Hindawi Publishing Corporation
International Journal of Engineering Mathematics
Volume 2013, Article ID 758729, 11 pages
http://dx.doi.org/10.1155/2013/758729

Research Article

Yoneda Philosophy in Engineering

Lambrini Seremeti[1] and Ioannis Kougias[2]

[1] *Faculty of Science and Technology, Hellenic Open University, 26222 Patras, Greece*
[2] *Department of Telecommunication Systems and Networks, Technological Educational Institute of Messolonghi, 30200 Messolonghi, Greece*

Correspondence should be addressed to Ioannis Kougias; kougias@teimes.gr

Received 25 March 2013; Revised 18 July 2013; Accepted 9 August 2013

Academic Editor: Shouming Zhong

Mathematical models, such as sets of equations, are used in engineering to represent and analyze the behaviour of physical systems. The conventional notations in formulating engineering models do not clearly provide all the details required in order to fully understand the equations, and, thus, artifacts such as ontologies, which are the building blocks of knowledge representation models, are used to fulfil this gap. Since ontologies are the outcome of an intersubjective agreement among a group of individuals about the same fragment of the objective world, their development and use are questions in debate with regard to their competencies and limitations to univocally conceptualize a domain of interest. This is related to the following question: "What is the criterion for delimiting the specification of the main identifiable entities in order to consistently build the conceptual framework of the domain in question?" This query motivates us to view the Yoneda philosophy as a fundamental concern of understanding the conceptualization phase of each ontology engineering methodology. In this way, we exploit the link between the notion of formal concepts of formal concept analysis and a concluding remark resulting from the Yoneda embedding lemma of category theory in order to establish a formal process.

1. Introduction

In the computer science context, the terms ontology and concept are defined in many various ways, and sometimes their use is a confused mixture of both terms. On the one hand, an ontology is defined as a common conceptualization of a general idea, or a domain of interest [1, 2], while, on the other hand, a concept is defined as a general idea, which is created by removing the uncommon characteristics from several particular ideas [3]. Thus, it is obvious that ontologies use concepts to convey their semantics and concepts are definitional structures that are explicitly encoded within ontologies.

In practice, an ontology consists of classes that are concepts of a particular domain, relations between classes, a hierarchy of classes, properties of classes that describe their attributes, instances of classes with specific values, and axioms that specify additional constraints over classes. So, the most crucial task of building an ontology is to consistently express its classes, that is, the concepts of a particular domain. An indispensable condition for formally expressing a concept that should carry a meaning, besides its name that does not carry any meaning, is to determine all its stakeholders, that is, other terms interrelated with the concept in question, properties attached to it, and specific values, directly or indirectly, affecting its noema, since, in natural language, a concept can be described by the objects belonging to it or by its determining features [4].

In order to make the ontological status of concepts apparent, independently of the language used to express them and of the techniques used for embedding them within ontologies, we have exploited the link between the notion of formal concepts of formal concept analysis (FCA) and a concluding remark resulting from the Yoneda embedding lemma of category theory. This enables the formalization of concepts by considering their semantics, since the notion of a concept in computer science is very subtle and does not rely on mathematical constructors. This step intends to eliminate any misunderstanding that occurs, due to ontological semantic heterogeneity, arising from synonymy (i.e., ontological concepts for a domain described with different terminologies) and polysemy (i.e., different meanings assigned to the same word in different contexts).

The basics and the role of formal concept analysis in defining ontology concepts are discussed in Sections 2 and 4, respectively, while the basics and a discussion of related work on category theory are presented in Sections 3 and 5, respectively. In Section 6, the main point of this work is illustrated, by means of a simple example. In Section 7, other possible meeting points of FCA and category theory, aiming at direct research into the critical path of formalizing tasks concerning ontology engineering and facilitating progress of the field, are discussed. The conclusion is drawn in the last section.

2. Formal Concept Analysis Background: Formal Context, Formal Concept, and Concept Lattice

This section introduces basic terms of formal concept analysis, among which are the fundamental notions of a formal context, formal concept, and concept lattice, as well as their basic properties.

Formal concept analysis (FCA), introduced by Wille [5], is a method for analyzing data which describe relationships between a set of objects and a set of attributes. From the input data, FCA produces a concept lattice, which is a collection of formal concepts in the data. Formal concepts are clusters of data drawn together by having common attributes [6] representing natural human-like concepts and are hierarchically ordered by a subconcept-superconcept relation. FCA also produces attribute implications which describe particular dependencies valid in the data. It integrates the discovery and reasoning with concepts in data, the discovery and reasoning with dependencies in data and the visualization of data, and concepts and dependencies with folding and unfolding capabilities, which are referred to as the reconstruction of a diagram of concepts and their dependencies, in order to be more simplified, but as informative as the original one. For more details see [7, 8].

In FCA, a cross-table with logical attributes is represented by a triplet $\langle X, Y, I\rangle$ (a so-called *context*), where the elements of the set X are called *objects* and correspond to the rows of the table, the elements of the set Y are called *attributes* and correspond to the columns of the table, and I is a binary (incidence) relation between X and Y. $\langle x, y\rangle \in I$, or xIy indicates that *object x has attribute y* or that y holds for x, while $\langle x, y\rangle \notin I$ indicates that object x does not have attribute y. An example of such a table is shown in Table 1.

Table 1: A table defining an incidence relation.

	y_1	y_2	y_3
x_1	×	×	×
x_2	×	×	
x_3		×	×

A partially ordered collection of particular clusters of objects and attributes is obtained out of a given context cross-table, and it is called a *concept lattice*. The clusters of a concept lattice, called *formal concepts*, are pairs $\langle A, B\rangle$, where $A \subseteq X$ is a set of objects and $B \subseteq Y$ is a set of attributes, such that A is the set of all objects which have all attributes from B and B is the set of all attributes which are common to all objects from A. For instance, $\langle\{x_1, x_2\}, \{y_1, y_2\}\rangle$ is an example of a formal concept for the table of Table 1. Under suitable permutations of the rows and the columns of the cross-table, a formal concept $\langle A, B\rangle$ is represented by a maximal rectangle full of crosses.

Moreover, formulas describing dependencies which are true in the table are obtained, which are called *attribute implications*. An attribute implication is an expression $A \Rightarrow B$, with A and B being sets of attributes. $A \Rightarrow B$ is true in the context $\langle X, Y, I\rangle$ if each object having all attributes from A has all attributes from B as well. For instance, $\{y_3\} \Rightarrow \{y_2\}$ is true for the context of Table 1, while $\{y_1, y_2\} \Rightarrow \{y_3\}$ is not, x_2 being a counterexample.

In order to produce the formal concepts of a formal context $\langle X, Y, I\rangle$, the so-called *concept-forming operators* are used, defined between the powersets of X and Y:

$$\begin{aligned}\uparrow : A \in 2^X &\longrightarrow B \in 2^Y : A^{\uparrow} \\ &= \{y \in Y / \text{for each } x \in A : \langle x, y\rangle \in I\},\end{aligned} \tag{1}$$

where $A^{\uparrow}$ is just the set of all attributes shared by all objects of A and

$$\begin{aligned}\downarrow : B \in 2^Y &\longrightarrow A \in 2^X : B^{\downarrow} \\ &= \{x \in X / \text{for each } y \in B : \langle x, y\rangle \in I\},\end{aligned} \tag{2}$$

where $B^{\downarrow}$ is just the set of all objects sharing all attributes from B.

A formal concept now in $\langle X, Y, I\rangle$ is just a pair $\langle A, B\rangle$ of $A \subseteq X$ and $B \subseteq Y$, such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$. A is called the *extent* and B the *intent* of $\langle A, B\rangle$, respectively. The formal concepts are naturally ordered by using a subconcept-superconcept relation, based on inclusion relation on objects and attributes; that is, for formal concepts $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$ of the context $\langle X, Y, I\rangle$, $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ if and only if $A_1 \subseteq A_2$ (if and only if $B_2 \subseteq B_1$).

By using basic mathematical structures behind FCA, it can be proven that the pair $\langle\uparrow, \downarrow\rangle$ of concept-forming operators induced by $\langle X, Y, I\rangle$ forms a *Galois connection* between X and Y, and the formal concepts $\langle A, B\rangle$ are just the fixed points of this Galois connection.

More precisely, given a pair of sets X and Y, a pair $\langle f, g\rangle$ of operators, $f : 2^X \to 2^Y$ and $g : 2^X \to 2^Y$, is a Galois connection if for $A, A_1, A_2 \subseteq X$, $B, B_1, B_2 \subseteq Y$ it satisfies the following:

$$\begin{gathered}A_1 \subseteq A_2 \Longrightarrow f(A_2) \subseteq f(A_1), \\ B_1 \subseteq B_2 \Longrightarrow g(B_2) \subseteq g(B_1), \\ A \subseteq g(f(A)), \\ B \subseteq f(g(B)).\end{gathered} \tag{3}$$

For a Galois connection $\langle f, g\rangle$ between the sets X and Y, the *fixpoint* set of $\langle f, g\rangle$ is defined as the set

$$\operatorname{fix}(\langle f, g\rangle) = \left\{ \langle A, B\rangle \in 2^X \times \frac{2^Y}{f(A)} = B,\ g(B) = A \right\}. \quad (4)$$

Also, for a Galois connection $\langle f, g\rangle$ between the sets X and Y, we have

$$f(A) = f(g(f(A))), \qquad g(B) = g(f(g(B))). \quad (5)$$

Now, given a set X, a mapping $C : 2^X \rightarrow 2^X$, such that, for each $A, A_1, A_2 \subseteq X$,

$$\begin{gathered} A_1 \subseteq C(A), \\ A_1 \subseteq A_2 \Longrightarrow C(A_1) \subseteq C(A_2), \\ C(A) \subseteq C(C(A)), \end{gathered} \quad (6)$$

is called a *closure operator* on X and its fixpoint set is given by

$$\operatorname{fix}(C) = \left\{ A \subseteq \frac{X}{C(A)} = A \right\}. \quad (7)$$

In addition, for a Galois connection $\langle f, g\rangle$ between the sets X and Y, the *composition* $C_X : f \circ g$ is the closure operator on X and $C_Y : g \circ f$ is the closure operator on Y.

Finally, since the pair $\langle \uparrow, \downarrow\rangle$ of concept-forming operators induced by $\langle X, Y, I\rangle$ forms a Galois connection between X and Y, the *compositions* $\uparrow\downarrow$ and $\downarrow\uparrow$ of the concept-forming operators are closure operators on X and Y, respectively.

Another fundamental notion in FCA is that of the set of all formal concepts of $\langle X, Y, I\rangle$, denoted by $\mathscr{B}(X, Y, I)$ the so-called *concept lattice*, where

$$\mathscr{B}(X, Y, I) = \left\{ \langle A, B\rangle \in 2^X \times \frac{2^Y}{A^{\uparrow}} = B,\ B^{\downarrow} = A \right\}. \quad (8)$$

Then, the fixpoints of the composition operator $\uparrow\downarrow$ give the set of all extents of the formal concepts of $\langle X, Y, I\rangle$ that is,

$$\begin{aligned} \operatorname{Ext}(X, Y, I) &= \left\{ A \in \frac{2^X}{\langle A, B\rangle} \in \mathbf{B}(X, Y, I) \text{ for some } B \right\} \\ &= \operatorname{fix}(\uparrow\downarrow), \end{aligned} \quad (9)$$

and the fixpoints of the composition operator $\downarrow\uparrow$ give the set of all intents of the formal concepts of $\langle X, Y, I\rangle$; that is,

$$\begin{aligned} \operatorname{Int}(X, Y, I) &= \left\{ B \in \frac{2^Y}{\langle A, B\rangle} \in \mathbf{B}(X, Y, I) \text{ for some } A \right\} \\ &= \operatorname{fix}(\downarrow\uparrow). \end{aligned} \quad (10)$$

The extents and intents are just the images under the concept-forming operators:

$$\begin{aligned} \operatorname{Ext}(X, Y, I) &= \left\{ \frac{B^{\downarrow}}{B} \subseteq Y \right\}, \\ \operatorname{Int}(X, Y, I) &= \left\{ \frac{A^{\uparrow}}{A} \subseteq X \right\}. \end{aligned} \quad (11)$$

Hence,

$$\begin{aligned} \mathscr{B}(X, Y, I) &= \left\{ \frac{\langle A, A^{\uparrow}\rangle}{A} \in \operatorname{Ext}(X, Y, I) \right\}, \\ &= \left\{ \frac{\langle B^{\downarrow}, B\rangle}{B} \in \operatorname{Int}(X, Y, I) \right\}. \end{aligned} \quad (12)$$

Now, a partial order $\leq$ can be defined on $\mathscr{B}(X, Y, I)$, since for $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathscr{B}(X, Y, I)$, $A_1 \subseteq A_2$ if and only if $B_2 \subseteq B_1$. Thus, one puts $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$, meaning that $\langle A_1, B_1\rangle$ is more specific than $\langle A_2, B_2\rangle$ or, equivalently, that $\langle A_2, B_2\rangle$ is more general than $\langle A_1, B_1\rangle$. Then $\langle \mathscr{B}(X, Y, I), \leq\rangle$ is a *complete lattice*, which means that, for any collection of formal concepts $K \subseteq \mathscr{B}(X, Y, I)$, $\mathscr{B}(X, Y, I)$ contains both their direct generalization (the supremum of K, noted as $\vee K$) and their direct specialization (the infimum of K, noted as $\wedge K$). The main theorem of concept lattices states that the supremum and infimum of a collection of formal concepts $\langle A_j, B_j\rangle$, $j \in J$, $J = 1, 2, 3, \ldots$, are given by

$$\begin{aligned} \bigwedge_{j \in J} \langle A_j, B_j\rangle &= \left\langle \bigcap_{j \in J} A_j, \left(\bigcup_{j \in J} B_j \right)^{\downarrow\uparrow} \right\rangle, \\ \bigvee_{j \in J} \langle A_j, B_j\rangle &= \left\langle \left(\bigcup_{j \in J} A_j \right)^{\uparrow\downarrow}, \bigcap_{j \in J} B_j \right\rangle. \end{aligned} \quad (13)$$

3. Category Theory Background: Categories, Functors, Natural Transformations, and Yoneda Embedding Lemma

In this section, we briefly refer to the basic notions of category theory, such as categories, functors, and natural transformations that guide to the comprehension of the Yoneda embedding lemma.

Category theory [9–12], which can, in general, be considered as a mathematical theory of structures and of systems of structures, occupies a central position in contemporary mathematics and theoretical computer science, since (a) it focuses on relationships (categorical morphisms, functors, and natural transformations) and not on entities (categorical objects and categories), (b) it allows the coexistence of heterogeneous entities, since it provides the ability to define several categories, according to the kind of entities to be described, which can be related by the definition of special morphisms (categorical functors), (c) it offers a set of categorical constructors for creating new categories, by using predefined ones, (d) it provides a means for the combination of categorical objects (colimits can be used to compose them and limits to decompose them) and for the combination of categorical functors (natural transformations), and (e) it provides a multilevel study of its categorical notions, by defining three interrelated levels (the level of categories, of functors, and of natural transformations).

A category is a mathematical structure consisting of entities of some kind, together with the relationships between

$$\begin{tikzcd}
A \arrow[r, "f"'] \arrow[rr, bend left, dotted, "g \circ f"] \arrow[rrr, bend left=50, dotted, "h \circ (g \circ f) = (h \circ g) \circ f"] & B \arrow[r, "g"'] \arrow[rr, bend right, dotted, "h \circ g"'] & C \arrow[r, "h"'] & D
\end{tikzcd}$$

Figure 1: The composition operation is associative.

$$\begin{tikzcd}
A \arrow[loop left, dotted] \arrow[r] \arrow[r, phantom, shift right=3, "1_B \circ f = f = f \circ 1_A"] & B \arrow[loop right, dotted]
\end{tikzcd}$$

Figure 2: The unit of the composition.

$$\begin{tikzcd}
& A_1 \\
& A_2 \\
0 \arrow[ruu, "!"] \arrow[ru, "!"] \arrow[r, "!"] & A_3 \\
& \vdots
\end{tikzcd}$$

Figure 3: An initial object.

$$\begin{tikzcd}
A_1 \arrow[rdd, "!"] & \\
A_2 \arrow[rd, "!"] & \\
A_3 \arrow[r, "!"] & 1 \\
\vdots &
\end{tikzcd}$$

Figure 4: A terminal object.

them that express the structure. Formally, a category consists of objects (entities) and morphisms (relationships), together with a law of composition for the morphisms, and a unique identity morphism for each object. More rigorously speaking, a *category* $\mathscr{C}$ is anything comprising the following data.

(i) A class $\mathscr{C}_0$ of objects is denoted by $A, B, C, \ldots$.

(ii) A class $\mathscr{C}_1$ of morphisms between objects is denoted by $f, g, h, \ldots$. In order to denote a morphism f from object A to object B, we write $f : A \rightarrow B$, where A is the domain of $f(\mathrm{dom}\, f = A)$ and B is the codomain of $f(\mathrm{cod} f = B)$. In general, there may exist more than one morphism from A to B. The collection of all morphisms from A to B is denoted by $\mathrm{Hom}_{\mathscr{C}}(A, B) = \mathscr{C}(A, B)$.

(iii) A composition operator is denoted by $\circ$, assigning to each pair of composable morphisms $f : A \rightarrow B$ and $g : B \rightarrow C$ a composite arrow $h : A \rightarrow C, h = g \circ f$. The composition of morphisms is associative, that is, for any morphisms $f : A \rightarrow B$, $g : B \rightarrow C$, and $h : C \rightarrow D : h \circ (g \circ f) = (h \circ g) \circ f$, as depicted in Figure 1.

(iv) For each object A in $\mathscr{C}_0$, a unique morphism $1_A : A \rightarrow A$, which acts as the unit of the composition; that is, for any morphism $f : A \rightarrow B$, $1_B \circ f = f = f \circ 1_A$, as depicted in Figure 2.

A category $\mathscr{C}$ is called *small* when the objects $\mathscr{C}_0$ and the morphisms $\mathscr{C}_1$ are sets and not proper classes. A less strict concept is that of a *locally small category*, when for all objects A and B in $\mathscr{C}_0$, the collection of morphisms between them, $\mathscr{C}(A, B)$, is a set, called a Hom-set (in this case, we may still have “too many” pairs of objects).

An important category is *Set*, the category having sets as objects, total functions as morphisms, and identity functions as identity morphisms. The composition of total functions, $f : A \rightarrow B$ and $g : B \rightarrow C$, is the total function $h : A \rightarrow C$, mapping $a \in A$ to $(g \circ f)(a) = g(f(a)) \in C$.

For each category $\mathscr{C}$, the dual category $\mathscr{C}^{\mathrm{op}}$ is defined as the category having the same objects as those of $\mathscr{C}$ and as morphisms the opposites of the morphisms in $\mathscr{C}$; that is, if $f : A \rightarrow B$ is a morphism in $\mathscr{C}_1$, then $f^{\mathrm{op}} : B \rightarrow A$ is a morphism in $\mathscr{C}_1^{\mathrm{op}}$.

Another essential feature of category theory is that of an isomorphism, which is mainly used in order to compare objects. An *isomorphism* is a morphism $f : A \rightarrow B$ that has an inverse, that is, a morphism $f^{-1} : B \rightarrow A$, such that $f \circ f^{-1} = 1_B$ and $f^{-1} \circ f = 1_A$. If $f : A \rightarrow B$ is an isomorphism, then its inverse $f^{-1} : B \rightarrow A$ is unique. By using this definition, two objects A and B are isomorphic ($A \cong B$) if there is a pair of inverse morphisms $f : A \rightarrow B$ and $f^{-1} : B \rightarrow A$ between them. Two isomorphic objects are said to be identical up to an isomorphism if every object satisfying a property p of an object A is isomorphic to A.

There are two special objects that are defined in a category: the initial and the terminal objects. More formally, an object of a category $\mathscr{C}$ is called an *initial object*, denoted by 0 in Figure 3, if for every object A in $\mathscr{C}$, there exists exactly a unique morphism $0 \rightarrow A$.

The symbol ! is used to characterize unique morphisms. On the other hand, an object of a category $\mathscr{C}$ is called a *terminal object*, denoted by 1 in Figure 4, if for every object A in $\mathscr{C}$, there exists exactly a unique morphism $A \rightarrow 1$.

In this mathematical theory, any part of a category can be pictured as a directed diagram, which is a set of objects and morphisms. A significant operation in category theory is the pushout, which is defined over a diagram. A *pushout* of a pair of morphisms, $f : A \rightarrow B$ and $g : A \rightarrow C$, is an object Q, together with a pair of morphisms $q_f : C \rightarrow Q$ and $q_g : B \rightarrow Q$, such that $q_f \circ g = q_g \circ f$, and with the universal property stating that, whenever $q'_f : C \rightarrow Q'$ and $q'_g : B \rightarrow Q'$ are such that $q'_f \circ g = q'_g \circ f$, there exists a unique morphism $s : Q \rightarrow Q'$, such that $s \circ q_f = q'_f$ and $s \circ q_g = q'_g$. In Figure 5, the pushout is indicated by drawing a square corner $\ulcorner$ next to it.

$$\begin{tikzcd}
A \arrow[r, "g"'] \arrow[d, "f"] & C \arrow[d, "q_f"'] \arrow[ddr, "q'_f = s \circ q_f"] & \\
B \arrow[r, "q_g"] \arrow[drr, "q'_g = s \circ q_g"'] & Q \arrow[ul, phantom, "\ulcorner", very near start] \arrow[dr, dotted, "!_s"] & \\
& & Q'
\end{tikzcd}$$

Figure 5: The pushout tuple (Q, q_f, q_g) of f and g.

$$\begin{tikzcd}
P' \arrow[dr, dotted, "!_m"] \arrow[drr, "p'_g = p_f \circ m"] \arrow[ddr, "p'_f = p_g \circ m"'] & & \\
& P \arrow[r, "p_f"'] \arrow[d, "p_g"] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "g"] \\
& A \arrow[r, "f"] & C
\end{tikzcd}$$

Figure 6: The pullback tuple (P, p_f, p_g) of f and g.

The pushout is the categorical dual of *pullback*, which is depicted in Figure 6, which is obtained by reversing arrows in the graph of pushout.

These categorical constructions are characterized as universal constructions, describing a class of objects and the accompanying morphisms that share a common property.

Another important notion of category theory is functors, which are structure-preserving mappings between categories. Given two categories $\mathscr{C}$ and $\mathscr{D}$, a *functor* $F : \mathscr{C} \rightarrow \mathscr{D}$ maps

(i) each object $A \in \mathscr{C}$ to an object $FA \in \mathscr{D}$,

(ii) each morphism $f : A \rightarrow B \in \mathscr{C}$ to a morphism $Ff : FA \rightarrow FB \in \mathscr{D}$,

(iii) identity morphisms of $\mathscr{C}$ to identity morphisms of $\mathscr{D}$; that is, $F(1_A) = 1_{FA}$,

(iv) the composition of morphisms f and g in $\mathscr{C}$ to the composition of their mappings in $\mathscr{D}$; that is, with f and g composable in $\mathscr{C}$, $F(g \circ f) = F(g) \circ F(f)$.

For a given category $\mathscr{C}$, the identity functor $1_{\mathscr{C}}$ maps each object and each arrow in $\mathscr{C}$ to itself. Moreover, functors can be composed. Given two functors $F : \mathscr{C} \rightarrow \mathscr{D}$ and $G : \mathscr{D} \rightarrow \mathscr{E}$, their composition $G \circ F : \mathscr{C} \rightarrow \mathscr{E}$ is a functor mapping an object $A \in \mathscr{C}_0$ to the object $G(FA) \in \mathscr{E}_0$ and mapping a morphism $f \in \mathscr{C}_1$ to the morphism $G(Ff) \in \mathscr{E}_1$.

Let us now consider two locally small categories $\mathscr{C}$ and $\mathscr{D}$. For every pair of objects $A, B \in \mathscr{C}_0$, a functor $F : \mathscr{C} \rightarrow \mathscr{D}$ defines a function $F_{A,B} : \mathscr{C}(A, B) \rightarrow \mathscr{D}(FA, FB)$ in *Set*. Then, the functor $F : \mathscr{C} \rightarrow \mathscr{D}$ is

(i) faithful if the function $F_{A,B}$ is injective,

(ii) full if the function $F_{A,B}$ is surjective,

(iii) fully faithful if the function $F_{A,B}$ is bijective.

A functor $F : \mathscr{C} \rightarrow \mathscr{D}$ is an embedding if it is full, faithful, and injective on objects. A functor from $\mathscr{C}$ to $\mathscr{D}$ is called a covariant functor, while a functor $F : \mathscr{C}^{\mathrm{op}} \rightarrow \mathscr{D}$ is called a contravariant functor from $\mathscr{C}$ to $\mathscr{D}$ if it reverses arrow; that is, it is a covariant functor $F : \mathscr{C}^{\mathrm{op}} \rightarrow \mathscr{D}$ from the opposite category $\mathscr{C}^{\mathrm{op}}$ to $\mathscr{D}$.

In order to comprehend the Yoneda embedding, the more elaborate categorical notions of representable functors are needed. The representable functors map a locally small category $\mathscr{C}$ to the category *Set*. They refer either to incoming morphisms to a fixed object in $\mathscr{C}$ (generalized elements) or to outgoing morphisms from a fixed object in $\mathscr{C}$ (generalized properties).

Given a category $\mathscr{C}$, the *covariant representable functor* $\mathscr{C}(A, -) : \mathscr{C} \rightarrow \mathit{Set}$, which is depicted in Figure 7, can be defined, for each fixed object $A \in \mathscr{C}_0$, and it takes each object $B \in \mathscr{C}_0$ to the set $\mathscr{C}(A, B) \in \mathit{Set}_0$ of morphisms from A to B ($\mathscr{C}(A, B)$ is an object of *Set*) and each morphism $f : B \rightarrow C \in \mathscr{C}_1$ to the function

$$\begin{aligned}\mathscr{C}(A, -)(f) &= \mathscr{C}(A, f) \\ &= f \circ - : \mathscr{C}(A, B) \longrightarrow \mathscr{C}(A, C) \in \mathit{Set}_1.\end{aligned} \tag{14}$$

This last function takes a morphism $g : A \rightarrow B \in \mathscr{C}_1$ (an element of the set $\mathscr{C}(A, B)$ in Set_0) to the morphism $\mathscr{C}(A, f)(g) = f \circ g \in \mathscr{C}_1$ (an element of the set $\mathscr{C}(A, C)$ in Set_0). The covariant representable functor refers to the outgoing morphisms from a fixed object A in $\mathscr{C}$ and to their compositions with other morphisms in $\mathscr{C}$.

Given a category $\mathscr{C}$, the *contravariant representable functor* $\mathscr{C}(-, B) : \mathscr{C}^{\mathrm{op}} \rightarrow \mathit{Set}$, which is depicted in Figure 8, is defined, for each fixed object $B \in \mathscr{C}_0$, taking each object $A \in \mathscr{C}_0$ to the set $\mathscr{C}(A, B) \in \mathit{Set}_0$ of morphisms from A to B ($\mathscr{C}(A, B)$ is an object of *Set*) and each morphism $f : A \rightarrow C \in \mathscr{C}_1$ to the function

$$\begin{aligned}\mathscr{C}(-, B)(f) &= \mathscr{C}(f, B) \\ &= - \circ f : \mathscr{C}(C, B) \longrightarrow \mathscr{C}(A, B) \in \mathit{Set}_1,\end{aligned} \tag{15}$$

which takes a morphism $g : C \rightarrow B \in \mathscr{C}_1$ (an element of the set $\mathscr{C}(C, B)$ in Set_0) to the morphism $\mathscr{C}(f, B)(g) = g \circ f \in \mathscr{C}_1$ (an element of the set $\mathscr{C}(A, B)$ in Set_0). The contravariant representable functor refers to the incoming morphisms to a fixed object B in $\mathscr{C}$ and their compositions with other morphisms in $\mathscr{C}$.

Moreover, given a category $\mathscr{C}$, the *Hom-set bifunctor* $\mathscr{C}(-, -) : \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \rightarrow \mathit{Set}$, which is depicted in Figure 9, is defined, for each pair of fixed objects $A \in \mathscr{C}_0$ and $B \in \mathscr{C}_0$, and it takes each such pair of fixed objects $A \in \mathscr{C}_0$ and $B \in \mathscr{C}_0$ to the set $\mathscr{C}(A, B) \in \mathit{Set}_0$ of morphisms from A to

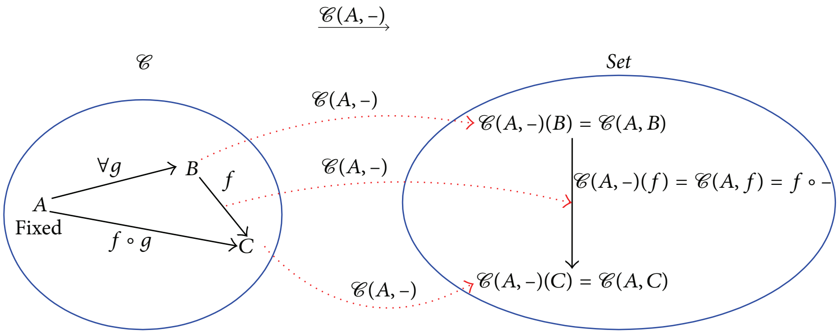

$\mathscr{C}(A,-)$
$\mathscr{C}$
Set
$\mathscr{C}(A,-)$
$\mathscr{C}(A,-)(B) = \mathscr{C}(A,B)$
$\forall g$
B
f
$\mathscr{C}(A,-)$
A
Fixed
$\mathscr{C}(A,-)(f) = \mathscr{C}(A,f) = f\circ -$
$f\circ g$
C
$\mathscr{C}(A,-)$
$\mathscr{C}(A,-)(C) = \mathscr{C}(A,C)$

FIGURE 7: The covariant representable functor $\mathscr{C}(A,-)$.

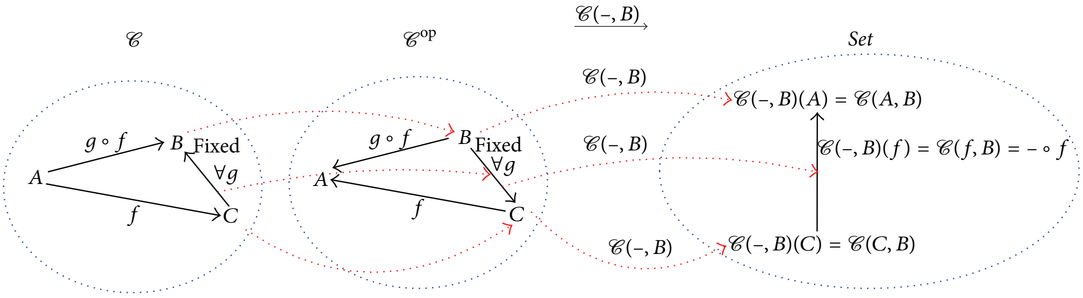


FIGURE 8: The contravariant representable functor $\mathscr{C}(-,B)$.

B ($\mathscr{C}(A,B)$ is an object of *Set*) and each pair of morphisms $f : A' \to A \in \mathscr{C}_1$ and $g : B' \to B \in \mathscr{C}_1$ to the function

$$\begin{aligned}\mathscr{C}(-,-)(f,g) &\\ &= \mathscr{C}(f,g)\\ &= g\circ - \circ f : \mathscr{C}(A,B') \longrightarrow \mathscr{C}(A',B) \in Set_1\end{aligned}\tag{16}$$

This last function takes a morphism $h : A \to B \in \mathscr{C}_1$ (an element of the set $\mathscr{C}(A,B)$ in Set_0) to the morphism $\mathscr{C}(f,g)(h) = g\circ h\circ f \in \mathscr{C}_1$ (an element of the set $\mathscr{C}(A',B')$ in Set_0). The contravariant representable functor refers to the incoming morphisms to a fixed object A in $\mathscr{C}$ and to the outgoing morphisms from a fixed point B in $\mathscr{C}$.

Another important notion of category theory is that of natural transformations, which relate two functors. Given two functors $F : \mathscr{C} \to \mathscr{D}$ and $G : \mathscr{C} \to \mathscr{D}$ from $\mathscr{C}$ to $\mathscr{D}$, a *natural transformation* $\eta : F \Rightarrow G$, which is depicted in Figure 10, is a map $h : \mathscr{C}_0 \to \mathscr{D}_1$, assigning to each object $A \in \mathscr{C}_0$ a morphism $\eta_A : FA \to GA \in \mathscr{D}_1$, called the component of η at A, such that, for every morphism $f : A \to B \in \mathscr{C}_1$, it holds that $\eta_B \circ Ff = Gf \circ \eta_A$. The components η_A are morphisms in $\mathscr{D}_1$ and can thus be composed.

Given two categories $\mathscr{C}$ and $\mathscr{D}$, a functor category denoted by $\mathscr{D}^{\mathscr{C}}$ is defined, having as objects the functors from $\mathscr{C}$ to $\mathscr{D}$ and having as morphisms the natural transformations between them. In this functor category, $\mathrm{Hom}_{\mathscr{D}^{\mathscr{C}}}(F,G) = \mathrm{Nat}(F,G)$ is the set of natural transformations from functor F to functor G (F and G are objects of $\mathscr{D}^{\mathscr{C}}$). The contravariant representable functor for $\mathscr{D}^{\mathscr{C}}$ is then denoted by $\mathrm{Nat}(-,G)$ and sends a functor $F \in \mathscr{D}^{\mathscr{C}}$ to the set $\mathrm{Nat}(F,G) \in Set$ of natural transformations between F and G and a natural transformation $\eta : F \Rightarrow H$ to the function

$$\begin{aligned}\mathrm{Nat}(-,G)(\eta) &= \mathrm{Nat}(\eta,G)\\ &= -\circ\eta : \mathrm{Nat}(H,G) \longrightarrow \mathrm{Nat}(F,G).\end{aligned}\tag{17}$$

The function $\mathrm{Nat}(\eta,G)$ sends a natural transformation $\theta : H \Rightarrow G$ to the natural transformation $\theta\circ\eta : F \Rightarrow G$ (which is a morphism in $\mathscr{D}^{\mathscr{C}}$ and an element of the set $\mathrm{Nat}(F,G) \in Set$).

An isomorphism can be defined in a functor category $\mathscr{D}^{\mathscr{C}}$. Given two functors $F, G : \mathscr{C} \to \mathscr{D}$ (objects of $\mathscr{D}^{\mathscr{C}}$), if a natural transformation $\eta : F \to G$ (morphism in $\mathscr{D}^{\mathscr{C}}$) has an inverse natural transformation $\eta^{-1} : G \to F$ such that for every object A in C it is $(\eta\circ\eta^{-1})_A = (1_F)_A = 1_{FA}$ and $(\eta^{-1}\circ\eta)_A = (1_G)_A = 1_{GA}$, then η is called a natural isomorphism, and F and G are said to be naturally isomorphic; that is, $F \cong G$.

Given an object A in a category $\mathscr{C}$, there is a functor $y : \mathscr{C} \to Set^{\mathscr{C}^{\mathrm{op}}}$ from $\mathscr{C}$ to the category $Set^{\mathscr{C}^{\mathrm{op}}}$ of functors from $\mathscr{C}^{\mathrm{op}}$ to *Set* (called the category of presheaves) that sends A to the functor $\mathscr{C}(-,A) \in Set^{\mathscr{C}^{\mathrm{op}}}$. This functor y is the *Yoneda embedding*. The Yoneda embedding also sends a morphism $f : A \to B \in \mathscr{C}_1$ to the natural transformation $y_f : \mathscr{C}(-,A) \to \mathscr{C}(-,B)$ between functors $\mathscr{C}(-,A)$ and $\mathscr{C}(-,B)$, having a component $(y_f)_X : \mathscr{C}(X,f) = f\circ -$ at $X \in C_0$, which sends a morphism $g : X \to A \in \mathscr{C}(X,A)$ to the morphism $f\circ g : X \to B \in \mathscr{C}(X,B)$.

The Yoneda lemma is an abstract result on functors of the type morphisms into a fixed object, which allows the

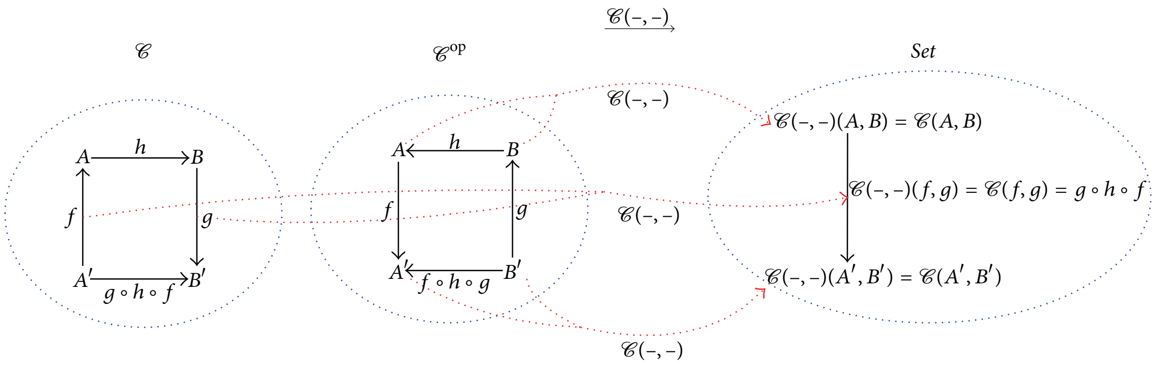


FIGURE 9: The Hom-set bifunctor $\mathscr{C}(-,-)$.

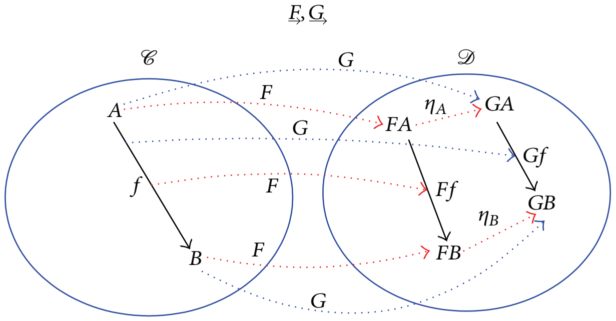


FIGURE 10: The definition of the natural transformation.

embedding of any category into a category of functors defined on that category. It states that every functor $F:\mathscr{C}^{\mathrm{op}}\rightarrow Set\in Set^{\mathscr{C}^{\mathrm{op}}}$ is naturally isomorphic to the functor $\mathrm{Nat}(y-,F)$; that is, for every $A\in\mathscr{C}_0$, $\mathrm{Nat}(yA,F)\cong FA$ and this isomorphism is natural both in A and F. One instance of Yoneda's lemma is the Yoneda embedding, which informally states that in order to determine an object of a category, it suffices to determine all its generated elements or all its generated properties.

4. The Interplay between FCA and Ontological Concepts

An ontology is the formal expression of relations between relevant notions in order to consistently describe a domain of interest, and in this perspective, it must clearly conceptualize these notions. At the conceptualization level, it is necessary to formally describe the relevant concepts, which are to be included in the ontology.

FCA, on the other hand, is a mathematical theory, which is based on the philosophical statement that a concept is constituted by two parts: its extension consisting of all objects belonging to the concept and its intension comprising all attributes shared by those objects [13]. In FCA, a formal concept is produced from a formal context such as $\langle X,Y,I\rangle$ of Section 2 by applying the *concept-forming operators* to its intension or to its extension; that is, a formal concept is a pair $\langle A,B\rangle$, where $A\subseteq X$ is a set of objects and $B\subseteq Y$ is a set of attributes, such that A is the set of all objects which have all attributes from B and B is the set of all attributes which are common to all objects from A.

Based on the assumption that an isolated concept has no meaning, since meaning is a product of the relationships among concepts in a specific context, an extension of FCA has been designed, the so-called relational concept analysis (RCA) [14], in order to deal with multirelational data [15]. RCA provides a generalization of FCA in the case of many-valued contexts. A many-valued context is defined as a context (G,M,W,I) consisting of three sets, namely, G, whose elements are called formal objects, M with many-valued attributes, and W, whose elements are values, and a ternary relation I between them, with the conditions $(g,m,w)\in I$ and $(g,m,v)\in I\Rightarrow w=v$, with (g,m,w) being read as "the attribute m has the value w for the object g." Thus, relational context schemas are useful for attributes that have not just only one but several values for an object.

Moreover, a triadic approach to FCA has been designed, the so-called triadic concept analysis (TCA) [16], as another FCA extension, which is based on a formalization of the triadic relation connecting objects, attributes, and conditions, where a concept is described by its extension, intention, and modus, respectively [17]. In this framework, instead of the dyadic relation between objects and attributes used in FCA, the triadic approach is based on a triadic relation saying that the object g has the attribute m under the condition b. Similar to the dyadic form of FCA, a triadic concept is produced from a triadic context, which is defined as a quadruple $\langle X,Y,Z,I\rangle$, where X, Y, and Z are sets of objects, attributes, and conditions, respectively, and I is a ternary relation between them, by applying derivation operators to its intension, extension, or modus. Thus, triadic concepts can be considered, for example, as those formal concepts that have the same extent but differ according to their intent and modus.

The interplay between FCA and ontologies has been studied to a considerable extent [18]. It turns out that there are basically two ways concerning how FCA and ontological concepts can be combined. The most obvious way is to identify the ontology concepts with the formal concepts of FCA. However, in many applications, the ontology concepts correspond to the FCA attributes [19, 20]. Since the ontological status of concepts has an internal structure, within the framework of FCA, this is sometimes generated by tertiary relations within multivalued contexts [21] or by using distributional formal contexts [22].

The crucial point in specifically describing a concept within an ontology is to find out all the attributes that are shared by this concept. Thus, from an FCA-based point of view, for formally describing a concept, its intentional description is required. In the case where a concept is described by a set of attributes under specific conditions, the notion of triadic contexts is used for its representation [23]. Moreover, each concept within an ontology denotes a specific role emerging from adding specific values to its intentional structure, that is, its attributes. In this perspective, ontology concepts are considered as the relational structures that the RCA process provides [24, 25].

5. The Interplay between Category Theory and Ontological Concepts

Since ontology can be regarded as a categorization of that which exists, issues involving the ontological status of concepts and the notion of a category are important. According to Aristotle, a class or collection of things all of one type, where typicality is determined by a concept, is a category. Later on, a category, as a purely mathematical notion, was defined as having an internal structure given by relationships among members. Thus, the exact match between the terms category and concept has emerged because a concept can be defined as having internal structure, due to their explanatory capabilities [26].

Category theory has been established as an appropriate mathematical basis for the formalization and study of ontologies [27, 28], since it focuses on relationships (categorical morphisms) between entities (categorical objects) in such a way that an entity can be defined according to its interaction with other entities [29]. There have been various category theoretic approaches to ontology engineering, which involves tasks, such as development, mapping, and merging of ontologies. In particular, in the case of ontology mapping and merging, the category *Ont* of ontologies is defined and the categorical constructions of pullback and pushout are used, respectively [30–32].

Formally speaking, ontology is a theory over logic. But many ontologies are described in different languages, each one with its own logic. Thus, in order to integrate ontologies, many approaches using the theory of institutions [33] have emerged, such as in [34, 35] for mapping between different logics to deal with heterogeneous formalization. Moreover, the constructions of category theory have been combined with information flow theory [36], which describes how information about one thing can convey information about something else, for integrating distributed ontologies [37, 38]. Finally, in [39] a rigorous foundation for integrating heterogeneous information is provided, by unifying information flow theory, institutions, formal concept analysis, and category theory.

By using a category theoretic language, ontologies can be represented as categories with finite limits and finite coproducts, where classes (concepts) are represented by nodes and functional relationships by arrows of a directed graph [40], and if an ontology has a representation by a category, then one may apply homological and homotopical algebra constructions to study the ontology [41]. Another view of the category theoretic perspective of ontologies and their underlying classes (concepts) is given in [42], where an ontology is regarded as a mathematical category of theories and a concept is represented by the ideas of theories and models that are borrowed from FCA. Moreover, in [43] an ontology can be described by formal concepts and thus it can be viewed as a category.

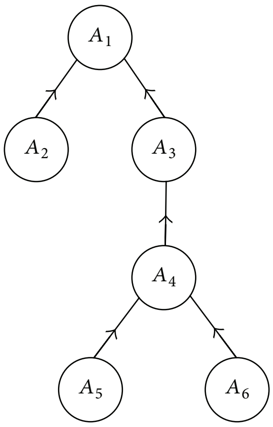


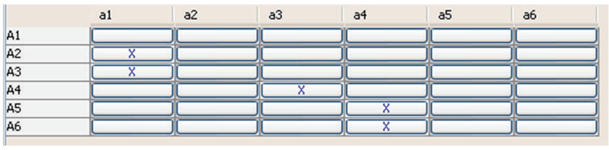
Figure 11: A simple example of an ontology.

6. From the Yoneda Embedding to Formal Concepts

Our main point is the fact that, from an ontological point of view, the formal determination of a concept is guided by the interconnection existing between a concluding remark of the Yoneda embedding lemma of category theory and the intents of the formal concepts of FCA. In order to illustrate this point, an example is used in this section, relying on the assumption that attributes can also be considered as concepts, according to the German standard DIN 2330, stating that attributes "are units of thought which are gained by abstraction, and hence they are also concepts" [19, 44].

Following this statement, we can represent binary relations between entities within an ontology, by using a suitable formal context. For this purpose, we consider the entities of the ontology, both as objects and as attributes of the context. A binary relation present in the context signifies that the respective entities are related. As an example, the ontology of Figure 11, where the concepts A_i, $i = 1, 2, \ldots, 6$, are hierarchically structured, is represented by the formal context of Figure 12, by using Lattice Miner [45], which is a formal concept analysis software tool for the construction of concept lattices.

In Figure 12, the concept lattice and the formal concepts associated with the formal context of Figure 11 are depicted. It is a representation of a formal context called cross-table. The context has six attributes, a_i, $i = 1, 2, \ldots, 6$. An "x" in a



	a1	a2	a3	a4	a5	a6
A1						
A2	X					
A3	X					
A4			X			
A5				X		
A6				X		

Figure 12: The formal context for the ontology of Figure 11.

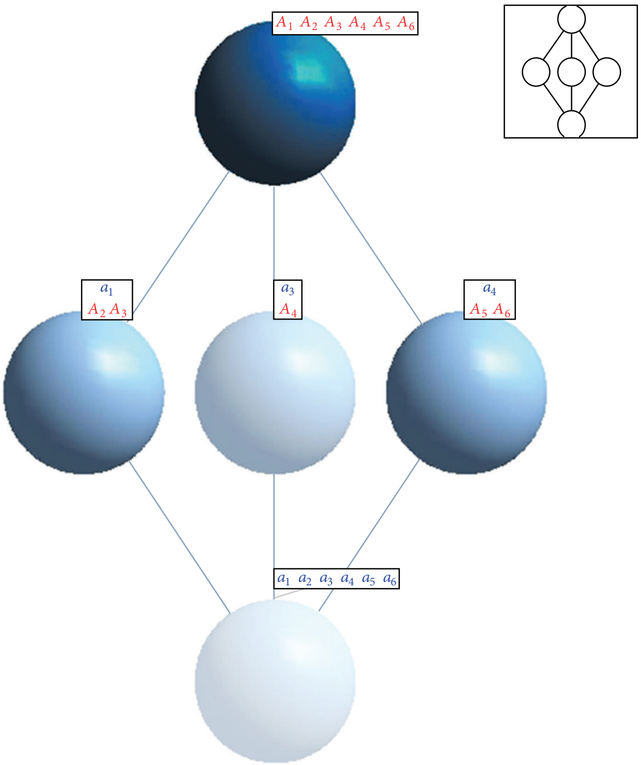

$A_1\ A_2\ A_3\ A_4\ A_5\ A_6$
a_1
$A_2\ A_3$
a_3
A_4
a_4
$A_5\ A_6$
$a_1\ a_2\ a_3\ a_4\ a_5\ a_6$

Figure 13: The formal concepts and the concept lattice for the ontology of Figure 11.

particular location or cell in the cross-table indicates that an object has an attribute, so, for example, the object "A_2" has the attribute "a_1." Conversely a blank location indicates that the object does not have that attribute.

The concept lattice of Figure 13 contains five concepts, the most general being $\langle\{A_1,\ldots,A_6\}\rangle$. The bottom concept is $\langle\{\},\{a_1,\ldots,a_6\}\rangle$ because it is the concept of objects that have all attributes. The remaining concepts are marked with their intent and their extent.

The diagram shows that there is an ordering over the concepts, which is a specialization ordering, denoted by $\leq$ [46].

It is interesting to notice here that the formal concepts represent the incoming relations to an ontology entity; that is, the intents of each formal concept correspond to the incoming arrows to each respective ontological concept. For example, the formal concept $\langle\{a_1\},\{A_2,A_3\}\rangle$ represents the incoming relations (arrows) to ontological entity (concept) A_1. Now, if we consider a categorical perspective of the ontology, where the ontology entities are the objects of a category and the relations between the entities are the morphisms (arrows), then the intent of a formal concept represents the incoming arrows to an entity (object). But, from a category theoretic point of view, this remark is equivalent to the remark concluded from the Yoneda embedding lemma, which states that an object is completely determined by the incoming arrows to this object.

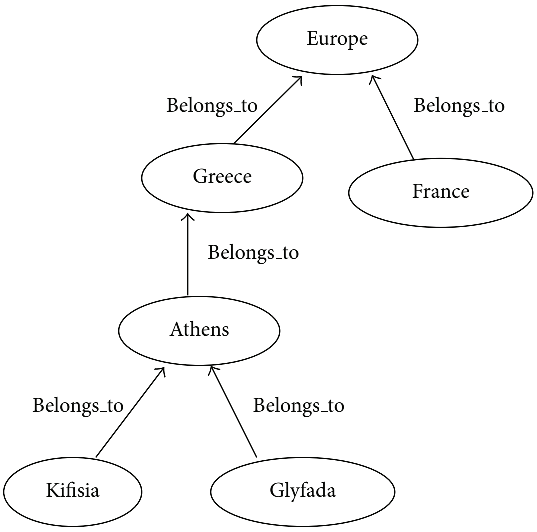


Figure 14: A segment of an ontology conceptualizing European places.

This statement yields from the Yoneda lemma, which is interpreted as follows: an object X of a category C is determined by the network of relationships that the object X has with all the other objects in C [47]. Thus, a concept can be clearly specified only by identifying the set of all its attributes and a set of attributes that are common for a concept determines completely a concept. This reciprocal condition can guide the building of the ontological status of concepts completely and consistently, since it eliminates polysemy and synonymy.

In a more illustrative example, such as that of the ontology of Figure 14, following the same process and taking into account its category theoretical representation, where concepts are objects of a category and their attributes/relations are morphisms, we observe the alignment between the intent of the formal concept $\langle\{belongs\_To\},\{Greece, France\}\rangle$ and the incoming arrows to the ontological entity *"Europe,"* that is, the Yoneda philosophy in unequivocally determining concepts in ontology engineering.

7. Equivalent Points of FCA and Category Theory

In this section, we refer to the possible meeting points of FCA and category theory aiming at direct research into the critical path of formalizing tasks concerning ontology engineering and facilitating progress of the field.

Due to the fact that we focus on the abstract notion of concepts, as well as their formal representation, and not on strict mathematical objects, it is incongruous to talk about equality between different points of mathematical theories, rather than equivalence, or isomorphism between them. Thus, the main question to be answered is not if two concepts are equivalent, but if they are isomorphic, that is, if they are interchangeable in the sense that they have the same relationships with all concepts within a context. In that sense, it will be interesting to investigate the correspondence between the basic notions and constructions of FCA and category theory, such as between (a) the supremum of any collection of formal concepts of a complete lattice and the initial object of a category, (b) the infimum of any collection of formal concepts of a complete lattice and the terminal object of a category, and (c) the application of the concept-forming operators to the intention or to the extension of a formal context and the fact that from an object of a category, one can define a specific functor that encodes the essence of these objects, as well as between other notions.

8. Conclusion

A formal view on ontologies or on ontological concepts can contribute a lot to the direction of achieving effective interoperability, since the coexistence of multiple heterogeneous ontologies within or among ontology-based applications is an objective reality. Thus, by carefully studying FCA and category theory, as the mathematical theories that provide frameworks for structuring, analyzing, and visualizing data to make them more understandable and for combining and maintaining heterogeneous collections of data to enable them to interoperate, respectively, we investigate some equivalent notions of both theories. This can lead, despite the abstract but quite accurate techniques of these theories, to create a unified framework for the ontological representation as well as their management.

Acknowledgments

The authors are indebted to the editor and the anonymous reviewers for their valuable suggestions and comments that greatly contributed to improving the paper.

References

[1] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing,” in *Formal Analysis in Conceptual Analysis and Knowledge Representation*, Kluwer Academic/Plenum, New York, NY, USA, 1993.

[2] W. N. Borst, *Construction of Engineering Ontologies, Centre for Telematica and Information Technology*, University of Tweenty, Ensschede, The Netherlands, 1997.

[3] J. Goguen, “What is a concept?” in *Proceedings of the 13th Conference on Conceptual Structures*, F. Dan and M. Mungier, Eds., vol. 3596 of *Lecture Notes in Artificial Intelligence*, pp. 52–77, Springer, 2005.

[4] L. Lambrechts, *Formal Concept Analysis*, Vrije Universtiteit, Brussels, Belgium, 2012.

[5] R. Wille, “Restructuring lattice theory: an approach based on hierarchies of concepts,” in *Ordered Sets*, I. Rival, Ed., pp. 445–470, Reidel, Dordrecht, The Netherlands, 1982.

[6] R. Bělohlávek and V. Sklenář, “Formal concept analysis over attributes with levels of granularity,” in *Proceedings of the International Conference on Communicational Intelligence for Modelling, Control and Automation & Inter. Conf. on Intelligent Agents, Web Technologies and Internet Commerce*, Vienna, Austria, November 2005.

[7] R. Bělohlávek, *Introduction to Formal Concept Analysis*, UP Olomouc, 2008.

[8] B. Ganter and R. Will, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin, Germany, 1999.

[9] S. Awodey, *Category Theory*, Oxford University Press, New York, NY, USA, 2010.

[10] B. Pierce, *Basic Category Theory for Computer Scientists*, The MIT Press, Cambridge, Mass, USA, 1991.

[11] W. Lawvere and S. Schanuel, *Conceptual Mathematics, A First Introduction to Categories*, Cambridge University Press, New York, NY, USA, 1997.

[12] J. Fiadeiro, *Categories for Software Engineering*, Springer, New York, NY, USA, 2004.

[13] M. Obitko, V. Snasel, and J. Smid, “Ontology design with formal concept analysis,” in *Proceedings of the International Conference on Concept Lattices and Their Applications (CLA ’04)*, V. Snasel and R. Belohlavek, Eds., pp. 111–119, 2004.

[14] U. E. Priss, *Relational concept analysis: semantic structures in dictionaries and lexical databases [Dissertation]*, TH-Darmstadt, 1996.

[15] M. Huchard, M. R. Hacene, C. Roume, and P. Valtchev, “Relational concept discovery in structured datasets,” *Annals of Mathematics and Artificial Intelligence*, vol. 49, no. 1–4, pp. 39–76, 2007.

[16] F. Lehmann and R. Will, “A triadic approach to formal concept analysis,” in *Conceptual Structures: Applications, Implementation and Theory*, vol. 954 of *Lecture Notes in Computer Science*, pp. 32–43, Springer, Berlin, Germany, 1995.

[17] J. Konecny and P. Osicka, “General approach to triadic concept analysis,” in *Proceedings of the 7th International Conference on Concept Lattices and Their Applications (CLA ’10)*, M. kryszkiewicz and S. A. Obiedkov, Eds., pp. 116–126, Sevilla, Spain, 2010.

[18] P. Hitzler, “What’s happening in semantic web—and what FCA could have to do with it?” in *Formal Concept Analysis*, P. Valtchev and R. Jäschke, Eds., vol. 6628 of *Lecture Notes in Computer Science*, pp. 18–23, Springer, Berlin, Germany, 2011.

[19] P. Cimiano, A. Hotho, G. Stumme, and J. Tane, “Conceptual knowledge processing with formal concept analysis and ontologies,” in *Proceedings of the 2nd International Conference on Formal Concept Analysis (ICFCA ’04)*, pp. 189–207, February 2004.

[20] R. Bendaoud, A. Napoli, and Y. Toussaint, “Formal concept analysis: a unified framework for building and refining ontologies,” in *Knowledge Engineering: Practice and Patterns*, A. Gangemi and J. Euzenat, Eds., vol. 5268 of *Lecture Notes in Computer Science*, pp. 156–171, Springer, Berlin, Germany, 2008.

[21] J. Nanda, T. W. Simpson, S. R. T. Kumara, and S. B. Shooter, “A methodology for product family ontology development using formal concept analysis and web ontology language,” *Journal of Computing and Information Science in Engineering*, vol. 6, no. 2, pp. 103–113, 2006.

[22] D. Jurkevicius and O. Vasilecas, “Formal concept analysis for concept collecting and their analysis,” *Scientific Papers, University of Latvia, Computer Science and Information Technologies*, vol. 751, pp. 22–39, 2009.

[23] R. Jäschke, A. Hotho, C. Schmitz, B. Ganter, and G. Stumme, "Discovering shared conceptualizations in folksonomies," *Journal of Web Semantics*, vol. 6, no. 1, pp. 38–53, 2008.

[24] M. Rouane-Hacene, M. Huchard, A. Napoli, and P. Valtchev, "Using formal concept analysis for discovering knowledge patterns," in *Proceedings of the 7th International Conference on Concept Lattices and Their Applications (CLA '10)*, Sevilla, Spain, October 2010.

[25] X. Dolques, M. Huchard, C. Nebut, and H. Saada, "Formal and relational concept analysis approaches in software engineering: an overview and an application to learn model transformation patterns in examples," in *Proceedings of the 1st ICESE Virtual Workshop on Software Engineering and Artificial Intelligence*, Doha, Qatar, 2011.

[26] M. J. Healy, "Category Theory as a Mathematics for formalizing ontologies," in *Theory and Applications of Ontology: Computer Applications*, R. Poli, M. Healy, and A. Kameas, Eds., pp. 487–510, Springer, New York, NY, USA, 2010.

[27] M. Krötzsch, P. Hitzler, M. Ehrig, and Y. Sure, "Category theory in ontology research: concrete gain from an abstract approach," Tech. Rep., AIFB, University of Karlsruhe, 2005.

[28] R. Brown, J. F. Glazebrook, and I. C. Baianu, "A conceptual construction of complexity levels theory in spacetime categorical ontology: non-abelian algebraic topology, many-valued logics and dynamic systems," *Axiomathes*, vol. 17, no. 3-4, pp. 409–493, 2007.

[29] I. Cafezeiro and E. H. Haueusler, "Semantic interoperability via Category Theory," in *Proceedings of the 26th International Conference on Conceptual Modeling*, J. Grundy, S. Hartman, A. H. F. Laender, L. Maciaszek, and J. F. Roddick, Eds., vol. 83, pp. 197–202, 2007.

[30] A. Zimmermann, M. Krötzsch, J. Euzenat, and P. Hitzler, "Formalizing ontology alignment and its operations with category theory," in *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS '06)*, B. Bennett and C. Fellbaum, Eds., pp. 277–288, IOS Press, Baltimore, Md, USA, 2006.

[31] P. Hitzler, M. Krötzsch, M. Ehrig, and Y. Sure, "What is ontology merging? A category-theoretical perspective using pushouts," in *Proceedings of the 1st International Workshop on Contexts and Ontologies: Theory, Practice and Applications*, pp. 104–107, AAAI Press, July 2005.

[32] L. Hu and J. Wang, "Geo-ontology integration based on category theory," in *Proceedings of the International Conference on Computer Design and Applications (ICCDA '10)*, pp. V15–V18, Qinhuangdao, China, June 2010.

[33] J. A. Goguen and R. M. Burstall, "Institutions: abstract model theory for specification and programming," *Journal of the ACM*, vol. 39, no. 1, pp. 95–146, 1992.

[34] T. Mossakowski, C. Maeder, and K. Lüttich, "The heterogeneous tool set," in *Tools and Algorithms for the Construction and Analysis of Systems*, O. Grumberg and M. Huth, Eds., vol. 4424 of *Lecture Notes in Computer Science*, pp. 519–522, Springer, Heidelberg, Germany, 2007.

[35] O. Kutz, D. Lücke, and T. Mossakowski, "Heterogeneously structured ontologies: integration, connection and refinement," in *Proceedings of the Knowledge Representation Ontology Workshop (KR '08)*, vol. 90 of *Conferences in Research and Practice in Information Technology*, Sydney, Australia, 2008.

[36] J. Barwise and J. Seligman, *Information Flow: The Logic of Distributed Systems*, vol. 44 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, New York, NY, USA, 1997.

[37] R. Kent, "The IFF foundation for ontological knowledge organization," *Cataloging & Classification Quarterly*, vol. 37, no. 1-2, 2003.

[38] Y. Kalfoglou and M. Schorlemmer, "The information flow approach to ontology-based semantic alignment," in *Theory and Applications of Ontology: Computer Applications*, chapter 4, pp. 101–114, Springer, New York, NY, USA, 2010.

[39] J. Goguen, *Information Integration in Institutions*, L. Moss, Ed., 2004.

[40] M. Johnson and R. Rosebrugh, "Ontology engineering, universal algebra, and category theory," in *Theory and Applications of Ontology: Computer Applications*, R. Poli, M. Healy, and A. Kameas, Eds., pp. 565–576, Springer, New York, NY, USA, 2010.

[41] N. Glazunov, "Semantic web and category theory models," in *Proceedings of the International Seminar "iscrete Mathematics and its Applications"*, pp. 309–311, MGU, Moscow, Russia, 2007.

[42] M. J. Healy and T. P. Caudell, "Ontologies and worlds in category theory: implications for neural systems," *Axiomathes*, vol. 16, no. 1-2, pp. 165–214, 2006.

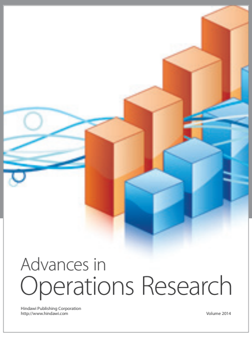
[43] P. Wendorff, "A formal approach to the assessment and improvement of terminological models used in information systems engineering," in *Proceedings of the 8th Eiropean Engineering Conference and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC '01 and FSE '01)*, V. Gruhn, Ed., pp. 83–87, Vienna, Australia, September 2001.

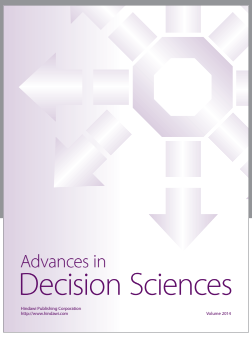
[44] Deutsches Institut für Normung, "Begriffe und Benennungen—Allgemeine Grundsätze," DIN 2330, 1993.

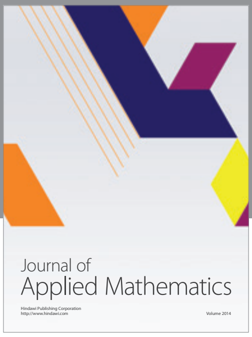
[45] 2012, http://lattice-miner.softpedia.com/.

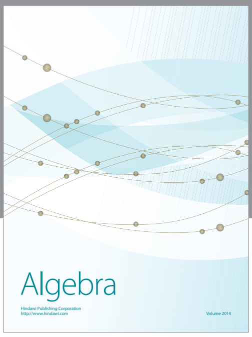
[46] T. Tilley, *Concept Analysis and Formal Methods*, 2000.

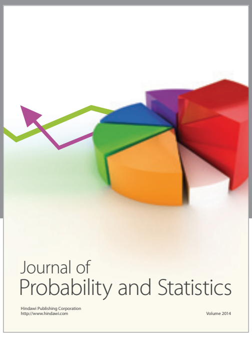
[47] B. Mazur, "When is one thing equal to other thing?" in *Proof and Other Dilemmas: Mathematics and Philosophy*, B. Gold and R. Simons, Eds., 2007.


Advances in Operations Research

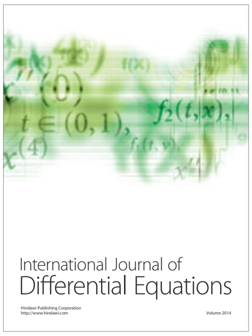

Advances in Decision Sciences


Journal of Applied Mathematics

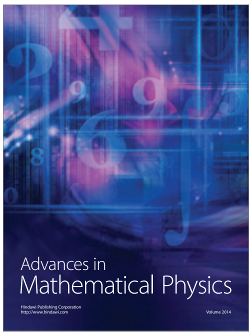

Algebra

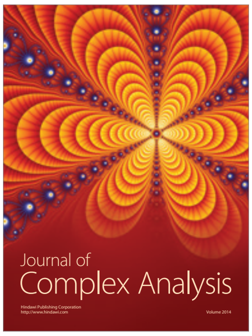

Journal of Probability and Statistics

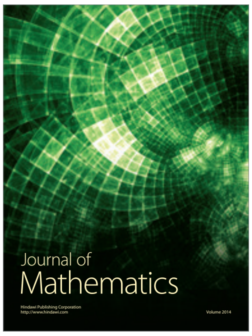
The Scientific World Journal

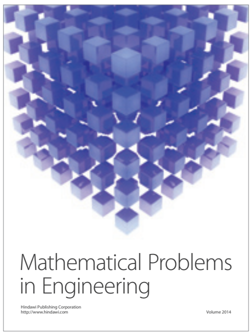

International Journal of Differential Equations

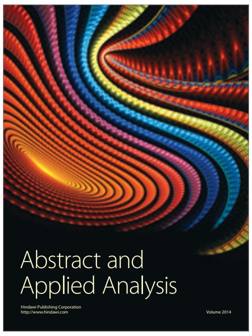
International Journal of Combinatorics

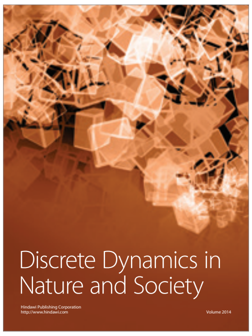

Advances in Mathematical Physics

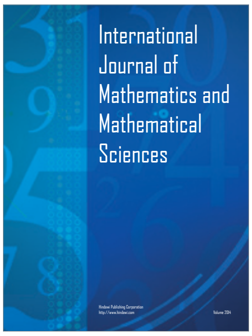

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

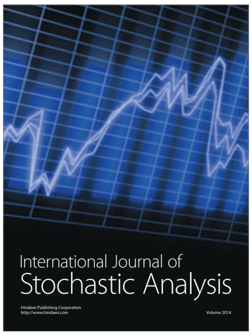

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization