated by the two restrictions

- a concept lattice to classification forgetful functor $c : \mathsf{CNLat} \to \mathsf{CLsn}$, and
- a classification to concept lattice construction functor $l : \mathsf{CLsn} \to \mathsf{CNLat}$.

```
 (5) (vlrg.cat$category CL$Concept-Lattice)
     (vlrg.cat$category CL$CNLAT)
     (= CL$CNLAT CL$Concept-Lattice)
     (= (vlrg.cat$object CL$Concept-Lattice) CL$concept-lattice)
     (= (vlrg.cat$morphism CL$Concept-Lattice) CL.MOR$concept-morphism)

 (6) (vlrg.cat$category CLS$Classification)
     (vlrg.cat$category CLS$CLSN)
     (= CLS$CLSN CLS$Classification)
     (= (vlrg.cat$object CLS$Classification) CLS$classification)
     (= (vlrg.cat$morphism CLS$Classification) CLS.INFO$infomorphism)

 (7) (vlrg.func$functor CLS$instance)
```

```
     (= (vlrg.func$source CLS$instance) (vlrg.cat$opposite CLS$CLSN))
     (= (vlrg.func$target CLS$instance) SET$SET)
     (= (vlrg.func$object CLS$instance) CL$instance)
     (= (vlrg.func$morphism CLS$instance) CL.MOR$instance

 (8) (vlrg.func$functor CLS$type)
     (= (vlrg.func$source CLS$type) CLS$CLSN)
     (= (vlrg.func$target CLS$type) set$Set)
     (= (vlrg.func$object CLS$type) CL$type)
     (= (vlrg.func$morphism CLS$type) CL.MOR$type)

 (9) (vlrg.func$functor CL$cnlat2clsn)
     (= (vlrg.func$source CL$cnlat2clsn) CL$CNLAT)
     (= (vlrg.func$target CL$cnlat2clsn) CLS$CLSN)
     (= (vlrg.func$object CL$cnlat2clsn) CL$classification)
     (= (vlrg.func$morphism CL$cnlat2clsn) CL.MOR$infomorphism)

(10) (vlrg.func$functor CLS$clsn2cnlat)
     (= (vlrg.func$source CLS$clsn2cnlat) CLS$CLSN)
     (= (vlrg.func$target CLS$clsn2cnlat) CL$CNLAT)
     (= (vlrg.func$object CLS$clsn2cnlat) CLS$concept-lattice)
     (= (vlrg.func$morphism CLS$clsn2cnlat) CLS.INFO$concept-morphism)
```

There is also an important intent functor *intent* : CLsn → CNLat, which is isomorphic to the classification to concept lattice construction functor *l*. The intent functor constructs only the intentional aspect of a formal concept. This can be used in place of the full construction *l*, whose formal concepts include both intents and extents. [Also, define the action of "intent" on morphisms.]

```
(11) (vlrg.func$functor CLS.CL$intent)
     (= (vlrg.func$source CLS.CL$intent) CLS$CLSN)
     (= (vlrg.func$target CLS.CL$intent) CL$CNLAT)
     (forall (?a (CLS$classification ?a))
         (and (LAT$isomorphic
                  ((vlrg.func$object intent) ?a)
                  (CLS.CL$complete-lattice ?a))
              (= (CL$class ((vlrg.func$object intent) ?a))
                 (SET.FTN$image (CLS.CL$intent ?a)))))
```

In order to represent institutions with either small or large sets for its sentence and model components, we should define four different categories for each classifications and concept lattices. We could number these 00, 01, 10, 11 corresponding to (small set of sentences, small set of models), etc. These comments extend to any functors and natural transformations based at such categories, including institutions themselves. As shorthand, the most common such would have an unnumbered synonym.

Institutions

INS.OBJ

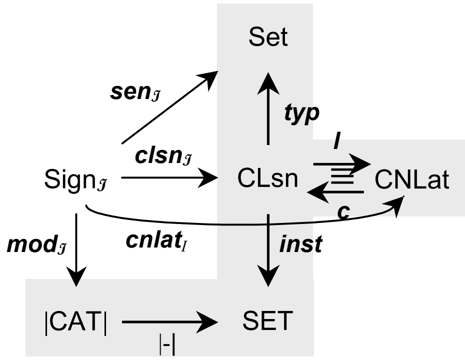

Figure 1a: An Institution

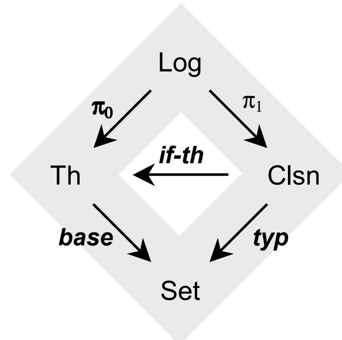

Figure 1b: The Logic Environment of the (large) IF Institution

Basics

In this document, institutions are viewed abstractly. Institutions abstract Tarski's model theory. In addition, institutions allow the translation of signatures. An institution $\mathcal{I} = \langle \mathsf{Sign}_{\mathcal{I}}, \mathit{sen}_{\mathcal{I}}, \mathit{mod}_{\mathcal{I}}, \models_{\mathcal{I}} \rangle$ consists of (Figure 1a)

- a category of *signatures* $\mathsf{Sign}_{\mathcal{I}}$,
- a *sentence* functor $\mathit{sen}_{\mathcal{I}} : \mathsf{Sign}_{\mathcal{I}} \rightarrow \mathsf{Set}$,
- a *model* functor $\mathit{mod}_{\mathcal{I}} : \mathsf{Sign}_{\mathcal{I}}^{\mathrm{op}} \rightarrow |\mathsf{CAT}|$, and
- an indexed collection $\{\models_{\Sigma} \subseteq |\mathit{mod}|_{\mathcal{I}}(\Sigma) \times \mathit{sen}_{\mathcal{I}}(\Sigma) \mid \Sigma \in |\mathsf{Sign}_{\mathcal{I}}|\}$ of *satisfaction* relations.

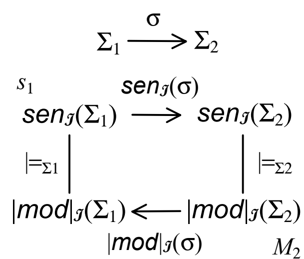

Figure 2: Satisfaction

Define the object model functor $|\mathit{mod}|_{\mathcal{I}} = \mathit{mod}_{\mathcal{I}} \cdot |\text{-}| : \mathsf{Sign}_{\mathcal{I}}^{\mathrm{op}} \rightarrow \mathsf{SET}$ by composition with the object functor $|\text{-}| : |\mathsf{CAT}| \rightarrow \mathsf{SET}$. The following condition expresses the invariance of truth under change of notation. Note that for signatures Σ the morphisms in the fiber categories $\mathit{mod}_{\mathcal{I}}(\Sigma)$ are not used in this satisfaction condition.

Satisfaction Condition: (Figure 2) For each signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$, each target model $M_2 \in |\mathit{mod}|_{\mathcal{I}}(\Sigma_2)$, and each source sentence $s_1 \in \mathit{sen}_{\mathcal{I}}(\Sigma_1)$,

$$|\mathit{mod}|_{\mathcal{I}}(\sigma)(M_2) \models_{\Sigma 1} s_1 \ \underline{\text{iff}}\ M_2 \models_{\Sigma 2} \mathit{sen}_{\mathcal{I}}(\sigma)(s_1).$$

For each signature $\Sigma \in |\mathsf{Sign}_{\mathcal{I}}|$, the sentence functor gives a set $\mathit{sen}_{\mathcal{I}}(\Sigma)$ whose elements are called sentences over that signature (Σ-sentences), and the model functor gives a category $\mathit{mod}_{\mathcal{I}}(\Sigma)$ whose objects are called Σ-models and whose morphisms are called Σ-model (homo)morphisms. For each signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$, the sentence functor gives a function $\mathit{sen}_{\mathcal{I}}(\sigma) = \sigma(\text{-}) : \mathit{sen}_{\mathcal{I}}(\Sigma_1) \rightarrow \mathit{sen}_{\mathcal{I}}(\Sigma_2)$ called the *sentence translation* function along σ, and the model functor gives a functor $\mathit{mod}_{\mathcal{I}}(\sigma) = (\text{-})\lceil\sigma : \mathit{mod}_{\mathcal{I}}(\Sigma_1) \leftarrow \mathit{mod}_{\mathcal{I}}(\Sigma_2)$ called the *model reduct* (*fiber*) functor along σ. When $M_1 = (M_2)\lceil\sigma$ we say that M_1 is an σ-*reduct* of M_2 and M_2 is an σ-*expansion* of M_1.

We need a constant function between Ur objects! These can be placed into the namespaces for the second Ur object (which must be concrete; i.e., contain elements).

```
(1)  (UR$object institution)

(2)  (UR$morphism signature)
     (= (UR$source signature) institution)
     (= (UR$target signature) CAT$category)
```

```
(3)  (UR$morphism sentence)
     (= (UR$source sentence) institution)
     (= (UR$target sentence) FUNC$functor)
     (= (UR$composition [sentence FUNC$source]) signature)
     (= (UR$composition [sentence FUNC$target])
        ((UR$constant [institution CAT$category]) set$Set))

(4)  (UR$morphism model)
     (= (UR$source model) institution)
     (= (UR$target model) vlrg.func$functor)
     (= (UR$composition [model vlrg.func$source]) signature)
     (= (UR$composition [model vlrg.func$target])
        ((UR$constant [institution KIF.COL$collection]) (2-cat$category CAT$CAT)))

(5)  (UR$morphism satisfaction)
     (= (UR$source satisfaction) institution)
     (= (UR$target satisfaction) KIF.FTN$function)
     (= (UR$composition [satisfaction KIF.FTN$source])
        (UR$composition [signature CAT$object]))
     (= (UR$composition [satisfaction KIF.FTN$target])
        ((UR$constant [institution KIF$collection]) REL$relation))
     (for all (?I (institution ?I))
         (and (= (KIF.FTN$composition [(satisfaction ?I) REL$class0])
                 (KIF.FTN$composition [(vlrg.func$object (model ?I)) CAT$object]))
              (= (KIF.FTN$composition [(satisfaction ?I) REL$class1])
                 (FUNC$object (sentence ?I)))
              (forall (?s ((CAT$morphism (signature I)) ?s)
                       ?e1 ((set.ftn$source ((FUNC$morphism (sentence I)) ?s)) ?e1)
                       ?M2 ((CAT$object (FUNC$target ((vlrg.func$morphism (model I)) ?s))) ?M2))
                  (<=> (((satisfaction ?I) ((CAT$source (signature I)) ?s))
                          ((FUNC$object ((vlrg.cat$morphism (model ?I)) ?s)) ?M2) ?e1)
                       (((satisfaction ?I) ((CAT$target (signature ?I)) ?s))
                          ?M2 (((FUNC$morphism (sentence ?I)) ?s) ?e1))))))
```

Conceptual Structures

For anyone conversant with the theory of Information Flow, the satisfaction condition is very familiar. In fact, it is clear that the satisfaction condition is in some form the fundamental condition of an infomorphism (see Figure 2). This observation formulates institutions in terms of a nice associated structure, a functor to the (large) category of classifications and infomorphisms CLsn, or equivalently, a functor to the (large) category of concept lattices and concept morphisms CNLat.

Truth is objectively represented. For any signature Σ, the triple $clsn_{\mathcal{J}}(\Sigma) = \langle |mod_{\mathcal{J}}(\Sigma)|, sen_{\mathcal{J}}(\Sigma), \models_{\mathcal{J}}(\Sigma)\rangle$ is called the truth classification of Σ. From the basic theorem of FCA, this is equivalent to a concept lattice $cnlat_{\mathcal{J}}(\Sigma) = \langle lat_{\mathcal{J}}(\Sigma), |mod_{\mathcal{J}}(\Sigma)|, sen_{\mathcal{J}}(\Sigma), \iota_{\mathcal{J}}(\Sigma), \tau_{\mathcal{J}}(\Sigma)\rangle$ called the truth concept lattice of Σ, which consists of

- a complete lattice $lat_{\mathcal{J}}(\Sigma) = \langle cloth_{\mathcal{J}}(\Sigma), \leq_{\mathcal{J}}(\Sigma), \bigvee_{\mathcal{J}}(\Sigma), \bigwedge_{\mathcal{J}}(\Sigma)\rangle$,
- a model embedding map $\iota_{\mathcal{J}}(\Sigma) : |mod_{\mathcal{J}}(\Sigma)| \rightarrow cloth_{\mathcal{J}}(\Sigma)$, and
- a sentence embedding map $\tau_{\mathcal{J}}(\Sigma)) : sen_{\mathcal{J}}(\Sigma) \rightarrow cloth_{\mathcal{J}}(\Sigma)$.

In general, the elements of any concept lattice can be represented by their intents. That is the case here. The truth concept lattice elements are represented by their intents $cloth_{\mathcal{J}}(\Sigma)$, the closed theories of Σ, and the lattice order $\leq_{\mathcal{J}}(\Sigma)$ is reverse subset inclusion: $T_1 \leq_{\mathcal{J}}(\Sigma)\ T_2$ <u>iff</u> $T_1 \supseteq T_2$. The join $\bigvee_{\mathcal{J}}(\Sigma)$ and meet $\bigwedge_{\mathcal{J}}(\Sigma)$ operators are described as follows: the join of a collection of closed theories $\mathcal{T}$ is their intersection $\bigvee_{\mathcal{J}}(\Sigma)\mathcal{T} = \bigcap\mathcal{T}$, and the meet is the closure of their union $\bigwedge_{\mathcal{J}}(\Sigma)\mathcal{T} = (\bigcup\mathcal{T})^{\bullet}$. Model embedding maps a model to its maximal theory consisting of the set of all sentences that it satisfies, and sentence embedding maps a sentence to its entailment theory consisting of the set of all sentences that it entails. A model theory is the embedding (maximal theory) of some model. A sentence theory is the embedding (entailment theory) of some sentence. Any concept (closed theory) is the join of a subset of instance concepts (model theories) and the meet of a subset of type concepts (sentence theories). In summary, associated with any institution $\mathcal{J}$ is

- a truth classification functor or $clsn_{\mathcal{J}} : \mathrm{Sign}_{\mathcal{J}} \rightarrow \mathrm{CLsn}$, and
- a truth concept lattice functor $cnlat_{\mathcal{J}} : \mathrm{Sign}_{\mathcal{J}} \rightarrow \mathrm{CNLat}$.

By composing the truth classification functor with these two projections, we can unpack it (Figure 1a) into

- the underlying model functor $|mod|_{\mathcal{I}} = clsn_{\mathcal{I}}^{op} \circ inst : \mathrm{Sign}_{\mathcal{I}}^{op} \rightarrow \mathrm{SET}$ and
- the sentence functor $sen_{\mathcal{I}} = clsn_{\mathcal{I}} \circ typ : \mathrm{Sign}_{\mathcal{I}} \rightarrow \mathrm{Set}$.

We define the classification functor from scratch (in terms of satisfaction). However, we use the intent functor to define the concept lattice functor as $cnlat_{\mathcal{I}} = clsn_{\mathcal{I}} \circ intent$. That is, the concept lattice of a signature is the intentional concept lattice generated by the (satisfaction) classification of the signature. Then $clsn_{\mathcal{I}} = cnlat_{\mathcal{I}} \circ c$.

```
(6)  (UR$morphism classification)
     (= (UR$source classification) institution)
     (= (UR$target classification) vlrg.func$functor)
     (forall (?I (institution ?I))
         (and (= (vlrg.func$source (classification ?I)) (signature ?I))
              (= (vlrg.func$target (classification ?I)) CLS$Classification)
              (= (sentence ?I) (vlrg.func$composition [(classification ?I) CLS$type]))
              (= (vlrg.func$composition [(model ?I) CAT$object])
                 (vlrg.func$composition [(classification ?I) CLS$instance]))
              (forall (?S ((CAT$object (signature ?I)) ?S))
                  (and (= (CLS$instance ((vlrg.func$object (classification ?I)) ?S))
                          (CAT$object ((vlrg.func$object (model ?I)) ?S)))
                       (= (CLS$type ((vlrg.func$object (classification ?I)) ?S))
                          ((FUNC$object (sentence ?I)) ?S))
                       (= ((vlrg.func$object (classification ?I)) ?S)
                          ((satisfaction ?I) ?S))))
              (forall (?s ((CAT$morphism (signature ?I)) ?s))
                  (and (= (CLS.INFO$instance ((vlrg.func$morphism (classification ?I)) ?s))
                          (FUNC$object ((vlrg.func$morphism (model ?I)) ?s)))
                       (= (CLS.INFO$type ((vlrg.func$morphism (classification ?I)) ?s))
                          ((FUNC$morphism (sentence ?I)) ?s))))))

(7)  (UR$morphism concept-lattice)
     (= (UR$source concept-lattice) institution)
     (= (UR$target concept-lattice) vlrg.func$functor)
     (forall (?I (institution ?I))
         (and (= (vlrg.func$source (concept-lattice ?I)) (signature ?I))
              (= (vlrg.func$target (concept-lattice ?I)) CL$Concept-Lattice)
              (= concept-lattice (vlrg.func$composition [classification intent]))))

(8)  (forall (?I (institution ?I))
         (= classification (vlrg.func$composition [concept-lattice cnlat2clsn])))
```

A (not necessarily closed) theory is just a set of sentences. For any signature Σ, the free theory order $th^{\circ}_{\mathcal{I}}(\Sigma) = \langle \wp sen_{\mathcal{I}}(\Sigma), \supseteq \rangle$ is just the powerset construction $\wp$ on sentences of Σ with the reverse subset inclusion order $\supseteq$. This is a complete lattice. For any signature Σ, the free classification $free_{\mathcal{I}}(\Sigma) = \langle \wp sen_{\mathcal{I}}(\Sigma), sen_{\mathcal{I}}(\Sigma), \ni \rangle$ has theories as instances, sentences as types and reverse membership as incidence relation. There is a canonical intent infomorphism $intent_{\mathcal{I}}(\Sigma) : free_{\mathcal{I}}(\Sigma) \rightarrow clsn_{\mathcal{I}}(\Sigma)$, whose instance function $max\text{-}th_{\mathcal{I}}(\Sigma) : |mod_{\mathcal{I}}(\Sigma)| \rightarrow \wp sen_{\mathcal{I}}(\Sigma)$ maps a model to its maximal theory and whose type function is the identity on sentences. The free concept lattice is the concept lattice associated with the free classification; it has the free theory order as its complete lattice, identity instance embedding, and singleton type embedding functions. The concept morphism associated with the intent infomorphism has maximal theory as its instance function, inclusion of closed theories as theories as its left adjoint, theory closure as its right adjoint, and singleton closure as its type function.

For any signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$, there are (at least) two maps between theory orders; the direct image function $dir_{\mathcal{I}}(\sigma) = \wp sen_{\mathcal{I}}(\sigma) : \wp sen_{\mathcal{I}}(\Sigma_1) \rightarrow \wp sen_{\mathcal{I}}(\Sigma_2)$ is the direct power operator applied to the sentence function, and the inverse image function $inv_{\mathcal{I}}(\sigma) = sen_{\mathcal{I}}(\sigma)^{-1} : \wp sen_{\mathcal{I}}(\Sigma_1) \leftarrow \wp sen_{\mathcal{I}}(\Sigma_2)$ is the inverse image operator applied to the sentence function. For the free theories these form a Galois connection – the inverse image function is left adjoint to the direct image function. Hence, associated with an institution $\mathcal{I}$ is a free theory functor for either adjoint. Let $th^{\circ}_{\mathcal{I}} : \mathrm{Sign}_{\mathcal{I}}^{op} \rightarrow \mathrm{Cat}$ represent the free theory functor associated with the inverse image operator. Using the Grothendieck construction to generate a category of theories, the adjointness properties mean that the associated signature projection is a bifibration.

Truth extends to morphisms. Based upon the satisfaction condition for institutions, associated with any signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$ is a *truth infomorphism*

- $clsn_{\mathcal{I}}(\sigma) = \langle |mod_{\mathcal{I}}(\sigma)|, sen_{\mathcal{I}}(\sigma) \rangle : clsn_{\mathcal{I}}(\Sigma_1) \rightarrow clsn_{\mathcal{I}}(\Sigma_2)$,

whose instance function is the model function $|mod_{\mathcal{I}}(\sigma)| : |mod_{\mathcal{I}}(\Sigma_1) \leftarrow |mod_{\mathcal{I}}(\Sigma_2)$ and whose type function is the sentence function $sen_{\mathcal{I}}(\sigma) : sen_{\mathcal{I}}(\Sigma_1) \rightarrow sen_{\mathcal{I}}(\Sigma_2)$. From the basic theorem of FCA, this is equivalent to a concept morphism

- $cnlat_{\mathcal{I}}(\sigma) = \langle left_{\mathcal{I}}(\sigma), right_{\mathcal{I}}(\sigma), |mod_{\mathcal{I}}(\sigma)|, sen_{\mathcal{I}}(\sigma) \rangle : cnlat_{\mathcal{I}}(\Sigma_1) \rightarrow cnlat_{\mathcal{I}}(\Sigma_2)$,

called the *truth concept morphism* of σ, which consists of the two component maps of $clsn_{\mathcal{I}}(\sigma)$, plus a Galois connection consisting of a left adjoint monotonic function $left_{\mathcal{I}}(\sigma) : lat_{\mathcal{I}}(\Sigma_1) \leftarrow lat_{\mathcal{I}}(\Sigma_2)$, and a right adjoint monotonic function $right_{\mathcal{I}}(\sigma) : lat_{\mathcal{I}}(\Sigma_1) \rightarrow lat_{\mathcal{I}}(\Sigma_2)$. Adjointness means that $left_{\mathcal{I}}(\sigma)(T_2) \leq_{\mathcal{I}}(\Sigma_1)\ T_1$ <u>iff</u> $T_2 \leq_{\mathcal{I}}(\Sigma_2)\ right_{\mathcal{I}}(\sigma)(T_1)$; that is, that $left_{\mathcal{I}}(\sigma)(T_2) \supseteq T_1$ <u>iff</u> $T_2 \supseteq right_{\mathcal{I}}(\sigma)(T_1)$ for every closed target theory $T_2 \in lat_{\mathcal{I}}(\Sigma_2)$ and every closed source theory $T_1 \in lat_{\mathcal{I}}(\Sigma_1)$. The left adjoint is compatible with the model function $\iota_{\mathcal{I}}(\Sigma_2) \cdot left_{\mathcal{I}}(\sigma) = |mod_{\mathcal{I}}(\sigma)| \cdot \iota_{\mathcal{I}}(\Sigma_1)$, and the right adjoint is compatible with the sentence function $sen_{\mathcal{I}}(\sigma) \cdot \tau_{\mathcal{I}}(\Sigma_2) = \tau_{\mathcal{I}}(\Sigma_1) \cdot right_{\mathcal{I}}(\sigma)$. Pointwise on theories, the left adjoint is the inverse image function $inv_{\mathcal{I}}(\sigma) = sen_{\mathcal{I}}(\sigma)^{-1}$ restricted to closed theories, and the right adjoint is the composition of the restriction of the direct image function $dir_{\mathcal{I}}(\sigma) = \wp sen_{\mathcal{I}}(\sigma)$ and the theory closure function. Hence, associated with an institution $\mathcal{I}$ is a theory functor for either adjoint. Let $th_{\mathcal{I}} : \mathsf{Sign}_{\mathcal{I}}^{op} \rightarrow \mathsf{Cat}$ represent the closed theory functor associated with the inverse image operator. Similar comments hold true about the Grothendieck category of closed theories.

Examples

The Information Flow Institution (IF)

As is clear from their definition and more explicitly the section on logical environments, in a general sense institutions are closely related to Information Flow. However, there is also a special sense in which these two metatheories are related. This sense is represented by a naturally defined institution for Information Flow denoted IF. Here, we discuss this institution in some detail. First, we do some review.

$$\begin{tikzcd} \mathbf{Clsn} \arrow[d, shift right=2, "inst"'] \arrow[d, phantom, "\eta\ \dashv\ 1"] \\ \mathbf{Set}^{op} \arrow[u, shift right=2, "\wp"'] \end{tikzcd} \qquad\qquad \begin{tikzcd} \mathbf{Clsn} \arrow[d, shift left=2, "typ"] \arrow[d, phantom, "1\ \dashv\ \varepsilon"] \\ \mathbf{Set} \arrow[u, shift left=2, "\wp"] \end{tikzcd}$$

Figure 3ie: The instance-extent reflection **Figure 3ie: The type-intent coreflection**

A *classification* $\boldsymbol{A} = \langle inst(\boldsymbol{A}), typ(\boldsymbol{A}), \models_{\boldsymbol{A}} \rangle$ consists of a set of instances or tokens $inst(\boldsymbol{A})$, a set of types $typ(\boldsymbol{A})$ and a binary incidence relation $\models_{\boldsymbol{A}} \subseteq inst(\boldsymbol{A}) \times typ(\boldsymbol{A})$ between instances and types. We can consider the type symbols $typ(\boldsymbol{A})$ to be a collection of unary predicate symbols. When $a \models_{\boldsymbol{A}} \alpha$ and for some instance $a \in inst(\boldsymbol{A})$ and some type $\alpha \in typ(\boldsymbol{A})$, then we say "$a$ is of type α". The power set construction $\wp$ allows us to define two special classifications: for each set A (of elements considered as instances), the instance power set classification is $\wp A = \langle A, \wp A, \in_A \rangle$, and for any set Σ (of elements considered as types), the type power classification $\wp\Sigma = \langle \wp\Sigma, \Sigma, \ni_A \rangle$. Every classification $\boldsymbol{A}$ determines an intent function $int_{\boldsymbol{A}} : inst(\boldsymbol{A}) \rightarrow \wp typ(\boldsymbol{A})$ where $inst_{\boldsymbol{A}}(a) = \{\alpha \in typ(\boldsymbol{A}) \mid a \models_{\boldsymbol{A}} \alpha\}$, and an extent function $ext_{\boldsymbol{A}} : typ(\boldsymbol{A}) \rightarrow \wp inst(\boldsymbol{A})$ where $ext_{\boldsymbol{A}}(\alpha) = \{a \in inst(\boldsymbol{A}) \mid a \models_{\boldsymbol{A}} \alpha\}$. Classifications are connected via infomorphisms.

An *infomorphism* $\boldsymbol{f} = \langle inst(\boldsymbol{f}), typ(\boldsymbol{f}) \rangle : \boldsymbol{A} \rightarrow \boldsymbol{B}$ from source classification $\boldsymbol{A}$ to target classification $\boldsymbol{B}$ consists of an instance function $inst(\boldsymbol{f}) : inst(\boldsymbol{B}) \rightarrow inst(\boldsymbol{A})$ in the reverse sense and a type function $typ(\boldsymbol{f}) : typ(\boldsymbol{A}) \rightarrow typ(\boldsymbol{B})$ in the direct sense, which satisfy the fundamental condition:

$inst(\boldsymbol{f})(b) \models_{\boldsymbol{A}} \alpha$ <u>iff</u> $b \models_{\boldsymbol{B}} typ(\boldsymbol{f})(\alpha)$, for any target instance $b \in \mathit{inst}(\boldsymbol{B})$ and for any source type $\alpha \in \mathit{typ}(\boldsymbol{A})$. Compositions and identities are defined in terms of the component instance and type functions. Therefore, there is a category Clsn of classifications and infomorphisms and two projection functors $\mathit{inst} : \mathsf{Clsn} \to \mathsf{Set}^{\mathrm{op}}$ and $\mathit{typ} : \mathsf{Clsn} \to \mathsf{Set}$. Every classification $\boldsymbol{A}$ determines an intent infomorphism $\varepsilon_{\boldsymbol{A}} = \langle \mathit{int}(\boldsymbol{A}), 1_{\mathit{typ}(\boldsymbol{A})} \rangle : \wp\mathit{typ}(\boldsymbol{A}) \to \boldsymbol{A}$ and an extent infomorphism $\eta_{\boldsymbol{A}} = \langle 1_{\mathit{inst}(\boldsymbol{A})}, \mathit{ext}(\boldsymbol{A}) \rangle : \boldsymbol{A} \to \wp\mathit{inst}(\boldsymbol{A})$. The intent infomorphism $\varepsilon_{\boldsymbol{A}}$ is the $\boldsymbol{A}^{\mathrm{th}}$ component of an intent natural transformation $\varepsilon : \mathit{typ} \circ \wp \to \mathit{id}_{\mathsf{Clsn}}$, which is the counit of a coreflection between the type power functor $\wp : \mathsf{Set} \to \mathsf{Clsn}$ and the type projection functor $\mathit{typ} : \mathsf{Clsn} \to \mathsf{Set}$. The extent infomorphism $\eta_{\boldsymbol{A}}$ is the $\boldsymbol{A}^{\mathrm{th}}$ component of an extent natural transformation $\eta : \mathit{id}_{\mathsf{Clsn}} \to \mathit{inst} \circ \wp$, which is the unit of a reflection between the instance projection functor $\mathit{inst} : \mathsf{Clsn} \to \mathsf{Set}^{\mathrm{op}}$ and the instance power functor $\wp : \mathsf{Set}^{\mathrm{op}} \to \mathsf{Clsn}$.

For any set A (of elements considered as instances), let $\mathit{Inst}(A) = \mathit{inst}^{-1}(A) \subseteq \mathsf{Clsn}$ denote the fiber subcategory of all classifications whose instance set is A and all infomorphisms whose instance function is 1_A. For any function $g : B \to A$ (regarded as an instance map), let $\mathit{Inst}(g) : \mathit{Inst}(B) \to \mathit{Inst}(A)$ denote the functor whose object function maps a classification $\boldsymbol{A} \in \mathit{Inst}(A)$ to the classification $\mathit{Inst}(g)(\mathrm{A}) = \langle B, \mathit{typ}(\boldsymbol{A}), \models \rangle \in \mathit{Inst}(B)$ where $b \models \alpha$ when $g(b) \models_{\boldsymbol{A}} \alpha$ in classification $\boldsymbol{A}$, and whose morphism function maps an infomorphism $\boldsymbol{f} = \langle 1_A, \mathit{typ}(\boldsymbol{f}) \rangle : \boldsymbol{A} \to \boldsymbol{B}$ in $\mathit{Inst}(A)$ to the infomorphism $\mathit{Inst}(g)(\boldsymbol{f}) = \langle 1_B, \mathit{typ}(\boldsymbol{f}) \rangle : \mathit{Inst}(g)(\boldsymbol{A}) \to \mathit{Inst}(g)(\boldsymbol{B})$ in $\mathit{Inst}(B)$. This instance construction forms an indexed category $\mathit{Inst} : \mathsf{Set}^{\mathrm{op}} \to \mathsf{CAT}$. [NEEDS WORK].

Dually, for any set Σ (of elements considered as types), let $\mathit{Typ}(\Sigma) = \mathit{typ}^{-1}(\Sigma) \subseteq \mathsf{Clsn}$ denote the fiber subcategory of all classifications whose type set is Σ and all infomorphisms whose type function is 1_Σ. Considering the type symbols Σ to be unary predicate symbols, a classification $\boldsymbol{A}$ in the fiber $\mathit{Typ}(\Sigma)$ can be regarded as a model for Σ consisting of a set of instances $\mathit{inst}(\boldsymbol{A})$ called the (relative) universe of $\boldsymbol{A}$ and a map $\mathit{ext}_{\boldsymbol{A}} : \Sigma \to \wp\mathit{inst}(\boldsymbol{A})$ called the extension function of $\boldsymbol{A}$, which that maps predicate symbols to subsets of the universe where those symbols are "true", and an infomorphism $\boldsymbol{f} = \langle \mathit{inst}(\boldsymbol{f}), 1_\Sigma \rangle : \boldsymbol{A} \to \boldsymbol{B}$ in $\mathit{Typ}(\Sigma)$ can be regarded as a model morphism consisting of a universe map $\mathit{inst}(\boldsymbol{f}) : \mathit{inst}(\boldsymbol{B}) \to \mathit{inst}(\boldsymbol{A})$ that preserves extensions in the sense that $\mathit{ext}_{\boldsymbol{B}}(\alpha) = \boldsymbol{f}^{-1}(\mathit{ext}_{\boldsymbol{A}}(\alpha))$ for all unary predicate symbols $\alpha \in \Sigma$. For any function $h : \Sigma \to \Omega$ (regarded as a type map), let $\mathit{Typ}(h) : \mathit{Typ}(\Omega) \to \mathit{Typ}(\Sigma)$ denote the functor whose object function maps a classification $\boldsymbol{B} \in \mathit{Typ}(\Omega)$ to the classification $\mathit{Typ}(h)(\boldsymbol{B}) = \langle \mathit{inst}(\boldsymbol{B}), \Sigma, \models \rangle \in \mathit{Typ}(\Sigma)$, where $b \models \alpha$ when $b \models_{\boldsymbol{B}} g(\alpha)$ in classification $\boldsymbol{B}$, and whose morphism function maps an infomorphism $\boldsymbol{f} = \langle \mathit{inst}(\boldsymbol{f}), 1_\Omega \rangle : \boldsymbol{A} \to \boldsymbol{B}$ in $\mathit{Typ}(\Omega)$ to the infomorphism $\mathit{Typ}(h)(\boldsymbol{f}) = \langle \mathit{inst}(\boldsymbol{f}), 1_\Sigma \rangle : \mathit{Typ}(h)(\boldsymbol{A}) \to \mathit{Typ}(h)(\boldsymbol{B})$ in $\mathit{Typ}(\Sigma)$. The type construction forms an indexed category $\mathit{Typ} : \mathsf{Set}^{\mathrm{op}} \to \mathsf{CAT}$. The Grothendieck category $\mathsf{Gr}(\mathit{Typ})$ is isomorphic to Clsn and the Grothendieck projection functor is isomorphic to $\mathit{typ} : \mathsf{Clsn} \to \mathsf{Set}$.

For any set Σ (of elements considered as types), a sequent of Σ is a pair $\sigma = (\Gamma, \Delta)$ of subsets of Σ. The subset $\Gamma \subseteq \Sigma$ is called the antecedent of σ and the subset $\Delta \subseteq \Sigma$ is called the consequent of σ. In the meaning of a sequent, the antecedent is regarded in a conjunctive sense, the consequent is regarded in a disjunctive sense, and the relation between them is regarded in an implicative sense. The binary power $\mathit{Seq}(\Sigma) = \wp\Sigma\times\wp\Sigma$ is the set of all sequents of Σ (Σ-sequents). For any function $h : \Sigma \to \Omega$ (regarded as a type map), the binary power $\mathit{Seq}(h) = \wp h\times\wp h : \mathit{Seq}(\Sigma) \to \mathit{Seq}(\Omega)$ denotes the function that maps a sequent $\sigma = \langle \Gamma, \Delta \rangle$ in $\mathit{Seq}(\Sigma)$ to the sequent $\mathit{Seq}(h)(\sigma) = \langle \wp h(\Gamma), \wp h(\Delta) \rangle$ in $\mathit{Seq}(\Omega)$. The sequent construction forms an indexed (discrete) category $\mathit{Seq} : \mathsf{Set} \to \mathsf{Set}$. [This construction is akin to the expression functor on the category Lang of 1st order languages.]

Let Σ be any set (of elements considered as types), let $\boldsymbol{A} = \langle \mathit{inst}(\boldsymbol{A}), \Sigma, \models_{\boldsymbol{A}} \rangle$ be any classification in $\mathit{Typ}(\Sigma)$ and let $\sigma = (\Gamma, \Delta)$ be any sequent in $\mathit{Seq}(\Sigma)$. An instance $a \in \mathit{inst}(\boldsymbol{A})$ satisfies σ when the following holds: if a is of every antecedent type then it is of some consequent type; or in symbols, $(\forall_{\alpha \in \Gamma}\ a \models_{\boldsymbol{A}} \alpha)$ implies $(\exists_{\beta \in \Delta}\ a \models_{\boldsymbol{A}} \beta)$. An instance not satisfying a sequent is called a counterexample to the sequent. The classification $\boldsymbol{A}$ satisfies the sequent σ, denoted by $\boldsymbol{A} \models_\Sigma \sigma$, when all instances of $\boldsymbol{A}$ satisfy σ (that is, there are no counterexamples to σ in $\boldsymbol{A}$. Hence, there is a binary satisfaction relation $\models_\Sigma \subseteq \mathit{Typ}(\Sigma)\times\mathit{Seq}(\Sigma)$. Satisfaction satisfies the following condition: for any function $h : \Sigma \to \Omega$ (regarded as a type map), $\mathit{Typ}(h)(\boldsymbol{B}) \models_\Sigma \sigma$ <u>iff</u> $\boldsymbol{B} \models_\Sigma \mathit{Seq}(h)(\sigma)$ for any source sequent σ in $\mathit{Seq}(\Sigma)$ and any target classification $\boldsymbol{B}$ in $\mathit{Typ}(\Omega)$.

Definition: The Information Flow (IF) institution is the quadruple $\langle \mathsf{Set}, \mathit{Typ}, \mathit{Seq}, \models \rangle$, whose category of abstract signatures is the category Set of sets and functions, whose model functor is the type functor $\mathit{Typ} : \mathsf{Set}^{\mathrm{op}} \rightarrow \mathsf{CAT}$, whose sentence functor is the sequent functor $\mathit{Seq} : \mathsf{Set} \rightarrow \mathsf{Set}$, and whose parameterize satisfaction relation is the binary satisfaction relation $\{\models_\Sigma \subseteq \mathit{Typ}(\Sigma) \times \mathit{Seq}(\Sigma) \mid \Sigma \in |\mathsf{Set}|\}$.

IF Theories

A theory $\boldsymbol{T} = \langle \mathit{typ}(\boldsymbol{T}), \vdash_{\boldsymbol{T}} \rangle$ is a set (of types) and a binary consequence relation $\vdash_{\boldsymbol{T}} \subseteq \mathit{Seq}(\mathit{typ}(\boldsymbol{T}))$ on subsets of types. An axiom or constraint of a theory $\boldsymbol{T}$ is a sequent $\sigma = (\Gamma, \Delta)$ of $\mathit{typ}(\boldsymbol{T})$ for which $\Gamma \vdash_{\boldsymbol{T}} \Delta$. The theory $\mathit{Th}(\boldsymbol{A}) = \langle \mathit{typ}(\boldsymbol{A}), \vdash_{\boldsymbol{A}} \rangle$ generated by a classification $\boldsymbol{A}$ is the theory whose types are the types of $\boldsymbol{A}$ and whose constraints are the set of sequents satisfied by every instance of $\boldsymbol{A}$. [MUCH MORE].

An *institution classification* $\langle M, S, \models \rangle$ consists of

- a category of models M,
- a set of sentences S, and
- a satisfaction relation $\models \subseteq |M| \times S$.

Classifications are connected via infomorphisms. These represent change of notation within one logic, and help define translations between logics. There is a semantic entailment relation $\models \subseteq \wp S \times \wp S$ defined as follows: for any two subsets of sentences $\Gamma, \Delta \subseteq S$, $\Gamma \models \Delta$ when any model that satisfies all sentences in Γ satisfies some sentence in Δ.

An *institution infomorphism* $(\mu, \sigma) : \langle M_1, S_1, \models_1 \rangle \rightarrow \langle M_2, S_2, \models_2 \rangle$ consists of

- a model reduction functor $\mu : M_1 \rightarrow M_2$, and
- a sentence translation function $\sigma : S_1 \rightarrow S_2$,

for which the following *satisfaction condition* holds:

$\mu(m_2) \models_1 s_1$ iff $m_2 \models_2 \sigma(s_1)$,

for all target models $m_2 \in M_2$ and all source sentences $s_1 \in S_1$.

An *institution theory* $\langle S, \vdash \rangle$ is a set of sentences S, equipped with an *entailment relation* $\vdash \subseteq \wp S \times \wp S$. A theory is *regular* when the following conditions are satisfied:

- *reflexivity (identity)*: for any sentence $\alpha \in S$, $\alpha \vdash \alpha$,
- *monotonicity*: if $\Gamma' \subseteq \Gamma$ and $\Gamma \vdash \alpha$ then $\Gamma' \vdash \alpha$,
- *weakening*: if $\Gamma \vdash \Delta$ then $\Gamma, \Gamma' \vdash \Delta, \Delta'$,
- *transitivity*: if $\Gamma \vdash \alpha_i$, $i \in I$ and $\Gamma \cup \{\alpha_i \mid i \in I\} \vdash \beta$, then $\Gamma \vdash \beta$, and
- *global cut*: if $\Gamma, \Sigma_0 \vdash \Delta, \Sigma_1$ for each partition $\{\Sigma_0, \Sigma_1\}$ of Σ', then $\Gamma \vdash \Delta$.

Here, we have used the usual notational conventions about consequence relations. For example, we write $\alpha, \beta \vdash \gamma$ for $\{\alpha, \beta\} \vdash \{\gamma\}$ and $\Gamma, \Gamma' \vdash \Delta, \alpha$ for $\Gamma \cup \Gamma' \vdash \Delta \cup \{\alpha\}$.

An *institution logic* $\langle M, S, \models, \vdash \rangle$ is an institution classification $\langle M, S, \models \rangle$ and an institution theory $\langle S, \vdash \rangle$ that share a set of sentences.

A logic is *sound* when it satisfies

- *soundness*: for $\Gamma \subseteq S$ and $\varphi \in S$, $\Gamma \vdash \varphi$ implies $\Gamma \models \varphi$.

A logic is *complete* when it satisfies

- *completeness*: for $\Gamma \subseteq S$ and $\varphi \in S$, $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$.

Institution Morphisms

INS.MOR

Basics

$$\begin{tikzcd} \mathrm{Sign}_{\mathcal{A}} \arrow[rr, "sign_{\mathcal{F}}"] \arrow[dr, "sen_{\mathcal{A}}"'] & \arrow[d, phantom, "\overset{sen_{\mathcal{F}}}{\Leftarrow}"] & \mathrm{Sign}_{\mathcal{B}} \arrow[dl, "sen_{\mathcal{B}}"] \\ & \mathrm{Set} & \end{tikzcd} \qquad \begin{tikzcd} \mathrm{Sign}_{\mathcal{A}}^{op} \arrow[rr, "sign_{\mathcal{F}}^{op}"] \arrow[dr, "|mod|_{\mathcal{A}}"'] & \arrow[d, phantom, "\overset{|mod|_{\mathcal{F}}}{\Rightarrow}"] & \mathrm{Sign}_{\mathcal{B}}^{op} \arrow[dl, "|mod|_{\mathcal{B}}"] \\ & \mathrm{SET} & \end{tikzcd}$$

Figure 4: Institution Morphism

An *institution morphism* $\mathcal{F} = \langle sign_{\mathcal{F}}, sen_{\mathcal{F}}, mod_{\mathcal{F}} \rangle : \mathcal{A} \rightarrow \mathcal{B}$ from source institution $\mathcal{A}$ to target institution $\mathcal{B}$ consists of

- a signature functor $sign_{\mathcal{F}} : \mathrm{Sign}_{\mathcal{A}} \rightarrow \mathrm{Sign}_{\mathcal{B}}$,
- a sentence natural transformation $sen_{\mathcal{F}} : sign_{\mathcal{F}} \circ sen_{\mathcal{B}} \Rightarrow sen_{\mathcal{A}} : \mathrm{Sign}_{\mathcal{A}} \rightarrow \mathrm{Set}$, and
- a model natural transformation $|mod|_{\mathcal{F}} : |mod|_{\mathcal{A}} \Rightarrow (sign_{\mathcal{F}}^{op} \circ |mod|_{\mathcal{B}}) : \mathrm{Sign}_{\mathcal{A}}^{op} \rightarrow \mathrm{SET}$,

such that the following *satisfaction condition* holds

$$M \models_{\mathcal{A}}(\Sigma)\ sen_{\mathcal{F}}(\Sigma)(e) \ \ \underline{\text{iff}} \ \ |mod|_{\mathcal{F}}(\Sigma)(M) \models_{\mathcal{B}}(sign_{\mathcal{F}}(\Sigma))\ e$$

for each signature $\Sigma \in |\mathrm{Sign}_{\mathcal{A}}|$, model $M \in |mod|_{\mathcal{A}}(\Sigma)$, and sentence $e \in sen_{\mathcal{B}}(sign_{\mathcal{F}}(\Sigma))$.

```
(1)  (UR$object institution-morphism)

(2)  (UR$morphism source)
     (= (UR$source source) institution-morphism)
     (= (UR$target source) INS.OBJ$institution)

(3)  (UR$morphism target)
     (= (UR$source target) institution-morphism)
     (= (UR$target target) INS.OBJ$institution)

(4)  (UR$morphism signature)
     (= (UR$source signature) institution-morphism)
     (= (UR$target signature) FUNC$functor)
     (= (UR$composition [signature FUNC$source]) signature)
        (UR$composition [source INS.OBJ$signature]))
     (= (UR$composition [signature FUNC$target]) signature)
        (UR$composition [target INS.OBJ$signature]))

(5)  (UR$morphism sentence)
     (= (UR$source sentence) institution-morphism)
     (= (UR$target sentence) NAT$natural-transformation)
     (forall (?F (institution-morphism ?F))
         (= (NAT$source-functor (sentence ?F))
            (FUNC$composition [(signature ?F) (INS.OBJ$sentence (target ?F))])))
     (= (UR$composition [sentence NAT$target-functor])
        (UR$composition [source INS.OBJ$sentence]))
     (= (UR$composition [sentence NAT$source-category])
        (UR$composition [source INS.OBJ$signature]))
     (= (UR$composition [sentence NAT$target-category])
        ((UR$constant [institution-morphism CAT$category]) set$Set))

(6)  (UR$morphism model)
     (= (UR$source model) institution-morphism)
     (= (UR$target model) vlrg.nat$natural-transformation)
     (= (UR$composition [model vlrg.nat$source-functor])
        (UR$composition [source INS.OBJ$model]))
     (forall (?F (institution-morphism ?F))
         (= (vlrg.nat$target-functor (model ?F))
            (vlrg.func$composition [(FUNC$opposite (signature ?F)) (INS.OBJ$model (target ?F))])))
     (= (UR$composition [model vlrg.func$source-category])
        (UR$composition [(UR$composition [source INS.OBJ$signature]) CAT$opposite]))
     (= (UR$composition [model vlrg.func$target-category])
        ((UR$constant [institution-morphism KIF.COL$collection]) (2-cat$category CAT$CAT)))

(7)  (forall (?F (institution-morphism ?F)
          ?S ((CAT$object (INS.OBJ$signature (source ?F))) ?S)
          ?M ((CAT$object ((vlrg.func$object (INS.OBJ$model (source ?F))) ?S)) ?M)
          ?e (((FUNC$object (INS.OBJ$sentence (target ?F)))
                 ((FUNC$object (signature ?F)) ?S)) ?e))
```

```
(<=> (((satisfaction (source ?F)) ?S) ?M (((sentence ?F) ?S) ?e))
     (((satisfaction (source ?F)) ((FUNC$object (signature ?F)) ?S)) (((model ?F) ?S) ?M) ?e)))
```

Two institution morphisms are composable when the target of the first is the source of the second. There is a composition operator defined from composable pairs of institution morphisms to institutions: if $\mathcal{F} = \langle sign_{\mathcal{F}}, sen_{\mathcal{F}}, mod_{\mathcal{F}} \rangle : \mathcal{A} \to \mathcal{B}$ and $\mathcal{G} = \langle sign_{\mathcal{G}}, sen_{\mathcal{G}}, mod_{\mathcal{G}} \rangle : \mathcal{B} \to \mathcal{C}$ are a composable pair of institution morphisms, the composition morphism is defined to be (expressed in detail without the usual abbreviations)

$$\mathcal{F} \circ \mathcal{G} = \langle sign_{\mathcal{F}} \circ sign_{\mathcal{G}}, (1_{sign_{\mathcal{F}}} \circ sen_{\mathcal{G}}) \bullet sen_{\mathcal{F}}, mod_{\mathcal{F}} \bullet ((1_{sign_{\mathcal{F}}})^{op} \circ mod_{\mathcal{G}}) \rangle : \mathcal{A} \to \mathcal{C}.$$

```
 (8) (UR$relation composable)
     (= (UR$object0 composable) institution-morphism)
     (= (UR$object1 composable) institution-morphism)

 (9) (UR$object composable-pair)
     (= composable-pair (UR$extent composable))

(10) (UR$morphism morphism0)
     (= (UR$source morphism0) composable-pair)
     (= (UR$target morphism0) institution-morphism)
     (= morphism0 (UR$projection0 composable))

(11) (UR$morphism morphism1)
     (= (UR$source morphism1) composable-pair)
     (= (UR$target morphism1) institution-morphism)
     (= morphism1 (UR$projection1 composable))

(12) (forall (?fg (morphism-morphism ?fg))
         (and (= ?fg [(morphism0 ?fg) (morphism1 ?fg)])
              (= (target (morphism0 ?fg)) (source (morphism1 ?fg)))))

(13) (UR$morphism composition)
     (= (UR$source composition) composable-pair)
     (= (UR$target composition) institution-morphism)
     (= (UR$composition [composition source])
        (UR$composition [morphism0 source]))
     (= (UR$composition [composition target])
        (UR$composition [morphism1 target]))
     (forall (?F (institution-morphism ?F) ?G (institution-morphism ?G)
             (composable ?F ?G))
         (and (= (signature (composition [?F ?G]))
                 (FUNC$composition [(signature ?F) (signature ?G)]))
              (= (sentence (composition [?F ?G]))
                 (NAT$vertical-composition
                     [(NAT$horizontal-composition
                          [(NAT$vertical-identity (signature ?F)) (sentence ?G)])
                      (sentence ?F)]))
              (= (model (composition [?F ?G]))
                 (vlrg.nat$vertical-composition
                     [(model ?F)
                      (vlrg.nat$horizontal-composition
                          [(vlrg.nat$vertical-identity (FUNC$opposite (signature ?F)))
                           (model ?G)])]))))
```

For any institution $\mathcal{A}$, there is an identity institution morphism $1_{\mathcal{A}} = \langle id_{sign\mathcal{A}}, 1_{sen\mathcal{A}}, 1_{|mod|\mathcal{A}} \rangle : \mathcal{A} \to \mathcal{A}$.

```
(14) (UR$morphism identity)
     (= (UR$source identity) institution)
     (= (UR$target identity) institution-morphism)
     (= (UR$composition [identity source]) (UR$identity institution))
     (= (UR$composition [identity target]) (UR$identity institution))
     (forall (?A (institution ?A))
         (and (= (signature (identity ?A))
                 (FUNC$identity (signature ?A)))
              (= (sentence (identity ?A))
                 (NAT$vertical-identity (sentence ?A)))
              (= (model (identity ?A))
                 (vlrg.nat$vertical-identity (vlrg.func$composition [(model ?A) CAT$object])))))
```

Composition satisfies the usual *associative law*.

```
(15) (forall (?f1 (institution-morphism ?f1)
              ?f2 (institution-morphism ?f2)
              ?f3 (institution-morphism ?f3)
             (composable ?f1 ?f2) (composable ?f2 ?f3))
         (= (composition [?f1 (composition [?f2 ?f3])])
            (composition [(composition [?f1 ?f2]) ?f3])))
```

Identity satisfies the usual *identity laws* with respect to composition.

```
(16) (forall (?f (institution-morphism ?f))
         (and (= (composition [(identity (source ?f)) ?f]) ?f)
              (= (composition [?f (identity (target ?f))]) ?f)))
```

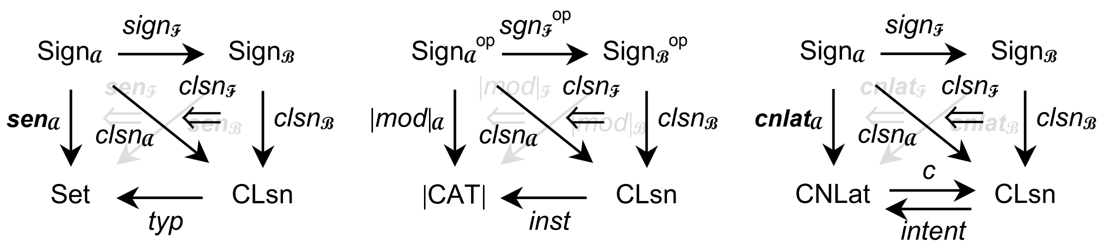

Figure 5: Classification and Concept Lattice of an Institution Morphism

Conceptual Structures

Associated with any institution morphism $\mathcal{F} : \mathcal{A} \rightarrow \mathcal{B}$ is

- a truth *classification* natural transformation $clsn_{\mathcal{F}} : sign_{\mathcal{F}} \circ clsn_{\mathcal{B}} \Rightarrow clsn_{\mathcal{A}} : \mathrm{Sign}_{\mathcal{A}} \rightarrow \mathrm{CLsn}$, and
- a truth *concept lattice* natural transformation $cnlat_{\mathcal{F}} : sign_{\mathcal{F}} \circ clsn_{\mathcal{B}} \Rightarrow clsn_{\mathcal{A}} : \mathrm{Sign}_{\mathcal{A}} \rightarrow \mathrm{CNLat}$.

By horizontally composing the truth classification natural transformation with the two projection functors, $inst : \mathrm{CLsn}^{op} \rightarrow \mathrm{SET}$ and $typ : \mathrm{Clsn} \rightarrow \mathrm{Set}$, we can unpack it (Figure 5) into

- the object model natural transformation $|\mathrm{mod}|_{\mathcal{F}} = \mathrm{clsn}_{\mathcal{F}}{}^{op} \circ \mathrm{inst} : \mathrm{Sign}_{\mathcal{F}}{}^{op} \rightarrow \mathrm{SET}$ and
- the sentence natural transformation $sen_{\mathcal{F}} = clsn_{\mathcal{F}}{}^{op} \circ res\text{-}typ : \mathrm{Sign}_{\mathcal{F}} \rightarrow \mathrm{Set}$.

We define the truth classification natural transformation from scratch (in terms of satisfaction). However, we use the intent functor to define the truth concept lattice natural transformation as $cnlat_{\mathcal{F}} = clsn_{\mathcal{F}} \circ 1_{intent}$. That is, the concept morphism of a signature is the intentional concept morphism generated by the (satisfaction) infomorphism of the signature. Then $clsn_{\mathcal{F}} = cnlat_{\mathcal{F}} \circ 1_{c}$.

```
(17) (UR$morphism classification)
     (= (UR$source classification) institution)
     (= (UR$target classification) vlrg.nat$natural-transformation)
     (forall (?F (institution-morphism ?F))
         (and (= (vlrg.nat$source-category (classification ?F))
                 (INS.OBJ$signature (source ?F)))
              (= (vlrg.nat$target-category (classification ?F)) CLS$Classification)
              (= (vlrg.nat$source-functor (classification ?F))
                 (vlrg.func$composition [(signature ?F) (INS.OBJ$classification (target ?F))]))
              (= (vlrg.nat$target-functor (classification ?F))
                 (INS.OBJ$classification (source ?F)))
              (forall (?S ((CAT$object (INS.OBJ$signature (source ?F))) ?S))
                  (and (= (CLS.INFO$instance ((vlrg.nat$component (classification ?F)) ?S))
                          (FUNC$object ((vlrg.nat$component (model ?F)) ?S)))
                       (= (CLS$type ((vlrg.nat$component (classification ?F)) ?S))
                          ((sentence ?F) ?S))))))

(18) (UR$morphism concept-lattice)
     (= (UR$source concept-lattice) institution)
     (= (UR$target concept-lattice) vlrg.nat$natural-transformation)
     (forall (?F (institution-morphism ?F))
         (and (= (vlrg.nat$source-category (concept-lattice ?F))
                 (INS.OBJ$signature (source ?F)))
              (= (vlrg.nat$target-category (concept-lattice ?F)) CL$Concept-Lattice)
```

```
              (= (vlrg.nat$source-functor (concept-lattice ?F))
                 (vlrg.func$composition [(signature ?F) (INS.OBJ$concept-lattice (target ?F))]))
              (= (vlrg.nat$target-functor (concept-lattice ?F))
                 (INS.OBJ$concept-lattice (source ?F)))
              (= concept-lattice
                 (vlrg.nat$horizontal-composition
                     [classification (vlrg.nat$vertical-identity intent)]))))

(19) (forall (?F (institution-morphism ?F))
         (= classification
            (vlrg.nat$horizontal-composition
               [concept-lattice (vlrg.nat$vertical-identity cnlat2clsn)]))))
```

Logical Environments

A central goal of the Information Flow Framework (IFF) project is to specify a theoretical framework for representing and managing ontologies. This theoretical framework is concentrated around the concept of a logical environment. The concept of a logical environment is the concept of a "structured institution". As stated on the institutions homepage, the concept of an institution formalizes, represents, implements and translates the notion of a "logic". The literature on institutions now comprises hundreds of papers. The concept of a logical environment has been discussed within the SUO IFF project during the past several years. The "structured" aspect of a logical environment is iconically illustrated in Figure 6i. This architecture con-

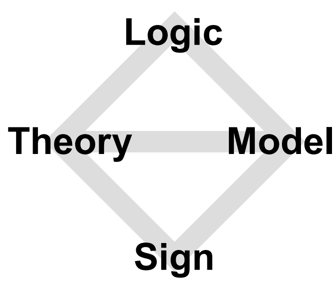

Figure 6i: Logical Environment - Iconic

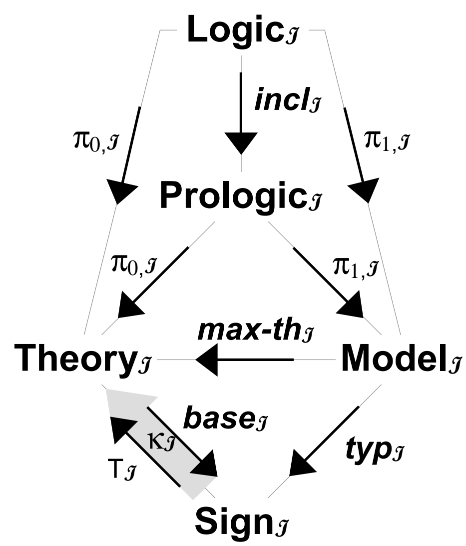

Figure 6d: Logical Environment - details

sists of the four representing concepts of signature, model, theory and (local) logic, and the five natural connections between these concepts. There is a formal-actual distinction present within this architecture. The type pole of formality consists of languages and theories. Signatures (aka languages) represent the basic formalism and theories add constraints. The instance pole of actuality consists of models and logics. Models contain instance data and classifications, and logics add constraints. Theories represent formal semantics, models represent actual/interpretive semantics, and logics represent combined semantics. Unpopulated ontologies manifesting formal semantics are represented in the IFF by theories, whereas populated ontologies incorporating actual semantics (aspects of the natural world) are represented in the IFF by logics.

- As discussed in the paper (Kent, 2000), a simple notion of a logical environment appears as the background architecture for Information Flow. Call this the IF logical environment. This was the original logical environment in the IFF. Figure 2 of (Kent, 2000) illustrates the categories and functors that form the IF logical environment. Models are IF classifications, sentences are IF sequents, theories are IF theories, and logics are IF normal logics. The lower part of Figure 2 of (Kent, 2000), regarded as an logical environment architectural diagram, is absent due to simplicity – languages are sets of type symbols, the signature projection functor for both theories and models is the underlying type set functor. A more detailed presentation of the IF institution is given below.
- The concept of a logical environment is pictured in the architecture diagram (Figure 1) on the introduction webpage for the IFF Ontology (meta) Ontology (IFF-ONT), and is theoretically discussed in the associated document "Category Theory of Ontologies". Call this the ONT logical environment. This environment, which is illustrated in Figure 1 of the document IFF-ONT work-in-progress, was briefly discussed in a presentation on the IFF metastack at the category theory conference CT04.

A goal of this paper is to make the presentation of logical environments a little more precise and understandable.

Model Variance and the Grothendieck Construction

The Grothendieck construction is discussed in Appendix A of the draft paper "Information Flow in Institutions". Let $\mathcal{J}$ be any institution. It is understood that satisfaction is at the heart of institutions, and the satisfaction condition expresses the invariance of truth under change of notation. However, more is desired. In formulating the conceptual structure of institutions, we also want satisfaction to vary appropriately along model morphisms as expressed by the following.

Model Variance Condition: For any signature Σ, satisfaction varies along model morphisms in the category $\mathit{mod}_{\mathcal{J}}(\Sigma)$ in the following sense: $N \models_{\mathcal{J}}(\Sigma)\ s$ implies $M \models_{\mathcal{J}}(\Sigma)\ s$, for any pair of models $M, N \in |\mathit{mod}_{\mathcal{J}}(\Sigma)|$, any model morphism $h : M \rightarrow N$ in $\mathit{mod}_{\mathcal{J}}(\Sigma)$, and any sentence $s \in \mathit{sen}_{\mathcal{J}}(\Sigma)$.

An institution $\mathcal{J}$ that satisfies the model variance condition is called an *orthovariant institution*. Assume that $\mathcal{J}$ is orthovariant. Define the Grothendieck model category $\mathsf{Model}_{\mathcal{J}} = \mathsf{Gr}(\mathit{mod}_{\mathcal{J}})$ with projection functor (fibration) $\mathit{typ}_{\mathcal{J}} : \mathsf{Model}_{\mathcal{J}} \rightarrow \mathsf{Sign}_{\mathcal{J}}$, and the Grothendieck theory category $\mathsf{Theory}_{\mathcal{J}} = \mathsf{Gr}(\mathit{th}_{\mathcal{J}})$ with projection functor (bifibration) $\mathit{base}_{\mathcal{J}} : \mathsf{Theory}_{\mathcal{J}} \rightarrow \mathsf{Sign}_{\mathcal{J}}$. These two Grothendieck categories are the vertices in the horizontal axis of the diagram in Figure 6d and these two projection functors form the opspan below this horizontal axis. The model variance condition is equivalent to the statement that the maximal theory operator has the following property: $\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma)(M) \supseteq \mathit{max\text{-}th}_{\mathcal{J}}(\Sigma)(N)$, which is expressible as $\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma)(M) \leq_{\mathcal{J}}(\Sigma)\ \mathit{max\text{-}th}_{\mathcal{J}}(\Sigma)(N)$ in the truth concept lattice order. Hence, for any signature Σ, the maximal theory operator is a functor $\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma) : \mathit{mod}_{\mathcal{J}}(\Sigma) \rightarrow \mathit{th}_{\mathcal{J}}(\Sigma)$. The satisfaction condition implies that maximal theory is a natural transformation $\mathit{max\text{-}th}_{\mathcal{J}} : \mathit{mod}_{\mathcal{J}} \Rightarrow \mathit{th}_{\mathcal{J}}$ from the model functor to the theory functor.

$$\begin{tikzcd}
\Sigma_1 \arrow[d, "\sigma"'] & {|\mathit{mod}_{\mathcal{J}}(\Sigma_1)|} \arrow[r, "\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma_1)"] & \mathit{th}_{\mathcal{J}}(\Sigma_1) \ s_1 \\
\Sigma_2 & {|\mathit{mod}_{\mathcal{J}}(\Sigma_2)|} \ M_2 \arrow[u, "|\mathit{mod}_{\mathcal{J}}(\sigma)|"] \arrow[r, "\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma_2)"'] & \mathit{th}_{\mathcal{J}}(\Sigma_2) \arrow[u, "\mathit{th}_{\mathcal{J}}(\sigma)"']
\end{tikzcd}$$

Proof: Let $\sigma : \Sigma_1 \rightarrow \Sigma_2$ be any signature morphism, and let $M_2 \in |\mathit{mod}_{\mathcal{J}}(\Sigma_2)|$ be any model of the target signature. For any sentence $s_1 \in \mathit{sen}_{\mathcal{J}}(\Sigma_1)$ of the source signature, $s_1 \in \mathit{max\text{-}th}_{\mathcal{J}}(\Sigma_1)(\mathit{mod}_{\mathcal{J}}(\sigma)(M_2))$ <u>iff</u> $\mathit{mod}_{\mathcal{J}}(\sigma)(M_2) \models_{\mathcal{J}}(\Sigma_1)\ s_1$ <u>iff</u> $M_2 \models_{\mathcal{J}}(\Sigma_2)\ \mathit{sen}_{\mathcal{J}}(\sigma)(s_1)$ <u>iff</u> $\mathit{sen}_{\mathcal{J}}(\sigma)(s_1) \in \mathit{max\text{-}th}_{\mathcal{J}}(\Sigma_2)(M_2)$ <u>iff</u> $s_1 \in \mathit{inv}_{\mathcal{J}}(\sigma)(\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma_2)(M_2))$ <u>iff</u> $s_1 \in \mathit{th}_{\mathcal{J}}(\sigma)(\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma_2)(M_2))$.■

Applying the Grothendieck construction to this maximal theory natural transformation results in a functor $\mathit{max\text{-}th}_{\mathcal{J}} : \mathsf{Model}_{\mathcal{J}} \rightarrow \mathsf{Theory}_{\mathcal{J}}$ that commutes with the signature projections (fibrations). This is the horizontal axis in the diagram in Figure 6d. As the proof shows, functoriality of the maximal theory operator is dependent upon the model variance condition.

Unfortunately, many institutions are not orthovariant. Fortunately, associated with any institution $\mathcal{J}$ is an orthovariant institution $|\mathcal{J}|$ called the *underlying institution* or *orthovariant associate* of $\mathcal{J}$. The underlying institution is defined as follows. For any signature Σ of $\mathcal{J}$, ignore the model morphisms in $\mathit{mod}_{\mathcal{J}}(\Sigma)$. Use the model embedding map $\iota_{\mathcal{J}}(\Sigma) : |\mathit{mod}_{\mathcal{J}}(\Sigma)| \rightarrow \mathit{cloth}_{\mathcal{J}}(\Sigma)$ to induce a preorder $\mathit{mod}_{|\mathcal{J}|}(\Sigma)$ on the class of models $|\mathit{mod}_{\mathcal{J}}(\Sigma)|$ as follows: $M \leq N$ when $\iota_{\mathcal{J}}(\Sigma)(M) \supseteq \iota_{\mathcal{J}}(\Sigma)(N)$; that is, $M \leq N$ when $\mathit{max\text{-}th}_{\mathcal{J}}(\Sigma)(M) \supseteq \mathit{max\text{-}th}_{\mathcal{J}}(\Sigma)(N)$; or equivalently, $M \leq N$ when $N \models_{\mathcal{J}}(\Sigma)\ s$ implies $M \models_{\mathcal{J}}(\Sigma)\ s$ for any sentence $s \in \mathit{sen}_{\mathcal{J}}(\Sigma)$. For any signature morphism $\sigma : \Sigma_1 \rightarrow \Sigma_2$, by using the fact (stated above) that the left adjoint is compatible with the model function, it is easy to see that the underlying model function is a monotonic function (functor) $\mathit{mod}_{|\mathcal{J}|}(\sigma) : \mathit{mod}_{|\mathcal{J}|}(\Sigma_1) \rightarrow \mathit{mod}_{|\mathcal{J}|}(\Sigma_2)$ between preorders (categories). Here is an elementary proof.

Proof: Assume that M_2 and N_2 are models in $\mathit{mod}_{\mathcal{J}}(\Sigma_2)$ and that $M_2 \leq N_2$. Also assume that s_1 is a sentence in $\mathit{sen}_{\mathcal{J}}(\Sigma_1)$ and that $\mathit{mod}_{\mathcal{J}}(\sigma)(N_2) \models_{\mathcal{J}}(\Sigma_1)\ s_1$. By the satisfaction condition, $N_2 \models_{\mathcal{J}}(\Sigma_2)\ \mathit{sen}_{\mathcal{J}}(\sigma)(s_1)$. By the induced order definition, $M_2 \models_{\mathcal{J}}(\Sigma_2)\ \mathit{sen}_{\mathcal{J}}(\sigma)(s_1)$. By the satisfaction condition, $\mathit{mod}_{\mathcal{J}}(\sigma)(M_2) \models_{\mathcal{J}}(\Sigma_1)\ s_1$.■

This preorder construction enriches the underlying model functor $|\mathit{mod}|_{\mathcal{J}} = \mathit{mod}_{\mathcal{J}} \cdot |\text{-}| : \mathsf{Sign}^{\mathrm{op}}{}_{\mathcal{J}} \rightarrow \mathsf{SET}$ from sets to (preorders as) categories, resulting in the associate model functor $\mathit{mod}_{|\mathcal{J}|} : \mathsf{Sign}_{\mathcal{J}} \rightarrow \mathsf{CAT}$. The underlying institution $|\mathcal{J}|$ has the same signature category, sentence functor, and satisfaction relations as $\mathcal{J}$, but has the associate model functor. Hence, $|\mathcal{J}|$ and $\mathcal{J}$ are the same on the object parts of their model fiber functors $|\mathit{mod}|_{\mathcal{J}}(\sigma)$ for signature morphisms $\sigma : \Sigma_1 \rightarrow \Sigma_2$. Define the associate Grothendieck category $\mathsf{Model}_{|\mathcal{J}|} = \mathsf{Gr}(\mathit{mod}_{|\mathcal{J}|})$. We can use the orthovariant associate $|\mathcal{J}|$ when the institution $\mathcal{J}$ is not orthovariant.

For orthovariant institutions $\mathcal{I}$, we can use either $\mathcal{I}$ or $|\mathcal{I}|$. Also for orthovariant institutions, there is a forgetful functor $mod_{\mathcal{I}}(\Sigma) \rightarrow mod_{|\mathcal{I}|}(\Sigma)$ that is natural in Σ; hence, using the Grothendieck construction, there is a quotient functor $\mathsf{Model}_{\mathcal{I}} \rightarrow \mathsf{Model}_{|\mathcal{I}|}$.

Prologics

A *prologic* $L = \langle T, M\rangle$ is a theory-model pair that share the same signature Σ. The category of prologics $\mathsf{Prologic}_{\mathcal{I}}$ is the pullback (fibered product) in CAT of the projection opspan under the categories $\mathsf{Theory}_{\mathcal{I}}$ and $\mathsf{Model}_{\mathcal{I}}$. Note that the model component of a prologic may satisfy only part of the theory component of the prologic; that is, only part of the model M is "normal" with respect to the theory T. A *logic* is a prologic that is normal; that is, a logic $L = \langle T, M\rangle$ is a prologic, where $M \models_{\mathcal{I}}(\Sigma)\ s$ for all $s \in thm_{\mathcal{I}}(\Sigma)(T) = T^{\bullet}$; that is, where $max\text{-}th_{\mathcal{I}}(\Sigma)(M) \supseteq axm_{\mathcal{I}}(\Sigma)(T)$; or equivalently, where $max\text{-}th_{\mathcal{I}}(\Sigma)(M) \leq T_2$. Let $\mathsf{Logic}_{\mathcal{I}}$ denote the full subcategory of logics in $\mathsf{Prologic}_{\mathcal{I}}$ with logic inclusion functor $incl_{|\mathcal{I}|} : \mathsf{Logic}_{\mathcal{I}} \rightarrow \mathsf{Prologic}_{\mathcal{I}}$.

References

- Jon Barwise and Jerry Seligman. *Information Flow: The Logic of Distributed Systems*, Cambridge Tracts in Theoretical Computer Science, vol. 44 (Cambridge, 1997).
- Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*, (Springer, 1999).
- Joseph Goguen. "A Categorical Manifesto", *Mathematical Structures in Computer Science*, 1, 1991, pages 49–67.
- Joseph Goguen and Rod Burstall. "Institutions: Abstract Model Theory for Specification and Programming". *Journal of the Association for Computing Machinery*, 39, no. 1, 1992, pages 95–146. Preprint, Report CSLI-85-30, Center for the Study of Language and Information, Stanford University, 1985.
- Joseph Goguen and Grigore Rosu. "Institution Morphisms". *Formal Aspects of Computing*, 13, 2002, pages 274–307.
- Robert Kent. "The Information Flow Foundation for Conceptual Knowledge Organization". In: *Dynamism and Stability in Knowledge Organization*. Proceedings of the Sixth International ISKO Conference. Advances in Knowledge Organization 7 (2000), pp. 111–117. Ergon Verlag, Würzburg.
- Robert Kent. "Distributed Conceptual Structures". In: Proceedings, Sixth International Workshop on Relational Methods in Computer Science, Harre de Swart, editor. *Lecture Notes in Computer Science*, vol. 2561 (Springer, 2002), pages 104–123.
- Robert Kent. "The IFF Foundation for Ontological Knowledge Organization". In: *Knowledge Organization and Classification in International Information Retrieval* (Haworth, 2003).
- Robert Kent. "Semantic Integration in the IFF". ISWC'03 Semantic Integration Workshop. Sanibel Island, Florida, USA, 2003.
- Till Mossakowski. "Comorphism-based Grothendieck logics". In K. Diks, W. Rytter (Eds.), Mathematical Foundations of Computer Science, Vol. 2420, pp. 593–604, *Lecture Notes in Computer Science*. Springer-Verlag, London, 2002.
- John Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations* (Brookes/Coles, 2000).
- Alfred Tarski. "The Semantic Conception of Truth", *Philosophy and Phenomenological Research*, 4, 1944, pages 13–47.