Algebraic Operations in Fuzzy Object-Oriented Databases Based on Hedge Algebras

Doan Van Thang(✉)

FPT University, Ho Chi Minh, Vietnam
vanthangdn@gmail.com

Abstract. The article is about blurry algebraic operations for classes blurred and objects blurred in blurry oriented object databases based on a semantic approximation approach of algebras of the hedge. It also defines the blurry algebraic operations for blurry classes and blurry objects. Finally, these blurred algebraic operations are applied to the process and query data blurred on the blurry object-oriented database model.

Keywords: Fuzzy object-oriented database · Hedge algebra
Algebraic operations

1 Introduction

The blurry relational database model (FRDB) and blurry object-oriented database (FOOD) model and related problems have been widely studied in recent years by many domestic and foreign authors [1–9]. To implement blurry information in the data model, there are several basic approaches: blurry set theory-based model [7], probability and blurry model [1], etc. All these approaches are designed to reach and treat blurry values to build evaluation methods and comparison among them to manipulate data more accessible and accurately.

Based on the advantages of the structure the algebra of the hedge (HA) [5–7], the authors studied the relational database model [5,6,8], and blurry object-oriented database model [2,3] based on the approaches of algebra of the hedge, in which semantic language quantified by quantitative semantic mapping of algebra of the hedge.

In this approach, the semantics of the language can be expressed in a neighborhood of intervals determined by the measure of fuzziness of language values of an attribute as a linguistic variable.

This article is based on approximate measures of the semantics of the two fuzzy data to define the blurry algebraic operations for objects blurry and classes blurry in FOOD model. These blurry algebraic operations are defined as bases for the treatmenting of blurry data in FOOD model.

P. Cong Vinh et al. (Eds.): ICCASA 2017/ICTCC 2017, LNICST 217, pp. 124–134, 2018.
https://doi.org/10.1007/978-3-319-77818-1_12

This article is organized as follows: Sect. 2 presents some basic concepts related to algebra of the hedge and FOOD as the basis for the following sections. Section 3 offers solutions to detect and manage redundant objects in FOOD. Section 4 defines blurry algebraic operations to classes blurred and objects blurred, and finally the conclusion.

2 The Basic Concepts

In this section, presents a general overview of the algebra of hedge linear full was proposed by Ho et al. And some related concepts on the for mapping quantification and how to determine neighboring semantic quantitative approach HA [5–9].

2.1 Hegde Algebra

Consider a complete hedge algebra (Comp-HA) $\underline{\mathbf{AX}}$ = (X, G, H, Φ, Σ, $\leq$), where G is a set of generators which are designed as primary terms denoted by c^- and c^+, and specific constants 0, W and 1 (zero, neutral and unit elements, respectively), H = $H^+ \cup H^-$ and two artificial hedges Σ, Φ, the meaning of which is, respectively, taking in the poset $\underline{\mathrm{X}}$ the supremum (sup, for short) or infimum (inf, for short) of the set H(x) - the set generated from x by using operations in H. The word "complete" means that certain elements are added to usual hedge algebras in order for the operations Σ and Φ will be defined for all x $\in$ X. Set $Lim(X) = \underline{X} \setminus H(G)$, the set of the so-called limit elements of $\underline{\mathbf{AX}}$.

Definition 1. *A Comp-HAs* $\underline{\boldsymbol{AX}}$ = *(*$\underline{X}$*, G, H,* Σ*,* Φ*,* $\neq$*) is said to be a linear hedge algebra (Lin-HA, for short) if the sets* $G = \{0,\ c^-,\ W,\ c^+,\ 1\}$, $H^+ = h_1, \ldots, h_p$ *and* $H^- = h_{-1}, \ldots, h_{-q}$ *are linearly ordered with* $h_1 < \ldots < h_p$ *and* $h_{-1} < \ldots < h_{-q}$*, where* $p, q > 1$*. Note that* $H = H^- \cup H^+$*.*

Proposition 1. *Fuzziness measures fm and fuzziness measures of* $\mu(h)$*,* $\forall h \in H$*, the following statements hold:*

(1) $fm(hx) = \mu(h)fm(x)$, $\forall x \in X$.
(2) $fm(c^-) + fm(c^+) = 1$.
(3) $\sum_{-q \leq i \leq p, i \neq 0} fm(h_i c) = fm(c)$, *where* $c \in \{c^-, c^+\}$.
(4) $\sum_{-q \leq i \leq p, i \neq 0} fm(h_i x) = fm(x), x \in X$.
(5) $\sum\{\mu(h_i) : -q \leq i \leq -1\} = \alpha$ *and* $\sum\{\mu(h_i) : 1 \leq i \leq p\} = \beta$, *where* $\alpha, \beta > 0$ *and* $\alpha + \beta = 1$.

Definition 2 *(Sign function). Sgn:* $X \to \{-1,\ 0,\ 1\}$ *is a function which is defined recursively as follows, where h, h'* $\in$ *H, and* $c \in \{c^-, c^+\}$*:*

(a) Sgn(c^-*) = −1, Sgn(c+) = +1,*
(b) Sgn(h'hx) = 0, if h'hx = hx otherwise:
Sgn(h'hx) = -Sgn(hx), if h'hx $\in$ *hx and h' is negative with h.*
Sgn(h'hx) = +Sgn(hx), if h'hx $\in$ *hx and h' is positive with h.*

Proposition 2. *With $\forall x \in X$, we have: $\forall h \in H$, if Sgn(hx)= +1 then $hx < x$, if $Sgn(hx) = -1$ then $hx < x$ and if $Sgn(hx) = 0$ then $hx = x$.*

From properties of fuzziness and sign function, semantically quantifying mapping of HA is defined as below:

Definition 3. *Let $\underline{\boldsymbol{AX}}$ = ($\underline{X}$, G, H, Σ, Φ, $\leq$) be a free linear complete HA, fm(x) and $\mu(h)$ are, respectively, the fuzziness measures of linguistic and the hedge h satisfying properties in Proposition 2. Then, v is a induced mapping by fuzziness measure fm of the linguistic if it is determined as follows:*

(1) $v(W) = k = fm(c^-), v(c^-) = k - \alpha fm(c^-) = \beta fm(c^-), v(c^+) = k + \alpha fm(c^+)$.
(2) $v(h_jx) = v(x) + Sgn(h_jx)\{\sum_{i=Sgn(j)}^{j} \mu(h_i)fm(x) - \omega(h_jx)\mu(h_j)fm(x)\}$, where $\omega(h_jx) = \frac{1}{2}[1 + Sgn(h_jx)Sgn(h_ph_jx)(\beta - \alpha)] \in \{\alpha, \beta\}$, for all j, $-q \leq j \leq p$ and $j \neq 0$.
(3) $v(\Phi c^-) = 0, v(\Sigma c^-) = k = v(\Phi c^+), v(\Sigma c^+) = 1$, and for all j, $-q \leq j \leq p$ and $j \neq 0$, we have $v(\Phi h_jx) = v(x) + Sgn(h_jx)\{\sum_{i=sign(j)}^{j-1} \mu(h_i)fm(x)\}$ and $v(\Sigma h_jx) = v(x) + Sgn(h_jx)\{\sum_{i=sign(j)}^{j} \mu(h_i)fm(x)\}$.

2.2 Neighborhood Level

The authors in [8], [10–12] took the blurry intervals of length k as similar long between the elements. This means that the elements of which the representative values belong to the same interval blurred level k which is similar as k-level. However, to build the level k blurry intervals, representative values of the elements of x have length that k is always in the ends of the blurry level k. Therefore, intervals in the determination of k of district level, we expect these representative values as the interior points of the district level k.

We always assume that each set of H^- and H^+ contains at least two hedges. We reviewed by X_k to be a collection of all elements of length k. On the basis of blurred interval on level k and k + 1. The authors [8], [10] built a domain partition [0, 1] as the following:

(1) Similarity level 1: with k = 1, blurry intervals of level 1 includes $I(c^-)$ and $I(c^+)$. Blurry intervals of level 2 on the interval $I(c^+)$ is $I(h_{-q}c^+) \leq I(h_{-q+1}c^+) \leq \ldots \leq I(h_{-2}c^+) \leq I(h_{-1}c^+) \leq v_A(c^+) \leq I(h_1c^+) \leq I(h_2c^+) \leq \ldots \leq I(h_{p-1}c^+) \leq I(h_pc^+)$. When that, we build partitions on same level level 1 consists of the equivalence classes as follows: $S(0) = I(h_pc^-)$; $S(c^-) = I(c^-)\ [I(h_{-q}c^-) \cup I(h_pc^-)]$; $S(W) = I(h_{-q}c^-) \cup I(h_{-q}c^+)$; $S(c^+) = I(c^+)\ [I(h_{-q}c^+) \cup I(h_pc^+)]$ and $S(1) = I(h_pc^+)$.
As we can see that except for the two end-points $v_A(0) = 0$ and $v_A(1) = 1$, the representative values $v_A(c^-)$, $v_A(W)$ and $v_A(c^+)$ are inner points corresponding of similarity classes level 1 $S(c^-)$, S(W) and $S(c^+)$.
(2) Similarity level 2: k = 2, we build partitions similarity classes of level 2. Such, on a blurry interval level 2, $I(h_ic^+) = (v_A(\Phi h_ic^+), v_A(\Sigma h_ic^+)]$ with

the two blurring interval of adjacent is $I(h_{i-1}c^{+})$ and $I(h_{i+1}c^{+})$, We will have form equivalence classes the following: $S(h_i c^{+}) = I(h_i c^{+})\ [I(h_p h_i c^{+}) \cup I(h_{-q} h_i c^{+})]$, $S(\Phi h_i c^{+}) = I(h_{-q} h_{i-1} c^{+}) \cup I(h_{-q} h_i c^{+})$ and $S(\Phi h_i c^{+}) = I(h_p h_i c^{+}) \cup I(h_p h_i c^{+})$, with i so that $-q \leq i \leq p$ and $i \neq 0$.
By similarity, can the construction of partitions the same classes any level k. However, in reality the application according to [6], $k \leq 4$, that is, there is maximum of 4 hegdes consecutive impacts the element up primitive c^{-} and c^{+}. The value of clear and translucent called has similar in the level k if the value represented by them along in a similar class in the level k.

2.3 Fuzzy Object-Oriented Database

Real-world entity applications or abstract concepts are often fairly complex objects. These objects contain a certain set of information on objects and behavior on the basis of the information it. Object attribute and its value to determine information about the object. The case for the value of this: (1) clear values: usually values are the values of the primitive data type such as string or number, or is the set of primitive values; (2) blurry value: this blur value is complex, the language label is used to demonstrate the value of this type. For example, the value of the height attribute of an object is said to be height about 180 cm tall, or maybe a language value "high ability"; (3) Object: in this case usually attributes value can refer to an object another. The object that it refers to may be blurred; (4) Collection: usually this attribute value is the set of values or objects. The inaccuracies of this attribute is the set can blur, or a member of the set is the value of the blur or blur objects.

Thus, an object is blurred because of the lack of information or incorrect information is caused by the value of that attribute the information incorrect, unclear, which collectively fuzzy information.

In FOOD model, a class is defined as a set of properties including attributes and methods for determining objects of this class, each method is represented as an operation function on the object's attribute values. On the other hand, attribute values are imprecise (or fuzzy), so methods for determining this class's objects also become fuzzy and uncertain.

The class to be reviewed is blurred caused by the following: (1) a some of objects of a class are determined can be blurry. (2): the domain of an attribute that is blurred, so a translucent class is formed when class definition this. (3): the class is a translucent class when it is inherited from one or more superclass, in which at least one superclass is a translucent class.

CLASS *class name*
  INHERITES
    class name 1 WITH LEVEL OF level 1
    ...
    class name n WITH LEVEL OF level n
  ATTRIBUTES
    attribute 1: [FUZZY] DOMAIN dom 1: TYPE OF type 1
    ...

attribute n: [FUZZY] DOMAIN dom n: TYPE OF type n
METHODS
...
END

3 Evaluation of Duplicates in Fuzzy Objects

A basic task of algebraic operations is used to determine the semantic relationship between two objects and evaluate if they are duplicates. In this part, we will present methods of evaluation and handling of redundant blurry objects.

3.1 Approximation Level k

Based on the concept of neighborhood level k, the paper offers a definition of approximation level k of the object attributes. Approximation level k is defined as follows:

Definition 4. *Let fuzzy class C defined on the set of attributes Attr(C) =* $a_1, a_2, \ldots, a_n$ *and methods M,* $o_1, o_2 \in C$*. We say that* $o_1.a_i$ *is approximation level k* $o_2.a_i$ *and denoted by* $o_1.a_i \approx_k o_2.a_i$ *if* $o_1.a_i$ *and* $o_2.a_i$ *belong to the equivalence class* $FRN_k(fu)$*. Where* $FRN_k(fu)$ *is a partition interval of the equivalence classes level k.*

Example 1: Suppose that in FOOD, a class named Employees, with a fuzzy attribute *salary*, the values of attribute *salary* are corresponding to one of four objects of class Employees are o_1.salary = high; o_2.salary = 80; o_3.salary = 70; o_4.salary = 90. The neighborhood system is built as follows:

Consider the hedge algebra of the linguistic variable *salary*, where D_{salary} = [0, 100], the generators are {0, low, W, high, 1}, the set of hedges is little, possibly, more, very, $FD_{salary} = H_{salary}$(high) ∪ H_{salary}(low).

Choose fm(high) = 0.60, fm(low) = 0.40, μ(possibly) = 0.15, μ(little) = 0.25, μ(more) = 0.25 and μ(very) = 0.35. [0, 100] is partitioned into five intervals similar *level 1* as follows: fm(very high) * 100 = 0.35*0.60*100 = 21. So S(1)*100 = (79, 100]. (fm(posibly high) + fm (more high))*100 = (0.25*0.60 + 0.15*0.60)*100 = 24, so S(high) = (55, 79]. (fm (little low) + fm (little high))*100 = (0.25*0.60 + 0.25*0.40) * 100 = 25, so S(W) = (30, 55]. (fm (posibly low) + fm (more low)) * 100 = (0.25*0.40 + 0.15*0.40)*100 = 16, so S (low) = (14, 30], and S(0)*100 = [0, 14].

Since then, we have the neighborhood *level 1* of the equivalence classes as follows: $FRN_1(0)$ = [0, 14], FRN_1(low) = (14, 30], FRN_1(W) = (30, 55], FRN_1(high) = (55, 79] and $FRN_1(1)$ = (79, 100].

So, we say that o_1.salary $\approx_1$ o_3.salary because o_1.salary = high $\in FRN_1$(high) and o_3.salary = 70 $\in FRN_1$(high); or o_2.salary $\approx_1$ o_4.salary because o_2.salary = 80 $\in FRN_1(1)$ and o_4.salary = 90 $\in FRN_1(1)$. With level k = 1.

3.2 Redundant Fuzzy Objects

In the precise object-oriented database, an object is considered to be redundant if and only if it is duplicated completely with another object. But in the FOOD model, due to the object is fuzzy, so to evaluate the redundancy of two fuzzy objects o_i and o_j, the paper offers the following definitions:

Definition 5. *Let fuzzy class C with the set of attributes $a_1, a_2, \ldots, a_n$. Let two objects o_i and o_j in the fuzzy class C, k is the partition level and $i \neq j$ considered to be redundant with respect to each other if $\forall k = 1, 2, \ldots, n$, $\forall o_i.a_k$ $\exists o_j.a_k$: $o_1.a_i \approx_k o_2.a_i$, and otherwise. Use denoted $o_1 \approx_k o_2$ to say that o_i is redundant to o_j based on the partition level k, where $k = k_1, k_2, \ldots, k_n$.*

Example 2: Give a fuzzy class C with the set of attributes Attr(C) = {name, age}, and o_1(C) = <name: An, age: 18>; o_2(C) = <name: Binh, age: young>; o_3(C) = <name: Huong, age: 32>; o_4(C) = <name: Nhan, age: 34>.

Suppose k ={0, 1}. Meanwhile, the level of partition for attribute *name* is a k = 0 and k = 1 for the attribute *age*. That is, we only construct the partition level for fuzzy attribute of the object class.

Consider the hedge algebra of the linguistic variable *age*, where D_{age} = [0, 100], the generators are {0, young, W, old, 1}, the set of hedges is {little, possibly, more, very}, $FD_{age} = H_{age}$(young) $\cup$ H_{age}(old).

Choose fm(young) = 0.4, fm(old) = 0.6, μ(possibly) = 0.25, μ(little) = 0.2, μ(more) = 0.15 and μ(very) = 0.4. [0, 100] is partitioned into five intervals similar level 1, and as the same way of calculation in Example 1 the intervals are as follows S(0) = [0, 16), S(young) = [16, 32), S(W) = [32, 52), S(old) = [52, 76), and S(1) = [76, 100].

Since then, we have the neighborhood level 1 of the equivalence classes as follows: FRN_1(0) = [0, 16), FRN_1(young) = [16, 32), FRN_1(W) = [32, 52), FRN_1(old) = [52, 76) and FRN_1(1) = [76, 100].

We have, $o_1.age$ = 18, $o_2.age$ = young $\in$ FRN_1(young), and $o_3.age$ = 32, $o_4.age$ = 34 $\in$ FRN_1(W). It is easy to see that $o_1 \approx_1 o_2$ and $o_3 \approx_1 o_4$, that is o_1 is redundant to o_2 and o_3 is redundant to o_4 with the partition level k= 1.

To remove the redundant fuzzy objects by the partition level k in class C, we combine the redundant objects together until there are no longer two fuzzy objects which are redundant to each other.

Let o_i and o_j are two redundant objects level k in the class C, to remove these redundancies, we will combine o_i and o_j into a new object o. There are three types of combination for fuzzy objects to meet different requirements in the object manipulations.

$$\begin{aligned} o = merge_{\cup_k}(o_i, o_j) &= < merge_{\cup_k}(o_i.a_1, o_j.a_1),\, merge_{\cup_k}(o_i.a_2, o_j.a_2), \\ &\quad \ldots, merge_{\cup_k}(o_i.a_n, o_j.a_n) > \\ o = merge_{-_k}(o_i, o_j) &= < merge_{-_k}(o_i.a_1, o_j.a_1),\, merge_{-_k}(o_i.a_2, o_j.a_2), \\ &\quad \ldots, merge_{-_k}(o_i.a_n, o_j.a_n) > \end{aligned}$$

$$o = merge_{\cap_k}(o_i, o_j) \; =< merge_{\cap_k}(o_i.a_1, o_j.a_1),\; merge_{\cap_k}(o_i.a_2, o_j.a_2),$$
$$\ldots, merge_{\cap_k}(o_i.a_n, o_j.a_n) >$$

In this paper, the object's fuzzy attributes are considered as the linguistic attributes and represented by structure of hedge algebra. Since then, we can construct the partition of equivalence classes level k for the linguistic attributes. Determine the linguistic value of the equivalence class level k and thus determine [a, b] corresponding to the attribute values.

Thus, the combinations of attribute values of two objects are the intersection, composition and subtraction on the intervals.

- $merge_{\cup_k}(o_i.a_1, o_j.a_1)$: union two intervals $o_i.a_1$ = [a, b] and $o_j.a_1$ = [c, d].
- $merge_{\cap_k}(o_i.a_1, o_j.a_1)$: intersect two intervals $o_i.a_1$ = [a, b] and $o_j.a_1$ = [c, d].
- $merge_{-_k}(o_i.a_1, o_j.a_1)$: subtract two intervals $o_i.a_1$ = [a, b] and $o_j.a_1$ = [c, d]. At this point, we have the following cases:
  1. if c ∈ [a, b], and d ∉ [a, b] then result of subtraction is [a, c].
  2. if c ∉ [a, b], and d ∈ [a, b] then result of subtraction is [d, b].
  3. if [c, d] ⊂ [a, b] then result of subtraction is [a, c] ∪ [d, b].
  4. if [c, d] ∩ [a, b] = ∅ then result of subtraction is [a, b].

4 Fuzzy Algebraic Operations

We will present the blurry algebraic operations for blurry classes based on the semantic neighborhood of algebra of the hedge. This paper divides the blurry algebraic operations on FOOD model into two categories: algebraic operations for blurry classes and algebraic operations for blurry objects.

In order to define the algebraic operations for fuzzy objects and fuzzy classes, we first introduction some notations being used below. Let C is a class with attributes $\{a_1, a_2, \ldots, a_n\}$ and denoted by Attr(C), and Attr'(C) is the set of attributes which is obtained from the construction of the partition to determine the equivalence classes for the attribute in the Attr(C). Class C contains the set of (fuzzy) objects, denoted by C = $\{o_1, o_2, \ldots, o_n\}$, and o(C) is the object o of class C.

4.1 Algebraic Operations for Fuzzy Objects

The algebraic operation for fuzzy objects is eventual the fuzzy selection. A selection operation refers to such a procedure that the objects of the classes satisfying a given selection condition are selected. Let C is a fuzzy class, P_f is a fuzzy predicate and denoted by a selection condition and k is the partition level. The selection of P_f in C with the partition level k is defined as follows:

$$\sigma_{P_f}(C) = \{o(C)|o(C) \wedge P_f(o)\}$$

4.2 Algebraic Operations for Fuzzy Classes

4.2.1 Fuzzy Product

Fuzzy product of C_1 and C_2 is a new class C, which is composed of these general attributes of C_1 and C_2, as well as member attributes of C_1 and C_2. Generally, it is required that Attr'(C_1) $\cap$ Attr'(C_2) = $\emptyset$ in the fuzzy product. The objects of class C are generated from the combination of objects from class C_1 and C_2, in which the class C contains attributes Attr'(C_1) and Attr'(C_2).

$$\begin{aligned} C = C_1 \times_k C_2 \; &= \{o(C) | (\forall o_1)(\forall o_2)(o_1(C_1) \wedge o_2(C_2) \wedge o[\mathrm{Attr}'(\mathrm{C}_1)] \\ &= o_1[\mathrm{Attr}'(\mathrm{C}_1]) \wedge \; o[\mathrm{Attr}'(\mathrm{C}_2)] = o_2[\mathrm{Attr}'(\mathrm{C}_2]))\} \end{aligned}$$

4.2.2 Fuzzy Join

For two fuzzy classes C_1 and C_2 with Attr'(C_1) $\cap$ Attr'(C_2) $\neq \emptyset$ and Attr'(C_1) $\neq$ Attr'(C_2). Then join between C_1 and C_2 will form a new class C, have Attr'(C) = Attr'(C_1)$\cap$(Attr'(C_2) - (Attr'(C_1)$\cap$Attr'(C_2))). The objects of class C are created by the composition of the objects from C_1 and C_2, whose semantics are equivalent on Attr'(C_1)$\cap$Attr'(C_2) according to a given partition level, Then:

$$\begin{aligned} C \; = C_1 \triangleright\!\triangleleft_k C_2 = \{&o(C) | (\exists o_1)(\exists o_2)(o_1(C_1) \wedge o_2(C_2) \wedge o[Attr'(C_1) - (Attr'(C_1) \\ &\cap Attr'(C_2))] = o_1[Attr'(C_1) - Attr'(C_1) \cap Attr'(C_2))] \wedge \\ &o[Attr'(C_1) \cap Attr'(C_2)] = merge_{\cap_k}(o_1[Attr'(C_1) \cap Attr'(C_2)], \\ &o_2[Attr'(C_1) \cap Attr'(C_2)]) \wedge o[Attr'(C_2) - (Attr'(C_1) \\ &\cap Attr'(C_2))] = o_2[Attr'(C_2) - (Attr'(C_1) \cap Attr'(C_2))]\} \end{aligned}$$

4.2.3 Fuzzy Union

The fuzzy union between C_1 and C_2 requires Attr'(C_1) = Attr'(C_2), which implies that all the corresponding attributes in C_1 and C_2 must be completely similar. Let a new class C is the fuzzy union of C_1 and C_2, and the objects of the class C are composed of three kinds of objects: the first two kinds are the objects are such objects that directly come from one componed class (for example, C_1) and are not redundant to any objects in another component classes (for example, C_2). Final objects are the objects that are the resulted of merging the redundant objects from two component classes, with k is the partition level. We have:

$$\begin{aligned} C = C_1 \cup_k C_2 \; &= \{o(C) | (\forall o_2)(\exists o_1)(o_2(C_2) \wedge o_1(C_1) \wedge o = o_1) \\ &\vee (\forall o_1)(\exists o_2)(o_1(C_1) \wedge o_2(C_2) \wedge o = o_2) \\ &\vee (\exists o_2)(\exists o_1)(o_1(C_1) \wedge o_2(C_2) \wedge o = merge_{\cup_k}(o_1, o_2)\} \end{aligned}$$

4.2.4 Fuzzy Subtraction

The fuzzy subtraction of C_1 and C_2, also requires Attr'(C_1) = Attr'(C_2), which implies that all the corresponding attributes in C_1 and C_2 must be completely similar. Let a new class C is the fuzzy subtraction of C_1 and C_2, and k is the partition level. Then we have

$$C = C_1 -_k C_2 = \{o(C)|(\forall o_2)(\exists o_1)(o_2(C_2) \wedge o_1(C_1) \wedge o = o_1) \\ \vee(\exists o_1)(\exists o_2)(o_1(C_1) \wedge o_2(C_2) \wedge o = merge_{-_k}(o_1, o_2)\}$$

4.2.5 Fuzzy Intersection

The fuzzy intersetion of C_1 and C_2 is to combine the common objects of these two classes, which requires Attr'(C_1) = Attr'(C_2), which implies that all corresponding attributes in C_1 and C_2 must be completely similar. Let a new class C is the fuzzy intersection of C_1 and C_2, and k is the partition level. We have

$$C = C_1 \cap_k C_2 = \{o(C)|(\exists o_2)(\exists o_1)(o_1(C_1) \wedge o_2(C_2) \wedge = merge_{\cap_k}(o_1, o_2)\}$$

4.2.6 Fuzzy Projection

Let a class C' and S are a subset of the set of attributes of class C'. A new class C is formed from the projection of C' on S is to remove the attributes Attr(C') - S from C' and only retain the attribute S in C'. It is clear that S ⊂Attr(C') and Attr(C) = S. Every object in C' becomes a new object, whose set of attributes only consists of attributes S and remove the attributes Attr(C') - S. Obviously, there may be redundancy in new objects. After removing the possible redundancies, the new objects constitute class C. The projection of C' on S is defined as follows:

$$C = \prod{}^k_S(C') = \{o(C)|(\forall o')(o'(C') \wedge o[S] = o'[S] \wedge o = merge_{\cup_k}(o[S])\}$$

4.3 Fuzzy Queries

The handling of queries in the object-oriented database refers to the method such that the objects that meet a certain condition are selected and distributed to the user according as formats the required. The format of the request includes the attributes that appear in the result and if the result is a group then the attributes will sort in an. A query can be viewed as containing two components that are query conditions and requested formats. In the interest of simple illustration, the formatting requirements will be ignored. An object-oriented query language (OQL) has regular structure like SQL query and is described as follows:

SELECT <list of attributes/methods> **FROM** <list of classes> **WHERE** <query conditions>

In which, <list of attributes/methods> lists the attributes (methods) will appear in the query results, this list has at least one attribute (method). The attributes (methods) in <list of attributes/methods> are select from the associated classes which are specified in the **FROM** statement <list of classes> contains the class names separated by commas: $class_1$, $class_2$,..., $class_n$, from which the attributes/methods are selected with the **SELECT** statement.

<Query conditions> is a logical expression, they always result in truth (1) or false (0), this is precise query conditions.

From above query structure, we can see that classical database suffer from a lacks of flexibility to query. The given query condition and the contents of the database are all crip. A query is flexible if the query condition is imprecise and uncertain information. For example, consider the following query *"show name all the students of possibly young age"*, and in this query *possibly young age* is fuzzy query condition.

Thus, a query in the fuzzy object-oriented database is structured as follows:

SELECT <list of attributes/methods> **FROM** <list of classes> **WHERE** <fuzzy query conditions>

The structure of this fuzzy OQL query is an extension of the structure OQL. Here, <fuzzy query conditions> is a fuzzy condition or combination of fuzzy conditions using the selection and association calculations.

As analyzed above, the domain of the fuzzy attributes of the classes and objects is very complex and can get the values such as numeric values, interval values or linguistic values. Then the identification of objects satisfying the query conditions is matching the attribute values of the objects and the fuzzy query conditions.

Consider a fuzzy class Student with the set of attributes Attr(Student) = name, age, and three objects: o_1(Student) = <name: an, age: young>; o_2(Student) = <name: binh, age: 34>; o_3(Student) = <name: department, age: middle-aged>

Assume have question as follow: Indicate the name the objects of class sinhvien have age is *young*.

Using the algebraic operations above, we can answer this question:

$$\prod\nolimits^{1}_{student.name}(\sigma_{student.age=young})$$

And, OQL statement corresponding: **SELECT** Student.name **FROM** Student **WHERE** Student.age = young.

This query will return all objects of the class Student that satisfy the query conditions is young. Since the age query condition is young, we should only construct a partition level k = 1 for this attribute age. From example 2, we have the neighborhood level 1 of the equivalence classes for the attribute age as follows: $FRN_1(0) = [0, 16)$, $FRN_1(\text{young}) = [16, 32)$, $FRN_1(\text{W}) = [32, 52)$, $FRN_1(\text{old}) = [52, 76)$ and $FRN_1(1) = [76, 100]$.

The received as a result: o_1(an, young), because o_1.young $\in FRN_1(\text{young})$.

5 Conclusion

To base oneself on the semantic quantity the algebra of the hedge, the paper proposes the definition of approximate the level of k of attribute values. Founded on that foundation, the paper proposed the definition of redundant objects and three operations for combining objects to get rid of the redundancy. It also gave the definition of blurry algebraic operations for blurry classes and blurry objects. It also shows the methods for processing blurry queries in FOOD model. The next chapters present case studies. We will proceed to extend blurry queries on the blurry object-oriented database such as preference queries, keyword queries, ranking queries, and so on.

References

1. Cao, T.H., Nguyen, H.: Uncertain and fuzzy object bases: a data model and algebraic operations. Int. J. Uncertainty: Fuzziness Knowl.-Based Syst. **19**(2), 275–305 (2011)
2. De Caluwe, R., Van Gyseghem, N., Cross, V.: Basic notions and rationale of the integration of uncertainty management and object-oriented databases. In: Fuzzy and Uncertain Object-Oriented Databases, pp. 1–20 (1997)
3. Van Ban, D., Tuan, T.C., Van Thang, D.: Querying data with fuzzy information in object-oriented databases based on hedge algebraic semantics. In: Proceedings of the 4th International Conference on Knowledge and Systems Engineering, Danang, pp. 39–45 (2012)
4. Ho, N.C.: Quantifying hedge algebras and interpolation methods in approximate reasoning. In: Proceedings of the 5th International Conference on Fuzzy Information Processing, Beijing, pp. 105–112 (2003)
5. Ho, N.C.: Fuzzy set theory and soft computing technology. Fuzzy system, neural network and application. Publishing science and technology, 37–74
6. Nguyen, C.H., Wechler, W.: Hedge algebras: an algebraic approach to structures of sets of linguistic domains of linguistic truth variable. Fuzzy Sets Syst. **35**(3), 281–293 (1990)
7. Ho, N.C.: A topological completion of refined hedge algebras and a model of fuzziness of linguistic terms and hedges. Fuzzy Sets Syst. **158**, 436–451 (2007)
8. Yan, L., Ma, Z.M., Zhang, F.: Algebraic Operations in Fuzzy Object-oriented Databases. @springer Science + Business Media, Berlin (2012)
9. Ma, Z.M.: Advances in Fuzzy Object-Oriented Databases: Modeling and Applications. Idea Group Publishing, Hershey (2004)