Analysis of the Metagraph Data Model in Terms of Metagraph Operations and Category Theory

Stepan Vinnikov, Anatoly Nardid, and Yuriy Gapanyuk

Bauman Moscow State Technical University, Moscow, Russia
vinnikovss@student.bmstu.ru, nardid@bmstu.ru, gapyu@bmstu.ru

Abstract. A metagraph data model is a type of complex graph model designed to describe complex subject areas. Category theory is a well-known and well-researched mathematical apparatus that studies the properties of relations between various mathematical objects that do not depend on the internal structure of objects. In this article, we make an attempt to analyze the metagraph data model in terms of category theory. Operations over the metagraph are introduced and the properties of operations are considered in detail. The category of metagraphs "MetGr" is proposed, which is a category in which the object class is the set of metagraphs. Morphisms in such a category are based on a union operation. The proposed operations over the metagraph and the category of metagraphs "MetGr" is a necessary basis for further development of a rewriting system on metagraphs.

Keywords: Complex graph · Metagraph · Metavertex · Category theory · Category · Morphism

1 Introduction

There are countless different systems of various degrees of complexity in the world, the implementation and maintenance of which would be impossible or extremely difficult without their preliminary modeling.

Modeling allows, on the one hand, to analyze various characteristics and properties of the object of study without significant resource costs, and on the other hand, to predict the behavior of the object with changes in both its parameters and structure.

One of the most commonly used tools for modeling systems and subject areas is graph theory. This type of modeling is called graph structural. A special case of such modeling is modeling with metagraph data model.

The metagraph model is a type of complex graph model designed to describe complex subject areas. The model was originally proposed by A. Bazu and R. Blanning in the monograph [1] and subsequently received a number of extensions independently proposed by different groups of researchers. In this article, the metagraph model is used in the form of annotated metagraph model proposed in [2].

The structure of the absolute majority of existing systems and subject areas is dynamic, that is, it changes over time – new components and connections are added, old ones are changed or removed. This process of changing the system is called its evolution.

The evolution of the system leads to the need for the evolution and models of this system, since they must reflect its current state. This statement is true, in particular, for the case when the model is described using metagraphs.

There are techniques that allow to work with graph transformations formally. One such technique is graph rewriting [3]. This approach is based on category theory – mathematical theory that studies the properties of relations between various mathematical objects. The same approach can be developed for metagraphs, but for this, first of all, it is necessary to consider operations over metagraph and describe metagraph as a category.

Currently, knowledge-based systems often use a Semantic Web approach. In particular, the RDF triple model and the OWL ontology description language are used [4]. Descriptive logics [5] are used as a basis for the formal-logical description of Semantic Web technologies. At the same time, paper [6] shows the fundamental difference between the RDF data structure and the metagraph model. Because of this, the results obtained in the logical study of Semantic Web technologies cannot be directly applied to the metagraph model.

Thus, the main goal this article is the creation of a formal apparatus that can later be used as a basis for development of a rewriting system on metagraphs. As such a formalization, we propose a way of defining the operations over metagraph and representing metagraph data model as a category.

The article is organized as follows. In section two, a formal definition of the metagraph model is given, operations on the metagraph are introduced and their properties are considered. In section three the category of metagraphs "MetGr" is introduced and its properties are considered. In section four the example of the application of the described category is given.

2 Operations Over Metagraphs

2.1 Definitions of the Metagraph Structures

According to [2] metagraphs and their internal structures are described as follows:

Definition 1. Metagraph *is a set* $MG = \{MV, E\}$ *where* MG *is a metagraph,* MV *is a set of metavertices of the metagraph,* E *is a set of edges of the metagraph.*

Definition 2. Empty metagraph *is a metagraph that has both empty* MV *and* E *sets. Empty metagraph is denoted as* $\varnothing = \{\varnothing, \varnothing\}$

Definition 3. Metavertex *is a set* $MV = \{id, mg\}$ *where* MV *is a metavertex,* id *is the identifier of the metavertex,* mg *is a nested metagraph of the metavertex,* mg *can be an empty metagraph.*

Definition 4. Edge *of the metagraph is a set* $E = \{id,\ (id_{begin}, id_{end})\}$ *where* E *is an edge,* id *is the identifier of the edge,* id_{begin} *is the identifier of the initial metavertex (the beginning of the edge),* id_{end} *is the identifier of the terminal metavertex (the ending of the edge).*

2.2 Theory of Categories Definitions

In this section and beyond, the definitions and properties of the elements of category theory are given, based both on the classical books of MacLane and Awodey ([7,8]) and on the works of contemporary authors ([9,10,11]).

Definition 5. *Category* $\mathbb{C}$ *consists of:*

1. *Class of objects* $Ob_{\mathbb{C}}$. *The fact that* X *is an object of* C *is written as* $X \in \mathbb{C}$*;*
2. *Sets* $hom\,(X, Y)$ *of morphisms from* X *to* Y *for each* $X \in \mathbb{C}$, $Y \in \mathbb{C}$.

The following axioms hold for any category:

1. *For* $\forall X \in Ob_{\mathbb{C}}$ *there exists an identity morphism* $1_X : X \to X$
2. *Morphisms can be composed: if there is* $f : X \to Y \in$ *and* $g : Y \to Z$ *then there exists their composition* $g \circ f : X \to Z$.
3. *Identity morphism is a neutral element for the composition operation:* $f \circ 1_X = f = 1_Y \circ f$ *where* $f \in hom\,(X, Y)$.
4. *Composition operation is associative:* $(h \circ g) \circ f = h \circ (g \circ f)$ *for arbitrary* f, g, h.

To apply the apparatus of category theory to the metagraph data model, it is necessary to determine the category carrier class and morphisms in this category. One of the possible categories is a category in which the class will be the set of all possible metagraphs, and the morphisms will be based on the union operation on the set of metagraphs. It is necessary to prove that sets with such morphisms are indeed categories.

2.3 Definition of the Operations

Let's describe operations and relations of metagraphs and metavertices:

1. Metagraph union:

$$mg_1 \cup mg_2 = mg_3 = \{MV_{mg_1} \cup MV_{mg_2},\ E_{mg_1} \cup E_{mg_2}\}, \tag{1}$$

   for $\forall\, mg_1, mg_2 \in MG$.
2. Metavertex union:

$$mv_1 \cup mv_2 = \begin{bmatrix} \{id, mg_{mv_1} \cup mg_{mv_2}\} \;\; if \;\; id_{MV_1} = id_{MV_2} \\ \{mv_1, mv_2\} \;\; if \;\; id_{mv_1} \neq id_{mv_2} \end{bmatrix}, \tag{2}$$

   for $\forall\, mv_1, mv_2 \in MV$.

3. Binary relation [12] $\subseteq$:

$$mg_1 \subseteq mg_2 \iff (mg_1 = \varnothing) \vee (\forall mv_i \in MV_{mg_1} \Rightarrow \exists! mv_j \in MV_{mg_2} : mv_i \subseteq mv_j \ \wedge \ (E_{mg_1} \subseteq E_{mg_1})), \tag{3}$$

for $\forall mg_1, mg_2 \in MG$.

$$mv_1 \subseteq mv_2 \iff (id_{mv_1} = id_{mv_2}) \wedge (mg_{mv_1} \subseteq mg_{mv_2}), \tag{4}$$

for $\forall mv_1, mv_2 \in MV$.

Let's note the special properties of metagraphs and metavertices:

1. Empty metagraph is a neutral element of the union operation:

$$\varnothing \cup mg = \{\varnothing \cup MV_{mg}, \varnothing \cup E_{mg}\} = \{MV_{mg}, E_{mg}\} =$$
$$\{MV_{mg} \cup \varnothing, E_{mg} \cup \varnothing\} = mg \cup \varnothing = mg$$

2. Each metavertex can be represented as a metagraph:

$$mv = \{\{mv\}, \varnothing\} \in MG, \forall mv \in MV$$

3. The MV structure can be represented as a MV' structure:

$$MV' = \{id, \{mv\}, \{e\}\},$$

and there is a bijection [12] between MV and MV':

$$f : MV \leftrightarrow MV'$$
$$f(mv) = \{id, \{mv\}_{mg_{mv}}, \{e\}_{mg_{mv}}\} \in MV'$$
$$f^{-1}(mv') = \{id, \{\{mv\}_{mv'}, \{e\}_{mv'}\}\} \in MV$$

Note that the MG and MV structures can be expanded with one or several additional structures, such as the string:

$$MG_{str} = \{MV,\ E,\ s\},$$
$$MV_{str} = \{id,\ mg,\ s\},$$

where s is the string.

Then it will be necessary to describe $\cup$ operation and $\subseteq$ relation for these additional structures. For example, $str_1 \cup str_2 = concatinate(str_1, str_2)$, and $str_1 \subseteq str_2 = str_1 \preccurlyeq str_2$ where $\preccurlyeq$ is a lexicographic order.

2.4 Property of the Union Operation

Operation $\cup$ is associative on MG and MV, if all the additional structures of MV and MG also have this property:

$$(mg_1 \cup mg_2) \cup mg_3 = mg_1 \cup (mg_2 \cup mg_3), \forall mg_1, mg_2, mg_3 \in MG \tag{5}$$

$$(mv_1 \cup mv_2) \cup mv_3 = mv_1 \cup (mv_2 \cup mv_3), \forall mv_1, mv_2, mv_3 \in MV \tag{6}$$

For the proof of the 5-6 properties, we introduce a special algebraic structure "simple recursive structure" for which we will prove the above properties.

2.5 Simple Recursive Structure Definition and Properties

One of the remarkable structural properties of metagraphs and metavertices is their nesting property, and such nesting is recursive in nature.

The simplest algebraic structure with such property is a simple recursive structure.

Let ID be a set of identifiers. ID can be any set, for example, the set of natural numbers $\mathbb{N}$.

Definition 6. Simple recursive structure *S is a set $S = \{id, \{s\}\}$ where $id \in ID$ is an identifier of the structure instance, $\{s\}$ is a nested set of the structure instances $s \in S$.*

Let's note the special properties of simple recursive structures:

1. Each element of a set of simple recursive structures has a unique identifier:
$$s_i \in \{s\} \Rightarrow \nexists s_j \in \{s\} : s_j \neq s_i \wedge id_{s_j} = id_{s_i}$$
2. There is an empty set of simple recursive structures which is neutral to the union operation and is a subset of any set:
$$\varnothing = \{\}$$
$$\{s\} \cup \varnothing = \varnothing \cup \{s\} = \{s\}$$
$$\varnothing \subseteq \forall \{s\}$$
3. If an instance of the simple recursive structure is an element of the set then a set, containing only this element is a subset of this set:
$$s_1 \in \{s_1, s_2\} \Rightarrow \{s_1\} \subseteq \{s_1, s_2\} \Rightarrow s_1 \cup \{s_1, s_2\} = \{s_1, s_2\}$$

Now we will describe operations and relations of simple recursive structures:

1. Binary relation "=":
$$\begin{aligned} s_1 = s_2 \iff & (id_{s_1} = id_{s_2}) \wedge (|\{s\}_{s_1}| = |\{s\}_{s_2}|) \wedge \\ & ((\{s\}_{s_1} = \{s\}_{s_2} = \varnothing) \vee (\forall s_i \in \{s\}_{s_1} \Rightarrow \exists! s_j \in \{s\}_{s_2} : s_i = s_j)), \end{aligned} \tag{7}$$
for $\forall s_1, s_2 \in S$.
2. Binary relation "$\subseteq$":
$$\begin{aligned} s_1 \subseteq s_2 \iff & (id_{s_1} = id_{s_2}) \wedge \\ & ((\{s\}_{s_1} = \varnothing) \vee (\forall s_i \in \{s\}_{s_1} \Rightarrow \exists! s_j \in \{s\}_{s_2} : s_i \subseteq s_j)), \end{aligned} \tag{8}$$
for $\forall s_1, s_2 \in S$.

Remark 1. By definition, a binary relation R over sets X and Y is a subset R of the Cartesian product $X \times Y$. We will denote $xRy = True$ if $(x, y) \in R$.

Note the property $s \subseteq s, \forall s \in S$ without proof

3. Binary operation "$\cup$":

$$s_1 \cup s_2 = \begin{bmatrix} \{s_1, s_2\} \;\; if \;\; id_{s_1} \neq id_{s_2} \\ \{id, \{s\}_{s_1} \cup \{s\}_{s_2}\} \;\; if \;\; id_{s_1} = id_{s_2} \end{bmatrix}, \forall s_1, s_2 \in S, \tag{9}$$

where $\{s\}_{s_1} \cup \{s\}_{s_2} = \{s_i \cup s_j\}, \forall s_i \in \{s\}_{s_1}, \forall s_j \in \{s\}_{s_2}$.

4. Function $h(s) \in \mathbb{N} \cup \{0\}$, that describes the nesting depth $s \in S$ and defined as follows:

$$h(s) = \begin{bmatrix} 0 \text{ if } \{s\}_s = \varnothing \\ 1 + h(\{s\}) \end{bmatrix}, \tag{10}$$

$\forall s \in S$, where $h(\{s\}) = max(\{h(s_i)\}), \forall s_i \in \{s\}_s$.

Before proving properties of the operations on S we distinguish a special predicate [13] class $P_r(S_{P_r})$ where $S_{P_r} = \{s_1, s_2, \dots\}$ is a set of variables of the P_r predicate.

Each predicate $p \in P_r$ is defined as follows:

$$(\exists s_i, s_j \in S_p : id_{s_i}! = id_{s_j}) \Rightarrow p(S_p) = True$$

$$(\exists s_i \in S_p : \{s\}_{s_i} = \varnothing) \Rightarrow p(S_p) = True$$

$$(\nexists s_i \in S_p : \{s\}_{s_i} = \varnothing) \Rightarrow p(S_p) = \bigwedge p(s_{1i}, s_{2j}, \dots),$$

for $\forall s_{1i} \in \{s\}_{s_1}, s_{2j} \in \{s\}_{s_2}, \dots$, where $s_1, s_2, \dots \in S_p$

The following lemma holds for this class of predicates

Lemma 1. *$p(S_p) = True$ for $\forall p \in P_r$, $\forall S_p$*

Proof. Take an arbitrary predicate $p \in P_r$ and let $P(h)$ be the statement $p = True$, $\forall S_p$. We induct on $h(S_p)$.

1. Base case:
   $P(0)$ is true by the definition of p.
   Let's show that $P(h)$ is true for $h(S_p) = 1$:
   1) if $(\exists s_i, s_j \in S_p : id_{s_i}! = id_{s_j})$, then $p(S_p) = True$.
   2) if $(\exists s_i \in S_p : \{s\}_{s_i} = \varnothing)$, then $p(S_p) = True$.
   3) if $(\nexists s_i \in S_p : \{s\}_{s_i} = \varnothing)$, then $p(S_p) = \bigwedge p(s_{1i}, s_{2j}, \dots)$, for $\forall s_{1i} \in \{s\}_{s_1}, s_{2j} \in \{s\}_{s_2}, \dots$, where $s_1, s_2, \dots \in S_p$.
   $h(s_{kj}) = 0, \forall s_{kj}, k = \overline{1, |S_p|}, j = \overline{1, |\{s\}_{s_k}|}$, hence $h(s_{1i}, s_{2j}, \dots) = 0$, hence $p(s_{1i}, s_{2j}, \dots) = True$ for $\forall s_{1i} \in \{s\}_{s_1}, s_{2j} \in \{s\}_{s_2}, \dots$, where $s_1, s_2, \dots \in S_p$, hence $\bigwedge p(s_{1i}, s_{2j}, \dots) = True$, hence $p(S_p) = True$.
2. Induction step:
   Let $p(S_p) = True$ for $S_p : h(S_p) = n \in \mathbb{N}$.
   then $p(S'_p) = True$ for $S'_p : h(S'_p) = n', n' \leq n$ by the definition pf p.
   Let's show that $p(S_p) = True$ for $S_p : h(S_p) = n + 1$:

1) if $(\exists s_i, s_j \in S_p : id_{s_i} != id_{s_j})$, then $p(S_p) = True$.
2) if $(\exists s_i \in S_p : \{s\}_{s_i} = \varnothing)$, then $p(S_p) = True$.
3) if $(\nexists s_i \in S_p : \{s\}_{s_i} = \varnothing)$, then $p(S_p) = \bigwedge p(s_{1i}, s_{2j}, \dots)$, for $\forall s_{1i} \in \{s\}_{s_1}, s_{2j} \in \{s\}_{s_2}, \dots$, where $s_1, s_2, \dots \in S_p$.
$h(s_{kj}) \leq n, \forall s_{kj}, k = \overline{1, |S_p|}, j = \overline{1, |\{s\}_{s_k}|}$, hence, $h(s_{1i}, s_{2j}, \dots) = n$, hence, $p(s_{1i}, s_{2j}, \dots) = True$ for $\forall s_{1i} \in \{s\}_{s_1}, s_{2j} \in \{s\}_{s_2}, \dots$, where $s_1, s_2, \dots \in S_p$, hence, $\bigwedge p(s_{1i}, s_{2j}, \dots) = True$, hence, $p(S_p) = True$.

Conclusion: Since both the base case and the induction step have been proved as true for the arbitrary $p \in P_r$, hence by mathematical induction the statement $P(h)$ holds for every $h(S_p)$, hence $p(S_p) = True$ for $\forall p \in P_r$, $\forall S_p$.

The following lemmas are valid for the S structure:

Lemma 2. $s_1 \subseteq s_1 \cup s_2, \forall s_1, s_2 \in S$

Proof. 1. Consider the expression $s_1 \subseteq s_1 \cup s_2$:
If $id_{s_1} \neq id_{s_2}$, then $s_1 \cup s_2 = \{s_1; s_2\}$. $s_1 \in \{s_1; s_2\} = s_1 \cup s_2 \Rightarrow s_1 \subseteq s_1 \cup s_2$.
If $id_{s_1} = id_{s_2} = id$, then $s_1 \cup s_2 = \{id, \{s\}_{s_1} \cup \{s\}_{s_2}\}$, hence $s_1 \subseteq s_1 \cup s_2 \iff (id = id) \wedge (\{s\}_{s_1} = \varnothing \vee (\forall s_i \in \{s\}_{s_1} \Rightarrow \exists! s_j \in \{s\}_{s_1 \cup s_2} : s_i \subseteq s_j))$.
2. Consider the expression $(id = id) \wedge (\{s\}_{s_1} = \varnothing \vee (\forall s_i \in \{s\}_{s_1} \Rightarrow \exists! s_j \in \{s\}_{s_1 \cup s_2} : s_i \subseteq s_j))$:
$id = id$ for $\forall id \in ID$.
If $\{s\}_{s_1} = \varnothing$, then $s_1 \subseteq s_1 \cup s_2 \overset{def}{=} True$.
3. Consider the case $\{s\}_{s_1} \neq \varnothing$:
First we prove that s_j exists.
If $\{s\}_{s_2} = \varnothing$, then $\{s\}_{s_1} \cup \{s\}_{s_2} = \{s\}_{s_1} \Rightarrow (\forall s_i \in \{s\}_{s_1} \Rightarrow \exists! s_j \in \{s\}_{s_1 \cup s_2} = s_i : s_i \subseteq s_j)$. $s_j = s_i \Rightarrow s_i \subseteq s_i, \forall s_i \in \{s\}_{s_1} \Rightarrow s_1 \subseteq s_1 \cup s_2$.
If $\{s\}_{s_2} \neq \varnothing$, then $\{s\}_{s_1} \cup \{s\}_{s_2} = \{s_k \cup s_l\}, \forall s_k \in \{s\}_{s_1}, \forall s_l \in \{s\}_{s_2}$

$$s_k \cup s_l \overset{def}{=} \begin{bmatrix} \{s_k, s_l\} \;\; if \;\; id_{s_k} \neq id_{s_l} \\ \{id, \{s\}_{s_k} \cup \{s\}_{s_l}\} \;\; if \;\; id_{s_1} = id_{s_2} \end{bmatrix} \Rightarrow$$

$\Rightarrow \forall s_i \in \{s\}_{s_1} \exists! s_j \in \{s\}_{s_1 \cup s_2} : id_{s_j} = id_{s_i}$
4. Consider the expression $s_i \subseteq s_j$ where $s_i \in \{s\}_{s_1}, s_j \in \{s\}_{s_1 \cup s_2}$:
$s_j = s_i \cup s_l, \forall s_i \in \{s\}_{s_1}, \forall s_l \in \{s\}_{s_2}$, then $s_i \subseteq s_j \iff s_i \subseteq s_i \cup s_l, \forall s_i \in \{s\}_{s_1}, \forall s_l \in \{s\}_{s_2}$, which is equivalent to the formulation of the lemma.
So,

$$s_1 \subseteq s_1 \cup s_2 \iff \begin{bmatrix} id_{s_1} \neq id_{s_2} \\ \{s\}_{s_1} = \varnothing \vee \{s\}_{s_2} = \varnothing \\ s_i \subseteq s_i \cup s_l, \forall s_i \in \{s\}_{s_1}, \forall s_l \in \{s\}_{s_2} \end{bmatrix} \Rightarrow$$

$$\Rightarrow (s_1 \subseteq s_1 \cup s_2) \in P_r \xRightarrow{Lemma1} s_1 \subseteq s_1 \cup s_2, \forall s_1, s_2 \in S$$

Lemma 3. *Operation* $\cup$ *is associative on* S.

$$(s_1 \cup s_2) \cup s_3 = s_1 \cup (s_2 \cup s_3), \forall s_1, s_2, s_3 \in S \tag{11}$$

Proof. Let $id_1, id_2, id_3 \in ID$. We will write down in the table all possible combinations of indexes, while if they match, we will denote them by id:

№	s_1	s_2	s_3
1	$\{id_1, \{s\}\}$	$\{id_2, \{s\}\}$	$\{id_3, \{s\}\}$
2	$\{id, \{s\}\}$	$\{id, \{s\}\}$	$\{id_3, \{s\}\}$
3	$\{id, \{s\}\}$	$\{id_2, \{s\}\}$	$\{id, \{s\}\}$
4	$\{id_1, \{s\}\}$	$\{id, \{s\}\}$	$\{id, \{s\}\}$
5	$\{id, \{s\}\}$	$\{id, \{s\}\}$	$\{id, \{s\}\}$

Consider the expression (11) for all of the five cases:

1. $(s_1 \cup s_2) \cup s_3 = \{s_1, s_2\} \cup s_3 = \{s_1, s_2, s_3\} = s_1 \cup \{s_2, s_3\} = s_1 \cup (s_2 \cup s_3)$
2. $(s_1 \cup s_2) \cup s_3 = \{s_1 \cup s_2\} \cup s_3 = \{s_1 \cup s_2, s_3\} \overset{\text{Lemma 2}}{=} \{s_1 \cup s_2, s_1, s_3\} = s_1 \cup \{s_2, s_3\} = s_1 \cup (s_2 \cup s_3)$
3. $(s_1 \cup s_2) \cup s_3 = \{s_1, s_2\} \cup s_3 = \{s_1 \cup s_3, s_2\} \overset{\text{Lemma 2}}{=} \{s_1 \cup s_3, s_1, s_2\} = s_1 \cup \{s_3, s_2\} = s_1 \cup (s_2 \cup s_3)$
4. $(s_1 \cup s_2) \cup s_3 = \{s_1, s_2\} \cup s_3 = \{s_1, s_2 \cup s_3\} = s_1 \cup \{s_2 \cup s_3\} = s_1 \cup (s_2 \cup s_3)$
5. $(s_1 \cup s_2) \cup s_3 = \{id, (\{s\}_{s_1} \cup \{s\}_{s_2}) \cup \{s\}_{s_3}\}$. On the other hand $s_1 \cup (s_2 \cup s_3) = \{id, \{s\}_{s_1} \cup (\{s\}_{s_2} \cup \{s\}_{s_3})\}$.
   In this case $(s_1 \cup s_2) \cup s_3 = s_1 \cup (s_2 \cup s_3) \iff (\{s\}_{s_1} \cup \{s\}_{s_2}) \cup \{s\}_{s_3} = \{s\}_{s_1} \cup (\{s\}_{s_2} \cup \{s\}_{s_3})$.

Consider the expression $(\{s\}_{s_1} \cup \{s\}_{s_2}) \cup \{s\}_{s_3} = \{s\}_{s_1} \cup (\{s\}_{s_2} \cup \{s\}_{s_3})$ for various $\{s\}_{s_1}, \{s\}_{s_2}, \{s\}_{s_3}$:

1. If $\{s\}_{s_1} = \varnothing$, then $(\{s\}_{s_1} \cup \{s\}_{s_2}) \cup \{s\}_{s_3} = (\varnothing \cup \{s\}_{s_2}) \cup \{s\}_{s_3} = \{s\}_{s_2} \cup \{s\}_{s_3} = \varnothing \cup (\{s\}_{s_2} \cup \{s\}_{s_3}) = \{s\}_{s_1} \cup (\{s\}_{s_2} \cup \{s\}_{s_3})$.
2. If $\{s\}_{s_2} = \varnothing$, then $(\{s\}_{s_1} \cup \{s\}_{s_2}) \cup \{s\}_{s_3} = (\{s\}_{s_1} \cup \varnothing) \cup \{s\}_{s_3} = \{s\}_{s_1} \cup \{s\}_{s_3} = \{s\}_{s_1} \cup (\varnothing \cup \{s\}_{s_3}) = \{s\}_{s_1} \cup (\{s\}_{s_2} \cup \{s\}_{s_3})$.
3. If $\{s\}_{s_3} = \varnothing$, then $(\{s\}_{s_1} \cup \{s\}_{s_2}) \cup \{s\}_{s_3} = (\{s\}_{s_1} \cup \cup \{s\}_{s_2}) \cup \varnothing = \{s\}_{s_1} \cup \{s\}_{s_2} = \{s\}_{s_1} \cup (\{s\}_{s_2} \cup \varnothing) = \{s\}_{s_1} \cup (\{s\}_{s_2} \cup \{s\}_{s_3})$.
4. If $\{s\}_{s_1} \neq \varnothing \wedge \{s\}_{s_2} \neq \varnothing \wedge \{s\}_{s_3} \neq \varnothing$, then:
   1) $(\{s\}_{s_1} \cup \{s\}_{s_2}) \cup \{s\}_{s_3} = \{\{s_i \cup s_j\} \cup s_k\}, \forall s_i \in \{s\}_{s_1}, \forall s_j \in \{s\}_{s_2}, \forall s_k \in \{s\}_{s_3}$
   2) $\{s\}_{s_1} \cup (\{s\}_{s_2} \cup \{s\}_{s_3}) = \{s_i \cup \{s_j \cup s_k\}\}, \forall s_i \in \{s\}_{s_1}, \forall s_j \in \{s\}_{s_2}, \forall s_k \in \{s\}_{s_3}$

   Then the expression (11) is true if

$$\{\{s_i \cup s_j\} \cup s_k\} = \{s_i \cup \{s_j \cup s_k\}\}, \forall s_i \in \{s\}_{s_1}, \forall s_j \in \{s\}_{s_2}, \forall s_k \in \{s\}_{s_3} \tag{12}$$

   We can fix s_i, s_j, s_k because if (12) is true for the fixed arbitrary combination, then it will be true for all other combinations s_l, s_m, s_n.
   Then (12) can be written as:

$$((s_i \cup s_j) \cup s_k) = (s_i \cup (s_j \cup s_k)), \forall s_i \in \{s\}_{s_1}, \forall s_j \in \{s\}_{s_2}, \forall s_k \in \{s\}_{s_3},$$

   which is equivalent to the formulation of the lemma.

So, $(s_1 \cup s_2) \cup s_3 = s_1 \cup (s_2 \cup s_3)$, if:

$$\begin{bmatrix} id_{s_1} \neq id_{s_2} \lor id_{s_1} \neq id_{s_3} \lor id_{s_2} \neq id_{s_3} \\ \{s\}.s_1 = \varnothing \lor \{s\}.s_2 = \varnothing \lor \{s\}.s_3 = \varnothing \Rightarrow \\ ((s_i \cup s_j) \cup s_k) = (s_i \cup (s_j \cup s_k)) \end{bmatrix}$$

$$\Rightarrow ((s_1 \cup s_2) \cup s_3 = s_1 \cup (s_2 \cup s_3)) \in P_r \xrightarrow{\text{Lemma 1}} (s_1 \cup s_2) \cup s_3 = s_1 \cup (s_2 \cup s_3),$$

for $\forall s_1, s_2, s_3 \in S$

Remark 2. If we expand S structure with an additional structure $p \in \mathbb{P}$ for which a binary relation $\subseteq$ and binary operation $\cup : \mathbb{P} \times \mathbb{P} \to \mathbb{P}$ are defined, and the statement $p_1 \subseteq p_1 \cup p_2, \forall p1, p2 \in \mathbb{P}$ is true, then lemma №2 will still hold true for S because:

$$s_1 \cup s_2 = \{id, p_{s_1} \cup p_{s_2}, \{s\}_{s_1} \cup \{s\}_{s_2}\}$$

Similarly, if lemmas №2,3 hold true for $\forall p_1, p_2 \in \mathbb{P}$, then they will hold for S.

The same conclusions can be done for S expanded with several additional structures $p_i \in \mathbb{P}_i$.

The MV' structure corresponds to the S structure expanded with a set of edges. Set of edges is the usual set, hence, lemmas №2,3 hold true for MV'. Since $MV \leftrightarrow MV'$, lemmas №2,3 hold true for MV. Each metagraph can be represented as a metavertex with the same $id = id_{mg}$, hence each metagraph holds all the properties of the S structure.

It follows from this that the statements 5 - 6 are proven for $\forall mg \in MG$ and $\forall mv \in MV$.

3 Category of Metagraphs

Having proved the associativity property of the union operation of metagraphs, we can describe a category of metagraphs.

Definition 7. *Category of metagraphs $MetGr_\cup$ is a category in which class of objects is a set of metagraphs MG and morphisms are expressions of the form $\cup mg$, that perform a union operation of the argument mg_1 (the beginning of the arrow) with mg and get metagraph mg_2 (the end of the arrow) as a result:*

$$mg_1 \xrightarrow{\cup mg} mg_2 \tag{13}$$

Proof of the category axioms for $MetGr_\cup$:

1. Identity morphism: $id_{MetGr_{\cup}} = \cup\varnothing$.
   Let $f = \cup mg_f$, then:

$$f \circ id = \cup\varnothing \cup mg_f = \cup mg_f = f$$
$$id \circ f = \cup mg_f \cup \varnothing = \cup mg_f = f$$

2. Composition of morphisms:
   Let $f = \cup mg_f,\ g = \cup mg_g$, then:

$$g \circ f = \cup mg_f \cup mg_g = \cup\{MV_{mg_f} \cup MV_{mg_g},\ E_{mg_f} \cup E_{mg_g}\}$$

3. Associativity of the composition operation:
   Let $f = \cup mg_f,\ g = \cup mg_g,\ h = \cup mg_h$, then:

$$\begin{gathered}(h \circ g) \circ f = \cup mg_f \cup (mg_g \cup mg_h) = \\ \cup mg_f \cup \{MV_{mg_g} \cup MV_{mg_h},\ E_{mg_g} \cup E_{mg_h}\} = \\ \cup\{MV_{mg_f} \cup (MV_{mg_g} \cup MV_{mg_h}),\ E_{mg_f} \cup (E_{mg_g} \cup E_{mg_h})\} = \\ \cup\{(MV_{mg_f} \cup MV_{mg_g}) \cup MV_{mg_h},\ (E_{mg_f} \cup E_{mg_g}) \cup E_{mg_h}\} = \\ \cup\{MV_{mg_f} \cup MV_{mg_g},\ E_{mg_f} \cup E_{mg_g}\} \cup mg_h = \\ \cup(mg_f \cup mg_g) \cup mg_h = h \circ (g \circ f)\end{gathered}$$

Category axioms are proven, hence $MetGr_{\cup}$ is a category.

4 Example

Let 's give an example of using operations over metagraphs.

Let's say there are two departments in some corporation called "Corp". The first department is called "Dep1", and Mike and Anna work in it, Anna is Mike's supervisor. The second department is called "Dep2", John and Nick work in it. John came to work recently, so he is not Nick's supervisor yet.

After the new year, the following changes are planned in "Corp":

1. A manager Harry comes to the corporation, and he will manage the "Dep1" and "Dep2" departments.
2. A new employee Alex comes to the "Dep1", and Anna will be his supervisor.
3. John becomes Nick's supervisor.

Let's simulate this situation using a $MetGr_{\cup}$ category. Let the metavertices denote departments, the vertices denote employees, and the edges denote the subordination relationship.

Metagraph in the figure 1 corresponds to the initial state of the "Corp". Figure 2 corresponds to the moment at which Harry comes to the company. In figure 3 Harry starts managing "Dep1" and "Dep2" departments. Alex comes to "Dep1" in figure 4. Anna becomes Alex's supervisor in figure 5. John becomes Nick's supervisor in figure 6.

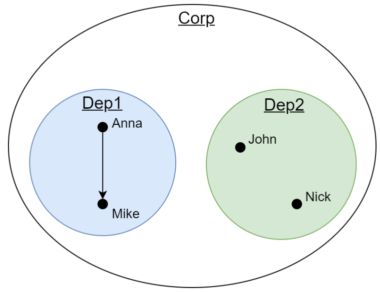

Fig. 1. initial state of the "Corp"

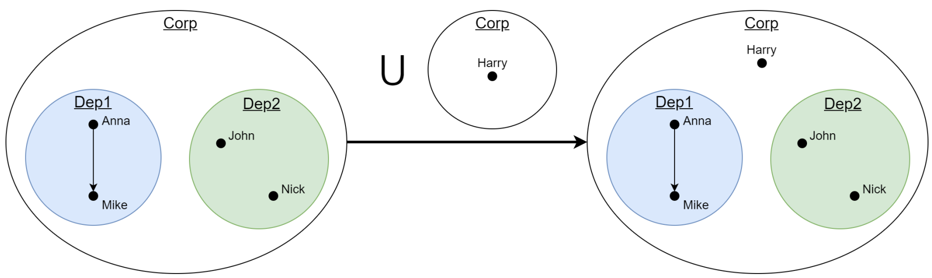

Fig. 2. Harry comes to the company

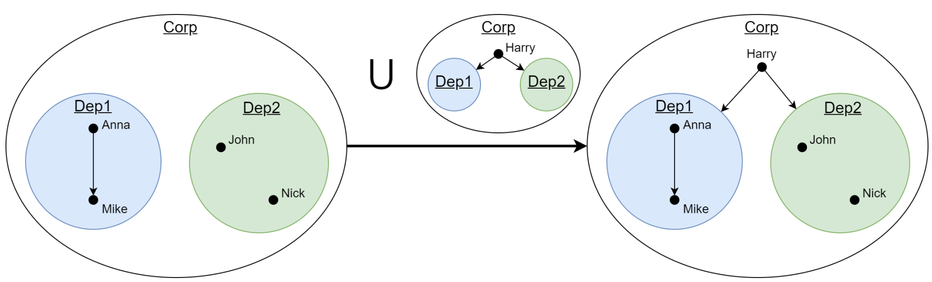

Fig. 3. Harry starts managing "Dep1" and "Dep2" departments

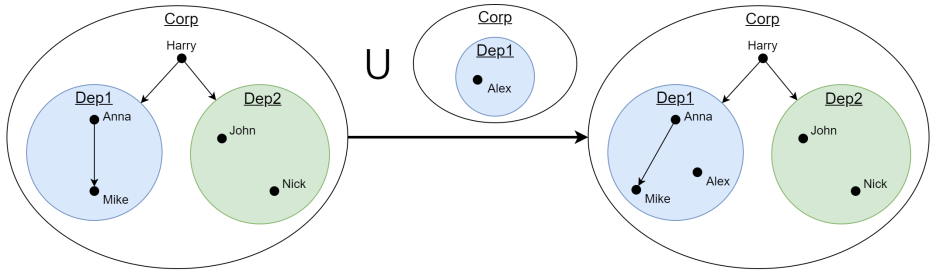

Fig. 4. Alex comes to "Dep1"

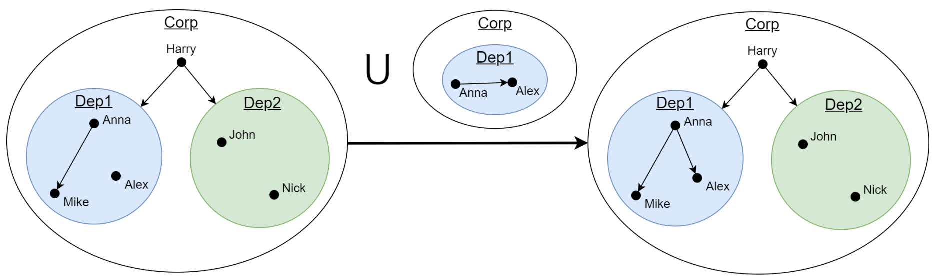

Fig. 5. Anna becomes Alex's supervisor

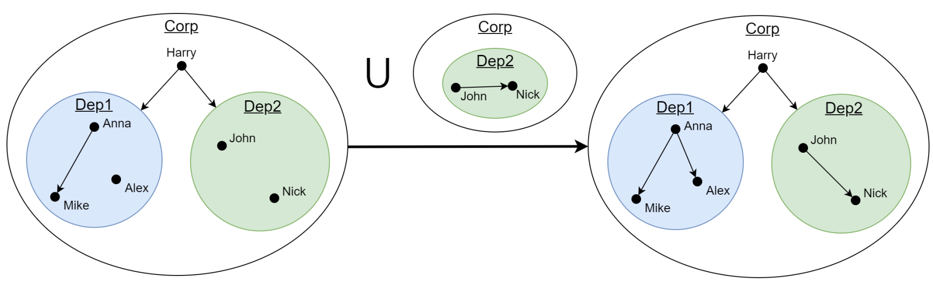

Fig. 6. John becomes Nick's supervisor

The right metagraph in figure 6 corresponds to the final state of the "Corp" after the new year. Note that we could achieve the same state with just one morphism using the composition operation in $MetGr_{\cup}$, as shown in figure 7.

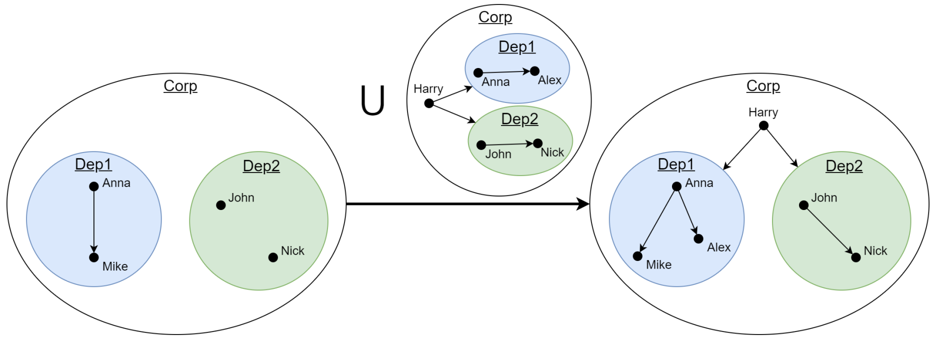

Fig. 7. transition from the initial state to the final one through a single morphism

Thus, using a metagraph data model and the $MetGr_{\cup}$ category, we are able to simulate real-life situations.

5 Conclusion

The metagraph data model is a modern tool for modeling systems of varying degrees of complexity. In addition to various data about the characteristics of the described system, metagraphs store information about the hierarchy of this system, which advantageously distinguishes it from other modeling methods.

The proof of the properties of the simple recursive structure allowed us to prove that a metagraph can be viewed as a category.

The obtained research results are applicable for the further development of the rewriting system on metagraphs. The rewriting system can use either metagraph operations or a categorical approach or their hybridization. The effectiveness of each of these approaches is the subject of further research.

References

1. Basu, A., Blanning, R.W.: Metagraphs and Their Applications. Springer (2007)
2. Gapanyuk, Yu.: The Development of the Metagraph Data and Knowledge Model. In: Russian Advances in Fuzzy Systems and Soft Computing: Selected Contributions to the 10th International Conference on “Integrated Models and Soft Computing in Artificial Intelligence (IMSC-2021)”, pp. 1–7. Kolomna, Russia, May 17-20 (2021)
3. Ehrig, H. et al.: Graph and Model Transformation. Springer (2015)
4. Allemang, D., Gandon, F., Hendler, J.A.: Semantic web for the working ontologist: effective modeling for linked data, RDFS, and OWL. ACM books (2020)
5. Baader, F., Horrocks, I., Sattler, U.: Description Logics as Ontology Languages for the Semantic Web. In: Hutter, D., Stephan, W. (eds.) Mechanizing Mathematical Reasoning 2005. LNCS, vol. 2605, pp. 228–248. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32254-2_14

6. Chernenkiy, V., Gapanyuk, Y., Nardid, A., Skvortsova, M., Gushcha, A., Fedorenko, Y., Picking, R.: Using the metagraph approach for addressing RDF knowledge representation limitations. In: 2017 Internet Technologies and Applications, ITA 2017 - Proceedings of the 7th International Conference, pp. 47–52. (2017)
7. MacLane, S. Categories for the Working Mathematician. Springer-Verlag, New York (1971)
8. Awodey, S.: Category Theory. Oxford University Press (2010)
9. Leinster, T.: Basic Category Theory. Cambridge University Press (2014)
10. Yanofsky, N.S.: Theoretical Computer Science for the Working Category Theorist. Cambridge University Press (2022)
11. Grandis, M.: Category Theory and Applications: A Textbook for Beginners. World Scientific (2018)
12. Grami, A.: Discrete Mathematics: Essentials and Applications. Academic Press (2022)
13. Mendelson, E.: Introduction to mathematical logic. CRC Press/Taylor & Francis Group (2015)