Difference of Fuzzy Homogeneous Classes of Objects

Dmytro O. Terletskyi[1], Sergey V. Yershov[2]

[1] V. M. Glushkov Institute of Cybernetics of NAS of Ukraine, Academician Glushkov Avenue, 40, Kyiv, 03187, Ukraine
[2] V. M. Glushkov Institute of Cybernetics of NAS of Ukraine, Academician Glushkov Avenue, 40, Kyiv, 03187, Ukraine

Abstract

Analysis of relevance, similarity, and the difference between extracted or acquired new knowledge items and previously obtained ones are important stages of the knowledge integration process for modern knowledge-based systems. These stages can be performed via the application of special operations defined over the knowledge representation structures provided by chosen representation model. Within the object-oriented approach, classes are one of the main knowledge representation structures, consequently, the analysis of the relevance, similarity, and difference between them requires the development of corresponding operations over classes. Therefore the concept of universal difference exploiter of fuzzy homogeneous classes of objects, within such a knowledge representation model as fuzzy object-oriented dynamic networks, was introduced in the paper. To implement the proposed exploiter, which allows computing of the difference of one fuzzy homogeneous class of objects from another one, we developed the corresponding algorithm and provided an example of its application. The proposed approach provides an opportunity to evaluate the relevance and novelty level for extracted or acquired fuzzy knowledge items, compared with previously obtained ones, by computing the difference between them.

Keywords

Fuzzy class, Fuzzy type, Universal difference exploiter, Difference of fuzzy classes

1. Introduction

The analysis of knowledge structures for their proper integration into the knowledge base and organizing future reasoning processes is a crucial and important task for modern knowledge-based systems (KBSs). To manage such a challenge, a KBS should be able to perform the comparative analysis of extracted or acquired new knowledge items with previously obtained ones. It allows a system to estimate the level of novelty for extracted knowledge items as well as to conclude about their similarity and differences with other knowledge items, which are already integrated within a knowledge base. Taking into account such parameters, it is possible to integrate new extracted or acquired knowledge items into the knowledge base avoiding their representation redundancy.

According to the concept of *knowledge integration* proposed by Murray and Porter in [1, 2, 3, 4, 5], it can be defined as a task of incorporating new information into a knowledge base, which requires elaborating new information and resolving inconsistencies with existing knowledge.

17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, September 28 – October 2, 2021, Kherson, Ukraine
dmytro.terletskyi@gmail.com (D. O. Terletskyi); ErshovSV@nas.gov.ua (S. V. Yershov)
0000-0002-7393-1426 (D. O. Terletskyi); 0000-0002-9895-777X (S. V. Yershov)

The corresponding computational model of knowledge integration incorporates the following main stages:

1. *Recognition*: Identification of relevance between new and previously obtained knowledge.
2. *Elaboration*: Determination of interactions of obtained new knowledge items on previously obtained ones, and how they can affect them.
3. *Adaptation*: Resolving of previously detected abnormalities within the knowledge base.

The model was implemented within the knowledge acquisition tool KI for management of a large Botany Knowledge Base, which incorporates knowledge about plant anatomy, physiology, and development.

The recognition stage of the knowledge integration process requires performing the verification of such relations between new and previously obtained knowledge items as *equivalence*, *inclusion*, *similarity*, *difference*, etc. However, each knowledge representation model provides a suite of particular representation structures and corresponding tools for their management. Therefore knowledge analysis approaches are oriented toward certain knowledge representation paradigms or models. Within the object-oriented knowledge representation approach such structures as *objects*, *classes*, *metaclasses*, and *relations* among them, form a basic representation suite. Consequently, knowledge analysis within such an approach can be interpreted as a comparison of corresponding representation structures. Since a class is a widely-used representation structure, thus the comparative analysis of classes is one of the important and priority knowledge analysis tasks within the object-oriented approach. One of the known approaches to solving this task is to define the special operations over classes, in particular set-theoretical ones. According to the approach, a class is considered as a collection of properties and methods, consequently, the application of basic set-theoretical operations such as *union*, *intersection*, *difference*, and *symmetric difference*, provides a corresponding framework for the analysis of knowledge items, which are represented in terms of classes.

However, different object-oriented knowledge representation models use distinct concepts of a class, consequently, all mentioned set-theoretical operations should be adapted for a particular knowledge representation model and appropriate interpretation of a class. In addition, the classical (crisp) object-oriented paradigm has some representation restrictions, which limit its application for modeling vague, imprecise as well as uncertain entities and domains. Therefore in this paper, we adapted the set-theoretical difference for such knowledge representation model as fuzzy object-oriented dynamic networks in the form of the corresponding universal exploiter of fuzzy homogeneous classes of objects. To implement the proposed universal difference exploiter, we developed an appropriate algorithm and provided a representative example of its application for the knowledge analysis within fuzzy object-oriented dynamic networks.

2. Fuzzy Classes and Types

A class is one of the main representation structures in class-based object-oriented programming, in a variety of object-oriented knowledge representation models as well as in object-oriented databases. According to the classical (crisp) definition, a class can be interpreted as the collection of properties (specification) and collection of methods (signature), which define a common structure and typical behavior for all instances of the class, respectively.

However, as it was noted in [6, 7], despite all benefits of crisp classes and objects, they can be inefficient for the construction of realistic representation models of *vague*, *imprecise*, or *uncertain* entities or domains, because of their descriptive restrictions. Therefore a crisp class-based object-oriented model was extended using notions of fuzzy sets and linguistic variables, which were introduced by Zade in [8, 9, 10, 11]. The main idea of such extension is to define values of class properties as fuzzy sets or as linguistic variables, which are more complex structures defined as the term-set interpreted using appropriate fuzzy variables defined by fuzzy sets. It led to the appearance of concepts of *fuzzy attributes* and *fuzzy objects*, described in [6]. The next step in fuzzification of the object-oriented paradigm was to introduce a *measure of truth* for each property of the class, which is defined on the interval of real numbers $[0, 1]$. Then, concepts of *fuzzy methods* and *fuzzy types* were introduced in [12, 13, 14, 15], as well as the concept of *fuzzy classes* and *fuzzy class hierarchies* were proposed in [7]. One more step was to define a membership degree for objects of a class as well as membership degree for subclasses of a class, which led to the appearance of *classification of fuzziness levels*, introduced in [16], and the notion of *fuzzy classes of fuzzy objects*, which were introduced in [17, 18, 19, 20].

Many of the introduced interpretations of fuzzy types and fuzzy classes were proposed as fuzzy extensions of the object-oriented entity-relationship model, used in databases, where attributes of a class are considered separately from each other. However, such interpretation of a class or a type is distinguished from corresponding interpretation within the class-based object-oriented programming and many object-oriented knowledge representation models, where properties and methods of a class can have internal dependencies from each other. As it was shown in [21], the internal structure of a class consists of structural and functional atoms as well as structural and functional molecules, created by properties and methods of a class. It specifies the main difference between the concept of a class within object-oriented programming as well as knowledge representation and similar concept within object-oriented databases.

Another extension of the class-based object-oriented model was implemented within such knowledge representation model, as fuzzy object-oriented dynamic networks (FOODNs), which was proposed in [22, 23] and later extended in [24, 25]. Similar to other object-oriented models, such concepts as *fuzzy objects*, *fuzzy classes*, and *fuzzy relations* are also used within the FOODNs. The structure of a fuzzy class also defined by a collection of crisp and (or) fuzzy properties, while its behavior is determined by a collection of crisp and (or) fuzzy methods. However, in contrast to other object-oriented models, the specification of a fuzzy class of objects consists of *quantitative* and *qualitative properties*. Quantitative properties represent evident numerical or symbolic single-valued or multi-valued characteristics, while qualitative properties represent more complex, not obvious features defined based on other properties and methods of the class [24, 25]. The signature of a fuzzy class of objects consists of methods, which define a common behavior and opportunities to check and (or) to modify the structure of a particular instance of the class. Similar to other object-oriented models, each fuzzy class within FOODNs defines the particular fuzzy type, which identifies a common structure and behavior for all fuzzy objects of the class. Therefore such classes can be called *homogeneous* ones. Let us consider the definition of the fuzzy homogeneous class of objects within FOODNs introduced in [24, 25].

Definition 1. *A fuzzy homogeneous class of objects is a collection*

$$T/M(T) = (P(t), F(t))/M(T) = ((p_1(t)/\mu(p_1(t)), \dots,$$

$$p_n(t)/\mu(p_n(t))), (f_1(t)/\mu(f_1(t)), \ldots, f_m(t)/\mu(f_m(t))))/M(T),$$

where t is a fuzzy type which is defined by the class T, $p_i(t)/\mu(p_i(t)) \in P(t)$ is a crisp or fuzzy property of the class T, $f_j(t)/\mu(f_j(t)) \in F(t)$ is its crisp or fuzzy method, $\mu(p_i(t)) : p_i(t) \to (0, 1]$ and $\mu(f_j(t)) : f_j(t) \to (0, 1]$ are measures of fuzziness of a property $p_i(t)$ and a method $f_j(t)$, and $M(T)$ is a measure of fuzziness of the class T, i.e.

$$M(T) = ((\mu(p_1(t)) + \cdots + \mu(p_n(t))) + (\mu(f_1(t)) + \cdots + \mu(f_m(t))))/(n + m).$$

Analyzing Definition 1, we can see that in the context of fuzziness of attributes and methods, a concept of a fuzzy homogeneous class of objects is similar to the concept of the fuzzy type described in [12, 13, 14, 15] as well as to the concept of the fuzzy class described in [17, 18, 19, 20], while the measure of fuzziness of the class itself is defined as the arithmetic mean of measures of the truth of all its properties and methods. Similar to homogeneous classes, fuzzy homogeneous classes of objects define only a single fuzzy type of objects, therefore as in class-based object-oriented programming, a fuzzy type of objects and a fuzzy homogeneous class of objects can be considered as equivalent concepts.

3. Difference of Fuzzy Homogeneous Classes

As it was mentioned above, to perform knowledge integration in the proper way, a KBS should be able to analyze the relevance, similarity, and differences between extracted or acquired new knowledge items and previously obtained ones, which are represented in terms of fuzzy classes. Therefore we propose to consider set-theoretical difference adapted for fuzzy classes as a tool for dynamic creation of new fuzzy classes of objects which represent of difference of one fuzzy class from another one. The idea to apply set-theoretical operations to fuzzy classes, fuzzy objects as well as fuzzy relations is widely used in many interpretations. As the result, it was used to implement algebraic operations for fuzzy object-oriented database language [26], query processing within the fuzzy relational object-oriented databases [27, 28, 19, 29, 17, 30], relational uncertain databases [31], the blurry classes within the fuzzy object-oriented databases [32], supporting fuzzy XML queries [33, 19] and handling uncertain spatiotemporal data [34, 35], biomedical fuzzy HBase databases [36], etc.

All mentioned algebras provide different signatures, however, all of them contain an adapted operation of set-theoretical difference, defined over fuzzy objects, or fuzzy classes, or fuzzy relations. However, in most cases, it is used for database querying on the level of objects, classes, and relations, when the data or knowledge are already integrated within the database or knowledge base. Therefore we propose to consider the application of difference operation, defined over the fuzzy homogeneous classes of objects, in the context of knowledge analysis as a part of the knowledge integration process.

The concept of difference of classes of objects was introduced in [22, 23] in a form of a corresponding universal exploiter for classes of fuzzy objects. Later the concept of a fuzzy class of objects was proposed in [24, 25]. Let us define the notion of difference exploiter for fuzzy homogeneous classes of objects via generalizing its versions for the classes of fuzzy objects.

Definition 2. *Difference $T_1 \setminus T_2$ of two fuzzy homogeneous classes of objects $T_1/M(T_1)$ and $T_2/M(T_2)$, which define fuzzy types of objects t_1 and t_2 correspondingly, is a fuzzy homogeneous class of objects $T_{1\setminus 2}/M(T_{1\setminus 2})$, which define fuzzy type of objects $t_{1\setminus 2}$, such that*

$$t_{1\setminus 2} \subseteq t_1 \mid \nexists \left(t_{1\setminus 2} \cap t_2\right) \wedge \left(\nexists t^*_{1\setminus 2} \mid \left(t_{1\setminus 2} \subseteq t^*_{1\setminus 2}\right) \wedge \left(t^*_{1\setminus 2} \subseteq t_1\right) \wedge \nexists \left(t^*_{1\setminus 2} \cap t_2\right)\right).$$

The fuzzy class $T_{1\setminus 2}/M(T_{1\setminus 2})$ exists if and only if $\exists p_{i_1}(t_1)$, $\exists p_{i_2}(t_2)$, such that $p_{i_1}(t_1) \not\equiv p_{i_2}(t_2)$, or $\exists f_{j_1}(t_1)$, $\exists f_{j_2}(t_2)$, such that $f_{j_1}(t_1) \not\equiv f_{j_2}(t_2)$, where $p_{i_1}(t_1)$ is an i_1-th property of the fuzzy type t_1, $i_1 = \overline{1, |P(t_1)|}$, $p_{i_2}(t_2)$ is an i_2-th property of the fuzzy type t_2, $i_2 = \overline{1, |P(t_2)|}$, $f_{j_1}(t_1)$ is an j_1-th method of the fuzzy type t_1, $j_1 = \overline{1, |F(t_1)|}$, $f_{j_2}(t_2)$ is an j_2-th method of the fuzzy type t_2, $j_2 = \overline{1, |F(t_2)|}$.

The universal difference exploiter creates a fuzzy homogeneous class of objects $T_{1\setminus 2}/M(T_{1\setminus 2})$, which represents the difference of the fuzzy class of objects $T_1/M(T_1)$ from the fuzzy class of objects $T_2/M(T_2)$ if such difference exists. The class $T_{1\setminus 2}/M(T_{1\setminus 2})$ defines a fuzzy type of objects $t_{1\setminus 2} \subseteq t_1$, which consists of crisp and (or) fuzzy properties and (or) methods, which are typical only for the fuzzy type of objects t_1.

To implement a universal exploiter of fuzzy homogeneous classes of objects, the corresponding algorithm should analyze specifications and signatures of fuzzy homogeneous classes of objects $T_1/M(T_1)$, $T_2/M(T_2)$ and find properties and methods of the fuzzy class $T_1/M(T_1)$, which are not typical for the fuzzy class of objects $T_2/M(T_2)$. For this purpose, we used the [25, Algorithm 1] for checking the equivalence of fuzzy quantitative properties, [25, Algorithm 2] for checking the equivalence of fuzzy qualitative properties, and [25, Algorithm 3] for checking the equivalence of fuzzy methods. As the result, we developed Algorithm 1, which implements the idea of universal difference exploiter for fuzzy homogeneous classes of objects.

Analyzing Algorithm 1, we can see that it uses fuzzy homogeneous classes of objects $T_1/M(T_1)$, $T_2/M(T_2)$ as the input data and computes the difference of the class $T_1/M(T_1)$ from the class $T_2/M(T_2)$ in a form of a new fuzzy homogeneous class of objects $T_{1\setminus 2}/M(T_{1\setminus 2}) = T_1/M(T_1) \setminus T_2/M(T_2)$ if such difference exists. The algorithm successively constructs the specification and signature of class $T_{1\setminus 2}/M(T_{1\setminus 2})$ and these stages are independent ones, consequently, such computations also can be performed in parallel mode. The polymorphic function `is_equivalent` checks the equivalence of two fuzzy properties $p_i(T_1)/\mu(p_i(T_1))$, $p_j(T_2)/\mu(p_j(T_2))$ or methods $f_i(T_1)/\mu(f_i(T_1))$, $f_j(T_2)/\mu(f_j(T_2))$ and if they are equivalent ones it returns 1, otherwise it returns 0. It can be implemented in various ways using corresponding algorithms for checking the equivalence of fuzzy properties and methods, which were proposed in [25].

Let us estimate the time and space complexity of Algorithm 1. As we can see, during the analysis of specifications and signatures of classes $T_1/M(T_1)$ and $T_2/M(T_2)$ the algorithm checks the equivalence of $|P(T_1)| \times |P(T_2)| = n \times m$ properties and $|F(T_1)| \times |F(T_2)| = k \times q$ methods. In addition, to construct the specification $P(T_{1\setminus 2})$ and signature $F(T_{1\setminus 2})$ of the fuzzy class $T_{1\setminus 2}/M(T_{1\setminus 2})$, it performs copying of w_1 properties and w_2 methods of the fuzzy class $T_1/M(T_1)$, which are not typical for the fuzzy class $T_2/M(T_2)$, where $0 \leq w_1 \leq |P(T_1)|$ and $0 \leq w_2 \leq |F(T_1)|$. Therefore, the time complexity of Algorithm 1 is equal to

$$O(n \times m) + O(k \times q) + O(w_1) + O(w_2) \approx O(n^2 + k^2 + w_1 + w_2),$$

Algorithm 1 Difference of fuzzy homogeneous classes.

Require: $T_1/M(T_1), T_2/M(T_2)$ – fuzzy homogeneous classes
Ensure: $T/M(T) = T_1/M(T_1) \setminus T_2/M(T_2)$

```
T := {};
unique := true;
for all p_i/μ(p_i) ∈ P(T_1) do
    for all p_j/μ(p_j) ∈ P(T_2) do
        if is_equivalent(p_i/μ(p_i), p_j/μ(p_j)) then
            unique := false;
            break;
    if unique then
        if P(T) ∉ T then
            P := {};
            T.add(P);
        P(T).add(p_i/μ(p_i))
    else
        unique := true;
unique := true;
for all f_i/μ(f_i) ∈ F(T_1) do
    for all f_j/μ(f_j) ∈ F(T_2) do
        if is_equivalent(f_i/μ(f_i), f_j/μ(f_j)) then
            unique := false;
            break;
    if unique then
        if F(T) ∉ T then
            F := {};
            T.add(F);
        F(T).add(f_i/μ(f_i))
    else
        unique := true;
return T.
```

where n^2 is a number of properties equivalence checks, k^2 is a number of methods equivalence checks, w_1 and w_2 is a number of copying operations of properties and methods of fuzzy class $T_1/M(T_1)$. To perform main computations Algorithm 1 uses s units of memory for storing fuzzy type $t_{1\setminus 2}$, therefore its space complexity is equal to $O(s)$, where $0 \leq s \leq |P(T_1)| + |F(T_1)|$.

4. Application Example

Let us consider a few fuzzy homogeneous classes of objects, represented in terms of fuzzy object-oriented dynamic networks, which simultaneously have equivalent and nonequivalent subclasses. Let us suppose that the first fuzzy homogeneous class of objects $HomeFridge$ defines the fuzzy type t_{hf1}, which describes a fuzzy concept of a home fridge, which has the following representation:

$$
\begin{aligned}
&HomeFridge(\\
&\quad p_1 = (refrigerator\_volume, (v \in T(r\_volume), str))/1, \\
&\quad p_2 = (freezer\_volume, (v \in T(f\_volume), str))/1, \\
&\quad p_3 = (cee, (v \in V_{cee}, str))/1, \\
&\quad p_4 = (aec, (v \in V_{aec}, kWh))/1, \\
&\quad p_5 = (sizes, ((v_1 \in V_{height}, cm), (v_2 \in V_{width}, cm), (v_3 \in V_{depth}, cm))/1, \\
&\quad p_6 = (compactness, (vf_6(HomeFridge.sizes), v \in [0,1]))/0.93, \\
&\quad p_7 = (color, (v \in V_{color}, str))/1, \\
&\quad p_8 = (weight, (v \in T(weight), str))/1, \\
&\quad p_9 = (noisiness, (v \in T(noisiness), str))/0.75, \\
&\quad p_{10} = (price, (v \in V_{price}, \mathbb{N}^+))/1, \\
&\quad f_1 = get\_crisp\_weight()/0.93, \\
&\quad f_2 = get\_fuzzy\_price(a, b, k)/0.87 \\
&)/0.96,
\end{aligned}
$$

refrigerator volume is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(r\_volume) = \{very\ small, small, medium, big, very\ big\},$$

where fuzzy variables *very small*, *small*, *medium*, *big*, and *very big* are defined over the interval of integer numbers $U = [40, 425]$, which means the volume of the refrigerator in cm^3, and have the following interpretation:

$$
\begin{aligned}
M(very\ small) &= \{40/1 + 50/0.95 + 60/0.85 + 70/0.7 + 80/0.65\}\ cm^3, \\
M(small) &= \{95/0.92 + 110/0.78 + 125/0.63 + 140/0.55 + 150/0.45\}\ cm^3, \\
M(medium) &= \{170/0.78 + 190/0.92 + 210/1 + 230/0.92 + 250/0.78\}\ cm^3, \\
M(big) &= \{270/0.82 + 290/0.94 + 310/1 + 330/0.94 + 350/0.82\}\ cm^3, \\
M(very\ big) &= \{365/0.65 + 380/0.72 + 395/0.86 + 410/0.93 + 425/1\}\ cm^3;
\end{aligned}
$$

freezer volume is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(f\_volume) = \{very\ small, small, medium, big, very\ big\},$$

where fuzzy variables *very small*, *small*, *medium*, *big*, and *very big* are defined over the interval of integer numbers $U = [10, 275]$, which means the volume of the freezer in cm^3, and have the following interpretation:

$$M(very\ small) = \{10/1 + 17/0.94 + 24/0.85 + 31/0.78 + 38/0.65\}\ cm^3,$$

$$M(small) = \{50/1 + 57/0.92 + 64/0.86 + 71/0.73 + 78/0.61\}\ cm^3,$$
$$M(medium) = \{90/0.85 + 100/0.93 + 110/1 + 120/0.93 + 130/0.85\}\ cm^3,$$
$$M(big) = \{140/0.82 + 155/0.93 + 170/1 + 185/0.93 + 200/0.82\}\ cm^3,$$
$$M(very\ big) = \{215/0.67 + 230/0.79 + 245/0.88 + 260/0.95 + 275/1\}\ cm^3;$$

cee is a crisp quantitative property, which means the class of energy efficiency to which the fridge belongs, and is defined over the set of string values $V_{cee} = \{A^{+++}, A^{++}, A^{+}, A, B, C, D, F\}$; *aec* is a crisp quantitative property, which means the annual energy consumption by the fridge in kWh, and is defined over the following interval of integer numbers $V_{aec} = [100, 360]$; *sizes* is a crisp multiple-valued quantitative property, which means dimensions of the fridge in cm, and is defined over the intervals of integer numbers $V_{height} = [45, 205]$, $V_{width} = [35, 95]$, $V_{depth} = [55, 85]$; *compactness* is a fuzzy qualitative property defined by verification function

$$vf_6(HomeFridge) : HomeFridge.sizes \rightarrow [0, 1],$$

where $vf_6(HomeFridge)$ is defined as follows

$$vf_6(HomeFridge) = \frac{V - V_{min}}{V_{max} - V_{min}},$$

where

$$V = sizes.v_1 \cdot sizes.v_2 \cdot sizes.v_3,$$
$$V_{min} = \min(V_{height}) \cdot \min(V_{width}) \cdot \min(V_{depth}),$$
$$V_{max} = \max(V_{height}) \cdot \max(V_{width}) \cdot \max(V_{depth});$$

color is a crisp quantitative property, which means the color of the fridge, and is defined over the following set of string values

$$V_{color} = \{beige, white, graphite, golden, brown, red, stainless\ steel,$$
$$silver, grey, titanium, black, bronze, blue, green, orange, pink, ivory, purple\};$$

weight is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(weight) = \{lightweight, medium, heavy, very\ heavy\},$$

where fuzzy variables *lightweight*, *medium*, *heavy*, and *very heavy* are defined over the interval of integer numbers $U = [10, 135]$, which means the weight of the fridge in kg, and have the following interpretation:

$$M(lightweight) = \{10/1 + 20/0.95 + 30/0.88 + 40/0.79 + 50/0.68\}\ kg,$$
$$M(medium) = \{55/0.83 + 60/0.94 + 65/1 + 70/0.94 + 75/0.83\}\ kg,$$
$$M(heavy) = \{80/0.86 + 85/0.95 + 90/1 + 95/0.95 + 100/0.86\}\ kg,$$
$$M(very\ heavy) = \{115/0.71 + 120/0.79 + 125/0.88 + 130/0.96 + 135/1\}\ kg;$$

noisiness is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(noisiness) = \{low, medium, high\},$$

where fuzzy variables *low*, *medium*, and *high* are defined over the interval of real numbers $U = [30, 45]$, which means the noisiness of the fridge in dB, and have the following meaning:

$$M(low) = \{30/1 + 31/0.97 + 32/0.93 + 33/0.89 + 34/0.82\}\ dB,$$
$$M(medium) = \{35/0.92 + 36/0.98 + 37/1 + 38/0.98 + 39/0.92\}\ dB,$$
$$M(high) = \{40/0.83 + 41/0.88 + 42/0.92 + 43/0.97 + 44/1\}\ dB;$$

price is a crisp quantitative property, which means the price of the fridge in UAH, and is defined over the interval of integer numbers $V_{price} = [2200, 255000]$; *get_crisp_weight* is a fuzzy method that computes defuzzification representation of the fuzzy quantitative property *weight* and defined in the following way:

$$get\_crisp\_weight() = \frac{\sum_{i=1}^{|weight.v|} \mu(weight.v) \cdot weight.v}{\sum_{i=1}^{|weight.v|} \mu(weight.v)};$$

get_fuzzy_price is a fuzzy method that computes fuzzification representation of the crisp quantitative property *price* and defined in the following way:

$$get\_fuzzy\_price(a, b, k) = \{x_i^-/\mu(x_i^-), price.v/1, x_i^+/\mu(x_i^+)\},$$

where $a < price.v < b$ and k is the incremental for the generation of x_i^- and x_i^+, $i = \overline{1, \ldots}$

$$x_i^- = price.v - k * i,\ \ a < price.v - k * i < price.v,$$
$$x_i^+ = price.v + k * i,\ \ price.v < price.v + k * i < b,$$

and where

$$\mu(x_i^-) = \frac{x_i^- - a}{price.v - a} - \delta_i^-,\ \ \delta_i^- = 1 - \mu(x_i^-) - \nu(x_i^-),\ \ \nu(x_i^-) = 1 - \mu(x_i^-),$$
$$\mu(x_i^+) = \frac{b - x_i^+}{b - price.v} - \delta_i^+,\ \ \delta_i^+ = 1 - \mu(x_i^+) - \nu(x_i^+),\ \ \nu(x_i^+) = 1 - \mu(x_i^+).$$

As we can see, the fuzzy class of objects $HomeFridge$ has a measure of its fuzziness, which is equal to 0.96 according to Definition 1.

Now let us suppose that the second fuzzy homogeneous class of objects $HotelFridge$ defines the fuzzy type t_{hf2}, which describes a fuzzy concept of a hotel fridge, which has the following representation:

$$HotelFridge($$
$$p_1 = (refrigerator\_volume, (v \in T(r\_volume), str))/1,$$
$$p_2 = (freezer\_volume, (v \in T(f\_volume), str))/0.78,$$

$$
\begin{aligned}
p_3 &= (cee, (v \in V_{cee}, str))/1,\\
p_4 &= (aec, (v \in V_{aec}, kWh))/1,\\
p_5 &= (sizes, ((v_1 \in V_{height}, cm), (v_2 \in V_{width}, cm), (v_3 \in V_{depth}, cm))/1,\\
p_6 &= (compactness, (vf_6(HotelFridge.sizes), v \in [0,1]))/0.93,\\
p_7 &= (color, (v \in V_{color}, str))/1,\\
p_8 &= (weight, (v \in T(weight), str))/1,\\
p_9 &= (noisiness, (v \in T(noisiness), str))/0.82,\\
p_{10} &= (price, (v \in V_{price}, \mathbb{N}^+))/1,\\
f_1 &= get\_crisp\_weight()/0.93,\\
f_2 &= get\_fuzzy\_price(a,b,k)/0.87\\
)&/0.94,
\end{aligned}
$$

refrigerator volume is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(r\_volume) = \{small, medium, big\},$$

where fuzzy variables *small*, *medium*, and *big* are defined over the interval of integer numbers $U = [30, 45]$, which means the volume of the refrigerator in cm^3, and have the following interpretation:

$$
\begin{aligned}
M(small) &= \{30/1 + 31/0.95 + 32/0.91 + 33/0.87 + 34/0.82\}\ cm^3,\\
M(medium) &= \{35/0.9 + 36/0.96 + 37/1 + 38/0.96 + 39/0.9\}\ cm^3,\\
M(big) &= \{40/0.78 + 41/0.83 + 42/0.89 + 43/0.95 + 44/1\}\ cm^3;
\end{aligned}
$$

freezer volume is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(f\_volume) = \{small, medium, big\},$$

where fuzzy variables *small*, *medium*, and *big* are defined over the interval of real numbers $U = [5, 6.5]$, which means the volume of the freezer in cm^3, and have the following interpretation:

$$
\begin{aligned}
M(small) &= \{5.0/1 + 5.1/0.92 + 5.2/0.87 + 5.3/0.82 + 5.4/0.78\}\ cm^3,\\
M(medium) &= \{5.5/0.8 + 5.6/0.93 + 5.7/1 + 5.8/0.93 + 5.9/0.8\}\ cm^3,\\
M(big) &= \{6.0/0.82 + 6.1/0.88 + 6.2/0.93 + 6.3/0.97 + 6.4/1\}\ cm^3;
\end{aligned}
$$

cee is a crisp quantitative property, which means the class of energy efficiency to which the fridge belongs, and is defined over the set of string values $V_{cee} = \{A^{+++}, A^{++}, A^{+}, A, B, C, D, F\}$; *aec* is a crisp quantitative property, which means the annual energy consumption by the fridge in kWh, and is defined over the following interval of integer numbers $V_{aec} = [95, 110]$; *sizes* is a crisp multiple-valued quantitative property, which means dimensions of the fridge in cm, and is defined over the intervals of integer numbers $V_{height} = [45, 55]$, $V_{width} = [40, 60]$, $V_{depth} = [40, 50]$; *compactness* is a fuzzy qualitative property defined by verification function

$$vf_6(HotelFridge) : HotelFridge.sizes \to [0, 1],$$

where $vf_6(HotelFridge)$ is defined as follows

$$vf_6(HotelFridge) = \frac{V - V_{min}}{V_{max} - V_{min}},$$

where

$$V = sizes.v_1 \cdot sizes.v_2 \cdot sizes.v_3,$$
$$V_{min} = \min(V_{height}) \cdot \min(V_{width}) \cdot \min(V_{depth}),$$
$$V_{max} = \max(V_{height}) \cdot \max(V_{width}) \cdot \max(V_{depth});$$

color is a crisp quantitative property, which means the color of the fridge, and is defined over the following set of string values

$$V_{color} = \{beige, white, graphite, golden, brown, red, stainless\ steel, silver, grey, titanium, black, bronze, blue, green, orange, pink, ivory, purple\};$$

weight is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(weight) = \{lightweight, medium\},$$

where fuzzy variables *lightweight*, and *medium* are defined over the interval of integer numbers $U = [14, 23]$, which means the weight of the fridge in kg, and have the following interpretation:

$$M(lightweight) = \{14/1 + 15/0.95 + 16/0.91 + 17/0.87 + 18/0.81\}\ kg,$$
$$M(medium) = \{19/0.96 + 20/1 + 21/0.95 + 22/0.91 + 23/0.87\}\ kg;$$

noisiness is a fuzzy quantitative property defined as a linguistic variable, which has the following term-set

$$T(noisiness) = \{low, medium\},$$

where fuzzy variables *low*, and *medium* are defined over the interval of real numbers $U = [35, 42]$, which means the noisiness of the fridge in dB, and have the following meaning:

$$M(low) = \{35/1 + 36/0.94 + 37/0.89 + 38/0.83\}\ dB,$$
$$M(medium) = \{39/0.95 + 40/1 + 41/0.95 + 42/0.93\}\ dB;$$

price is a crisp quantitative property, which means the price of the fridge in UAH, and is defined over the interval of integer numbers $V_{price} = [2800, 4700]$; *get_crisp_weight* is a fuzzy method that computes defuzzification representation of the fuzzy quantitative property *weight* and defined in the following way:

$$get\_crisp\_weight() = \frac{\sum_{i=1}^{|weight.v|} \mu(weight.v) \cdot weight.v}{\sum_{i=1}^{|weight.v|} \mu(weight.v)};$$

get_fuzzy_price is a fuzzy method that computes fuzzification representation of the crisp quantitative property *price* and defined in the following way:

$$get\_fuzzy\_price(a, b, k) = \{x_i^-/\mu(x_i^-), price.v/1, x_i^+/\mu(x_i^+)\},$$

where $a < price.v < b$ and k is the incremental for the generation of x_i^- and x_i^+, $i = \overline{1, \ldots}$

$$x_i^- = price.v - k * i, \;\; a < price.v - k * i < price.v,$$
$$x_i^+ = price.v + k * i, \;\; price.v < price.v + k * i < b,$$

and where

$$\mu(x_i^-) = \frac{x_i^- - a}{price.v - a} - \delta_i^-, \;\; \delta_i^- = 1 - \mu(x_i^-) - \nu(x_i^-), \;\; \nu(x_i^-) = 1 - \mu(x_i^-),$$
$$\mu(x_i^+) = \frac{b - x_i^+}{b - price.v} - \delta_i^+, \;\; \delta_i^+ = 1 - \mu(x_i^+) - \nu(x_i^+), \;\; \nu(x_i^+) = 1 - \mu(x_i^+).$$

As we can see, the fuzzy class of objects $HotelFridge$ has a measure of its fuzziness, which is equal to 0.94 according to Definition 1.

Now let us compute the difference between the fuzzy homogeneous classes of objects $HomeFridge$ and $HotelFridge$ and then vice versa, i.e. $HomeFridge \setminus HotelFridge$ and $HotelFridge \setminus HomeFridge$, using Algorithm 1. Analyzing the specifications and signatures of the fuzzy homogeneous classes of objects $HomeFridge$ and $HotelFridge$, we can conclude that

$$HomeFridge.cee \equiv HotelFridge.cee,$$
$$HomeFridge.compactness \equiv HotelFridge.compactness,$$
$$HomeFridge.color \equiv HotelFridge.color,$$
$$HomeFridge.get\_crisp\_weight() \equiv HotelFridge.get\_crisp\_weight(),$$
$$HomeFridge.get\_fuzzy\_price(a, b, k) \equiv HotelFridge.get\_fuzzy\_price(a, b, k).$$

Therefore, using Algorithm 1 we have constructed two fuzzy homogeneous classes of objects, which have the following representations:

$$\begin{aligned}
&HomeFridge \setminus HotelFridge(\\
&\quad p_1 = (refrigerator_volume, (v \in T(r_volume), str))/1, \\
&\quad p_2 = (freezer_volume, (v \in T(f_volume), str))/1, \\
&\quad p_3 = (aec, (v \in V_{aec}, kWh))/1, \\
&\quad p_4 = (sizes, ((v_1 \in V_{height}, cm), (v_2 \in V_{width}, cm), (v_3 \in V_{depth}, cm))/1, \\
&\quad p_5 = (weight, (v \in T(weight), str))/1, \\
&\quad p_6 = (noisiness, (v \in T(noisiness), str))/0.75, \\
&\quad p_7 = (price, (v \in V_{price}, \mathbb{N}^+))/1, \\
&)/0.96,
\end{aligned}$$

$$\begin{aligned}
&HotelFridge \setminus HomeFridge(\\
&\quad p_1 = (refrigerator_volume, (v \in T(r_volume), str))/1, \\
&\quad p_2 = (freezer_volume, (v \in T(f_volume), str))/0.78,
\end{aligned}$$

$$
\begin{aligned}
&\quad p_3 = (aec, (v \in V_{aec}, kWh))/1,\\
&\quad p_4 = (sizes, ((v_1 \in V_{height}, cm), (v_2 \in V_{width}, cm), (v_3 \in V_{depth}, cm))/1,\\
&\quad p_5 = (weight, (v \in T(weight), str))/1,\\
&\quad p_6 = (noisiness, (v \in T(noisiness), str))/0.82,\\
&\quad p_7 = (price, (v \in V_{price}, \mathbb{N}^+))/1,\\
&)/0.94,
\end{aligned}
$$

Created classes represent unique parts of fuzzy homogeneous classes of objects $HomeFridge$ and $HotelFridge$. Class $HomeFridge \setminus HotelFridge$ has a measure of fuzziness, which is equal to 0.96 according to Definition 1, and defines the fuzzy type $t_{hf1\setminus hf2} \subseteq t_{hf1}$, which consists of properties and methods which are typical only for the fuzzy homogeneous class of objects $HomeFridge/0.96$. Class $HotelFridge \setminus HomeFridge$ has a measure of fuzziness, which is equal to 0.94 according to Definition 1, and defines the fuzzy type $t_{hf2\setminus hf1} \subseteq t_{hf2}$, which consists of properties and methods which are typical only for the fuzzy homogeneous class of objects $HotelFridge/0.94$.

As the result, Algorithm 1 provides an opportunity to verify the difference of one fuzzy homogeneous class of objects from another one, as well as to compute it in the form of a new fuzzy homogeneous class of objects if such difference exists. Fuzzy homogeneous classes of objects, which are dynamically created by the algorithm, allow a KBS to estimate both the similarity and the difference between extracted or acquired new knowledge items and those ones, which already integrated within the knowledge base since the difference and similarity are inverse concepts. Results of such analysis can be used for the efficient integration of new knowledge into the knowledge base.

5. Conclusions

To perform the integration of new knowledge into the knowledge base efficiently, a KBS should be able to analyze and to compare extracted or acquired new knowledge items with those ones, which were integrated previously. A system should verify the difference and similarity between new knowledge items and previously obtained ones to perform the recognition stage of the knowledge integration process. For this purpose, we defined the concept of the universal difference exploiter for fuzzy homogeneous classes of objects and developed a corresponding algorithm for its implementation. The developed algorithm provides an opportunity to verify as well as to compute the difference and the similarity between extracted or acquired new knowledge items and previously obtained ones, in terms of fuzzy homogeneous classes of objects, within such knowledge representation model as fuzzy object-oriented dynamic networks.

Similar to other universal exploiters, difference exploiter can be adapted to compute the difference of fuzzy inhomogeneous classes of objects as well as for the difference of fuzzy homogeneous and inhomogeneous classes of objects. However, such extensions require the development of the appropriate algorithms for their implementation.

References

[1] R. Bareiss, B. W. Porter, K. S. Murray, Supporting Start-to-Finish Development of Knowledge Bases, Mach. Learn. 4 (1989) 259–283. doi:10.1007/BF00130714.

[2] K. S. Murray, B. W. Porter, Controlling Search for the Consequences of New Information during Knowledge Integration, in: Proceedings of the 6th International Workshop on Machine Learning, New York, USA, 1989, pp. 290–295.

[3] K. S. Murray, B. W. Porter, Developing a tool for knowledge integration: initial results, Int. J. Man-Machine Studies 33 (1990) 373–383.

[4] K. S. Murray, Learning as Knowledge Integration, Ph.D. thesis, Faculty of the Graduate School, University of Texas at Austin, Austin, Texas, USA, 1995.

[5] K. S. Murray, KI: A Tool for Knowledge Integration, in: Proceedings of the 13th National Conference on Artificial Intelligence, AAAI 1996, Portland, Oregon, USA, 1996, pp. 835–842.

[6] T. D. Ndousse, Intelligent Systems Modeling with Reusable Fuzzy Objects, Int. J. Intell. Syst. 12 (1997) 137–152. doi:10.1002/(SICI)1098-111X(199702)12:2<137::AID-INT2>3.0.CO;2-R.

[7] G. Bordogna, G. Pasi, D. Lucarella, A Fuzzy Object-Oriented Data Model for Managing Vague and Uncertain Information, Int. J. Intell. Syst. 14 (1999) 623–651. doi:10.1002/(SICI)1098-111X(199907)14:7<623::AID-INT1>3.0.CO;2-G.

[8] L. A. Zadeh, Fuzzy sets, Inform. Control 8 (1965) 338–353. doi:10.1016/S0019-9958(65)90241-X.

[9] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – I, Inform. Sci. 8 (1975) 199–249. doi:10.1016/0020-0255(75)90036-5.

[10] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – II, Inform. Sci. 8 (1975) 301–357. doi:10.1016/0020-0255(75)90046-8.

[11] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – III, Inform. Sci. 9 (1975) 43–80. doi:10.1016/0020-0255(75)90017-1.

[12] N. Marín, O. Pons, M. A. Vila, Fuzzy Types: A New Concept of Type for Managing Vague Structures, Int. J. Intell. Syst. 15 (2000) 1061–1085. doi:10.1002/1098-111X(200011)15:11<1061::AID-INT5>3.0.CO;2-A.

[13] N. Marín, O. Pons, M. A. Vila, A Strategy for Adding Fuzzy Types to an Object-Oriented Database System, Int. J. Intell. Syst. 16 (2001) 863–880. doi:10.1002/int.1039.

[14] F. Berzal, N. Marín, O. Pons, M. A. Vila, Managing Fuzziness on Conventional Object-Oriented Platforms, Int. J. Intell. Syst. 22 (2007) 781–803. doi:10.1002/int.20228.

[15] F. Berzal, N. Marín, O. Pons, M. A. Vila, Using Classical Object-Oriented Features to Build a Fuzzy O-O Database System, in: J. Lee (Ed.), Software Engineering with Computational Intelligence, volume 121 of *Studies in Fuzziness and Soft Computing*, Springer, 2003, pp. 131–155. doi:10.1007/978-3-540-36423-8_6.

[16] A. Zvieli, P. P. Chen, Entity – Relationship modeling and fuzzy databases, in: Proc. IEEE 2nd Int. Conf. Data Eng., Los Angeles, CA, USA, 1986, pp. 320–327. doi:10.1109/ICDE.1986.7266236.

[17] Z. M. Ma, W. J. Zhang, W. Y. Ma, Extending object-oriented databases for fuzzy information modeling, Information Systems 29 (2004) 421–435. doi:10.1016/S0306-4379(03)00038-3.

[18] Z. M. Ma, L. Yan, F. Zhang, Modeling fuzzy information in UML class diagrams and object-oriented database models, Fuzzy Sets Syst. 186 (2012) 26–46. doi:10.1016/j.fss.2011.06.015.

[19] L. Yan, Z. Ma, F. Zhang, Fuzzy XML Data Management, volume 311 of *Studies in Fuzziness and Soft Computing*, Springer, Berlin, 2014. doi:10.1007/978-3-642-44899-7.

[20] Z. Ma, F. Zhang, L. Yan, J. Cheng, Fuzzy Knowledge Management for the Semantic Web, volume 306 of *Studies in Fuzziness and Soft Computing*, Springer, Berlin, 2014. doi:10.1007/978-3-642-39283-2.

[21] D. O. Terletskyi, Run-Time Class Generation: Algorithms for Decomposition of Homogeneous Classes, in: A. Lopata, R. Butkienė, D. Gudonienė, V. Sukackė (Eds.), Information and Software Technologies. ICIST 2020, volume 1283 of *CCIS*, Springer, 2020, pp. 243–254. doi:10.1007/978-3-030-59506-7_20.

[22] D. A. Terletskyi, A. I. Provotar, Fuzzy Object-Oriented Dynamic Networks. I, Cybern. Syst. Anal. 51 (2015) 34–40. doi:10.1007/s10559-015-9694-0.

[23] D. A. Terletskyi, A. I. Provotar, Fuzzy Object-Oriented Dynamic Networks. II, Cybern. Syst. Anal. 52 (2016) 38–45. doi:10.1007/s10559-016-9797-2.

[24] D. O. Terletskyi, O. I. Provotar, Algorithm for Intersection of Fuzzy Homogeneous Classes of Objects, in: Proc. IEEE 2020 15th Int. Sci. Tech. Conf. Comput. Sci. Inform. Technol. (CSIT), volume 2, Zbarazh, Ukraine, 2020, pp. 314–317. doi:10.1109/CSIT49958.2020.9321914.

[25] D. O. Terletskyi, O. I. Provotar, Intersection of Fuzzy Homogeneous Classes of Objects, in: N. Shakhovska, M. O. Medykovskyy (Eds.), Advances in Intelligent Systems and Computing V, volume 1293 of *AISC*, Springer, 2020, pp. 306–323. doi:10.1007/978-3-030-63270-0_21.

[26] P. K. Panigrahi, A. Goswami, Algebra for Fuzzy Object Oriented Database Language, Int. J. Comput. Appl. 26 (2004) 1–9. doi:10.1080/1206212X.2004.11441721.

[27] L. Yan, Z. M. Ma, Operations in Fuzzy Object-Oriented Databases, in: Proc. 2009 Int. Conf. Comput. Intell. Softw. Eng., Wuhan, China, 2009, pp. 1–4. doi:10.1109/CISE.2009.5364610.

[28] L. Yan, Z. M. Ma, F. Zhang, Algebraic operations in fuzzy object-oriented databases, Inf. Syst. Front. 16 (2014) 543–556. doi:10.1007/s10796-012-9359-8.

[29] Z. M. Ma, F. Mili, Handling Fuzzy Information in Extended Possibility-Based Fuzzy Relational Databases, Int. J. Intell. Syst. 17 (2002) 925–942. doi:10.1002/int.10057.

[30] T. T. Nguyen, B. V. Doan, C. N. Truong, T. T. T. Tran, A New Approach for Query Processing and Optimization in Fuzzy Object-Oriented Database, in: V. Bhateja, B. L. Nguyen, N. G. Nguyen, S. C. Satapathy, D.-N. Le (Eds.), Information Systems Design and Intelligent Applications, volume 672 of *AISC*, Springer, 2018, pp. 49–63. doi:10.1007/978-981-10-7512-4_6.

[31] O. Pivert, H. Prade, A Certainty-Based Model for Uncertain Databases, IEEE Trans. Fuzzy Syst. 23 (2015) 1181–1196. doi:10.1109/TFUZZ.2014.2347994.

[32] D. V. Thang, Algebraic Operations in Fuzzy Object-Oriented Databases Based on Hedge Algebras, in: V. E. Cong Vinh P., Ha Huy Cuong N. (Ed.), Context-Aware Systems and Applications, and Nature of Computation and Communication. ICTCC 2017, ICCASA 2017, volume 217 of *LNICST*, Springer, 2018, pp. 124–134. doi:10.1007/978-3-319-77818-1_

12.
[33] Z. M. Ma, J. Liu, L. Yan, Fuzzy Data Modeling and Algebraic Operations in XML, Int. J. Intell. Syst. 25 (2010) 925–947. doi:10.1002/int.20424.
[34] L. Bai, L. Zhu, An Algebra for Fuzzy Spatiotemporal Data in XML, IEEE Access 7 (2019) 22914–22926. doi:10.1109/ACCESS.2019.2898228.
[35] L. Bai, X. Cao, W. Jia, Uncertain spatiotemporal data modeling and algebraic operations based on XML, Earth. Sci. Inform. 11 (2018) 109–127. doi:10.1007/s12145-017-0322-6.
[36] L. Zhang, J. Sun, S. Su, Q. Liu, J. Liu, Uncertainty Modeling of Object-Oriented Biomedical Information in HBase, IEEE Access 8 (2020) 51219–51229. doi:10.1109/ACCESS.2020.2980553.