DOI: 10.2478/ijasitels-2024-0014

A Categorical Formalism for Conceptual Graphs

Daniel C. Crăciunean[1]

[1]*Computer Science and Electrical and Electronics Engineering Department, Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania*
daniel.craciunean@ulbsibiu.ro

Abstract

In this paper, we propose a formalism for knowledge manipulation, based on conceptual graphs, which is supported by category theory, and which can be the basis for the development of knowledge-based systems. The model we propose is an extension of the model based on conceptual graphs. In this model, the application of an inference rule between logical formulas represented by two conceptual graphs is reduced to the identification of an arrow in a category. To this end, we introduce several new notions such as: the category of conceptual graphs, the category of classes of conceptual graphs, the conceptual category of a model and the conceptual category of inference of a model.

Keywords: conceptual graphs, coreference relation, category theory, subsumption functor, category of conceptual graphs, conceptual category

1 Introduction

In order to model logical inference through algorithmizable and accessible mechanisms, in the 1970s, a series of graphical languages [6] were developed to specify first-order logic. In this context, in 1976 Sowa [11] introduced a version of a graphical language for specifying knowledge, questions, and statements in natural language in terms representable in a relational database, called conceptual graphs.

The graphical atoms of the language are the rectangles that represent concepts, the circles that represent conceptual relations, and the edges that represent the connecting elements between concepts and relations.

A conceptual graph is a bipartite graph such that the concepts neighbouring a conceptual relation node are always nodes that represent concepts. The conceptual relations in a conceptual graph represent atomic formulas in first-order logic, and the neighbouring concepts represent the arguments of these atomic formulas. To specify the order of these arguments, the edges connecting a conceptual relation and its neighbouring concepts are marked with natural numbers; 1, . . . , n , where n is the number of neighbouring concepts.

In this paper, we propose a formalism for knowledge manipulation, based on category theory, which can be the basis for the development of knowledge-based systems [9, 13]. The model we propose is an extension of the model based on conceptual graphs [11].

In this model, the application of an inference rule between logical formulas represented by two conceptual graphs reduces to identifying an arrow in a category.

The main novelty of this paper is the formalization of conceptual graphs and reasoning based on conceptual graphs using category theory as support. For this purpose, we introduce notions such as: category of conceptual graphs, category of classes of conceptual graphs, conceptual category of the model and conceptual category of model inference.

Section 2 of the paper contains Background Notions, section 3 introduces the category of conceptual graphs and the category of classes of conceptual graphs, section 4 introduces the conceptual category of the model and the conceptual category of model inference. The paper ends with conclusions and observations.

2 Background Notions

Conceptual graphs are specified by a graphical language and represent different types of knowledge such as: facts, objectives, rules and queries [1]. Facts are statements about the existence of entities, about the properties of an entity or about the relationships between them. Objectives represent the goal pursued by an evolving system. Rules can describe knowledge and constraints, implicit in the model, as well as the evolution of processes.

A conceptual graph is a bipartite multigraph, whose nodes represent concepts and conceptual relationships. The edges of the multigraph represent the connection between the two types of nodes.

Each conceptual node is represented by a pair of labels, one specifying the type of concept and one specifying an individual of the type specified by the first label. If the individual label is missing, they will be represented by a variable. Conceptual relationship nodes are marked with labels that represent types of relationships. The edges, in turn, are labelled with natural numbers in the order in which the neighbouring concepts will become the parameters of these conceptual relationships.

These labels with which a conceptual graph is endowed form a vocabulary, which is denoted by $T=(T_C, T_R, E)$ where:

T_C is a set of concept types,

T_R is a set of relationship types,

E is the set of individual labels.

The connection between a conceptual graph and the associated vocabulary is made by an application that distributes the vocabulary to the components of the graph.

Therefore, a conceptual graph G, is a tuple (T, G, $\boldsymbol{\eta}$), where T is a vocabulary, $G=(C,R,\boldsymbol{\Gamma})$, is a bipartite multigraph, and $\eta=(\eta_C,\eta_R,\eta_\Gamma)$ is the application that distributes the vocabulary to the components of the graph, thus: $\eta_C:C\rightarrow T_C\times (E\cup\{*\})$, $\eta_R:R\rightarrow T_R$, and $\eta_\Gamma:\Gamma\rightarrow N$.

The basic support for inference based on conceptual graphs consists of two relations defined on both the set of concepts and the set of relations [1]. The first relation is the coreference relation which is a total equivalence relation and which we denote by $\boldsymbol{\rho}$. The second relation is the generalization relation which is a partial order relation and which we denote by ≥. Based on the generalization relation we introduce the subsummation functors that define logical deduction.

In this paper we will use category theory to formalize knowledge representation and reasoning based on conceptual graphs [12]. A category is a mathematical structure composed of a set of objects and a set of arrows between these objects, to which is added an operation of composing the arrows. The composition operation is associative and with neutral elements.

3 The Category of Conceptual Graphs

A subsummation homomorphism φ is a mapping from a conceptual graph G to a conceptual graph H, which:

- maps each concept in G to a concept in H and each relation in G to a relation in H,
- preserves the conceptual graph structure, that is, maps each edge (A, B), which connects a pair of nodes A and B in G, to an edge (φ(A), φ(B)), which connects nodes φ(A) and φ(B) in H,
- φ is a monotonically decreasing mapping, that is, $\forall e \in C_G \cup R_G$, $\eta_G(e) \geq \eta_H(\varphi(e))$.

We can define the composition of two subsummation homomorphisms as follows:

Let G, H and K be three conceptual graphs, and φ:G →H and ψ:H→K, then the subsummation homomorphism resulting from the composition of φ with ψ is $\delta=\psi(\varphi)$, δ:G→H. It is obvious that δ is a subsumption homomorphism.

If G, H, K, S are conceptual graphs and φ :G →H, ψ: H → K, μ:K→S, then obviously $(\varphi\circ\psi)\circ\mu=\varphi\circ(\psi\circ\mu)$:G→S, and therefore the operation of composing subsumption homomorphisms is associative.

For every conceptual graph G , there exists the identity subsumption homomorphism: ι:G→G, which takes each component of G into the same component.

Therefore, the set of conceptual graphs together with the set of subsumption homomorphisms, among them, form a category, in which the objects are conceptual graphs and the arrows are the subsumption morphisms. We call this construction, the category of conceptual graphs.

The ultimate goal of conceptual graphs is to allow the specification in natural language of different types of knowledge such as: facts, objectives, rules and queries. But natural language is flexible, and allows the use of multiple concepts for the same real entity, at least in terms of the vocabulary used.

To solve this problem, we will introduce an equivalence relation on the set of concepts that we call the coreference relation. In [8], the coreference relation is defined on the set of conceptual nodes. We introduce this relation on the set of concepts that could contain both multiple concepts representing the same real entity and multiple names for the same concept.

The same thing happens with the set of conceptual relations, ultimately the components, called conceptual relations, of the conceptual graphs are also concepts. Natural language also allows a multimer of syntactic formulas for expressing them. It follows that the coreference relation will have to refer to these as well.

Therefore, the coreference relation that we use here has two components, one that refers to the coreference of concepts and one that refers to the coreference of conceptual relations.

We denote the coreference relation with $\rho=(\rho_C,\rho_R)$, where: ρ_C, groups all syntactic forms that represent an entity in a set of concepts that we call the class of coreferent concepts, and ρ_R, groups all syntactic forms that represent a conceptual relation in a set of conceptual relations that we call the class of coreferent conceptual relations.

If C is a set of concepts, and R is a set of conceptual relations, we define the coreference relation $\rho=(\rho_C,\rho_R)$ as follows: $\rho_C \subseteq C \times C$, $(c_1,c_2) \in \rho_C$, if and only if c_1 and c_2 refer to the same real entity; $\rho_R \subseteq R \times R$, $(r_1,r_2) \in \rho_R$, if and only if r_1 and r_2 refer to the same real conceptual relation.

Next, we will define the notion of a class of conceptual graphs, in which each conceptual node will represent a class of coreferent concepts and each node, conceptual relation, will represent a class of coreferent conceptual relations.

If we have a conceptual graph G, then the class of conceptual graphs associated with G, which we denote by G*, is the image of the functor ϕ:G→Rel, where Rel is the category of sets and relations, with the properties:

- For each conceptual node $A \in G_C$, $\phi(A)$ is the set of all coreferent concepts with A.
- For each node, thenconceptual relation $R \in G_R$, $\phi(R)$ is the set of all coreferent conceptual relations with R.
- For each edge, (R,A) $R \in G_R$, $A \in G_C$, $\phi((R,A))$ is the set of all total relations, $\boldsymbol{\rho} \subseteq R \times A$.

We observe that the conceptual graph G, plays the role of a categorical sketch graph for a class of conceptual graphs, and therefore can serve us to impose various conditions on the structure of the graph [3, 4], but we will not address this issue in the present paper.

If G*, is a class of conceptual graphs then we will define the operation: G=Slice(G*)=G*/$\boldsymbol{\rho}$, thus: $G_C=G_C^*/\rho_C$, $G_R=G_R^*/\rho_R$, and the edges between relations and concepts will be those that exist between the concepts and conceptual relations selected as representatives in the Slice(G*) operation. We observe that the graph G=Slice(G*) is a conceptual graph.

To define subsummation homomorphisms between two classes of conceptual graphs, we will overload the ≥ relation, for coreferent concept classes and coreferent conceptual relation classes, as follows:

- If A and B are two coreferent conceptual classes then A≥B, if and only if $\forall$ $a \in A$ and $\forall$ $b \in B$, $\Rightarrow a \geq b$,
- If R and P are two coreferent conceptual relation classes then R≥P, if and only if $\forall$ $r \in R$ si $\forall$ $p \in P$, $\Rightarrow r \geq p$.

We can now define a subsummation homomorphism φ^*, between two classes of conceptual graphs, as a map from a class of conceptual graphs G*, to a class of conceptual graphs H*, which:

- maps each conceptual class in G*, to a conceptual class in H*, and each relation in G* to a relation in H*,

- preserves the class structure of conceptual graphs, that is, maps each edge (A*,B*), connecting a pair of nodes A* and B* in G, to an edge (φ(A^*), φ(B^*)), connecting nodes φ(A^*) and φ(B^*) in H*,
- φ is a monotonically increasing map, that is, ∀e ∈ $C_{G^*}^* \cup R_{G^*}^*$, η_{G^*}(e)≥ η_{H^*}(φ(e)).

We can define the composition of two subsummation homomorphisms as follows:

Let G, H, and K be three classes of conceptual graphs, and φ:G→H and ψ:H→K, then the subsummation homomorphism resulting from the composition of φ with ψ is δ=ψ(φ), δ:G→H. It is obvious that δ is a subsummation homomorphism.

If G, H, K, S are classes of conceptual graphs and φ :G →H, ψ: H → K, μ:K→S, then obviously (φ∘ψ)∘μ=φ∘(ψ∘μ):G→S, and therefore the operation of composing subsummation homomorphisms is associative.

For every conceptual graph G, there exists the identity subsummation homomorphism: ι:G→G, which takes each component of G to the same component, through the identity function.

Therefore, the set of classes of conceptual graphs together with the set of subsummation homomorphisms between them form a category, in which objects are classes of conceptual graphs and arrows are subsummation morphisms between them. We call this construction the category of classes of conceptual graphs and denote it by CCGC.

If $\mathcal{C}$*, is the category of classes of conceptual graphs then we will define the operation: $\mathcal{C}$=Slice($\mathcal{C}$*)= $\mathcal{C}$*/**ρ**, thus: ob($\mathcal{C}$)= ob($\mathcal{C}$)/ρ_C , arrow($\mathcal{C}$)=arrow($\mathcal{C}$*)/ρ_R. It can be easily demonstrated that Slice($\mathcal{C}$*) is a category. We note that this category is not isomorphic to the category of conceptual graphs because the latter also contains conceptual graphs with coreferent nodes. We will call the category Slice($\mathcal{C}$*), the inference category of conceptual graphs and denote it by CIGC.

4 Categorical Knowledge Modelling

The process of modelling knowledge in a domain begins with the conceptualization of the domain in question [2]. In knowledge modelling, concepts can represent real entities or can be generic in which case they represent classes of entities. Conceptualizing a domain of knowledge involves identifying the real entities involved in this domain and the relationships between these entities and replacing them with concepts that represent them.

Therefore, the conceptual metamodel of a domain of knowledge is a set of concepts that allow the specification of knowledge of interest in the domain in question [5].

The entities and the relationships between them, involved in the domain, must be replaced with concepts. Concepts must be introduced through clear and precise definitions. Defining a concept involves the precise delimitation of a family of real objects, which it is to represent, from a larger, already known family, by adding new properties, common to all objects of the new concept and only it. The generic name given to the new family of objects is the name of the defined concept. Therefore, each newly defined concept retains all the properties of the concepts on the basis of which it was defined, to which the new properties are added.

As we mentioned, to define a new concept, we must rely on another already defined one. But initially, when we do not have any defined concept, we must start from primary concepts, which we do not define, or from concepts defined in other models. This way of defining concepts provides us with a natural hierarchy of these concepts. Therefore, we can introduce the relation $\geq$, on the set of concepts, which we call the generalization relation. Obviously, this relation is a partial order relation.

If C, is a set of contexts and $\geq$, is a generalization relation then it respects the properties of:

- reflexivity : $\forall$ c$\in$C, c $\geq$ c;
- transitivity : $\forall$ c,d,e$\in$C, c $\geq$ d and d $\geq$ e $\Rightarrow$c $\geq$ e;
- antisymmetry : if e $\geq$ f and f $\geq$ e $\Rightarrow$ e = f.

It follows that the relation $\geq$ is a partial order relation. This relation is the basic support for logical inference in conceptual graphs. It is obvious that if c, d are concepts and c$\geq$d, then any logical formula satisfied, on the properties of c, is also satisfied for the properties of d. But conceptual relations are also concepts and therefore the generalization relation $\geq$ can be overloaded for conceptual relations.

The concepts involved in knowledge modelling, most of the time, represent classes of entities and therefore are generic. Generic concepts can have potential properties and can undergo potential transformations depending on certain contexts, which can also be potential.

4.1 Conceptual Category of Model Inference

Suppose that, for a specific domain, we have identified all the concepts involved in the representation of knowledge and all the conceptual relations between them, including the lexical varieties that represent them. We denote the set of these concepts by C and the set of conceptual relations by R. On the set C, we introduce the coreference relation $\boldsymbol{\rho}$. The coreference relation $\boldsymbol{\rho}$ is an equivalence relation and therefore partitions the set C into a set Č of classes of concepts, Č=C/$\boldsymbol{\rho}$, and the set R into a set of classes of conceptual relations, Ř=R/$\boldsymbol{\rho}$.

We saw in Section 3 that the set of classes of conceptual graphs together with the set of subsummation homomorphisms between them form a category, which we called the category of classes of conceptual graphs and denoted it by CCGC.

We consider the set of all classes of conceptual graphs, which have as conceptual nodes, elements from Č, and as relational nodes, elements from Ř, and we denote this set of classes of conceptual graphs by CCG(Č,Ř).

We will now construct a subcategory of the category CCGC. For this we will observe that if we have a category $\mathcal{C}$, and a set of objects O, we can define a subcategory $\mathcal{D}$, of $\mathcal{C}$, which has ob($\mathcal{D}$)=O, $\mathcal{D}$(X,Y)=$\mathcal{C}$(X,Y), where X,Y$\in$O , with the identity and composition in $\mathcal{C}$.

Based on this observation, we will now construct a subcategory of the category CCGC, which we call the conceptual category of the model (CCM), in the following way:

The set of objects of the category CCM is: ob(CCM)=ob(CCGC)$\cap$CCG(Č,Ř)

The set of arrows of the category CCM is: arrow(CCM)={CCGC(X,Y)|X,Y$\in$ CCG(Č,Ř).

It is easy to prove that the CCM construction is a category. The objects of this category are classes of conceptual graphs, which model the specific domain, and the arrows of the category are subsummation homomorphisms that model the logical inference between these classes of conceptual graphs.

Starting from the CCM category we can construct the conceptual category of inference of the model, which we denote with CCIM= Slice(CCM)=CCM/**ρ**. The CCIM category has the components: ob(CCIM)=ob(CCM)/**ρ**, arrow(CCIM)=arrow(CCM)/**ρ**. Obviously the CCIM category is a subcategory of the category of inference of conceptual graphs (CIGC).

4.2 Logical Inference in the Conceptual Category of Model Inference

The terms, in first-order logic, for a domain are constants, variables and functions that model certain phenomena specific to the domain [10]. An atomic formula, in the domain, is a predicate that has as parameters terms specific to the domain. Formulas are built on the basis of atomic formulas by introducing logical operators.

The diagrammatic language, used to specify conceptual graphs, can be naturally associated with the language of first-order logic, as follows:

- Individual labels of conceptual nodes become constants.
- Unspecified individual labels of conceptual nodes become variables.
- Conceptual node type labels become unary predicates, which have as parameters the individual label associated with the conceptual node.
- Conceptual relation node labels become n-ary predicates that have as parameters the type labels of neighbouring conceptual nodes.

In the following, we will use the same names for logical constants and predicates as those in the vocabulary of the conceptual graph, that is, the logical constant c will represent the individual label c and the logical predicate p will represent the type of concept or conceptual relation p. The individual labels that are not specified, we will denote with $x_1,x_2,\ldots,x_n$, and will represent the variables of the logical formula associated with a conceptual graph.

In this way, to each conceptual graph G, we can attach:

- A logical formula corresponding to the conceptual nodes:

$\Phi(G_E)=\exists\ x_1,\ldots x_n \bigwedge_{p\in E} p(e)$, where e is the individual label of the concept type p or the corresponding variable x_i, if the individual type is not specified.

- A logical formula corresponding to the nodes of conceptual relations:

$\Phi(G_R)=\exists\ x_1,\ldots x_n \bigwedge_{r\in R} r(t_{1,\ldots,}t_k)$ where $t_1,..,t_k\in\{x_1,\ldots,x_n\}$

- The logical formula [8], which represents the semantics of the conceptual graph G is:

$\Phi(G)= \Phi(G_E) \wedge \Phi(G_R)$.

Thus, the semantics of conceptual graphs is given by logical formulas of order 1 [8]. We note that these formulas use only the universal logical quantifier ∃, and the logical connector ∧, and, therefore, represent a subset of logical formulas of order 1. We denote by FOL(∧, ∃). We also denote by FOL(∧, ∃, CCIM), the set of logical formulas that represent the semantics of all conceptual graphs that are objects in the conceptual category of model inference (CCIM).

It is shown that for any logical formula in FOL(∧, ∃), a conceptual graph can be constructed that has this formula as its semantics [8]. Therefore, for any formula in FOL(∧, ∃), the corresponding conceptual graph can be constructed and conversely for any conceptual graph G, the formula in FOL(∧, ∃), which represents the semantics of G, can be constructed.

In our categorical model, subsummation homomorphisms are the fundamental ingredient for inference based on conceptual graphs. Inference in the CCIM category is based on the following proposition:

If G and H are objects in the CCIM category, then $\Phi(G) \vdash \Phi(H)$, if and only if CCIM(G,H)∈arrow(CCIM), that is, if and only if there is a subsummation homomorphism from the conceptual graph G to the conceptual graph H.

If we want to verify an implication of the type $\phi \vdash \psi$, where $\phi, \psi \in$ FOL(∧, ∃, CCIM), we will construct the conceptual graphs CG(ϕ) and CG(ψ), and we will verify whether CCIM(CG(ϕ),CG(ψ))∈arrow(CCIM).

5 Observations and conclusions

We note that the CCIM category is a finite category, and therefore it can be constructed algorithmically. The objects of the CCIM category are classes of concepts ob(CCIM)=$\check{C}\cup\check{R}$. It follows that the model can be permanently enriched with new concepts or new syntactic forms, without modifying this category if they can be included in the existing classes.

Adding concepts that are not coreferential with any existing class requires transforming the category, cases that are not rare enough if the model is well designed. In this case, graph transformation mechanisms must be used [7].

In our categorical model, subsummation homomorphisms are the fundamental ingredient for inference based on conceptual graphs. Finding a homomorphism between two graphs is, in general, an NP problem if the graph of the domain of definition is not acyclic. Therefore, this problem, which was not addressed in this paper, must be treated seriously. These are just a few problems related to this model, which we will treat in future works.

References

[1] Jamie Caine, Simon Polovina, ”From Enterprise Concepts to Formal Concepts: A University Case Study, in Graph Structures for Knowledge Representation and Reasoning”, 5th International Workshop, GKR 2017 Melbourne, VIC, Australia, August 21, 2017, Revised Selected Papers, Lecture Notes in Artificial Intelligence.

[2] Daniel-Cristian Craciunean, Daniel Volovici, “Conceptualization of Modeling Method in the Context of Categorical Mechanism”, in Dimitris Karagiannis and others, Domain Specific Conceptual Modeling, Springer Nature Switzerland AG (2022).

[3] Daniel-Cristian Craciunean, “Categorical Mechanisms in Multi-level Modeling Methods”, Publishing House of the "Lucian Blaga" University of Sibiu, (2023).
[4] Daniel-Cristian Crăciunean, D. Karagiannis, “A categorical model of process cosimulation”, Journal of Advanced Computer Science and Applications (IJACSA), 10, (2019)
[5] D. Karagiannis, H.C. Mayr, J. Mylopoulos, “Domain-Specific Conceptual Modeling Concepts, Methods and Tools”, Springer International Publishing Switzerland (2016)
[6] Dominik Bork , Dimitris Karagiannis, Benedikt Pittl, “A survey of modeling language specification techniques”, Information Systems 87 (2020) 101425, journal homepage: www.elsevier.com/locate/is
[7] Hartmut Ehrig, Claudia Ermel, Ulrike Golas, Frank Hermann, “Graph and Model Transformation General Framework and Applications”, Springer-Verlag Berlin Heidelberg (2015)
[8] Michel Chein , Marie-Laure Mugnier, “Graph-based Knowledge Representation, Computational Foundations of Conceptual Graphs”, Springer-Verlag London (2009).
[9] Michael Barr, Charles Wells, “Category Theory For Computing Science”, Reprints in Theory and Applications of Categories, No. 22, (2012).
[10] M. Huth, M. Ryan, “Logic in Computer Science, Modelling and Reasoning about Systems”, Published in the United States of America by Cambridge University Press, New York, (2004).
[11] Sowa, John F., “Conceptual graphs for a database interface”, IBM Journal of Research and Development **20:4**, 336-357, (1976).
[12] Zinovy Diskin, Uwe Wolter, “A Diagrammatic Logic for Object-Oriented Visual Modeling”, Electronic Notes in Theoretical Computer Science, Volume 203, Issue 6, 21 November (2008).
[13] Uwe Wolter, Zinovy Diskin, “The Next Hundred Diagrammatic Specification Techniques, A Gentle Introduction to Generalized Sketches”, 02 September (2015), https://www.researchgate.net/publication /253963677.