UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CAUÃ ROCA ANTUNES

A proposal for ontology formalization based on category theory

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Science.

Advisor: Prof. Dr. Mara Abel

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Rui Vicente Oppermann
Vice-Reitora: Prof.ª Jane Fraga Tutikian
Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento
Diretora do Instituto de Informática: Prof.ª Carla Maria Dal Sasso Freitas
Coordenador do Curso de Ciência da Computação: Prof. Sergio Luis Cechin
Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

ABSTRACT

This work proposes a formalization of domain ontologies based in category theory as a framework for the study and representation of conceptual models. Category theory is a branch of mathematics that studies the structure in systems of composable relations. Domain ontologies are conceptual models that enable the reuse of domain knowledge and the execution of inferential processes over said knowledge. In order to achieve such goals, concepts must be modeled intensionally. The established set-theoretic foundations of current conceptual models are incompatible with the intended intensionality of ontological models. Category theory, on the other hand, does not default to extensionality in the same way set theory does, and offers, therefore, a better-suited mathematical foundation. Additionally, category theory's focus on relations matches the primary attention of construction and representation of ontologies, which is turned towards the relations between the domain concepts. The present work builds upon these motivations and formalizes ontologies as categories of concepts and conceptual relations. We subsequently analyze the categorical constructions present in ontologies, and the consequences of this formalization for categories of ontologies.

Keywords: Ontology. Category theory. Conceptual modeling.

Uma proposta de formalização de ontologias baseada em teoria das categorias

RESUMO

O presente trabalho propõe uma formalização de ontologias de domínio baseada em teoria das categorias como um arcabouço para o estudo e representação de modelos conceituais. A teoria das categorias é uma área da matemática que estuda a estrutura presente em sistemas de relações componíveis. Ontologias de domínio são modelos conceituais que permitem o reuso de conhecimento de domínio e a execução de processos de inferência sobre tal conhecimento. Para atingir tais objetivos, os conceitos devem ser modelados de forma intensional. As fundamentações dos modelos conceituais baseadas em teoria dos conjuntos atualmente aceitas são incompatíveis com a pretendida intensionalidade de modelos ontológicos. A teoria das categorias, por outro lado, não está comprometida com extensionalidade da mesma forma que a teoria dos conjuntos e, portanto, mostra-se uma fundamentação matemática mais adequada. Ainda, o fato de que a teoria das categorias tem seu foco principalmente em relações melhor se relaciona à atenção primária presente na construção e representação de ontologias, que é orientada às relações entre os conceitos do domínio. Este trabalho parte destas motivações e formaliza ontologias como categorias de conceitos e relações conceituais. Subsequentemente são analisadas as construções categoriais presentes em ontologias e as consequências desta formalização para categorias de ontologias.

Palavras-chave: Ontologia. Teoria das categorias. Modelagem conceitual.

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

ER	Entity-Relationship
OWL	Web Ontology Language
RDF	Resource Description Framework
UFO	Unified Foundational Ontology
UFRGS	Universidade Federal do Rio Grande do Sul

CONTENTS

1 INTRODUCTION

Ontologies have been used in computer science as both aid and subject for inference in the field of artificial intelligence, and as disambiguation mechanisms to enable semantic interoperability between information systems. As such, ontology engineering aims to elicit the knowledge on a given domain and model said knowledge in a precise manner, in order to share it among several different persons and systems and to apply it in diverse circumstances that are sometimes unpredictable. To achieve this goal, the domain concepts and their relations must be defined formally and independently from any single world state.

There are two opposed approaches to define concepts, that is, extensional and intensional. Extensional definitions list everything that is comprised by the concept, while intensional definitions give its underlying meaning (CARNAP, 1988). Since ontologies are supposed to represent knowledge that is independent from any specific state of affairs and in a way that resembles human thinking, we consider that intensional definitions are a better option for modeling.

However, the established foundations of ontology engineering have their bases on set theory. Set theory has, among its foundations, the principle of extensionality, which states that for any two sets x and y, $\forall z\ (z \in x \leftrightarrow z \in y) \rightarrow x = y$, that is, two sets are equal (i.e., the same) if they contain exactly the same elements. Consequently, if an enterprise system models its employees as a set, for example, and any new employee is hired, the model needs to be changed to account for the substitution of the underlying set.

So far, the solution to this issue has been to define the model over a set of possible worlds, which indexes every possible configuration of the domain. Thus, a function maps concepts to their respective instances in each possible domain configuration. This approach is still based on sets, and therefore still relies on extensional definitions. Additionally, it does not reflect how we model ontologies in reality. We don't build ontologies by assigning to each term a set of its instances indexed by possible worlds. Instead, we build them upon the properties and relations of concepts, which express their meanings.

This work presents an alternative formalization of ontologies that do not rely upon sets, but is instead defined in terms of categories. Categories are mathematical constructions where the structure formed by composable relations between objects may be studied. This focus on relations mirrors the process of ontology engineering more closely than any possible set-theoretic approach.

Additionally, category theory provides a multi-layered framework upon which the relations between disparate ontologies may also be scrutinized. This property is extremely useful for distributed contexts such as the Semantic Web, where applications may need to access, operate and make inferences over the knowledge content spread over multiple ontologies that requires an integrated access or view. Hence, category theory also provides a sound formal framework for the study of ontology mappings, alignments, and merges.

In chapter 2, we introduce the concepts related to ontologies and ontology engineering. Chapter 3 outlines category theory and its constructions, along with some examples and diagrams. We review similar previous works on the formalization of ontologies as categories and related works on the study of operations defined in categories of ontologies in chapter 4. Chapter 5 describes the proposed framework, first using category-theoric foundations to define several notions commonly found in ontology engineering. Subsequently, we formalize ontologies as categories and we peer into their constructions, before delineating a category of ontologies and their relations. Finally, we discuss the associated categorical concepts and how the proposed framework preserves the meanings attributed to them in earlier works. Chapter 6 presents a brief overview along with some concluding remarks.

2 ONTOLOGY

The term “ontology” has a threefold meaning. When written with a capital O, it refers to the philosophical discipline that studies the nature of reality and dates back to Aristotle. With the initial in lowercase but still in a philosophical sense, an ontology is a particular system of categories that accounts for a certain vision of the world. The third meaning, which prevails in Computer Science, refers to a computational artifact that models the structure of a certain part of reality (GUARINO; OBERLE; STAAB, 2009).

Ontology as a computational artifact was defined by Studer, Benjamins and Fensel (1998) as “a formal, explicit specification of a shared conceptualization”. “Formal” here means that it should be machine-readable. “Explicit” reflects that the concepts and constraints on their use must be defined explicitly. “Shared” means that the knowledge contained in an ontology is consensual among a group of people and not private to some individual. Finally, “conceptualization” refers to an abstract model of a portion of reality. This last concept is the one most closely related to the philosophical definition of ontologies, and is defined by Guarino (1998) as a triple $C = (D, W, R)$ consisting of a universe of discourse D, a set W of possible worlds and a set R of conceptual relations on the domain space $\langle D, W \rangle$.

This work does not focus on the representational aspects of ontologies, and hence refers to “ontology” in its philosophical sense, which is language independent. Thus, we differentiate the ontology from its (possibly multiple) implementations in diverse ontology modeling languages.

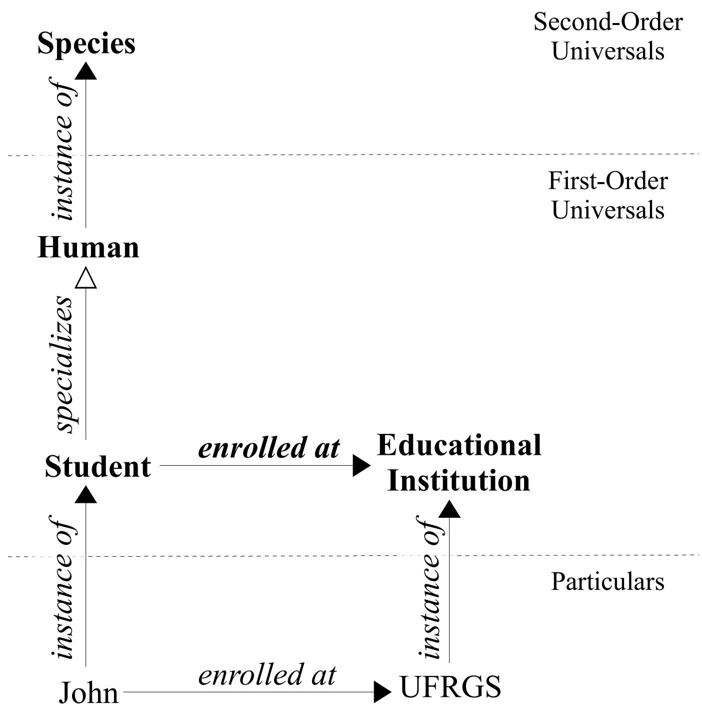
2.1 UNIVERSALS AND PARTICULARS

A core distinction to be made among the kinds of entities present in ontological modeling is the one between universals and particulars. Particulars, also called individuals, are entities that exist in reality (or in some counterfactual reality), while universals (which are sometimes called types, sorts, classes or concepts) are generalizations of other entities that encompass common properties and may be manifested in multiple different entities.

Each universal provides a principle of application that guides the judgment on whether it is realized in a specific entity. Those entities that manifest a certain universal are said to instantiate it. Therefore, we define particulars as the entities that cannot have instances, and universals as the ones that can. For example, a person called *John* is a particular that instantiates the universal *Human*. Usually, we consider instances as particulars. Models that follow this assumption are called two-level models, since they account for exactly two levels of abstraction, one of particulars and one of universals. However, in multi-level modeling universals may have universals as instances (ALMEIDA; FONSECA; CARVALHO, 2017). Following the previous example, we may say that the universal *Human* is an instance of the universal *Species*.

It is important to notice that this distinction between universals and particulars is not limited to monadic entities (that is, entities that exist by themselves), but also applies to relational entities. Thus, there are universal relations, which may have instances, and particular relations, which may not. For example, the universal relation *enrolled at* that holds between the universals *Student* and *Educational Institution*, is instantiated by the particular relation that holds between a particular student *John* and a particular educational institution *UFRGS*. Figure 2.1 shows the concepts and relations that we have discussed in this section, with universals in bold text and dashed lines dividing the different abstraction levels.

Figure 2.1 – Sample ontology



Source: the author.

Due to the intended intensionality of ontological models, ontologies usually do not contain monadic particulars, but particular relations that hold between universals, such as specialization relations (discussed in subsection 5.2), are almost always present.

2.2 ONTOLOGICAL METAPROPERTIES

Ontological engineering in computer science has as foundational basis the work in the philosophical discipline of Ontology. This philosophical support guides the modeling of ontologies towards coherent descriptions of the world. One example of such intersections between the ontological studies in philosophy and in computer science is the OntoClean framework (GUARINO; WELTY, 2009). This framework directs the modeling of ontologies through the analysis of four metaproperties, i.e., properties of properties. The authors of Ontoclean define properties as the meanings (or intensions) of expressions that correspond to unary predicates in first order logic, such as *being a human*. The metaproperties are those of rigidity, identity, unity and dependence.

A property is essential for an entity if it must be true for that entity in any possible state of the world. Said property is additionally rigid if it is essential to all its possible instances. That is, instances of rigid properties cannot cease to be instances of those properties. The property of *being a human*, for example, is essential for all humans, while the property of *being a student* is not essential for students. Further, rigid properties cannot specialize non-rigid properties, since such specialization would lead to contradictions.

The metaproperty of identity refers to how we recognize a specific instance of some property. Properties may provide or carry identity criteria that guide the identification of its instances. These criteria are those that allow us to recognize the same person in childhood, as an adult, and in old age, even through the changes performed by time. Every particular must instantiate at least one universal that provides an identity criterion for it.

Properties may also carry unity criteria that allow us to differentiate what is part of some entity from what is not. If a property specializes some other property that carries a unity criterion, it must also carry the same criterion. For example, the unity criteria of *Amount of Water* tells us that its instances are homeomerous, that is, their parts are also amounts of water. The parts of an *Ocean*, however, are defined geographically. Thus, *Ocean* and *Amount of Water* cannot specialize each other, since they carry incompatible unity criteria.

Finally, if for all instances of some property there must exist an instance of a second property that is neither part or constituent of the first, then we say it is externally dependent. A *Student*, for example, can only exist as such as long as there exists an *Educational Institution* at which he or she is enrolled. Properties that are not externally dependent cannot specialize properties that are.

2.3 INFERENCE

Inference is the manipulation of knowledge to produce new information. The knowledge contained in ontologies provides a basis for the automatic execution of such manipulations. One type of inferential process is that of logical entailment. A piece of information *entails* a certain proposition if it includes implicitly the truth of said proposition. For example, given that *John* is a *Student* and that every *Student* is enrolled at an *Educational Institution*, an ontology-based system may infer that *John* is enrolled at some *Educational Institution*, even without the knowledge of which institution fills that role.

Other inference processes include the finding of the *least common subsumer* of a set of concepts, that is, their most specific common generalization, and of the *greatest common subsumee*, i.e., the most general common specialization (BORGIDA, 1995). Given the meaning of concepts *Child* and *Man*, a reasoner may find that their least common subsumer is the concept *Person*, since it contains precisely all properties that are shared by the two concepts. Similarly, the greatest common subsumee of the same concepts is the concept *Boy*, given that it contains all properties from each concept.

We shall present category-theoretic formalizations of these processes throughout the remaining of the work. Particularly, we describe entailment along with several examples in subsection 5.3.1, the least common subsumer in subsection 5.5.2 and the greatest common subsumee in subsection 5.5.3. Other reasoning tasks, which are not subject of this work, include satisfiability checks, i.e., to test if a concept can be instantiated without a logic contradiction, subsumption checks, that is, given two concepts, test if one is a specialization of the other, and equivalence checks.

3 CATEGORY THEORY

Category theory is a branch of mathematics that studies the structure present in systems via abstractions of mappings and relations between mathematical objects. These abstractions take the form of morphisms, the basic primitives upon which we define the constructions in category theory.

By focusing on morphisms rather than on the objects they relate, category theory provides general constructions that are common to several mathematical domains, including set theory, topology, algebras and vector spaces.

Adámek et al. (1990) define a category as a quadruple $C = (O, hom, id, \circ)$, consisting of:

- a class O whose members are C-objects,
- for each pair (A,B) of C-objects, a set $hom(A,B)$, whose members are C-morphisms from A to B,
- for each C-object A, a morphism $id_A{:}A\rightarrow A$, called the C-identity on A, and
- a composition law $\circ$ associating each pair of C-morphisms $f{:}A\rightarrow B$ and $g{:}B\rightarrow C$ to a C-morphism $g \circ f{:}A\rightarrow C$, called the composite of f and g.

Such that composition is associative, that is, for any three morphisms $f{:}A\rightarrow B$, $g{:}B\rightarrow C$ and $h{:}C\rightarrow d$, $h \circ (f \circ g) = (h \circ f) \circ g$, C-identities are neutral with respect to composition, i.e., for any morphism $f{:}A\rightarrow B$, $id_B \circ f = f = f \circ id_A$, and the sets $hom(A,B)$ are pairwise disjoint.

Some examples of categories include the category *Set* of sets with functions as morphisms, the category *Rel*, which also has sets as objects but whose morphisms are mathematical relations, and the category *Prop* of logical propositions with formal proofs, i.e., derivation of propositions, as morphisms.

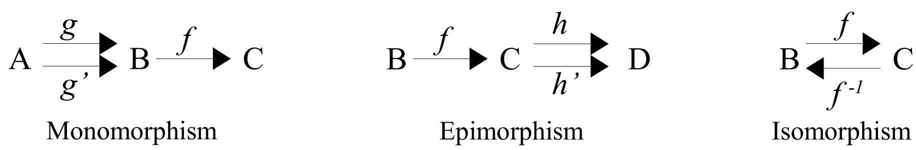
3.1 DUALITY

The concept of duality plays an important role in category theory by allowing both definitions and proofs to be simplified, since every statement about a category can be easily translated to its logical equivalent concerning the dual for that category. A dual for any category-theoretic construction is obtained by reversing the domain and codomain of its morphisms while mantaining the objects. Therefore, given a category $C = (O, hom, id, \circ)$, its dual is a category $C^{op} = (O, hom^{op}, id, \circ^{op})$, where $hom^{op}(A,B) = hom(B,A)$ and $f \circ^{op} g = g \circ f$. Some categories are self-dual, i.e., $C = C^{op}$. An example of a self-dual category is the category *Rel* as previously defined.

3.2 MONO-, EPI- AND ISOMORPHISMS

A monomorphism (or monic morphism) is a morphism $f{:}B \to C$ such that for any two morphisms $g{:}A \to B$ and $g'{:}A \to B$, $f \circ g = f \circ g'$ only if $g = g'$. The dual concept to monomorphism is that of epimorphism (or epic morphism), i.e., a morphism $f{:}B \to C$ such that for any two morphisms $h{:}C \to D$ and $h'{:}C \to D$, $h \circ f = h' \circ f$ only if $h = h'$. Figure 2.1 depicts the related diagrams, along with the diagram for isomorphisms.

Figure 3.1 – Monomorphisms, epimorphisms and isomorphisms



Source: the author.

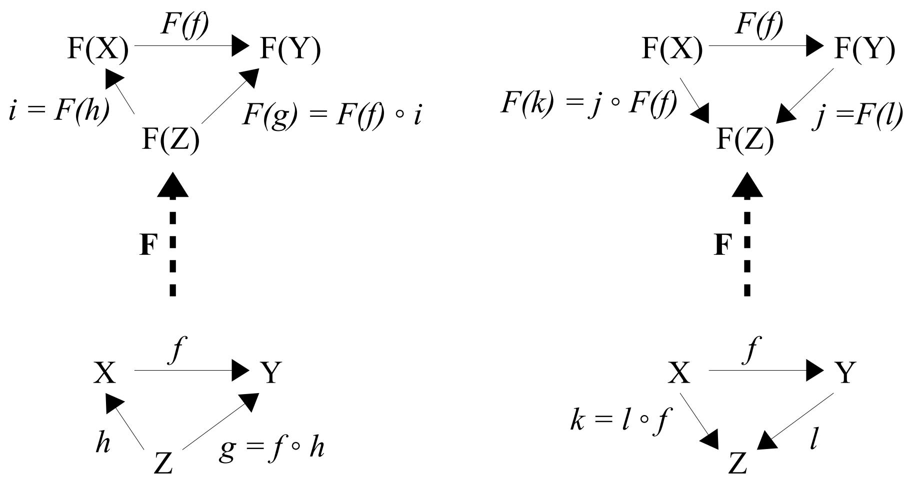
An isomorphism is a morphism $f{:}B \to C$ such that there exists a morphism $f^{-1}{:}C \to B$ with $f \circ f^{-1} = id_B$ and $f^{-1} \circ f = id_C$. Every isomorphism is both monic and epic, but not every morphism that is both monic and epic is an isomorphism. In the category *Set*, monomorphisms correspond to injective functions, epimorphisms to surjective functions and isomorphisms to bijective functions but this does not generalize to any category.

3.3 FUNCTORS AND NATURAL TRANSFORMATIONS

Equivalently to how different mathematical structures may be formalized as objects in a category and studied along with their respective morphisms, it is possible to define a category *Cat* with small categories[1] as objects. The morphisms in *Cat* are mappings between categories called functors. Functors preserve composition and indentities, i.e., any functor $F:A \to B$ maps each A-morphism $f:x \to y$ to a B-morphism $F(f):F(x) \to F(y)$ in such a way that for any two composable A-morphisms f and g, $F(f \circ g) = F(f) \circ F(g)$ and for each A-object x, $F(id_x)=id_{F(x)}$.

A morphism $f:X \to Y$ is said to be cartesian regarding a functor $F:A \to B$ if, for each A-morphism $g:Z \to Y$ and each B-morphism $i:F(Z) \to F(X)$ such that $F(f) \circ i = F(g)$, there exists a unique A-morphism $h:Z \to X$ with $f \circ h = g$ and $F(h) = i$. We show the related diagrams on the left side of figure 3.2. A fibration is a functor $F:A \to B$ for which every morphism is cartesian. Dually, $f:X \to Y$ is op-cartesian regarding $F:A \to B$ if for any $k:X \to Z$ and any $j:F(Y) \to F(Z)$ such that $j \circ F(f) = F(k)$, there exists a unique morphism $l:Y \to Z$ with $l \circ f = k$ and $F(l) = j$. Figure 3.2 depicts the diagram for an op-cartesian morphism on the right side. A functor is an op-fibration if every morphism is op-cartesian in regards to it.

Figure 3.2 – Cartesian and op-cartesian morphisms

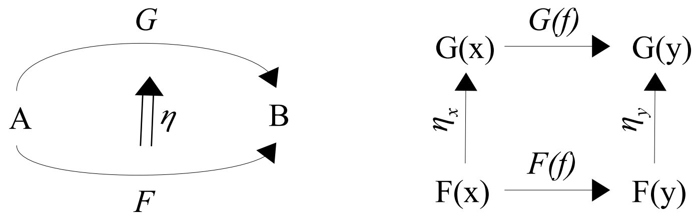


Source: the author.

Functors may, in turn, take the place of objects in a category. The category *Hom(A,B)* has as objects the functors from *A* to *B* and natural transformations as morphisms. A natural transformation η from a functor $F:A \to B$ to a functor $G:A \to B$ assigns for each A-object x a B-morphism $\eta_x:F(A) \to G(A)$ such that for every A-morphism $f:x \to y$ the square formed by $F(f)$, $G(f)$, η_x and η_y commutes, i.e., $G(f) \circ \eta_x = \eta_y \circ F(f)$.

[1]Small categories are those whose classes of objects are actually sets. This is not true for *Cat*, and therefore it is not an object of itself, avoiding Russel-like paradoxes.

Figure 3.3 – Natural transformation



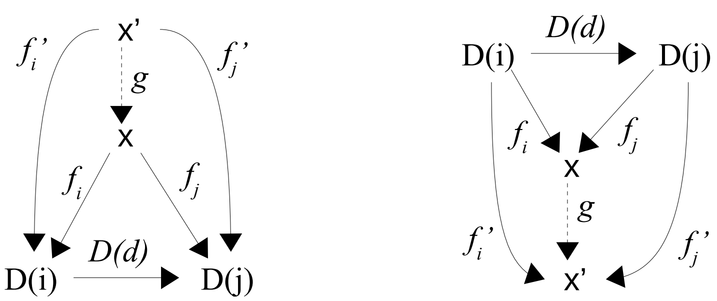
Source: the author.

3.4 LIMITS AND COLIMITS

A diagram in a category *A* is a functor $D:I\rightarrow A$. Intuitively, it is a selection of objects and morphisms in *A*. A source for *D* is a pair (x, f_i) consisting of an A-object *x* and a family of morphisms $f_i:x\rightarrow D(i)$ with domain *x* indexed by *I*. If for any I-morphism $d:i\rightarrow j$ the triangle formed by $D(d)$, f_i and f_j commutes, i.e., $D(d) \circ f_i = f_j$, the source (x, f_i) is called a cone. If (x, f_i) is a terminal cone, that is, for every other cone (x', f_i') there exists a unique morphism $g:x\rightarrow x'$ such that the resulting diagram commutes, (x, f_i) is a limit. Figure 3.4 depicts these objects on the left side, and the dual constructions on the right.

The dual to a source is a sink. A commutative sink is a cocone, which is dual to a cone. An initial sink is a colimit, that is, the dual to a limit. Several specific limits and colimits have meaningful interpretations in many categories, as we shall discuss over the next sections.

Figure 3.4 – Cones, limits, cocones and colimits

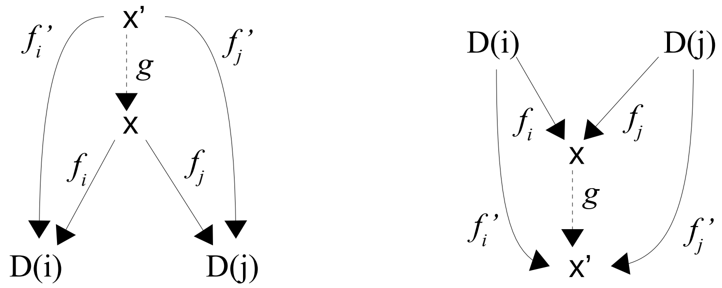


Source: the author.

3.5 PRODUCTS AND COPRODUCTS

The limit over a discrete diagram, that is, a diagram without non-identity morphisms, is called a product. Examples of product include cartesian products in *Set*, logical conjunctions in *Prop* and product categories in *Cat*. The colimit over the same diagram is called a coproduct. The coproducts in *Set* are disjoint set unions, while in *Prop* they are logical disjunctions and in *Cat* they are disjoint unions of categories. We describe products and coproducts in *Cat* in detail in section 5.6.2. As a consequence of being self-dual, both products and coproducts have the same meaning in *Rel* as disjoint set unions. For the entirety of this work, we will use binary (co)products as exemplars for all (co)products. Figure 3.5 shows a binary product and a corresponding cone on the left, and a coproduct and a cocone over the same diagram on the right.

Figure 3.5 – Products and coproducts

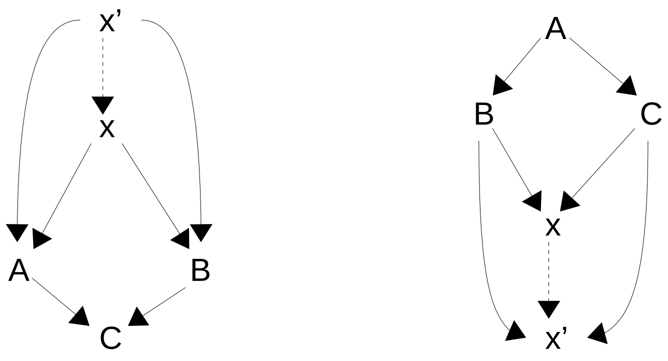


Source: the author.

3.6 PULLBACKS AND PUSHOUTS

A pullback is a limit for a diagram containing two morphisms $f{:}A \to C$ and $g{:}B \to C$ with the same codomain. As such, the precise meaning of each pullback depends on the selected morphisms. In *Set* the pullback is a subset of the Cartesian product of *A* and *B* consisting of pairs *(x,y)* such that *f(x)=g(y)*, i.e., it is the categorical equivalent to an equation. In *Cat*, if both morphisms are inclusions from subcategories, the pullback is the intersection of *A* and *B*. Figure 3.6 shows the diagram for a pullback on the left side.

Figure 3.6 – Pullbacks and pushouts



Source: the author.

The dual to a pullback is a pushout, i.e., a colimit over a diagram containing two morphisms *f:A→B* and *g:A→C* with the same domain. Again, the meaning depends on the selected morphisms. If *A* is the intersection of *B* and *C* in *Set*, the pushout is their union. Similarly, if both morphisms are inclusions in *Cat*, the pushout is the amalgamation of *B* and *C*. Figure 3.5 depicts the diagram for a pushout on the right side.

4 RELATED WORK

In one of the earliest works to use category theory to represent the semantic knowledge present in information systems, Colomb, Dampney and Johnson (2001) used the category-theoretic concepts of cartesian morphisms and fibrations to define an abstraction framework between enterprise models and the implementation models that instantiate them. In the context of the proposed framework, this restriction means that, given a fibration F from the implementation model to the enterprise model, for any two objects X and Z that are related to the same object Y in the implementation model, if there exists a morphism between their respective abstractions given by $F(X)$ and $F(Z)$, then there must exist a unique relation between X and Z such that the diagram commutes and whose abstraction is the morphism between $F(X)$ and $F(Z)$. The authors used this formalization to require that instances of the codomain of a relation in the enterprise model are codomain of an instance of such relation. However, this formalization leads to strong modeling restrictions, as we discuss in section 5.1. The authors also pointed out how this approach helped to solve intensionality issues originated from set-theoretic interpretations.

Later, Lu (2005) argued for a category-theoretic formalization of knowledge that abstracts from many different kinds of knowledge representation mechanisms, including database models and ontologies. Lu's proposal defined categories enriched with morphism "types", along with type composition rules. That is, given two composable morphisms and their types, rules on how to determine the type of their composition. The set of types must be closed under composition, that is, for any two types t and v their composition $v \circ t$ must be among the existing types, and must contain a unity type that is neutral regarding composition, i.e., for a unity type u and any other type t, $u \circ t = t$. We expand on this definition in section 5.1 with the use of functors to map each morphism to its type.

Peruzzi (2006) has provided a strong argument for the replacement of set theory by category theory as a foundation in many areas of philosophy. His work precisely identifies the inherent extensionality of set theory, even when used in conjunction with modal logic or possible-world semantics. The author then goes on to show how category theory may be utilized to allow several different forms of extensionality distinct from the classical set-theoretic well-pointedness. Additionally, Peruzzi describes (scientific and philosophical) theories as categories, rather than as sets of logical formulae, and models for one such theory as functors from it to other categories. Since ontologies are theories that account for a certain world view, his work supports our proposal of ontologies as categories, which we detail in chapter 5.

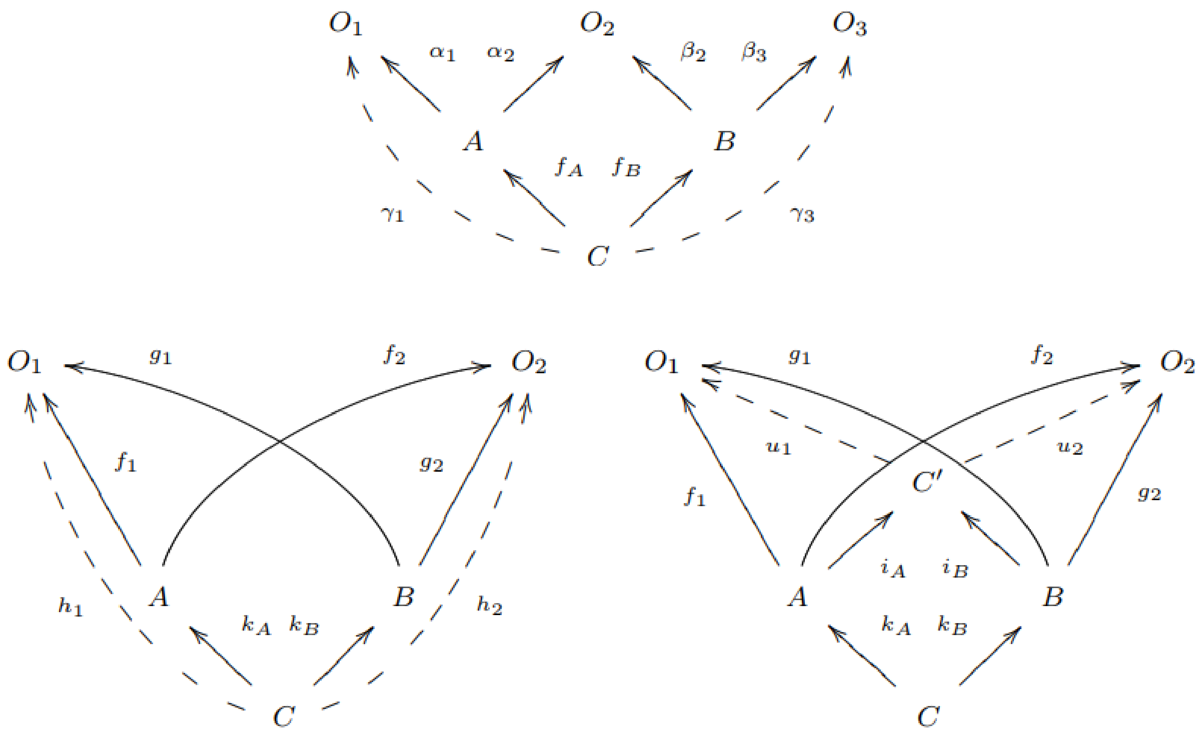
Johnson and Rosebrugh (2010) used category theory for the representation of ontologies, defining the coproduct of concepts to be their disjoint union. They have also proposed the pullback over a diagram with a specialization $s{:}A{\rightarrow}C$ and another conceptual relation $r{:}B{\rightarrow}C$ as the inverse image of s along r, i.e., a specialization of B defined by being related to A. Later, Spivak and Kent (2012) found the same constructions in their ologs (ontology logs, defined as labeled categories), in addition to the pullback over two specializations $s_A{:}A{\rightarrow}C$ and $s_B{:}B{\rightarrow}C$ as the greatest common subsumee. In section 5.5, we argue against the definition of the coproduct as a disjoint union but maintain both types of pullback. The root for this disagreement lies in the view of concepts as essentially sets of instances, present in the referred works, and in the resulting extensionality of the models.

The work of Seremeti and Kougias (2013) described how the composition of morphisms between ontologies facilitates the integration of new ontologies into preexisting ontology networks, requiring a single anchor-morphism to be constructed manually and them composing it with the other morphisms in the network to produce all possible mappings. The authors also proposed the representation of ontologies as path categories with morphisms in the form of lists of the vertices (concepts) and edges (relations) in the path between two concepts. They defined the composition of morphisms as the concatenation of consecutive paths. We propose a similar composition rule in section 5.3, with the caveat that paths with specific meanings are “compressed”, such as paths composed by transitive relations.

More recently, Aliyu et al. (2015) defined a category-theoretic formalization of the Resource Description Framework (RDF) with resources as objects and properties as morphisms. Their work did not make clear, however, how the composition of properties is supposed to behave.

Apart from formalizations of ontologies as categories, several works discuss the relations between ontologies in category-theoretic frameworks. Bench-Capon and Malcom (1999) defined relations between two ontologies O_1 and O_2 as pairs of morphisms $f_i{:}O{\rightarrow}O_i$ for $i = 1, 2$ in a category with ontologies as objects. Zimmerman et al. (2006) rebranded these relations as V-alignments, due to their shape, and defined the operations of composition, intersection and union via limits and colimits over these alignments. Figure 4.1 depicts the composition of two alignments (A,α_i) and (B,β_i) on the top, the intersection between alignments (A,f_i) and (B,g_i) on the bottom left and the union of the same alignments on the bottom right.

Figure 4.1 – Composition, intersection and union of V-alignments



Source: Zimmerman et al. (2006, p. 4-5).

The authors also defined the operation of ontology merging as a pushout over the alignment, but found the V-alignment lacking the expressiveness necessary for merging ontologies that are disjoint, i.e., that do not share concepts, but that contain concepts related by specialization relations. The authors propose two possible methods to obtain the desired merge. The first method utilizes W-alignments, which require the construction of a bridge ontology and the computation of three distinct pushouts to find the merge. The second method is based on a redefinition of the category of ontologies to accommodate morphisms that are sets of triples expressing relations between the concepts in each ontology. Composition is then defined as $(x,z,R) \in g \circ f \leftrightarrow \exists\, y\ (x,y,R_1) \in f \wedge (y,z,R_2) \in g \wedge R = R_2\ \varphi\ R_1$, where φ is the composition of conceptual relations defined through a composition table. In section 5.6.3 we discuss how our proposal leads to the same formalization of the merge as a pushout over an alignment of non-disjoint ontologies, and in section 5.6.5 we propose a different solution to the problem of merging disjoint ontologies.

Cafezeiro and Haeusler (2007) expanded the category-theoretic constructions in categories of ontologies to include the pullback from two ontology mappings as a search for similarity in the context of a broader ontology. The authors also described how equalizers, that is, limits over a diagram with two morphisms f and $g:A \rightarrow B$ with the same domain and codomain, could be used to hide sensitive information inside an ontology. Following works expanded on this basis with the introduction of "contextualized entities", which are pairs of ontologies where one provides context to the other (CAFEZEIRO et al., 2014; CAFEZEIRO; HAEUSLER; RADEMAKER, 2008). An algebra for operating such pairs is then defined, including the operations of entity integration, context integration, relative intersection and collapsing union. Entity integration is given by a pullback over two entities that share a context, while context integration is a pushout over a single entity in two different contexts. The authors define the two remaining operations respectively as the pullback and the pushout in the category whose objects are the entity-context morphisms from the original ontology.

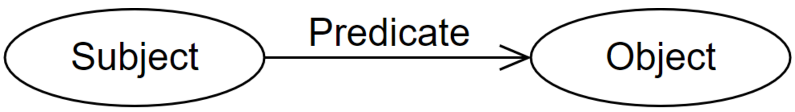
Noticing a gap in the preexisting literature, Antunes and Abel (2018) specified a set of guidelines for the evaluation of the semantic soundness and expressiveness of ontology morphisms. Five criteria where defined: first, that the association of concepts should be injective; second, that the association of relations should also be injective; third, that the mapping of relations should preserve their respective domains and codomains; fourth, that the mapping of concepts should preserve their metaproperties; and fifth, that the ontology morphisms should represent relations apart from equivalence between concepts. The metaproperties referred to here are the ones we have presented in section 2.2. We discuss in the conclusion (chapter 6) how our proposal fulfills these requirements.

5 ONTOLOGIES AS CATEGORIES

That the knowledge of entities is nothing apart from the knowledge of their qualities and relations is a proposition that has persisted in the philosophical literature since Hume's bundle theory (1888). Intuitively, this proposition seems to be indeed true: when describing things, we either describe their characteristics (an apple is red, a student is smart, a house is large) and their relations to other entities (an apple is *produced by* an apple tree, a student is *enrolled at an* educational institution, a house is *located in* some neighborhood). We can also identify broader concepts that generalize them (an apple is a fruit, a student is a person, a house is a building) that are, in turn, described in similar manner. Given that for long we have understood qualities as entities on their own rights, that are associated with their bearers by relations such as *inherence*, and since the connection of a universal to its more generic counterpart is also a relation, that of *specialization,* we can see the knowledge of entities as that of bundles of relations in which they participate. Apart from these bundles, we can say nothing on the subjects: the entities by themselves, i.e., without relations, appear as black boxes from which we can extract no information[2].

Accordingly, the practical construction of ontologies generally focuses on the relations between concepts. This focus can be testified by the usual representation of ontologies as graphs, as well as by the constructs used by ontology representation languages to model the knowledge, such as OWL's properties (ANTONIOU; VAN HARMELEN, 2009) and RDF's triples (CYGANIAK; WOOD; LANTHALER, 2014), which represent relations (shown in figure 5.1). Thus, tasks of ontological analysis and engineering are largely dependent on the scrutiny of relations.

Figure 5.1 – Representation of an RDF triple consisting of a subject, a predicate and an object



Source: Cyganiak, Wood and Lanthaler (2014, p. 3).

[2]This perspective is akin to a weak version of Hume's bundle theory. In Hume's theory, substances are considered to be nothing apart from collections (or bundles) of properties, with no underlying substratum. The notion presented here makes no judgement on the existence (or lack thereof) of such substrata, depending only on the impossibility of attaining, and consequently modeling, knowledge of it. This impossibility is supported by Hume's original argument.

As previously stated, category theory provides a sound formal basis for examining relations in the form of morphisms. This aspect leads to the conclusion that category theory is a suitable framework for the study of ontologies. Additionally, category theory provides a solution to intensionality issues present in previous set theoretic approaches. Set theory's axiom of extensionality states that two sets are equal if they have the same elements, i.e., a set is determined solely by its members. The issues this brings to conceptual modeling are easily noticed. If we define a conceptual relation over a universe of discourse that is a set, the inclusion or exclusion of any entity in the domain requires a new universe of discourse (since the two sets have different extensions and thus are different), and hence a new conceptual relation would need to be defined. Diversely, when a conceptual model is specified, it is intended to outline the underlying meaning of the concepts independently from any specific state of the world.

The current solution, introduced to the field of ontology engineering by Guarino (1998), does not get rid of set theory. His work defines a new set (of possible worlds) over which the universe of discourse is indexed. The entities and relations are then defined extensionally over this indexed domain space and the ontology "approximates as well as possible the set of intended models" according to such definitions. This formalization says nothing about how the ontology expresses the meanings of concepts, leaving this important aspect hidden behind an "approximation" in first order logic, and, while it solves the issue of allowing the real world to change without the need of new conceptualizations to account for it, it does not reflect how we model conceptual knowledge in reality. According to this definition, in order to build the best possible approximation of the set of intended models of the conceptualization, full knowledge of the extensions of every entity and relation in each possible world would be required, which is an unreasonable requirement. Furthermore, concepts with different intensions but equal extensions cannot be distinguished, as was noted by the author.

With category theory, on the other hand, concepts are not anchored to their extensions, but to the conceptual relations in which they participate. Intuitively, an entity is said to instantiate a universal when it participates in the relations that define the universal. That is, the meaning (its intension) of the universal is expressed in the relations, independently from its possible instances (its extension). In the following sections, we shall present formalizations based on category theory for several notions related to ontology engineering.

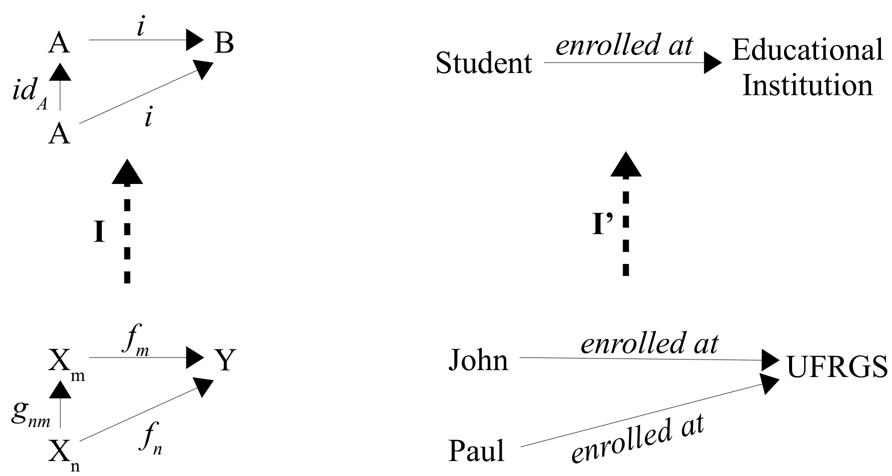
5.1 FORMALIZATION OF INSTANTIATION RELATIONS

In this section, we propose category-theoretic formalizations for instantiation relations. As discussed in chapter 2, instantiation is the relation that holds between entities and their respective types, i.e., it holds whenever a given entity is covered by a universal's principle of application. In traditional two-level modeling, we formalize these relations as a functor $I{:}S \rightarrow O$, where:

- S is a state of affairs formalized as a category with monadic particulars as objects and particular relations as morphisms, and
- O is an ontology formalized as a category with monadic universals as objects and relations (both universal and particular) as morphisms.

Thus, we "map" each particular entity in S to its universal in O through I. If S is a typed category as defined by Lu (2005), we define the type of each S-morphism by its image in O and type composition is simply composition in O. This definition respects associativity and the other axioms of type composition. The unit type is given by O-identities, which are trivially neutral regarding composition. Additionally, given two S-morphisms $f{:}X \rightarrow Y$ and $g{:}Y \rightarrow Z$, due to the definition of functor, there exists an O-morphism $i = I(g) \circ I(f)$ such that $i = I(g \circ f)$, i.e., the composition of the types of g and f is the type of the composition $g \circ f$.

Figure 5.2 – Example of instantiation functor that fails fibration rules

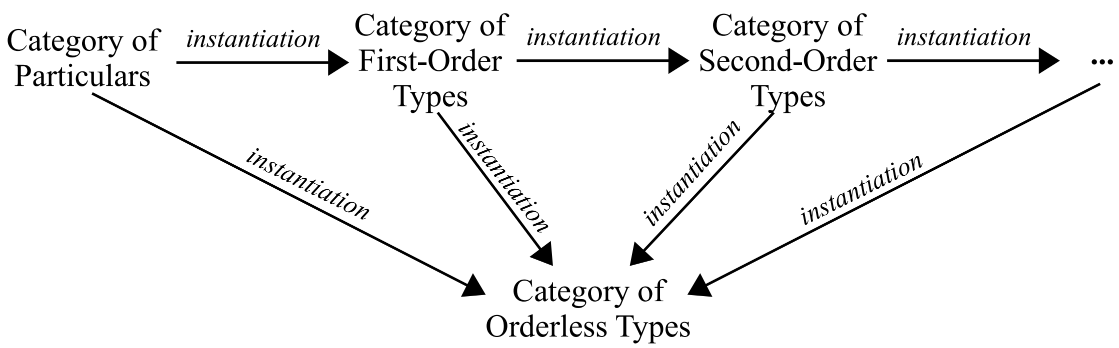


Source: the author.

This approach does not employ fibrations (or op-fibrations) as proposed by Colomb, Dampney and Johnson (2001). This is due to the fact that requiring I to be a fibration would entail that, for any many-to-one relation $i{:}A \rightarrow B$, given Y instance of B, each X_n instance of A with $f_n{:}X_n \rightarrow Y$ instance of i must be related to each other X_m by a relation $g_{nm}{:}X_n \rightarrow X_m$ such that $f_m \circ g_{nm} = f_n$ and $I(g_{nm}) = id_A{:}A \rightarrow A$. This definition would preclude, for example, the modeling of a state of affairs where two distinct students *John* and *Paul* are enrolled at the same educational institution *UFRGS* without being related to each other. Figure 5.2 shows this situation, where I is a fibration but I' is not. Similar world states frequently occur in reality. Conversely, requiring I to be an op-fibration would similarly restrain one-to-many relations concerning their codomains.

We can easily generalize this formalization to multi-level modeling by lifting the particularity restriction on S-objects and S-morphisms. Thus, an ontology could contain first-order universals that instantiate second-order universals from another ontology, which, in turn, may instantiate third-order universals and so on. The approach connects each level in this stratified scheme to the next through an instantiation functior. Orderless types, i.e., types with instances in multiple different orders, may be included with the addition of new instantiation functors from each tiered category to the category of orderless types, as presented in figure 5.3. It is important to note that, while instantiation is intransitive, it is not antitransitive, since some orderless types such as Type and Entity are instances of themselves (ALMEIDA; FONSECA; CARVALHO, 2017), and thus violate the antitransitivity condition $\forall$ *x, y, z: (x instance of y* $\wedge$ *y instance of z)* $\rightarrow$ $\neg$ *(x instance of z)* when $x = y = z =$ Type.

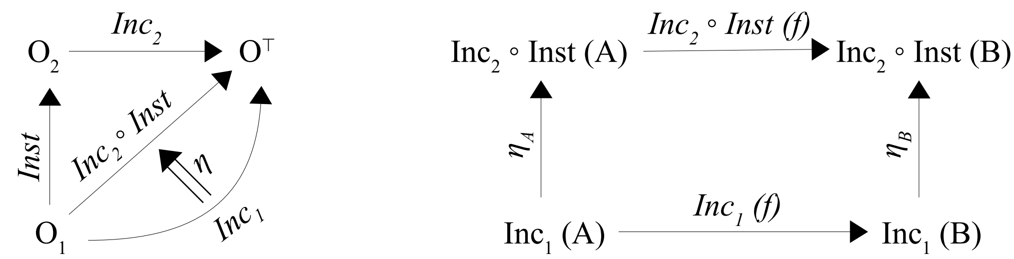
Figure 5.3 – Scheme of categories in different abstraction levels and their instantiation functors



Source: the author.

Since each component of an instantiation functor relates two entities, we may model it as a family of particular relations between entities of a single ontology to allow the study of other types of cross-level relations. To enable such modeling, a definition of instantiation as a conceptual relation is necessary, which may be given in terms of the instantiation functor and a natural transformation. Suppose an ontology $O^{\top}$ with all possible monadic universals and all possible relations between them. This ontology is an object in the category of ontologies (which we shall discuss in section 5.6) such that for every other ontology O there exists at least one inclusion functor $Inc{:}O \rightarrow O^{\top}$, that is, a functor that is both faithfull and injective on objects. Given two ontologies O_1 and O_2, an instantiation functor $Inst{:}O_1 \rightarrow O_2$ and inclusion functors $Inc_1{:}O_1 \rightarrow O^{\top}$ and $Inc_2{:}O_2 \rightarrow O^{\top}$, there is a natural transformation $\eta{:}Inc_1 \rightarrow Inc_2 \circ Inst$ whose components η_A are instantiation relations in $O^{\top}$. Figure 5.4 shows, on the left, the diagram with the three ontologies and the referred functors, and, on the right, the commuting square consisting of the images of a morphism f from O_1 via Inc_1 and via $Inc_2 \circ Inst$ and the components of the natural transformation η.

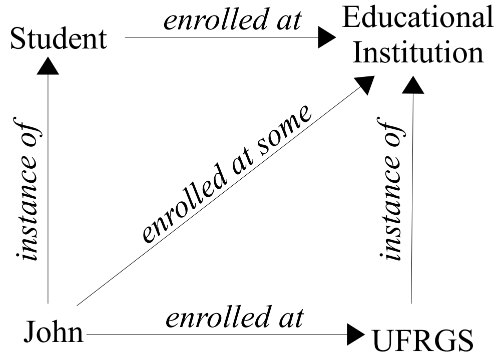
Figure 5.4 – Instantiation as natural transformation



Source: the author.

This formalization means that, for any universal relation $u{:}A \rightarrow B$ and any relation $r{:}X \rightarrow Y$ that instantiates it, there must exist two instantiation relations $i_1{:}X \rightarrow A$ and $i_2{:}Y \rightarrow B$, such that the square formed by the four relations commutes, i.e., $u \circ i_1 = i_2 \circ r$, as in the example given in figure 5.5, where u is the universal relation *enrolled at* between the universals *Student* and *Educational Institution*, r is the particular relation *enrolled at* between *John* and *UFRGS*, i_1 and i_2 are the instantiation relations between, respectively, *John* and *Student* and *UFRGS* and *Educational Institution*, and the composition $u \circ i_1 = i_2 \circ r$ is the relation *enrolled at some*.

Figure 5.5 – Example of composition with instantiation



Source: the author.

A separate analysis of each composition elucidates this definition. The composition $u \circ i_1$ is an instance-bound (by *X*) specialization of *u*, while the composition $i_2 \circ r$ is an instance-bound universalization of *r*. In these operations, the resulting composition preserves the real-world semantics from both *r* and *u*, while restricting the domain of $u \circ i_1$ to a single instance and generalizing the codomain of $i_2 \circ r$ to a (higher-order) universal. Consequently, other relations may instantiate $i_2 \circ r$. In fact, *r* is an instance of $i_2 \circ r$, and, if the diagram commutes, also of *u*.

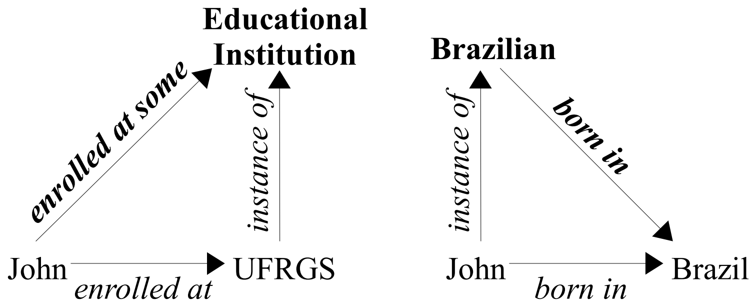
5.1.1 Instance-bound Universal Relations

An instance-bound universal relation is defined here as a universal relation between a particular or lower-order universal and a higher-order universal, i.e., a universal relation that crosses the boundary between different levels of classification. We say that a universal relation is "bound by *X*" when *X* is the lower-order entity related by it.

An instance-bound universal relation is a generalization of similar relations in which *X* takes part. As discussed previously, every unbound universal relation $u{:}A \rightarrow B$ gives rise to an instance-bound specialization of itself for each instantiation $i_n{:}X_n \rightarrow A$ through the composition $u \circ i_n$. Differently from unbound relations, we do not define the instantiation of instance-bound relations over a commutative square, but over a commutative triangle. That is, given an instance-bound relation $b{:}X \rightarrow B$ and an instantiation $i{:}Y \rightarrow B$, an instance of *b* is a relation $r{:}X \rightarrow Y$ such that $i \circ r = b$.

So far, the focus of this discussion has been on relations that are bound by the domain, but one can easily imagine relations directed in the opposite direction, that is, from the higher-order universal to the lower-order entity. Due to the definitional nature of relations with universals as domain, all instances of the universal would necessarily participate in an instance of such relation, which would lead to an instance-bound monadic universal. One possible example of such concepts is the universal *Brazilian*, which is defined by a relation with the particular *Brazil*. Previous works have referred to similar universals as "dependent types" (XI, 1998). Figure 5.6 shows examples for both domain-bound and codomain-bound universal relations, where universals are emphasized in bold text.

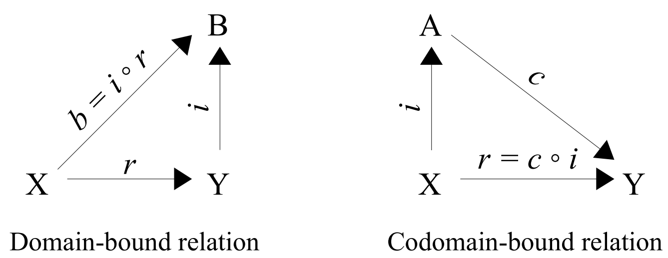
Figure 5.6 – Examples of instance-bound universal relations



Source: the author.

The instantiation triangle for codomain-bound relations is somewhat reversed. Given an instance-bound relation *c:A→Y* and an instantiation *i:X→A*, an instance of *c* is a relation *r:X→Y* such that $r = c \circ i$. We present the instantiation of both types of instance-bound relations in figure 5.7. We note that, while the formalizations presented in this work allow the inclusion of codomain-bound universal relations in the model, they are entirely inessential for the framework.

Figure 5.7 – Instantiation of instance-bound universal relations



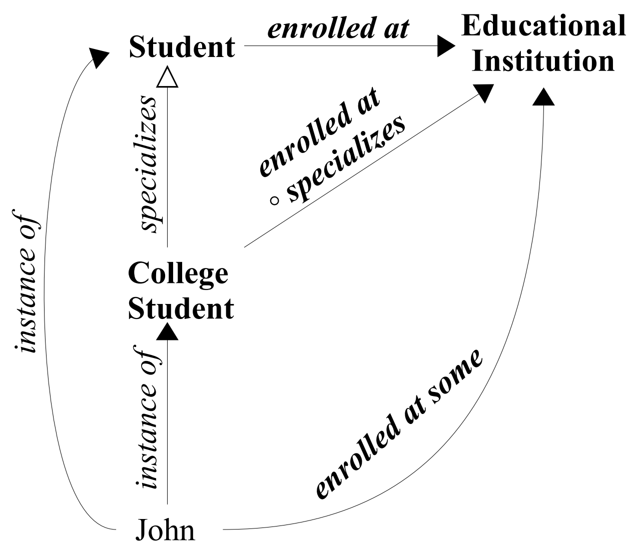
Source: the author.

5.2 FORMALIZATION OF SPECIALIZATION RELATIONS

Another kind of relation that is present in most ontologies is that of specialization. Relations of specialization are particular relations between universal entities that are usually defined as *A specializes B* $\leftrightarrow \forall$ *X (X instance of A* $\rightarrow$ *X instance of B)*. This definition appears, at first, to contradict the formalization of instantiation as a functor, since it would map each entity to a unique universal that it instantiates and prevent, in a way, the instantiation of multiple universals by a single entity. However, a closer inspection of the intuitive notions of instantiation, presented at the beginning of this chapter, clarifies the issue.

It is required for an instance to take part in the relations that define its corresponding universal. Thus, stating that every instance of a universal *A* also instantiates a second universal *B* means that *A* is defined upon at least all the relations that define *B*. We model this proposition by the composition of the specialization relation $s{:}A \rightarrow B$ and each universal relation $u_i{:}B \rightarrow C_i$ with *B* as domain. Each composition $u_i \circ s : A \rightarrow C_i$ is a specific (but still universal) form of u_i that preserves its meaning while restricting the domain. Figure 5.8 shows an example of such compositions, where we indicate universals in bolded text and specialization by a hollow-pointed arrow. Analogously, given a universal relation $v{:}D \rightarrow A$, the composition $s \circ v : D \rightarrow B$ is a generalization of *v* with a broader domain that maintains its meaning. That property has implications on the cardinality of universal relations that we shall discuss in section 5.4. It also means that specialization is both monic and epic.

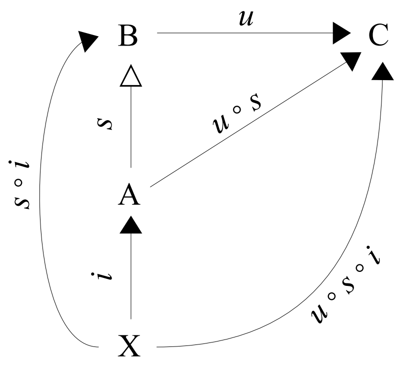
Figure 5.8 – Example of composition with specialization, instantiation and universal relations



Source: the author.

With the definition of instantiation as a family of relations, we can achieve the same conclusions by specifying instantiation as transitive over specialization, i.e., for any instantiation $i{:}X \rightarrow A$ and any specialization $s{:}A \rightarrow B$, $s \circ i : X \rightarrow B$ is also an instantiation. Indeed, it is easy to see that the diagram formed by the three relations and their compositions respects associativity, that is, $u \circ (s \circ i) = (u \circ s) \circ i$. Therefore, both formalizations yield equivalent results. While figure 5.8 gives an example of this property, figure 5.9 shows the formal diagram..

Figure 5.9 – Associativity on composition of instantiation, specialization and universal relations



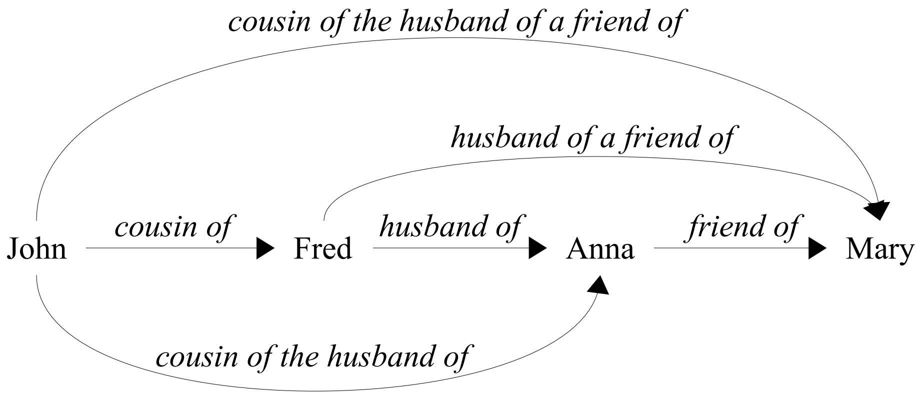
Source: the author.

Furthermore, the definition of specialization clearly implies that it is transitive, i.e., given two specialization relations $s_1{:}A \rightarrow B$ and $s_2{:}B \rightarrow C$, the composition $s_2 \circ s_1 : A \rightarrow C$ is also a specialization. We model this transitivity in typed categories as a type *specialization* subject to the type composition rule (denoted by $\circ^T$) *specialization* $\circ^T$ *specialization* = *specialization*. Similarly, we model the transitivity of instantiation over specialization with the addition of a type *instantiation* and the rule *specialization* $\circ^T$ *instantiation* = *instantiation*.

5.3 COMPOSITION OF RELATIONS

In order to define a category with relations as morphisms, we need to define a composition operation over relations, i.e., an operation that maps each pair of relations $f{:}A \rightarrow B$ and $g{:}B \rightarrow C$ to a relation $f \circ g{:}\ A \rightarrow C$. Previous sections included discussions on the behavior of specialization and instantiation relations on composition, but we will provide a more comprehensive description. Generically, relations are composable simply through the explicitation and connection of each constituent's meaning. Constructions of this sort are common in natural language, as exemplified in figure 5.10, in relations such as John is *cousin of the husband of a friend of* Mary, that is the composition of the relations John is *cousin of* Fred, Fred is *husband of* Anna and Anna is *friend of* Mary. We note that this diagram both commutes and respects associativity. This method for the composition of conceptual relations is similar to the one defined by Seremeti and Kougias (2013) for ontologies represented as path categories.

Figure 5.10 – Example of composition of relations in the domain of human relationships



Source: the author.

While this procedure is enough for an overall description of the composition of relations, each ontology should further specify composition for domain specific relations with singular meanings, such as transitive relations like *ancestor of* in the genealogical domain and *on top of* in the domain of spatial position.

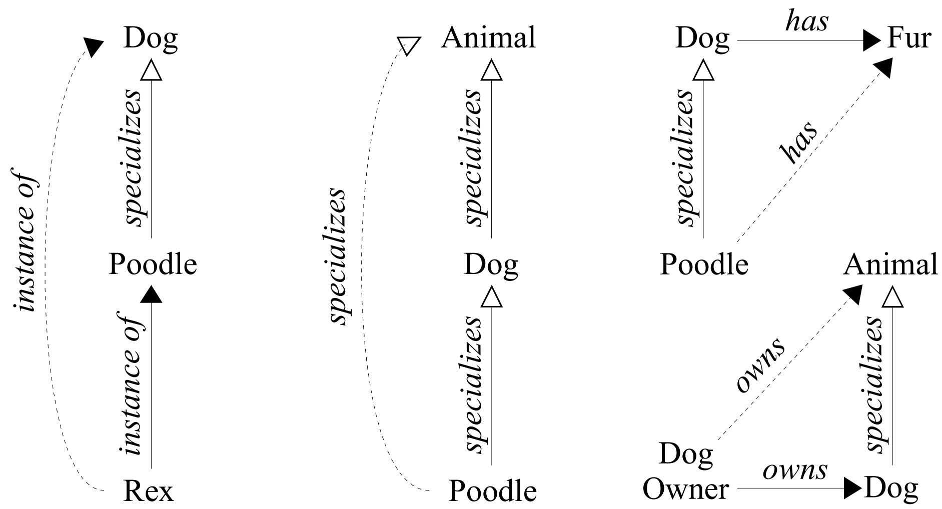
It is noteworthy that the composition of two unbound universal relations always yields an unbound universal relation. This proposition is a trivial consequence of both definitions of instantiation presented in section 5.1. Assume otherwise, i.e., that there exists two universal relations $f{:}B\to C$ and $g{:}A\to B$, such that their composition $f \circ g{:}A\to C$ is a particular relation. For an instantiation functor I, it is easy to find as counterexample two composable relations $a{:}Y\to Z$ and $b{:}X\to Y$ such that $I(a)=f$ and $I(b)=g$. If $I(a \circ b) = f \circ g$, the definition of particular entities (as those that may not be instantiated) is denied. Conversely, if $I(a \circ b) \neq f \circ g$, the functor axioms are contradicted. Analogously, for a family of instantiation relations $i_1{:}X\to A$, $i_2{:}Y\to B$ and $i_3{:}Z\to C$, associativity implies that $f \circ (i_2 \circ b) = (f \circ i_2) \circ b$ and, if the corresponding diagram commutes (as required by the definition of instantiation), that is, $g \circ i_1 = i_2 \circ b$ and $f \circ i_2 = i_3 \circ a$, we have $f \circ (g \circ i_1) =(i_3 \circ a) \circ b$, which means that the square formed by the relations $f \circ g$, $a \circ b$, i_1 and i_3 also commutes, and thus $a \circ b$ instantiates $f \circ g$. Since the assumption leads to contradictions, it must be false.

5.3.1 Modeling Inference as Composition

Most of the composition rules discussed so far emulate inference processes commonly found in ontology-based systems. In the case shown in figure 5.5, for example, the composition of instantiation and universal relations materializes the knowledge that there exists some educational institution at which *John* is enrolled, even if there is missing information on which particular institution plays that part. This is precisely a logic entailment, as discussed in section 2.3.

The behavior of specialization on composition is another example of this phenomenon. Figure 5.11 shows several examples, with dashed arrows denoting the relations "inferred" through composition. It is plain to see how both the transitivity of instantiation over specialization and the transitivity of specialization itself lead to the discovery of previously implicit knowledge, as well as the composition of specialization and universal relations.

Figure 5.11 – Examples of composition as inference



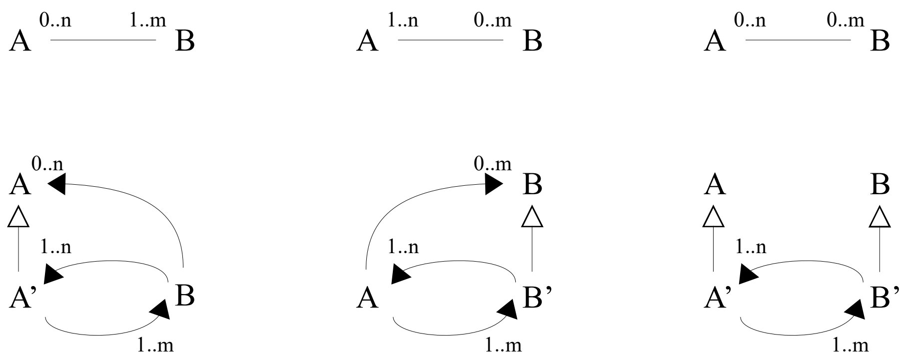
Source: the author.

5.4 CARDINALITY OF RELATIONS

In conceptual modeling, the cardinality of a relation is a restriction on the minimum and maximum numbers of entities associated through that relation. Cardinality may impose different restrictions for the relation's domain and codomain. Usually, minimum restrictions take a value of 1 or 0 (no restriction) and maximum restrictions of 1 or any (no restriction, usually denoted by '*'), but that is not always the case. Some parthood relations, for example, have minimum cardinality 2 on the part end of the relation, due to the supplementation principle. In our framework, cardinality materializes as restrictions on the instantiation of the relation. Accordingly, only universal relations can have cardinalities, given that particular relations hold directly between the two associated entities.

The definition of instantiation and the composition rules presented in section 5.1 presuppose that the universal relations present in the ontology are obligatory for every instance of their domains. Therefore, the formalization of an ontology as a category does not accept relations that are optional for their domains, i.e., whose minimum cardinality on the domain is 0. Instead, we must formalize non-mandatory relations through a specialization of the domain over which the relation is obligatory. Figure 5.12 shows how to formalize relations with different minimum cardinalities. We note that the relation *B→A* in the first diagram is the result of composing the relation *B→A'* with the specialization *A'→A*. Similarly, in the second diagram the relation *A→B* is the composition of *A→B'* with the specialization *B'→B*.

Figure 5.12 – Non-mandatory relations and their corresponding formalizations

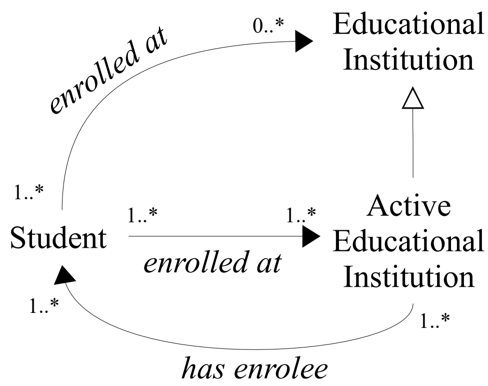


Source: the author.

We demonstrate here the broadening of the codomain of universal relations when composed with specializations, as discussed in section 5.2, along with its effects on cardinality. The composition is optional for the codomain, i.e., it has minimum cardinality 0, even if the original relation has a higher minimum. This optionality comes naturally from the fact that not every instance of *A* is an instance of *A'*, and the original relation is mandatory only for instances of *A'*.

Evidently, this kind of formalization is only necessary if we intend to model the relation bidirectionally. If, instead, we regard it as a single relation directed from the universal on which it is obligatory to the one on which it is not, the straight-forward formalization as a single morphism is enough. For example, if the enrollment relation between students and educational institutions is only mandatory for students, being optional for the institutions (i.e., there may exist educational institutions without any enrolled student), we may choose to model the relation only in the *Student→Educational Institution* direction, or we may define a specialization of *Educational Institution* called *Active Educational Institution*, which has a mandatory relation in the opposite direction and, therefore, whose instances must have enrolled students. We show this example in figure 5.13.

Figure 5.13 – Example of non-mandatory relation



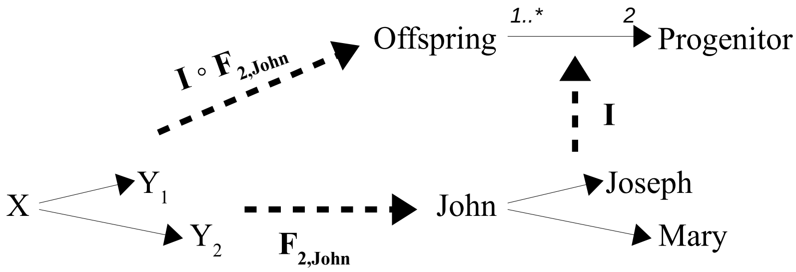
Source: the author.

5.4.1 Formalizing Cardinalities as Restrictions on Instantiation

As previously stated, cardinality manifests as restrictions on instantiation. That is, given a universal relation $u:A \rightarrow B$ in O with maximum cardinality n in the codomain and an instantiation functor $I:S \rightarrow O$, at most n instances of u may share a single instance of A as codomain. We formalize this restriction with injective functors and a family of categories C_k whose only non-identity morphisms are $f_i:X \rightarrow Y_i$ for $i = 1, 2, \ldots, k$. If u has maximum cardinality n in the codomain, then for any $k > n$ there exists no injective functor $F_k:C_k \rightarrow S$ such that $I \circ F_k (f_i) = u$ for all f_i. Pragmatically, we must check the restriction only for the category C_{n+1}, since for any C_{n+j} with $j > 1$ there exists an obvious inclusion $H:C_{n+1} \rightarrow C_{n+j}$ that composed with an injective functor $F_{n+j}:C_{n+j} \rightarrow S$ would result in an injective functor from C_{n+1} to S. If we prove that no such injective functor exists, all others with higher k consequently do not exist as well. Conversely, if there is an injection functor for C_{n+1}, then the instantiation I is not valid.

We formalize restrictions on the maximum cardinality of the domain in an analogous way, with the exception that we exchange the family of categories C_k for its dual, that is, a family D_k with morphisms f_i:$X_i \rightarrow Y$ for $i = 1, 2, ..., k$. Thus, if u:$A \rightarrow B$ has maximum cardinality m in the domain, then there is no injective functor G_{m+1}:$D_{m+1} \rightarrow S$ such that $I \circ G_k (f_i) = u$ for all f_i.

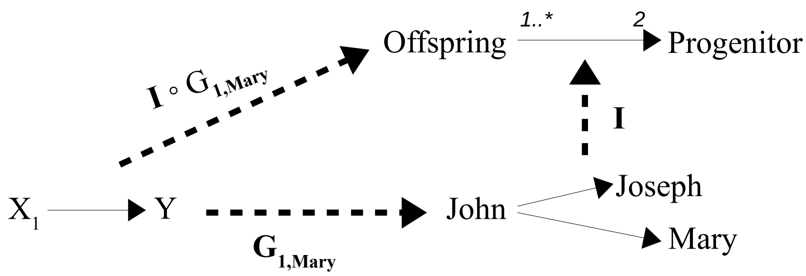
Figure 5.14 – Minimum codomain cardinality over instantiation as a functor



Source: the author.

Similarly, we formalize minimum restrictions as mandatory injective functors. If u:$A \rightarrow B$ has minimum cardinality p on the codomain, then for each V_i instance of A, i.e., $I(V_i) = A$, there exists an injective functor $F_{p,i}$:$C_p \rightarrow S$ such that $F_{p,i} (X) = V_i$. Figure 5.14 depicts one such situation, where we represent functors by bold dashed arrows. Accordingly, for a universal relation u:$A \rightarrow B$ with minimum cardinality q on the domain, for each W_i such that $I(W_i) = B$ there exists an injective functor $G_{q,i}$:$D_q \rightarrow S$ with $G_{q,i} (Y) = W_i$. We show an example in figure 5.15.

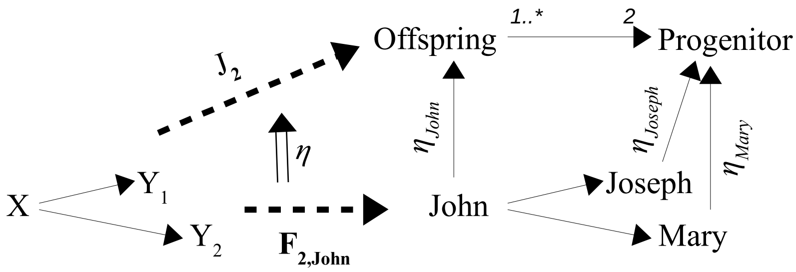
Figure 5.15 – Minimum domain cardinality over instantiation as a functor



Source: the author.

In order to check the restrictions in a multi-level model, we may adapt all formalizations presented in this subsection for the use of a family of instantiation relations in O rather than an instantiation functor I:$S \rightarrow O$. To achieve this, we exchange the compositions $I \circ F_k$ for functors J_k:$C_k \rightarrow O$ such that $J_k (f_i) = u$ for every f_i and the F_k of interest are the ones for which there is a natural transformation η:$F_k \rightarrow J_k$ whose components η_x are instantiation relations. Figure 5.16 represents this adaptation. For domain cardinalities, we follow the same process exchanging F_k for G_k and C_k for D_k.

Figure 5.16 – Minimum codomain cardinality over instantiation as a natural transformation



Source: the author.

The same process is applicable for instance-bound universal relations, with the singularities that the cardinality in the (co)domain for a (co)domain-bound relation is exactly one, for the obvious reason, and that the η_x component at the instance-bound end of the relation is an identity. Additionally, if the maximum cardinality in the other end of the relation is also of 1, then there exists exactly one functor F_k or G_k for each V_i or W_i.

5.5 CATEGORICAL CONSTRUCTIONS IN ONTOLOGIES

Based on the deliberations that we presented thus far, we can formalize an ontology as a category $O = (E, R, id, \circ)$ composed of:

- a set of objects E whose members are monadic entities,
- for each pair (A,B) of monadic entities in E, a set $R(A,B)$ of morphisms from A to B whose members are relational entities,
- for each entity A in E, an identity relation $id_A{:}A \rightarrow A$, and
- an associative composition operator $\circ$ that maps each pair of relations $f{:}A \rightarrow B$ and $g{:}B \rightarrow C$ to a relation $g \circ f{:}A \rightarrow C$, such that identities are neutral regarding $\circ$.

Additionally, the operator $\circ$ is subject to the rules of composition discussed so far. That is, the composition of two relations is the connection of their respective meanings, with the condition that, for any specialization relations s_1 and s_2, any instantiation relation i, any universal relation u and any particular relation p:

- $s_2 \circ s_1$ is a specialization relation;
- $s_1 \circ i$ is an instantiation relation;

- $s_2 \circ u$ is a generalization of u;
- $u \circ s_1$ is a specialization of u;
- $u \circ i$ is an instance-bound specialization of u;
- $i \circ p$ is an instance-bound universalization of p.

In the light of this definition, the following subsections shall discuss the meaning of some categorical constuctions, such as limits and colimits, in categories delineated in this manner, that is, ontologies formalized as categories.

5.5.1 Initial and Terminal Objects

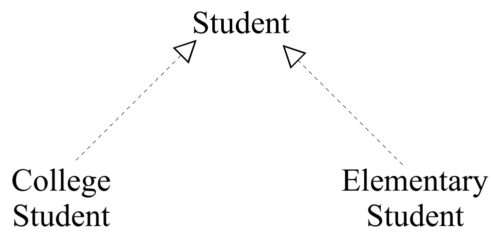
The terminal object of any ontology, if it exists, is its most generic universal. Common examples include OWL's *Thing* (ANTONIOU; VAN HARMELEN, 2009) and UFO's *Entity* (GUIZZARDI; WAGNER, 2010). We will refer to this concept as $\top$ in the following subsections. If an ontology contains no terminal object, then it contains at least two different universals that do not specialize any other universal.

Some ontologies have a concept that cannot be instantiated and specializes every other concept in the ontology, usually denoted $\bot$. When it exists, $\bot$ is the initial object of the ontology. Otherwise, there is no initial object unless the ontology contains no disjoint universals. In that case, the initial object is the most specific universal in the ontology.

5.5.2 Products and Coproducts

The coproduct of two universals A and B is a universal $A+B$ that relates to every entity to which both A and B relate. Furthermore, it relates to those entities in a way that approximates the meaning from both relations. This means that every instance of A and every instance of B is involved in relations analogous to the ones that define $A+B$. This formalization matches the definition of specialization given in section 5.2. Therefore, the coproduct $A+B$, when it exists, is the least common subsumer of A and B, that is, a universal whose intension is given precisely by their shared meaning, as we described in section 2.3. Figure 5.17 shows an example, where dashed arrows represent the cocone morphisms. The proposed formalization also implies that, given the specializations $s_A{:}A\rightarrow A{+}B$ and $s_B{:}B\rightarrow A{+}B$, for any $f{:}A\rightarrow C$ and $g{:}B\rightarrow C$, the relation $h{:}A{+}B\rightarrow C$, subject to $h \circ s_A = f$ and $h \circ s_B = g$, has as cardinalities on the (co)domain the lowest minimum (co)domain cardinality and the highest maximum (co)domain cardinality between f and g. Moreover, the initial object $\bot$ is neutral regarding the coproduct, while the terminal object $\top$ is absorbing with respect to it. That is, for any A, $\bot{+}A = A$, and $\top{+}A = \top$.

Figure 5.17 – Coproduct example



Source: the author.

This definition is opposed to the conceptions presented by Johnson and Rosebrugh (2010) and Spivak and Kent (2012), that interpret the coproduct as a “disjoint union” of classes. It fits here to pose the question on what a disjoint union of universals is supposed to be. If it is a universal whose set of instances is the disjoint union of the set of instances of A and the set of instances of B, then any entity that instantiates both A and B should be doubly present in the disjoint union. However, if the only relations that it participates as an instance of the disjoint union are the common relations from A and B, then there is no way of differentiating its A counterpart in the disjoint union from its B twin. If otherwise there are relations that are mandatory for some but not all of the instances of the disjoint union, the basic assumptions on how universals are defined are contradicted, as well as the restriction on optional relations discussed in section 5.4.

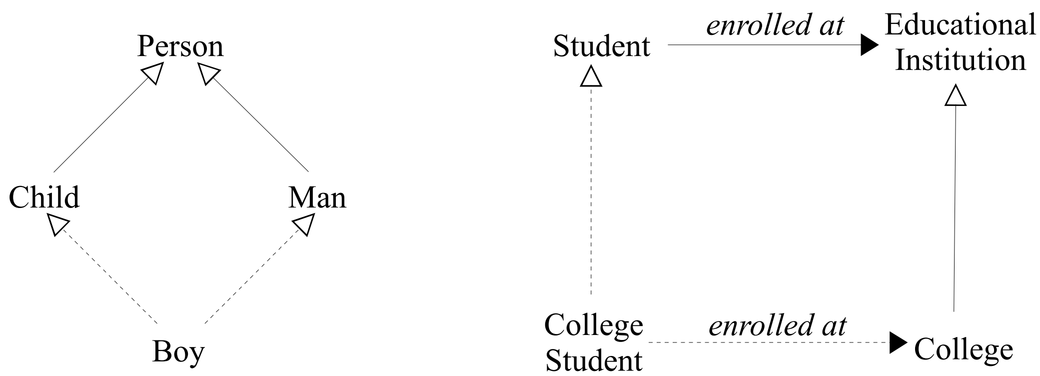
Further, this description is extensional, since it is defined not over the meaning of such universal but on the enumeration of its instances, while, as previously discussed, we consider ontological models as intensional. This extensionality implies that any instance that takes part in every relation that is common to A and B but is not an instance of either A or B is also not an instance of the disjoint union, even though its application principle would, theoretically, cover it. Finally, this definition would imply different categorical interpretations of the same universal for different world states. If A and B are not necessarily disjoint, then their generalization would be the categorical coproduct in every state of affairs where no instance that is shared by A and B exists, while it would not in states with any such instance.

The product of two universals A and B is a universal $A\times B$ to which every entity that is related to both A and B is related. No common specialization of A and B can fulfill this role, even when they are not disjoint, since there may be universals with instances that are related both to instances of A that are not instances of B and to instances of B that are not instances of A. Instead, the product $A\times B$ is the universal whose instances are ordered pairs of an instance of A and an instance of B, along with the relations *has A-part*:$A\times B\rightarrow A$ and *has B-part*:$A\times B\rightarrow B$. Any entity that is related to an instance of A and an instance of B is also related to their corresponding pair, by being related to both its A-part and its B-part. Thus, for any two relations $f{:}C\rightarrow A$ and $g{:}C\rightarrow B$, both f and g factor through a relation $h{:}C\rightarrow A\times B$ that comprises both their meanings.

5.5.3 Pullbacks

Given two monadic universals A and B with specializations $s_A{:}A\rightarrow C$ and $s_B{:}B\rightarrow C$, the pullback $A\times_C B$ of the corresponding diagram, when A and B are not disjoint, is the greatest common subsumee of A and B, as we delineated in section 2.3. Otherwise, the pullback either does not exist or is $\perp$ (in this last case, the only existing cone is given by $s_{\perp A}{:}\perp\rightarrow A$ and $s_{\perp B}{:}\perp\rightarrow B$). For any cone $f{:}Y\rightarrow A$ and $g{:}Y\rightarrow B$, the diagram must commute, that is, $s_A \circ f = s_B \circ g$. The commutativity here means that f and g carry the same real-world semantics, and, thus may both be instantiated by the same relation in reality. Therefore, there exists some common specialization of A and B such that both f and g factor through it. If this specialization cannot be further generalized, it is the pullback. The diagram presented on the left of figure 5.18 is an example of a pullback of this kind.

Figure 5.18 – Pullback examples



Source: the author.

For a diagram containing a universal relation $u:A \rightarrow C$ and a specialization $s_B:B \rightarrow C$, the pullback is a specialization of A whose instances are exactly all instances of A that are related through u to instances of B. For any cone $f:Y \rightarrow A$ and $g:Y \rightarrow B$, the diagram must commute, that is, $u \circ f = s_B \circ g$. As discussed in section 5.2, the composition $s_B \circ g$ is a generalization of g. Thus, $u \circ f$ should be of a similar generalization level, which requires that f is either a generalized relation or a specialization. If it is a generalized relation, then it factors through another relation $r:Y \rightarrow Y'$ and a specialization $s:Y' \rightarrow A$, and, since $u \circ f$ and $s_B \circ g$ are supposed to have the same real-world meaning, it follows that g should also contain the meaning of r, that is, factor through it. In that case, Y cannot be the pullback since there is a unique arrow $r:Y \rightarrow Y'$ for which the diagram commutes. If, however, f is a specialization, then Y either has as instances all instances of A that are related to instances of B or is a specialization of the universal that has. Intuitively, this pullback comes from a process similar to the one described in section 5.4. Since A has a mandatory relation with C and B is a specialization of C, then there is a (hidden) optional relation from A to B which is formalized by the pullback, as exemplified in the right side of figure 5.18, where A is the universal *Student*, B is *College* and C is *Educational Institution*. This definition matches the pullbacks found by Johnson and Rosebrugh (2010).

Both types of pullback described in this subsection emulate the discovery of new concepts that may be implicit in the original ontology. In the examples, the discovered concepts are the universals *Boy* and *College Student*. Similarly to the composition rules as discussed in subsection 5.3.1, these constructions are akin to inferential processes executed by ontology-based systems. The knowledge needed to infer the existence of such concepts is present in the original diagrams, while the pullback provides their meaning.

5.6 A CATEGORY OF ONTOLOGIES

Just as easily as we have analyzed monadic and relational entities in a category-theoretic context, we may take ontologies themselves as the objects of a category. Since we formalize ontologies as categories, we consider functors as morphisms between ontologies. Additionally, it is also desirable to study the instantiation functors from world states to ontologies, and therefore our category shall also admit world states (formalized as categories, of course) as objects. Thus, we define this category as $Ont = (O, F, id, \circ)$, consisting of:

- a class of objects O whose members are ontologies and world states formalized as categories,
- for each pair (A,B) of objects in O, a set $F(A,B)$ of functors from A to B,
- for each entity A in E, an identity functor $id_A:A\rightarrow A$, associating each object and each relation in A to itself, and
- the usual functor composition operator $\circ$.

It follows effortlessly from this definition that *Ont* is a subcategory of *Cat*, the category of small categories.

5.6.1 Initial and Terminal Objects

As mentioned previously, the ontology $O^{\top}$, that contains every monadic universal and all relations between them, is an object in the category of ontologies. Even though there exists an inclusion functor $Inc:O\rightarrow O^{\top}$ for every ontology O and an instantiation functor $I:S\rightarrow O^{\top}$ for every world state, $O^{\top}$ is not the terminal object in the category, since those functors are seldom unique. The ontology O^1 containing a single object and its identity morphism, for example, may be mapped to $O^{\top}$ in as many different ways as there are objects in it. Instead, the title of terminal object falls precisely upon O^1, since every ontology and every world state may be mapped to it by a unique functor that collapses all objects into O^1's single object and all morphisms into its identity. The initial object of *Ont* is the empty ontology O^0 that has no objects and, consequently, no morphisms. O^0 is mapped to every other object in *Ont* via the empty functor.

5.6.2 Products and Coproducts

As discussed in chapter 4, cones for a discrete diagram with two ontologies as objects were described in the related literature as ontology alignments. The terminal cone over such a diagram is a product. Products are not, however, very useful as alignments, since, like its counterpart in *Cat*, products in *Ont* are product categories, which would "align" every object from each ontology to all objects in the other. A product of two small categories A and B is a category $A\times B$ whose objects are ordered pairs (a,b) and whose morphisms are ordered pairs *(f,g)*, where *a* is an object and *f* is a morphism in *A* while *b* is an object and *g* a morphism in *B*. The product $A \times B$ is trivially isomorphic to $B \times A$.

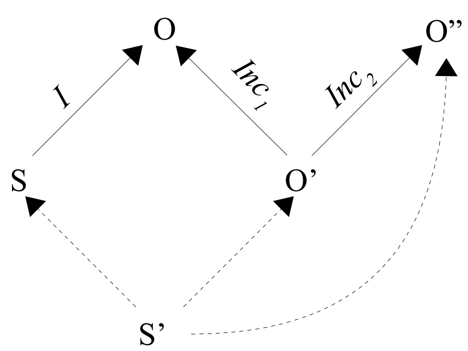
Coproducts in *Ont*, as in *Cat*, are disjoint unions of categories. The disjoint union A+B is a category whose collection of objects is the disjoint union of the objects from A and B. Also, the set $R_{A+B}(X,Y)$ of morphisms between two objects *X* and *Y* is equal to $R_A(X,Y)$ if both *X* and *Y* are *A*-objects, equal to $R_B(X,Y)$ if both *X* and *Y* are *B*-objects or empty otherwise.

5.6.3 Pullbacks and Pushouts

Previous authors have demonstrated in the category-theoretic literature (TRNKOVÁ, 1965) that a pushout over a diagram with inclusion functors $Inc_1{:}A \rightarrow B$ and $Inc_2{:}A \rightarrow C$ is the amalgamation of *B* and *C* with respect to Inc_1 and Inc_2. This proposition matches the prevailing notion that a pushout over an ontology alignment is a merge of the involved ontologies. If the chosen alignment is a product, however, the resulting merge is necessarily the terminal object $O^{`1}$.

The pullback over a diagram with two inclusion functors $Inc_1{:}A \rightarrow C$ and $Inc_2{:}B \rightarrow C$ is the intersection of *A* and *B* in respect to *C*. That is, a category whose objects are those objects in *A* and *B* that are mapped to the same object in *C* and whose morphisms are those morphisms in *A* and *B* that are mapped to the same morphism in *C*. This also agrees with the meaning usually attributed to this construction in the related literature.

Figure 5.19 – Instance exchange as a limit



Source: the author.

For a diagram with a world state *S*, ontologies *O* and *O’*, an instantiation functor $I{:}S \rightarrow O$ and an inclusion functor $Inc_1{:}O' \rightarrow O$, the pullback is a subcategory *S’* of *S* with an instantiation functor $I':S'\rightarrow O'$ such that S' contains every entity in S that instantiates a universal in O that is mapped from a universal in O' by Inc_1. This construction has useful properties for semantic interoperability between information systems, since it allows the exchange of instances between distinct ontologies. In particular, if O' is an ontology alignment, along with Inc_1 and another inclusion functor $Inc_2:O'\rightarrow O''$, the composition $Inc_2 \circ I':S'\rightarrow O''$ is an instantiation functor. Thus, ontology alignments allow for the transference of instances from one ontology to the other through a pullback and a composition. This entire operation is actually a limit over the diagram containing both the alignment and the instantiation functor. Figure 5.19 depicts this limit.

5.6.4 Modeling Enriched Functors

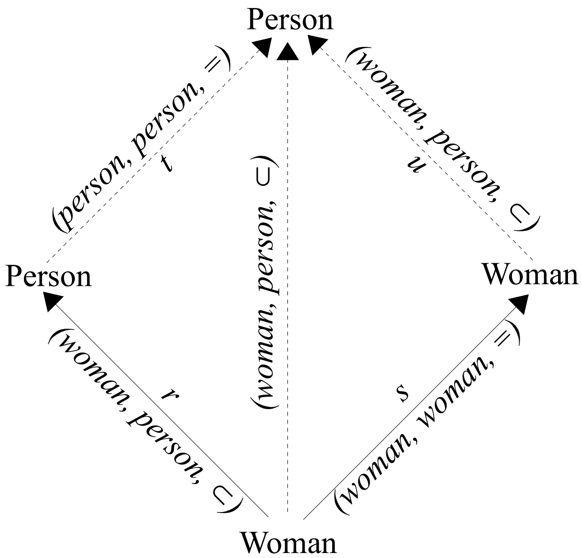
Since functors relate entities, each component of a functor may be interpreted as a conceptual relation and labeled accordingly. However, not every functor map entities in a semantically sound manner. That is, some mappings lead to absurd interpretations. We may identify semantically plausible functors $F:O_1\rightarrow O_2$ by the existence of a natural transformation $\eta:Inc_1 \rightarrow Inc_2 \circ F$, where $Inc_1:O_1\rightarrow O^{\top}$ and $Inc_2:O_2\rightarrow O^{\top}$ are inclusion functors into $O^{\top}$. For a simpler verification of this property, we may replace $O^{\top}$ by the ontology obtained through the disjoint merge of O_1 and O_2 with the addition of the components of F, as described in subsection 5.6.5.

We may then label each component F_i of the functor F with the corresponding component η_i of the natural transformation. Therefore, similarly to instantiation functors, it is possible to model specialization functors, parthood functors, or yet functors where each component is a different sort of conceptual relation. Whenever a component η_i is an identity morphism, the corresponding F_i is a conceptual equivalence. Remarkably, this is the case for every component of an inclusion functor. The composition of functors enriched in this manner is given by the composition of their respective natural transformations.

5.6.5 Merging Disjoint Ontologies

Even though the functors in *Ont* are capable of representing complex relations, as discussed in subsection 5.6.4, they are not enough to concretize the merge of disjoint ontologies through a single pushout as described by Zimmerman et al. (2006). In reality, the solution proposed in that work is also not suitable for such operations. Take, for example, the alignment composed by morphisms $r:A\rightarrow B$ and $s:A\rightarrow C$ such that r = *(woman, person, ⊂)* and s = *(woman, woman, =)*, where ⊂ means strict specialization and = means equivalence, following the original notation. With the provided definition of composition, in order for the diagram to commute, the pushout over such alignment must be given by a pair of morphisms $t:B\rightarrow B+_AC$ and $u:C\rightarrow B+_AC$ with t = *(person, person, =)* and u = *(woman, person, ⊂)* so that $t \circ r = u \circ s$ = *(woman, person, ⊂)*, as shown in figure 5.20. This is not the intended meaning for the pushout, which, according to the authors, should be an ontology containing both concepts *woman* and *person* and a specialization relation between them.

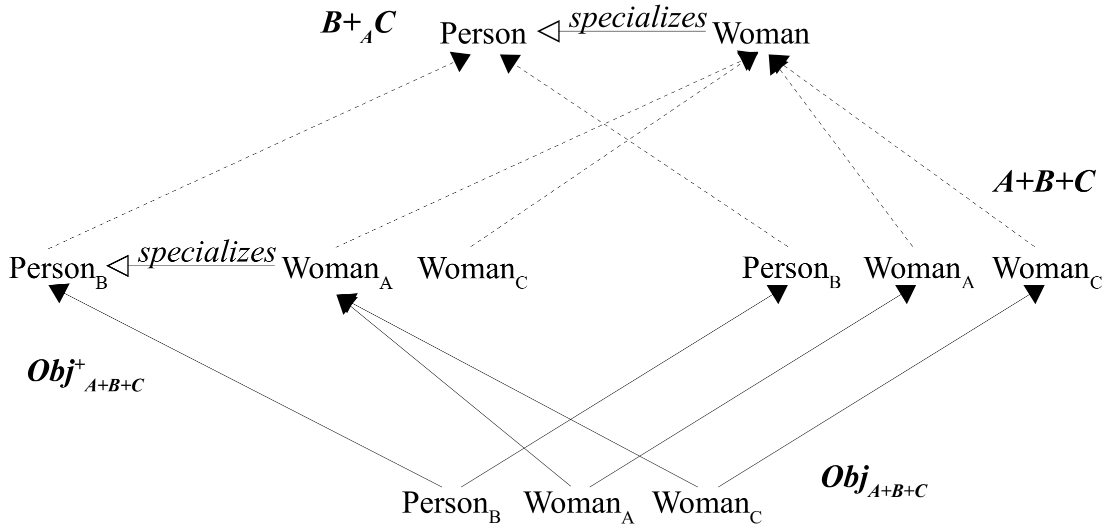
Figure 5.20 – Pushout over an alignment of disjoint ontologies



Source: the author.

Instead, a merge of two disjoint ontologies B and C aligned through functors $F:A\rightarrow B$ and $G:A\rightarrow C$ with components whose meanings are different from equivalence may be computed through a pushout over a different diagram, containing the disjoint union $A+B+C$ (given by a coproduct), a category Obj_{A+B+C} whose objects are the same ones from the disjoint union but without any non-identity morphism, a category Obj^+_{A+B+C} that is Obj_{A+B+C} enriched with the non-equivalence components of F and G and functors $Inc:Obj_{A+B+C}\rightarrow A+B+C$ mapping each object to itself and $H:Obj_{A+B+C}\rightarrow Obj^+_{A+B+C}$ mapping each object that is the image of an equivalence component of F or G to its inverse image and the remaining objects to themselves. The resulting diagram for the same previous example is given in figure 5.21. The inclusions from the original ontologies to the final merged ontology $B+_AC$ are the result of the composition $I \circ Inc_i$ of the functor $I{:}A{+}B{+}C{\rightarrow}B{+}_AC$ with the inclusions $Inc_i{:}i{\rightarrow}A{+}B{+}C$ for $i = A$, B or C.

Figure 5.21 – Pushout over the modified alignment

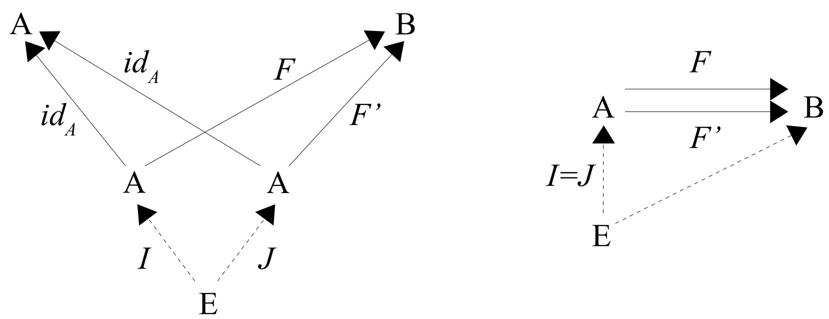


Source: the author.

The process of construction of a diagram for the disjoint merge pushout just described may be generalized for any alignment, since alignments of functors with only equivalence components will have all equivalent objects collapsed by H. Additionally, we may also apply the same process over single mappings $F{:}A{\rightarrow}B$ by removing every reference to C in the previous definition. This procedure is equivalent to exchanging G for $id_A{:}A{\rightarrow}A$, since, again, H collapses both copies of A in the disjoint union.

The operations of alignment intersection and union, defined over V-alignments, may be computed over the original alignment given by F and G before the construction of the final pushout diagram. For disjoint unions of alignments defined over single functors $F{:}A{\rightarrow}B$ and $F'{:}A{\rightarrow}B$, the intersection is given by an equalizer, as shown in figure 5.18. In order for the diagram on the left to commute, $id_A \circ I = id_A \circ J$ and consequently $I = J$. Therefore, E is an equalizer for the diagram on the right. It is not possible to find a union of alignments given by single functors, since F and F' are contradictory total mappings (unless, of course, $F{=}F'$).

Figure 5.22 – Intersection of single-functor alignments as an equalizer



Source: the author.

6 CONCLUSION

This work proposes category theory as a suitable formalism for the study and description of ontologies. First, we have shown how the modeling based on relations of ontology engineering is equivalent to the focus of category theory on morphisms. Then, we identified the issues regarding the incompatibility of the intended intensionality of ontological models with the extensional set-theoretic foundations currently accepted in ontology engineering, and addressed said issues with the use of concepts from category theory. Particularly, we have shown how the meaning of ontological concepts is better expressed through morphisms and composition rules than as a mapping function over a set indexed by possible worlds.

Following, we described how category theory and its constructions allow the formalization of notions commonly present in ontologies and ontology engineering. Among said formalizations, we defined instantiation both as a functor and as a natural transformation, supporting the description of multi-level models. We also formalized cardinalities of conceptual relations with injective functors, and some inferential processes as the application of the composition operator.

We then combined the proposed formalizations into a description of a category of ontologies, which we examined in search for common category-theoretic constructions and their meanings. The analysis has lead to new formalizations, such as disjointness of universals as the inexistence of the pullback over a diagram with specializations.

Subsequently, we discussed how the interpretation of ontologies as categories affects categories of ontologies. We have demonstrated how the category-theoretic constructions in the proposed framework have the same meaning found in the related literature, such as ontology merging as a pushout over an alignment. We have also shown how this interpretation supports semantic interoperability and the exchange of instances between distinct ontologies through pullbacks and composition. Additionally, we have described how the category-theoretic formalization of ontologies allows the merge of disjoint ontologies with the addition of new relations through the construction of a specific pushout diagram.

We have analyzed the categorical constructions in ontologies in more depth than the previous works on the subject. Additionally, most other works do not discern the extensionality issues of set theory and, consequently, fail to present category theory as an adequate alternative.

The proposed framework easily satisfies most of the guidelines for ontology morphisms defined by Antunes and Abel (2018). The third criterion, i.e., that the domain and codomain of conceptual relations should be preserved, is guaranteed by the definition of functors. Since this is the main aspect where most categories of ontologies agree, it comes without surprise that the categorical construction found here match the ones defined previously in the literature. We may enforce the first and second criteria, that is, that concepts and relations must be related injectively, by the selection of injective functors. The enriched functors defined in subsection 5.6.5 assure the fifth condition, i.e., the potential for expressing complex relations between the concepts, and additionally guarantee the semantic soundness of the resulting map as intended by the authors. Only the fourth guideline, which concerns the preservation of metaproperties, is not applicable to the framework as it currently exists. The integration of a future categorical formalization of such metaproperties into the present framework may contemplate this last criterion.

In addition to the definition of a category-theoretic formalism for metaproperties of concepts, we consider some possible paths for future research. Among them, we contemplate the scrutiny of the relation between the ontology defined as a category and its expression in an ontology representation language. Additionally, we intend to investigate category-theoretic definitions for other foundational concepts in ontology engineering, such as ontological commitment. Finally, we consider the development of an extensive and thorough analysis of the behavior of conceptual relations on composition, especially of the general composition rules for particular and universal relations (both instance-bound and otherwise), possibly with the definition of metaproperties for relations besides those of transitivity, reflexivity and symmetry.

REFERENCES

ADÁMEK, J.; HERRLICH, H.; STRECKER, G. E. **Abstract and concrete categories. The joy of cats**. New York: John Wiley and Sons, Inc, 1990.

ALIYU, S.; JUNAIDU, S. B.; KANA, A. F. D. A Category Theoretic Model of RDF Ontology. **International Journal of Web & Semantic Technology (IJWesT)**, [s. l.], v. 6, n. 3, 2015.

ALMEIDA, J. P. A.; FONSECA, C. M.; CARVALHO, V. A. A Comprehensive Formal Theory for Multi-level Conceptual Modeling. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING 2017, Valencia. **Proceedings**... Valencia: Springer, 2017.

ANTONIOU, G.; VAN HARMELEN, F. Web Ontology Language: OWL. In: **Handbook on Ontologies**. Heidelberg: Springer, 2009. p. 91–110.

ANTUNES, C.; ABEL, M. Ontologies in Category Theory: A search for meaningful morphisms. In: SEMINAR ON ONTOLOGY RESEARCH IN BRAZIL 2018, São Paulo. **Proceedings**... São Paulo

BORGIDA, ALEXANDER. Description Logics in data management. **IEEE Transactions on Knowledge and Data Engineering**, [s. l.], v. 7, n. 5, p. 671-682, 1995.

BENCH-CAPON, T.; MALCOLM, G. Formalising ontologies and their relations. In: INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS 1999, Florence. **Proceedings**... Florence: Springer, 1999.

CAFEZEIRO, I. et al. Specifying ubiquitous systems through the algebra of contextualized ontologies. **The Knowledge Engineering Review**, [s. l.], v. 29, n. 2, p. 171–185, 2014.

CAFEZEIRO, I.; HAEUSLER, E. H. Semantic interoperability via category theory. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING 2007, Auckland. **Proceedings**... Auckland: Springer, 2007.

CAFEZEIRO, I.; HAEUSLER, E. H.; RADEMAKER, A. Ontology and context. In: ANNUAL IEEE INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS 2008, Hong Kong. **Proceedings**... Hong Kong: IEEE, 2008.

CARNAP, R. **Meaning and necessity: a study in semantics and modal logic**. Chicago: University of Chicago Press, 1988.

COLOMB, R. M.; DAMPNEY, C. N. G.; JOHNSON, M. Category-theoretic fibration as an abstraction mechanism in information systems. **Acta Informatica**, [s. l.], v. 38, n. 1, p. 1–44, 2001.

CYGANIAK, R.; WOOD, D.; LANTHALER, M. RDF 1.1 Concepts and Abstract Syntax. **W3C Recommendation**, [s. l.], 2014.

GUARINO, N. Formal Ontology and Information Systems. In: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGY IN INFORMATION SYSTEMS 1998, Trento. **Proceedings**... Trento: IOS Press, 1998.

GUARINO, N.; OBERLE, D.; STAAB, S. What is an ontology? In: **Handbook on ontologies**. Heidelberg: Springer, 2009. p. 1–17.

GUARINO, N.; WELTY, C. A. An overview of OntoClean. In: **Handbook on ontologies**. Heidelberg: Springer, 2009. p. 201–220.

GUIZZARDI, G.; WAGNER, G. Using the unified foundational ontology (UFO) as a foundation for general conceptual modeling languages. In: **Theory and Applications of Ontology: Computer Applications**. Vancouver: Springer, 2010. p. 175–196.

HUME, D. **A Treatise of Human Nature**. Oxford: Clarendon Press, 1888.

JOHNSON, M.; ROSEBRUGH, R. Ontology Engineering, Universal Algebra, and Category Theory. In: **Theory and Applications of Ontology: Computer Applications**. Vancouver: Springer, 2010. p. 565–576.

LU, R.-Q. Towards a Mathematical Theory of Knowledge. **Journal of Computer Science and Technology**, [s. l.], v. 20, n. 6, p. 751–757, 2005.

PERUZZI, A. The Meaning of Category Theory for 21st Century Philosophy. **Axiomathes**, [s. l.], v. 16, n. 4, p. 424–459, 2006.

SEREMETI, L.; KOUGIAS, I. Networks of Aligned Ontologies Through a Category Theoretic Approach. **Journal of Emerging Trends in Computing and Information Sciences**, [s. l.], v. 4, n. 10, 2013.

SPIVAK, D. I.; KENT, R. E. Ologs: a categorical framework for knowledge representation. **PLoS One**, [s.l.], v. 7, n. 1, 2012.

STUDER, R. et al. Knowledge engineering: principles and methods. **Data and knowledge engineering**, [s. l.], v. 25, n. 1, p. 161–198, 1998.

TRNKOVÁ, V. Sum of categories with amalgamated subcategory. **Commentationes Mathematicae Universitatis Carolinae**, [s. l.], v. 6, n. 4, p. 449–474, 1965.

XI, H. **Dependent types in practical programming**. 1998. Thesis (Doctorate of Philosophy) – Departament of Mathematical Sciences, Canergie Mellon University, Pittsburgh, 1998.

ZIMMERMANN, A. et al. Formalizing ontology alignment and its operations with category theory. In: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGY IN INFORMATION SYSTEMS 2006, Baltimore. **Proceedings**... Baltimore: IOS Press, 2006.