Towards arrow-theoretic semantics of ontologies: conceptories

Osman Bineev  
binnev@gmail.com

15 July 2010

Abstract

In context of efforts of composing category-theoretic and logical methods in the area of knowledge representation we propose the notion of conceptory. We consider intersection/union and other constructions in conceptories as expressive alternative to category-theoretic (co)limits and show they have features similar to (pro-, in-)jections. Then we briefly discuss approaches to development of formal systems built on the base of conceptories and describe possible application of such system to the specific ontology.

Contents

1 Introduction 1
2 f-categories 3
3 f-lattices 6
4 Conceptories and their language 8
5 Appendix A 11

1 Introduction

Ontologies [1] are used in computer science for representing and sharing knowledge about the real world. Usually ontological structures are described in terms of classes (of things) and relationships (between things). This is rather similar to category-theoretic notions of objects and morphisms (see [2,3] for information about the algebraic category theory). Since the category theory already brings us many benefits in other areas of computer science, it is desirable to find arrow-theoretic approaches in the area of knowledge representation.
Some authors proposed category-theoretic techniques helpful in different aspects of knowledge representation\cite{5,6}. Usually they operate with (co)limits that are convenient for merging and interoperating between existing models and meta-models. Our aim is to find a category-theoretic tools that would be useful for description of ontological models from the very beginning.

In order of informal discussion, consider relations between people, like following:

\[ \text{Managers} \xrightarrow{\text{management}} \text{Employees} \]

Here both ‘Managers’ and ‘Employees’ are classes of individuals (within presumed classification) and ‘management’ can be considered as relation, something similar to morphism in category Rel of sets and relations. We can also consider some wider relations, for example ‘co-working’:

\[ \begin{array}{c}
\text{Managers} \\
\xRightarrow{\text{co-working}} \\
\text{Employees}
\end{array} \]

Here the bold arrow shows something new: the ‘co-working’ relation includes ‘management’ (considered as simply institutional relation between people, not a process!).

But we also know how to express this fact in the language of category theory. In order to consider ‘arrows between arrows’ we can use 2-categories\cite{3}.

Something really interesting happens when we consider arrow-theoretically the following picture:

\[ \begin{array}{c}
\text{Managers} \\
\xRightarrow{\text{management}} \\
\text{Employees} \\
\xforall{\text{management}} \\
\text{Engineers}
\end{array} \]

This picture is intuitively correct: the ‘management of employees’ includes, in some sense, ‘management of engineers’. But the problem with this picture is that targets of arrows connected by bold arrow should coincide, if we wish to continue using the theory of 2-categories.

Experienced mathematician would say there is the arrow between ‘Engineers’ and ‘Employees’ that makes the picture correct in 2-categories. However from the point of view of knowledge representation this is mistake: there is no significant relationship, like ‘management’ or ‘co-working’ or something else. But certainly there is some inclusion of the other nature.

Starting from this point we develop the theory of conceptories, where it is possible to express such situations and other things convenient in knowledge representation problems.

Conceptories and informal description of their language, together with discussion of possible application to the ontology of international standard ISO 15926, are introduced in the\cite{section4} of this paper; reader who is not interested in mathematical details may skip to there.
2 f-categories

Before introducing conceptories we should define wider notion of f-category. The definition is rather complicated, but gives us almost all the instruments needed to work with ontologies. Besides, this complexity is typical for (⩾ 2)-dimensional constructions in category theory.

What is f-category?

f-category C consists of

- Category C_1 with objects (0-cells of C) denoted A, B, A_1, A_2, ... and arrows (1-cells of C) denoted f, g, f_1, f_2, ...

- Category C_v where objects are 1-cells of C and arrows (2-cells of C) are denoted α, β, α_1, α_2, ... Note that we do not require domain and codomain of 2-cell to be parallel 1-cells. Composition of arrows in C_v is called vertical composition of 2-cells.

- For any pairs (f_1, f_2), (g_1, g_2) of composable 1-cells and for any 2-cells α: f_1 → g_1, β: f_2 → g_2 the associative horizontal composition

\[ \beta \star \alpha: f_2 \circ f_1 \rightarrow g_2 \circ g_1, \]

subject to the following interchange law:

\[ (\alpha_2 \star \beta_2) \circ (\alpha_1 \star \beta_1) = (\alpha_2 \circ \alpha_1) \star (\beta_2 \circ \beta_1) \] (1)

- For any 2-cells α: id_{A'} → id_A and β: id_{B'} → id_B operations \( \nabla_{\alpha,\beta} \) and \( \Delta_{\alpha,\beta} \) stated as follow.

Denote \( C_{\alpha,\beta} \) the subcategory of \( C_v \), consisting of full subcategories \( \text{Hom}_{C_1}(A, B) \), \( \text{Hom}_{C_1}(A', B') \) and all 2-cells \( \gamma \) such that

\[ \gamma \star \alpha = \beta \star \gamma = \gamma \] (2)
Consider inclusion functors:

\[ U: \text{Hom}_{C_1}(A, B) \to C_{\alpha, \beta} \]
\[ U': \text{Hom}_{C_1}(A', B') \to C_{\alpha, \beta}. \]

In category \(C_{\alpha, \beta}\) operation \(\nabla_{\alpha, \beta}\) provides for any 1-cell \(f: A \to B\) universal arrow from \(U'\) to \(f\).

\[ \begin{array}{c}
A \\
\downarrow \alpha \quad \downarrow \nabla_{\alpha, \beta}(f) \quad \downarrow \beta \\
A' \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
B \\
\downarrow \beta \\
B' \\
\end{array} \]

And operation \(\Delta_{\alpha, \beta}\) provides for any 1-cell \(f': A' \to B'\) universal arrow from \(f'\) to \(U\).

\[ \begin{array}{c}
A \\
\downarrow \alpha \\
A' \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
B \\
\downarrow \beta \\
B' \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
A \\
\downarrow \Delta_{\alpha, \beta}(f') \\
A' \\
\end{array} \]

Operations \(\nabla\) and \(\Delta\) are subjects to following restrictions:

\[ \nabla_{\text{id}_A, \beta}(f) \bullet \nabla_{\alpha, \text{id}_B}(g) = \nabla_{\alpha, \beta}(f \circ g) \] (3)
\[ \Delta_{\text{id}_A, \beta}(f) \bullet \Delta_{\alpha, \text{id}_B}(g) = \Delta_{\alpha, \beta}(f \circ g) \] (4)

That’s all about the definition of \(f\)-category. In addition, denote

\[ f_{\downarrow \alpha, \beta} \equiv \text{dom}_{C_1}(\nabla_{\alpha, \beta}(f)) \quad f_{\uparrow \alpha, \beta} \equiv \text{cod}_{C_1}(\Delta_{\alpha, \beta}(f)) \]
\[ f_{\downarrow \alpha} \equiv \text{dom}_{C_1}(\nabla_{\alpha, \text{id}_B}(f)) \quad f_{\uparrow \alpha} \equiv \text{cod}_{C_1}(\Delta_{\alpha, \text{id}_B}(f)) \]
\[ f_{\downarrow \beta} \equiv \text{dom}_{C_1}(\nabla_{\text{id}_A, \beta}(f)) \quad f_{\uparrow \beta} \equiv \text{cod}_{C_1}(\Delta_{\text{id}_A, \beta}(f)) \]

Example: \(f\text{-Rel}\)

Let \(X\) be a set and \(\text{Rel}(X)\) be a full subcategory of \(\text{Rel}\) generated by all subsets of \(X\). \(\text{Rel}(X)\) can be turned to \(f\)-category \(f\text{-Rel}(X)\) taking \(f\text{-Rel}(X)_1 = \text{Rel}(X)\).

Recall that morphism \(f: A \to B\) of \(\text{Rel}\) is a triple \(\langle r_f, A, B \rangle\) where \(A\) and \(B\) are sets and \(r_f \subseteq A \times B\). The only 2-cell \(f \to g\) exists iff \(r_f \subseteq r_g\). Operations \(\uparrow\) and \(\downarrow\) for \(f\) are introduces as follows:

\[ \langle r_f, A, B \rangle \downarrow_{A', B'} \equiv \langle (A' \times B') \cap r_f, A', B' \rangle \], where \(A' \subseteq A, B' \subseteq B\)
\[ \langle r_f, A, B \rangle \uparrow_{A', B'} \equiv \langle r_f, A', B' \rangle \], where \(A \subseteq A', B \subseteq B'\)

Operation ‘\(\uparrow\)’ takes place: if \(r_{f_1} \subseteq r_{g_1}\) and \(r_{f_2} \subseteq r_{g_2}\) then obviously \(r_{f_2} \circ r_{f_1} \subseteq r_{g_2} \circ r_{g_1}\). It is associative and satisfies interchange law just because there is no alternative.
Functoriality of ↓ and ↑

Note that for given 0-cell $A$ the category $\text{Hom}_C(A,A)$ can be considered as monoidal, where action of tensor on objects is given by $\circ$ (composition of 1-cells), on arrows by $\star$ (horizontal composition of 2-cells) and tensor unit $\text{id}_A$. Denote this monoidal category $H_A$.

**Proposition 2.1** For any 2-cells $\alpha: \text{id}_{A'} \to \text{id}_A$, $\beta: \text{id}_{B'} \to \text{id}_B$ and $\gamma: \text{id}_{C'} \to \text{id}_C$, any 1-cells $f: A \to B$, $g: B \to C$, $f': A' \to B'$, $g': B' \to C'$

there is at least one 2-cell in $\text{Hom}_C(g\downarrow_\beta \circ f\downarrow_\alpha \circ \gamma)$ and at least one 2-cell in $\text{Hom}_C((g \circ f')\downarrow^{\alpha,\gamma}, (g' \circ f')\downarrow^{\alpha,\beta})$.

**Proof:** First 2-cell exists by universality of $\nabla_{\alpha,\gamma}$ as a factorizing arrow for $\nabla_{\beta,\gamma}(g) \ast \nabla_{\alpha,\beta}(f)$. And second by universality of $\Delta_{\alpha,\gamma}$ as a factorizing arrow for $\Delta_{\beta,\gamma}(g') \ast \Delta_{\alpha,\beta}(f')$. □

**Proposition 2.2** (Functoriality of ↓ and ↑)

1. For given 2-cells $\alpha: \text{id}_{A'} \to \text{id}_A$ and $\beta: \text{id}_{B'} \to \text{id}_B$ operation $\downarrow_{\alpha,\beta}$ induces functor 

   $\downarrow_{\alpha,\beta}: \text{Hom}_C(A,B) \to \text{Hom}_C(A',B')$.

2. For given 2-cell $\alpha: \text{id}_{A'} \to \text{id}_A$ the functor $\downarrow_{\alpha,\alpha}$ has monoidal structure:

   $(\downarrow_{\alpha,\alpha}, \phi_{\alpha,\alpha}, \phi_{\alpha,0}): H_A \to H_{A'}$.

3. Same for ↑, except that $\uparrow^{\alpha,\alpha}$ is comonoidal functor.

Proof can be found in the Appendix A of this paper.

It is obvious that restricting the class of 2-cells to those between parallel 1-cells in $\mathcal{F}$-category gives us some 2-category. Therefore we may adopt some notions of 2-categories like, for example, adjunctions [3].

Although research in this direction is not in goals of this paper, we would like to state hypothesis, that could unite both propositions 2.1 and 2.2 in one:

**Hypothesis 2.1** (2-functoriality of ↓ and ↑)

For any function $\xi: \text{Ob}(\mathcal{C}_1) \to \text{Mor}(\mathcal{C}_v)$, such that always

$\text{dom}(\xi(A)) = \text{id}_A \land \exists B : \text{cod}(\xi(A)) = \text{id}_B$

the operation $f \mapsto f\downarrow_{\xi(\text{dom}(f))}\xi(\text{cod}(f))$ has structure of 2-endofunctor over $\mathcal{C}$ taken as 2-category.
3 f-lattices

Recall the notion of thin category: it is a category having at most one morphism in each homset. The f-category $\mathcal{C}$ is called f-lattice iff $\mathcal{C}_v$ is thin category with finite products and finite coproducts, where bifunctors of binary product ‘$\cup$’ and of binary product ‘$\cap$’ have following properties:

\[
\begin{align*}
\text{id}_{\text{dom}(f \cup g)} &= \text{id}_{\text{dom}(f)} \cup \text{id}_{\text{dom}(g)} & (5) \\
\text{id}_{\text{cod}(f \cup g)} &= \text{id}_{\text{cod}(f)} \cup \text{id}_{\text{cod}(g)} & (6) \\
\text{id}_{\text{dom}(f \cap g)} &= \text{id}_{\text{dom}(f)} \cap \text{id}_{\text{dom}(g)} & (7) \\
\text{id}_{\text{cod}(f \cap g)} &= \text{id}_{\text{cod}(f)} \cap \text{id}_{\text{cod}(g)} & (8)
\end{align*}
\]

We will write $A \cup B$ instead of $\text{dom}(\text{id}_A \cup \text{id}_B)$ and $A \cap B$ instead of $\text{dom}(\text{id}_A \cap \text{id}_B)$. It is also convenient to use expression $A \leq B$ (for 0-cells $A$ and $B$) as shortened form of $\text{id}_A \leq \text{id}_B$.

Since, when work with f-lattices, there is at most one 2-cell in each $\mathcal{C}_v$-homset, we will use domains (codomains) of id’s of 2-cells in indices of operation $\nabla$. For instance, $\nabla_{A,B}(f)$ is defined when $A \leq \text{dom}(f)$ and $B \leq \text{cod}(f)$. And same for $\nabla$, $\nabla$, $\nabla$, $\nabla$.

Operations $|$ and $\nabla$

Call the predicate $X$ over 1-cells $|$-preserved ($\nabla$-preserved) iff for any $f : A \rightarrow B$ proposition $X(f \mid A)$ (proposition $X(f \mid B)$) follows from $X(f)$ whenever $A \leq A$ (whenever $B \leq B$). Using our results 2.1 and 2.2 it’s easy to prove the following proposition:

**Proposition 3.1** In any f-lattice following predicates (of argument $f : A \rightarrow B$) are both $|$-preserved and $\nabla$-preserved:

1. $\exists g : \text{id}_A \leq f \circ g$
2. $\exists g : g \circ f \leq \text{id}_B$
3. $\exists g : (\text{id}_A \leq f \circ g) \land (f \circ g \circ f = f)$
4. $\exists g : (g \circ f \leq \text{id}_B) \land (g \circ f \circ g = g)$

**Sketch proof**: $\text{id}_A \leq (\text{id}_A)_{\nabla A,A}$ follows from existing of $\phi_{A,I}$ in monoidal structure of functor $\nabla_{A,A}$ and $(\text{id}_B)_{\nabla B,B} \leq \text{id}_B$ dually from comonoidal structure. Next, $g_{\nabla A} \circ f_{\nabla A} = (g \circ f)_{\nabla A,A}$ by property 3 and $f_{\nabla B} \circ g_{\nabla B} = (f \circ g)_{\nabla B,B}$ by property 4 from the definition of f-category. These facts together with functoriality of $\nabla$ and $\nabla$ are enough to prove $|$-preserving of the first inequality and $\nabla$-preserving of the second.

Then for $|$-preserving of $\text{id}_A \leq f \circ g$ and $\nabla$-preserving of $g \circ f \leq \text{id}_B$ we need the proposition 2.1 in order to prove $g_{\nabla A} \circ f_{\nabla A} \leq (g \circ f)_{\nabla A,A}$ and $f_{\nabla B} \circ g_{\nabla B} \geq (f \circ g)_{\nabla B,B}$.

Preserving of the last two inequalities follows easily from the properties 4 and 2 of f-categories, from proposition 2.1 and using the $\star$ operation for 2-cells. $\square
Call the 1-cell $f: A \to B$ a map iff it has right adjoint. As easy to see, in $f$-lattices it is equal to existing of $g: B \to A$ such that

$$(\text{id}_A \leq f \circ g) \land (g \circ f \leq \text{id}_B) \land (f \circ g \circ f = f) \land (g \circ f \circ g = g)$$

**Corollary 3.1** The predicate ‘to be a map’ is both $|$-preserved and $\|$-preserved.

**Corollary 3.2** For any 0-cells $A \leq B$ both $(\text{id}_A)|^B$ and $(\text{id}_B)|_A$ are maps.

Two last results give us reasons of importance of operations $|$ and $\|$: they both preserve maps and make maps from id’s. Another reason is that these operations are enough to introduce *hybrid composition* ‘$*$’ of 2-cells and 1-cells (again preserving maps!), not only in $f$-lattices but in general $f$-categories, as shown on the following picture:

**Involution** $(−)^\circ$

In some cases it is useful to have for each 1-cell $f: A \to B$ it’s ‘transposition’ $f^\circ: B \to A$. Maybe definition of $f$-lattice should be extended by such operation, by analogy with *allegories*\footnote{4}. In order to do that we should equip our $f$-lattice $\mathcal{C}$ with involutive contravariant endofunctor $(−)^\circ$ over $\mathcal{C}_1$ such that $\text{dom}(f^\circ) = \text{cod}(f)$ and $\text{cod}(f^\circ) = \text{dom}(f)$. In addition it should satisfy the *modular law*:

$$(f \circ g) \cap h \leq (f \cap (h \circ g^\circ)) \circ g$$

The useful property of such involution is that whenever $f$ has right adjoint $g$ it coincides with $f^\circ$: $g = f^\circ$ \footnote{4}.

**Constructions in $f$-lattices**

Following constructions can be useful in real tasks of knowledge representation:

- $A \cap B$ and $A \cup B$: 

![Diagram of constructions in $f$-lattices](image-url)
Note that, by analogy with categorical binary products and sums (coproducts) \[2\], we have ‘projections’

\[
pr_A \equiv (id_A)_{|A \cap B} \\
pr_B \equiv (id_B)_{|A \cap B}
\]

and ‘injections’

\[
in_A \equiv (id_A)_{|A \cup B} \\
in_B \equiv (id_B)_{|A \cup B}
\]

By corollary \[3.2\] all of them are maps. Surely we may consider other operations (⇂ and ↾) but they are not bounded to produce maps.

- Intuitively, if ordering in \(f\)-lattice is understood as subclassing, we can understand \(⇂\) as inheritance of features, relationships, methods, etc. For example, when \(A\) and \(B\) have 1-cells \(f: A \to A'\) and \(f: B \to B'\), \(A \cap B\) inherits both via \(⇂\):

- Consider situation when \(A\) and \(B\) have 1-cells \(f: A \to C\) and \(g: B \to C\) and we wish to inherit them and unite in one 1-cell. We could use \(f \cap g\), since by definition \(\text{dom}(f \cap g) = A \cap B\). But this is not convenient, because \(f \cap g\) is not guaranteed to be a map. In fact we need another operation \(f \cap_C g\) such that

\[
(f|_{\text{dom}(f \cap_C g)} = f \cap_C g) \land (g|_{\text{dom}(f \cap_C g)} = f \cap_C g)
\]

and \(h \leq f \cap_C g\) for any other \(h\) with these properties.

Call it logical pullback. Logical pullbacks preserve good properties, but as far as we can see this construction does not follow from the definition of \(f\)-lattice. This operation is similar to pullback square \[2\] from the usual category theory, so it is good subject for the future research.
4 Conceptories and their language

If a \( f \)-lattice is, in addition, complete heyting (boolean) algebra, then we call it **conceptory (boolean conceptory)**. But this section is dedicated not so much to mathematical properties of conceptories, but to (rather informal) description of possible formal system that this notion induces and that could be used in ontological applications.

**Formal system**

Thus, 0-cells \((A, B, C, A_1, A_2, \ldots)\) of conceptory become _classes_ \((A, B, C, A_1, A_2, \ldots)\) of our ontological language and 1-cells \((f, g, h, f_1, f_2, \ldots)\) become typed _relationships_ \((f, g, h, f_1, f_2, \ldots)\). As before we may describe domain and codomain of relationship, for example \(f: A \rightarrow B\), and compose them, \(f \circ g\), in associative way. The apparatus of heyting or boolean algebras gives us the full set of logical connectives over relationships and classes, with the usual collection of axioms. It is not necessary to describe them here — we are going to concentrate on specific axioms and rules of conceptories.

First introduce some auxiliary axioms:

\[
\begin{array}{c}
\text{def}_{A,B}(f \downarrow_{A',B'}) \\
\text{def}_{A,B}(f \uparrow_{A',B'}) \\
\end{array}
\]

Now describe specific axioms and rules of \( f \)-categories. The first one comes from the \( \star \) operation over 2-cells:

\[
\begin{array}{c}
f_1 \Rightarrow g_1 \\
f_2 \Rightarrow g_2 \\
\end{array}
\]

Note that associativity of \( \star \) and interchange law \( \square \) are guaranteed, since there are no alternatives.

Then we have several axioms for \( \downarrow \) and \( \uparrow \), as consequences of definitions of \( \lor \) and \( \land \):

\[
\begin{array}{c}
\text{def}_{A,B}(f \downarrow_{A',B'}) (\downarrow) \\
\text{def}_{A,B}(f \uparrow_{A',B'}) (\uparrow) \\
\end{array}
\]

Sometimes we will omit \( \text{def}_{A,B}(f \downarrow_{A',B'}) \) and \( \text{def}_{A,B}(f \uparrow_{A',B'}) \) for simplicity of formulae. Following axioms come from the definition of \( f \)-lattice:

\[
\begin{array}{c}
f: A \rightarrow B \quad g: C \rightarrow D \\
(f \land g):(A \land C) \rightarrow (B \land D) \quad (\text{bounds}\land) \\
f: A \rightarrow B \quad g: C \rightarrow D \\
(f \lor g):(A \lor C) \rightarrow (B \lor D) \quad (\text{bounds}\lor)
\end{array}
\]
In order to introduce elements to our language we could add distinguished class \( \top \) and use \( x : A \) and \( f(x) \) instead of \( x : \top \to A \) and \( f \circ x \) correspondently. But usually we don’t need such extension of language thanks to the power of algebraic representation.

Proposition 2.1 and 2.2 give us following theorems:

\[
\begin{align*}
(g \circ f) \downarrow A', C' &\Rightarrow (g \circ f) \downarrow A', C' \\
(g \circ f) \uparrow A', C' &\Rightarrow g \uparrow A', C' \circ f \uparrow A', B'
\end{align*}
\]

(9) \hspace{1cm} (10)

\[
\begin{align*}
f \Rightarrow g \\
\hline
f \downarrow A', B' &\Rightarrow g \downarrow A', B'
\end{align*}
\]

(11)

\[
\begin{align*}
f \Rightarrow g \\
\hline
f \uparrow A', B' &\Rightarrow g \uparrow A', B'
\end{align*}
\]

(12)

\[
\begin{align*}
id_{A'} &\Rightarrow (id_A) \downarrow A', A' \\
(id_A) \uparrow A', A' &\Rightarrow id_A
\end{align*}
\]

(13) \hspace{1cm} (14)

By convention we will use sometimes \( A \cdot f : B \) instead of \( f : A \rightarrow B \) and \( A \cdot f \) instead of \( f \) when domain of \( f \) is proven to be \( A \).

Towards application to ISO 15926

ISO 15926 [7] is international standard for industrial automation systems and integration. Part 2 of this standard contains description of ontology consisting of 201 entity types, with EXPRESS language. Part ISO 15926-7 describes the same information in the language of first order logic (FOL).

Texts of [EXPRESS representation] and [FOL representation] of the standard are available online.

Although currently we don’t have any consistency checking algorithms for concepts, we hope to obtain such algorithms and below we make preliminary notes of how to describe ISO 15926 entities in conceptual language.

Let \( \text{abcd} \) is EXPRESS entity type. We present this in conceptual language as class (something corresponding to 0-cell) \( A\text{bcd} \). For example, entity type \( \text{class.of_relationship} \) becomes conceptual class \( \text{ClassOfRelationship} \).

Then, attributes will be presented in our language as 1-cells. For example, entity type \( \text{classification} \) declares two attributes: \( \text{classified} \) of type \( \text{thing} \) and \( \text{classifier} \) of type \( \text{class} \). In our language this facts are presented as follow:

\[
\begin{align*}
\text{classified:Classification} &\rightarrow \text{Thing} \\
\text{classifier:Classification} &\rightarrow \text{Class}
\end{align*}
\]

or, using syntactic sugar described in previous paragraph, simply

\[
\begin{align*}
\text{Classification.classified:Thing} \\
\text{Classification.classifier:Class}
\end{align*}
\]
Several types of information are represented using almost same principles as ISO 15926-7 uses in case of FOL. Subtyping among entity types is represented using implication; $A \Rightarrow B$ means that $A$ is subclass of $B$. If entity type $a$ is ABSTRACT and has immediate subtypes $b, c, d$, we present this in form

$$A \Rightarrow B \lor C \lor D$$

The EXPRESS $\text{ONEOF}(a, b, c)$ is represented with formulae

$$\neg(A \land (B \lor C))$$
$$\neg(B \land C)$$

Next, by convention, if we have $A.f: B$ and $C \Rightarrow A$ then we understand $C.f$ as $f|_C$. But if $C$ redeclares codomain of $f$ (to $B'$ for instance) it should be described distinctly:

$$C.f \leftrightarrow (A.f)|_{C'}$$

where $\bullet$ signs adopting name of domain $C$ from the name of 1-cell. Here is an example from the standard:

$$\text{UpperBoundOfNumberRange.classified} \leftrightarrow \text{Classification.classified}_{\bullet, \text{ArithmeticNumber}}$$

Till now we didn’t use any variables over elements of conceptual classes. Cardinality constraints can also be expressed in this style using involutive operation $(\cdot)^\circ$ (described in previous section) and in terms of proposition 3.1. For example, cardinality constraint $[1, \ast)$ for 1-cell $A.f: B$ is presented in form

$$\text{id}_A \Rightarrow A.f^\circ \circ A.f,$$

Next, the cardinality constraint $[0, 1]$ is presented as

$$A.f \circ A.f^\circ \Rightarrow \text{id}_B$$

And finally, EXPRESS UNIQUE restriction is presented as

$$A.f^\circ \circ A.f \Rightarrow \text{id}_A$$

5 Appendix A

Proof of the proposition 2.2 (functoriality of $\downarrow$ and $\uparrow$):

1. Consider 1-cells $f, g: A \to B$ and 2-cell $\gamma: f \to g$. With $\nabla$ we obtain 2-cells $\nabla_{\alpha, \beta}(g): g_{\downarrow_{\alpha, \beta}} \to g$ and $\gamma \circ \nabla_{\alpha, \beta}(f): f_{\downarrow_{\alpha, \beta}} \to g$. By universality of $\nabla$ there is unique 2-cell $\gamma': f_{\downarrow_{\alpha, \beta}} \to g_{\downarrow_{\alpha, \beta}}$ such that

$$\nabla_{\alpha, \beta}(g) \circ \gamma' = \gamma \circ \nabla_{\alpha, \beta}(f).$$

(15)
Define $\gamma_{\alpha,\beta} \equiv \gamma'$. It’s obvious from definition that $(\text{id}_f)_{\alpha,\beta} = \text{id}_{(f)_{\alpha,\beta}}$.

The rule

$$(\gamma_1 \circ \gamma_2)_{\alpha,\beta} = \gamma_1_{\alpha,\beta} \circ \gamma_2_{\alpha,\beta}$$

can be derived from commutativity of the outer square of the following diagram:

$$\begin{array}{ccc}
f & \xrightarrow{\gamma_2} & g & \xrightarrow{\gamma_1} & h \\
\downarrow_{\nabla_{\alpha,\beta}(f)} & & \downarrow_{\nabla_{\alpha,\beta}(g)} & & \downarrow_{\nabla_{\alpha,\beta}(h)} \\
f_{\downarrow_{\alpha,\beta}} & \xrightarrow{\gamma_2} & g_{\downarrow_{\alpha,\beta}} & \xrightarrow{\gamma_1} & h_{\downarrow_{\alpha,\beta}}
\end{array}$$

2. First derive $\phi_{\alpha,0}$. Both 2-cells $\alpha$ and $\nabla_{\alpha,\alpha}(\text{id}_A)$ have codomain $\text{id}_A$, so by universality there is unique $\gamma'' : \text{id}'_A \rightarrow (\text{id}_A)_{\downarrow_{\alpha,\alpha}}$ such that

$$\alpha = \nabla_{\alpha,\alpha}(\text{id}_A) \circ \gamma'' \quad (16)$$

Now define $\phi_{\alpha,0} \equiv \gamma''$.

We also need $\phi_{\alpha}$. Consider 1-cells $f : A \rightarrow A$ and $g : A \rightarrow A$. Together with $\nabla_{\alpha,\alpha}(f \circ g)$ we have $\nabla_{\alpha,\alpha}(f) * \nabla_{\alpha,\alpha}(g)$ and, by universality again, the 2-cell $\gamma''' : f_{\downarrow_{\alpha,\alpha}} \circ g_{\downarrow_{\alpha,\alpha}} \rightarrow (f \circ g)_{\downarrow_{\alpha,\alpha}}$ with property

$$\nabla_{\alpha,\alpha}(f \circ g) \circ \gamma''' = \nabla_{\alpha,\alpha}(f) \ast \nabla_{\alpha,\alpha}(g) \quad (17)$$

Define $\phi_{\alpha}(g, f) \equiv \gamma'''$. Naturality of $\phi_{\alpha}$ by $g$ means that

$$\phi_{\alpha}(g, f) \circ (\text{id}_f)_{\downarrow_{\alpha,\alpha}} \ast \beta'_{\downarrow_{\alpha,\alpha}} = (\text{id}_f \ast \beta')_{\downarrow_{\alpha,\alpha}} \circ \phi_{\alpha}(h, f) \quad (18)$$
for any \( \beta': h \to g \). Let’s prove it. First, by (17) we have
\[
\nabla_{a,a}(f \circ g) \circ \phi_a(g, f) \circ (\id_f \downarrow_{a,a} \ast \beta' \downarrow_{a,a}) = \\
(\nabla_{a,a}(f) \ast \nabla_{a,a}(g)) \circ (\id_f \downarrow_{a,a} \ast \beta' \downarrow_{a,a})
\]
Let’s modify the right side of this equality, using (15):
\[
\nabla_{a,a}(f \circ g) \circ \phi_a(g, f) \circ (\id_f \downarrow_{a,a} \ast \beta' \downarrow_{a,a}) = (\id_f \ast \beta') \circ (\nabla_{a,a}(f) \ast \nabla_{a,a}(h))
\]
Now, again by property (17) (but with different components)
\[
\nabla_{a,a}(f \circ g) \circ \phi_a(g, f) \circ (\id_f \downarrow_{a,a} \ast \beta' \downarrow_{a,a}) = (\id_f \ast \beta') \circ \nabla_{a,a}(f \circ h) \circ \phi_a(h, f)
\]
And by (15)
\[
\nabla_{a,a}(f \circ g) \circ \phi_a((g, f) \circ (\id_f \downarrow_{a,a} \ast \beta' \downarrow_{a,a}) = \\
\nabla_{a,a}(f \circ g) \circ (\id_f \ast \beta') \downarrow_{a,a} \ast \phi_a(h, f)
\]
Since in the last equation both factors of \( \nabla_{a,a}(f \circ g) \) belong the image of \( U' \), we may reduce \( \nabla_{a,a} \) by it’s universality to obtain (18).

Next we should prove
\[
\phi_a(g \circ h, f) \circ (\id_f \downarrow_{a,a} \ast \phi_a(h, g)) = \phi_a(h, f \circ g) \circ (\phi_a(g, f) \ast \id_h \downarrow_{a,a}) \quad (19)
\]
Composing the left part of (19) with \( \nabla_{a,a}(f \circ g \circ h) \) and modifying using (17) twice, we have
\[
\nabla_{a,a}(f \circ g \circ h) \circ \phi_a(g \circ h, f) \circ (\id_f \downarrow_{a,a} \ast \phi_a(h, g)) = \\
(\nabla_{a,a}(f) \ast \nabla_{a,a}(g)) \ast \nabla_{a,a}(h)
\]
The same result \( (\nabla_{a,a}(f) \ast \nabla_{a,a}(g) \ast \nabla_{a,a}(h)) \) can be obtained from the right part of (19) composed with \( \nabla_{a,a}(f \circ g \circ h) \) using similar procedures. It means that
\[
\nabla_{a,a}(f \circ g \circ h) \circ \phi_a(g \circ h, f) \circ (\id_f \downarrow_{a,a} \ast \phi_a(h, g)) = \\
(\nabla_{a,a}(f \circ g \circ h) \circ \phi_a(h, f \circ g) \circ (\phi_a(g, f) \ast \id_h \downarrow_{a,a})
\]
But this factor \( \nabla_{a,a}(f \circ g \circ h) \) can be reduced by it’s universality and we obtain (19).

Finally we need following equations:
\[
\phi_a(\id_A, f) \circ (\id_f \downarrow_{a,a} \ast \phi_a, 0) = \id_f \downarrow_{a,a} \quad (20)
\]
\[
\phi_a(f, \id_A) \circ (\phi_a, 0 \ast \id_f \downarrow_{a,a}) = \id_f \downarrow_{a,a} \quad (21)
\]
Prove only last one, because proofs are similar. Composing the left part with \( \nabla_{a,a}(f) \) and modifying it with (17) we obtain \( (\nabla_{a,a} \circ \phi_a, 0) \ast \nabla_{a,a}(f) \), which is equal to \( \alpha \ast \nabla_{a,a}(f) \) by (10). But by (2) it is equal to \( \nabla_{a,a}(f) \). And again we may reduce \( \nabla_{a,a}(f) \) to obtain the desirable result.

3. The proof of this part of proposition is very similar to proofs of previous parts, so we don’t repeat that. □
References

[1] Staab S and Studer R. (editors) *Handbook on Ontologies* 2nd ed. Springer, Berlin, 2009.

[2] M. Barr and C. Wells, *Category Theory. Lecture Notes for ESSLLI* 1999.

[3] S. Mac Lane, *Categories for Working Mathematician* 2nd Ed. Springer, 2008.

[4] P.T. Johnstone, *Sketches of an elephant: a topos theory compendium. Vol. 1*. Oxford University Press, Oxford, 2002.

[5] I. Cafezeiro and E.H. Haeusler, *Semantic interoperability via category theory*, in *26th international conference on Conceptual modeling*. 2007: Auckland, New Zealand.

[6] A. Zimmermann, M. Krotzsch, J. Euzenat, P. Hitzler, *Formalizing Ontology Alignment and its Operations with Category Theory*, in *Proc. 4th International conference on Formal ontology in information systems (FOIS)*, Baltimore pp. 277-288, 2006.

[7] International Organization for Standardization (ISO) *ISO 15926. Industrial automation systems and integration. Integration of life-cycle data for process plants including oil and gas production facilities*