Annales Univ. Sci. Budapest., Sect. Comp. 44 (2015) 165–182

DATA DEPENDENCIES IN FUZZY OBJECT ORIENTED DATABASES MODEL BASED ON HEDGE ALGEBRA

Nguyen Cong Hao (Hue University, Vietnam)
Truong Thi My Le (Quy Nhon, Vietnam)

Communicated by Bui Minh Phong

(Received July 18, 2015; accepted October 10, 2015)

Abstract. The single most important concept in database schemas design is that of a functional dependency. A functional dependency describes the relationship among attributes and is one of the key concepts used in the normalization. On the basic of fuzzy object-oriented databases model with hedge algebra has been studied in [2], in this paper, we introduce the study of the form of data dependencies, including fuzzy functional dependency in object class, fully fuzzy functional dependency, partial fuzzy functional dependency, transitive fuzzy functional dependency and fuzzy functional dependency with quanifier in natural languages. Based on the concepts of fuzzy functional dependencies, the paper also propose the normal forms of fuzzy object-class in the model.

1. Introduction

Data dependencies are theoretical foundation to define normal forms of database schema, in order to limit to the minimum data redundancy, the main

Key words and phrases: Fuzzy functional dependency, fuzzy object-oriented databases, hedge algebra, fuzzy normal forms.
2010 Mathematics Subject Classification: 11A07, 11A25, 11N25, 11N64.

cause breaks in the integrity of data in the database system. When the semantics of the database was extended, and allows storage of uncertainty and incomplete information (called fuzzy information), the semantics of the data dependencies has changed, i.e. to expand types of data dependencies. There have been many projects focused on expanding the types of data dependencies, especially in fuzzy database model [3], [4], [6], [7], [11] and fuzzy object oriented databases [7], [13].

Various fuzzy object oriented databases models have been developed mainly based on fuzzy set theory [13], theoretical possibility [9], [13], similar relationship [13], hedge algebra [2]. In particular, fuzzy object oriented database model was built using hedge algebra is a new approach to help process fuzzy information more efficiently, simply and intuitively. With this approach, a new model of fuzzy object oriented databases have been developed, in which the language semantics is quantified by quantitative mapping of hedge algebra. Meanwhile, the value of language is data, not the labels of fuzzy set representation semantic of linguistic values, and fundamental advantage that it allows to assess semantics of fuzzy information and classical data in a consistent manner [1]. Based on the new approach [2], we proposed several fuzzy data dependences, normal forms of fuzzy object-class and related issues.

2. Fundamental concepts

2.1. Hedge algebra

Hedge algebra is one approach to detecting algebraic structure of the value domain of the linguistic variable. In view of algebra, each value domain of the linguistic variable X can be interpreted as an algebra $AX = (X, G, H, \leq)$, in which Dom (X) is the terms domain of linguistic variable X is generated from a set of primary generators $G = \{c^-, c^+)$ by the impact of the hedges $H = H^- \cup H^+$, W is a neutral element; $\leq$ is an semantically ordering relation on X, it is induced from the natural qualitative meaning of terms. Order structure induced directly so is the difference compared to other approaches. When we add some special elements, then edge algebra become an abstract algebra $\underline{X} = (X, G, H, \Sigma, \Phi, \leq)$, which Σ, Φ are two operators taking the limit of the set terms is generated when affected by the hedges in H. Alternatively, if the symbol $H(x) = \{h_1...h_px/h_1, ...h_p \in H\}$, then $\Phi x = infimumH(x)$ and $\Sigma x = supremumH(x)$. Thus, hedge algebra $\underline{X}$ is built on foundation of Hedge algebra $AX = (X, G, H, \leq)$, where $X = H(G)$, Σ and Φ are two additional operators. Then $X = X \cup Lim(G)$ with $Lim(G)$ is the set of elements limited: $\forall x \in Lim(G), \exists u \in X : x = \Phi u$ or $x = \Sigma u$. The limitation elements are added to hedge algebra X to make the new calculation meant and so $\underline{X}$ =$(X, G, H, \Sigma, \Phi, \leq)$ called complete hedge algebra. The quantitative semantics function (ν), fuzziness measure function (fm), sign function (SGN) and the properties of hedge algebra can reference in the relevant documents [1], [2].

2.2. Similarity level k

When defining the neighborhood level k, we expect such representative value it must be inner point of neighborhood level k. So we define similarity level k as follows: We always assume that each set H^- and H^+ contains at least two hedges. At the X_k is the set of all elements of length k. Based on the fuzzy interval level k and $k+1$ we describe in not form the construction a partition of the domain $0, 1]$ as follows: With $k = 1$, fuzzy interval level 1 includes $I(c^-)$ and $I(c^+)$. The fuzzy interval level 2 on interval $I(c^-)$ is $I(h_p c^-) = = I(h_{p-1}c^-) = \ldots = I(h_2 c^-) = I(h_1 c^-) = \nu_A(c^-) = I(h_{-1}c^-) = = I(h_{-2}c^-) = \ldots = I(h_{-q+1}c^-) = I(h_{-q}c^-)$. Then, we construct a partition at similar level 1 consists of equivalence classes follows: $S(0) = I(h_p c^-)$, $S(c^-) = I(c^-)\backslash[I(h_{-q}c^-) \cup I(h_p c^-]$; $S(W) = I(h_{-q}c^-) \cup I(h_{-q}c^+)$; $S(c^+) = = I(c^+)\backslash[I(h_{-q}c^+) \cup I(h_p c^+)]$ and $S(1) = I(h_p c^+)$.

Similarly, with $k = 2$, we can construct a partition of similar classes level 2. Such on fuzzy interval level 2, $I(h_i c^+) = (\nu_A(\Phi h_i c^+), \nu_A(\Phi h_i c^+)]$ with two nearby fuzzy interval are $I(h_{i-1}c^+)$ and $I(h_{i+1}c^+)$, we will have equivalence classes following: $S(h_i c^+) = I(h_i c^+)\backslash[I(h_p h_i c^+) \cup I(h_{-q}h_i c^+)]$, $S(\Phi h_i c^+) = = I(h_{-q}h_{i-1}c^+) \cup I(h_{-q}h_i c^+)$ and $S(\Sigma h_i c^+) = I(h_p h_{i-1}c^+) \cup I(h_p h_i c^+))$, with $-q \leq i \leq p$ and $i \neq 0$. By the same, we can construct a partition of the equivalence classes level k at any. However, in fact we can limit hedges consecutive action onto primary terms c^- and c^+, denoted by $k^*(k^* \in Z)$. Classic and fuzzy values is called the similar level k if the representative value of their in the same a class similar level k.

3. Fuzzy oriented–object database base on hedge algebra

3.1. Fuzzy objects

The entities in the real world or abstract concepts are often complex objects. These objects contain a certain set of information about objects and actions based on such information. Information about the object is called object properties and is determined by the specific value, this value can be clearly value (the exact value) or for any reason we can not determin its exact value. For example, the value of the attribute "age" of an object is said to be "about 18", or can be a valuable language "very young", here is the fuzzy information. Formally, objects have at least one attribute whose value is a fuzzy set are fuzzy objects.

3.2. Fuzzy classes

The objects have the same properties are gathered into classes that are organized into hierarchies. Theoretically, a class can be considered from two different viewpoints: (a) an extensional class, where the class is defined by the list of its object instances, and (b) an intensional class, where the class is defined by a set of attributes and their admissible values. In addition, a subclass defined from its superclass by means of inheritance mechanism in object oriented database can be seen as the special case of (b) above.

Thus, a class is fuzzy because of the following several reasons: First, some objects of a class are fuzzy, these objects belong to the class with certain degree. Second, when a class is defined, the domain of an attribute may be fuzzy and a fuzzy class is formed.

For example, a class Picture is fuzzy because the domain of its attribute Year is a set of fuzzy values such as long, very long, and about 50 years. Third, a subclass inherits one or more superclasses, in which at least one superclass is fuzzy, then the class is also fuzzy.

In the fuzzy object oriented database, classes are fuzzy because their attribute domain are fuzzy. The issue that an object belongs to a class with a degree level k ($k \in Z$) occurs since a class or an object is fuzzy. Similarly, a class is a subclass of another class as well with a certain degree level k because of the class fuzziness. Therefore, the evaluations of fuzzy object-class relationships and fuzzy inheritance hierarchies are the core of fuzzy object oriented database model.

3.3. Fuzzy object-class relationships

In fuzzy object oriented database, the following four situations can be distinguished for object-class relationships: (a) Crisp class and crisp object: this situation is the same as the object oriented database, ie the object belongs or not to the class certainly; (b) Crisp class and fuzzy object: the class is precisely defined and has precise boundary, the object is fuzzy since its attribute value may be fuzzy. In this case, the object may be a member of the class under certain degree; (c) Fuzzy class and crisp object: being the same as the case in (b), the object may belong to the class with degree level k. For example, a Ph.D. student and a young student class; (d) Fuzzy class and fuzzy object: in this case, the object also belongs to the class with degree level k. The object-class relationships in (b), (c) and (d) above is called fuzzy object-class relationships. In fact, the case (a) can be seen as a special case of the fuzzy object-class relationship, with the degree of the object to the class is one.

According to [1], with a fuzzy linguistic value x, we will define a interval representation for x. In fact, hedges in linguistic values is limited, so there exists a positive integer k^*, such that $0 < |x| \leq k^*$, $\forall x \in X$. For every $x \in X$, put $j = |x|$, for every integer k, $1 \leq k \leq k^*$, minimum neighborhood level k of x, denoted by $O_{min,k}(x)$ is defined as follows: if $k = j$ then $Omin$, $k(x) = I_{k+1}(h_{-1}x) \cup I_{k+1}(h_1 x)$, if $1 \leq k < j$ then $O_{min,k}(x) = I_j(x)$ and if $j+1 \leq k \leq k^*$ then $O_{min,k}(x) = I_{k+1}(h_{-1}y) \cup I_{k+1}(h_1 y)$. Since then, we represent fuzzy linguistic data the following definition:

Definition 3.1. (see [1]) Let $x \in X \cup C$, a interval representation of x is a set intervals $IRp(x)$ is defined:

$$IRp(x) = \{O_{min,k}(x) | 1 \leq k \leq k^*\}.$$

The data representation as above can be used to represent different types of data. For numeric values, this is crisp data, the fuzzy degree of data equal to 0, then each numeric value a is denoted by $[a, a]$, and $O_{min,k}(a) = \{[a, a]\}$, $\forall k : 1 \leq k \leq k^*$ and $IRp(a) = \{[a, a]\}$. As each interval value a is represented by $[a-\varepsilon, a+\varepsilon]$, with ε be considered the radius from center a. Since $[a-\varepsilon, a+\varepsilon]$ is crisp data, so $O_{min,k}([a - \varepsilon, a + \varepsilon]) = \{[a - \varepsilon, a + \varepsilon]\}, \forall k : 1 \leq k \leq k^*$ and $IRp([a - \varepsilon, a + \varepsilon]) = \{[a - \varepsilon, a + \varepsilon]\}$. When x has length less than k, the value $\nu(x)$ is the tip of an equivalence class $I(u)$ in P_k. This can lead to values in the neighborhood of x is not similarity level k. Therefore, we will build a different partition so $\nu(x)$ is the topology of the partition for all x, $|x| \leq k$, as follows:

Let X is complete and linear hedge algebra, with $H^+ = \{h_1, \ldots, h_p\}$ and $H^- = \{h_{-1}, \ldots, h_{-q}\}$, where $p, q > 1$. Put H_1 is a set of weak hedges, H_2 is the set of powerful hedges: $H_1 = \{h_i, h_{-j} | 1 \leq i \leq [p/2], 1 \leq j \leq [q/2]\}$, $H_2 = \{h_i, h_{-j} | [p/2] \leq i \leq p, [q/2] \leq j \leq q\}$. Put $P_{k+1}(H_n) = \{I(h_i y) | y \in X_k, h_i \in H_n\}$, with $n = 1, 2$. Two interval $I(x)$ and $I(y)$ in $P_{k+1}(H_n)$ is called connected together if there exist consecutive intervals in $P_{k+1}(H_n)$ from $I(x)$ to $I(y)$. This relationship will divides $P_{k+1}(H_n)$ into connected components. We have, with each $y \in X_k, P_{k+1}(H_1)$ is divided into clusters of the form $\{I(h_i y) | h_i \in H_1\}$. Furthermore, as $I(h_{-1}y) \leq \nu(y) \leq I(h_1 y)$ or $I(h_1 y) \leq \nu(y) \leq I(h_{-1}y)$ should always have $\nu(y) \in \{I(h_i y) | h_i \in H_1\}$. Now we clustering fuzzy intervals of $P_{k+1}(H_2)$. Suppose $X_k = \{x_s | s = 0, ..., m - 1\}$

consists of m elements are arranged in a sequence such that $x_i \leq x_j, i \leq j$. Symbols $H_2^- = H_2 \cap H^-$ and $H_2^+ = H_2 \cap H^+$. Note that $h_{-q} \in H_2^-$ and $h_q \in H_2^+$. The clusters were born from the fuzzy intervals of $P_{k+1}(H_2)$ there are three types of clusters: cluster on the left $x_0\{I(h_i x_0)|h_i \in H_2^+\}$; cluster on the right $x_{m-1} : \{I(h_i x_{m-1})|h_i \in H_2^+\}$; cluster between x_s and x_{s+1} with $s = 0, ..., m-2$: depends on $Sgn(h_p x_s)$ and $Sgn(h_p x_{s+1})$ as follows:

$$C = \{I(h_i x_s), I(h_j^{'} x_{s+1})|h_i \in H_2^+, h_j^{'} \in H_2^-\}, \text{ if } Sgn(h_p x_s) = +1 \text{ and } Sgn(h_p x_{s+1}) = +1.$$

$$C = \{I(h_i x_s), I(h_j^{'} x_{s+1})|h_i \in H_2^+, h_j^{'} \in H_2^+\}, \text{ if } Sgn(h_p x_s) = +1 \text{ and } Sgn(h_p x_{s+1}) = -1.$$

$$C = \{I(h_i x_s), I(h_j^{'} x_{s+1})|h_i \in H_2^-, h_j^{'} \in H_2^-\}, \text{ if } Sgn(h_p x_s) = -1 \text{ and } Sgn(h_p x_{s+1}) = +1.$$

$$C = \{I(h_i x_s), I(h_j^{'} x_{s+1})|h_i \in H_2^-, h_j^{'} \in H_2^+\}, \text{ if } Sgn(h_p x_s) = -1 \text{ and } Sgn(h_p x_{s+1}) = -1.$$

Set of all the clusters is denoted C. Since $\{S_k(C)|C \in C\}$ is a partition on domain reference, it determines an equivalence relationship and we will call the similar relationship level k. Due to the feature of the partition so with each value of attribute x, there exists only one cluster that $\nu(x) \in S_k(C)$ and we define similar interval level k as follows:

Definition 3.2. (see [1]) For each $C \in C$, we called similar interval level k corresponding to C is : $S_k(C) = \cup\{I(u)|I(u) \in C\}$, then $S_k(x) = S_k(C)$.

Proposition 3.1. (see [1]) *Let $\underline{X}$ be a complete and linear hedge algebra, in which H^+ and H^- has at least two elements. Then:*

1. *For each k, $\{S_k(u)|u \in X \cup C\}$ is uniquely identified and is a partition of the interval* $[0, 1]$*;*
2. *For every $x, u \in X \cup C$, if $\nu(x) \in S_k(u)$ then smallest neighborhood level k of x within $S_k(u)$, ie $O_{min,k}(x) \in S_k(u)$.*

Definition 3.3. For any object o has attributes $\{A_1, A_2, ..., A_n\}$ of class C, $\underline{X}$ is a complete and linear hedge algebra, with each k, $1 \leq k \leq k^*$, S_k is the similar relationship level k on domain of attribute value A_i of the class C. Then, for each $u \in \underline{X}$, the value $o(A_i)$ and u is called the equal level k, denoted $o(A_i) =_k u$, if and only if $O_{min,k}(o(A_i)) \in S_k(u)$.

Definition 3.4. For two any objects o_1, o_2 on attributes $\{A_1, A_2, ..., A_n\}$ of class C, X is a complete and linear hedge algebra, for each k, $1 \leq k \leq k*$, S_k

is similar relationship level k on domain of attribute value A_i of the class C. Then:

1. Two values $o_1(A_i)$ and $o_2(A_i)$ are called the equal level k, denoted $o_1(A_i) =_k o_2(A_i)$, if and only if exist an equivalence class $S_k(u)$ of similar relationship S_k so that $O_{min,k}(o_1(A_i)) \in S_k(u)$ and $O_{min,k}(o_2(A_i)) \in$ $\in S_k(u)$;
2. Two values $o_1(A_i)$ and $o_2(A_i)$ are called the different level k, denoted $o_1(A_i) \neq_k o_2(A_i)$, if not exist an equivalence class $S_k(u)$ of similar relationship S_k so that $O_{min,k}(o_1(A_i)) \in S_k(u)$ and $O_{min,k}(o_2(A_i)) \in S_k(u)$.

Lemma 3.1. (see [1]) *The equal relationship level $k(=_k)$ is an equivalence relationship.*

Corollary 3.1. Let o_1, o_2 are any two objects on attributes $\{A_1, A_2, \ldots, A_n\}$ of class C, S_k is similar relationship level $k(0 < k \leq k^*)$ on the domain of attribute value A_i of class C,

1. If $o_1(A_i) =_k o_2(A_i)$ then $o_1(A_i) =_{k'} o_2(A_i), \forall k^{'} < k$;
2. If $o_1(A_i) \neq_k o_2(A_i)$ then $o_1(A_i) \neq_{k'} o_2(A_i), \forall k^{'} < k$.

4. Data dependency in fuzzy object oriented database model

4.1. Fuzzy functional dependency in object-class

Two objects o_1 and o_2 is called equal level k on set X, denoted $o_1(X) =_k$ $=_k o_2(X)$, if for every $A \in X$, we have $o_1(A) =_k o_2(A)$. Two objects o_1 and o_2 are called different level k on the set X, denoted $o_1(X) \neq_k o_2(X)$, if exists $X \in A : o_1(A) \neq_k o_2(A)$.

Definition 4.1. Let fuzzy object class C with attributes $U, X, Y \subseteq U$, for every integer k and $1 \leq k \leq k^*$. We say that the class C satisfies fuzzy functional dependency X functionally determines Y with level k, denoted $X \sim>_k Y$, if $\forall o_1, o_2 \in C, o_1(X) =_k o_2(X) \Rightarrow o_1(Y) =_k o_2(Y)$.

Algorithm 4.1. Check the class C satisfies fuzzy functional dependency $X \sim>_k Y$?

Input: Class C with set of attributes X, Y and set of objects $\{o_i, i = 1, ..., m\}$.

Output: *True* if C satisfies $X \sim>_k Y$, else *False*.

Method:

1. Construct hedge algebras for fuzzy attributes of X and Y.

2. Construct minimum neighborhood level k for values of objects of X and Y.

3. Browse in turn pairs objects of class C to detect pairs objects do not satisfy the functional dependency:

   For (each object $o_i \in C, i = 1, ...m - 1)$

   For (each object $o_j \in C, j = i + 1, ..., m)$

   If $(o_i(X) =_k o_j(X))$ and $(o_i(Y) \neq_k o_j(Y))$ Return *False*;

   Return *True*.

Algorithm 4.1 ensures stop because number of attribute (n) and number of objects (m) of class C are finite and complexity of the algorithm is $O(m^2 * n)$.

Example 4.1. We consider the class *"Employee"* includes the following attributes:

Class **Employee** {

Oid : allID

Name : string

Department : string

Job : string

Experience :[fuzzy] domain [0 .. 40]: float

Salary : [fuzzy] domain [2..30]: float

Income tax : [fuzzy] domain [0 .. 4.15]: float

}

Where Name, Department and Job are classic attributes, but Experience, Salary and Income tax are fuzzy attributes. In fact, between attributes may exist inexact relationships such as "The employees in the same parts have similar job and experience must have nearly equal salary" ..., such relationships are called fuzzy functional dependencies. Let us consider some of the objects of class *"Employee"* as follows:

Oid	Name	Department	**Job**	Experience	Salary	Income tax
oid1	Binh	Technical	Engineer	25	15	0.36
oid2	Lan	Accounting	Accountancy	more low	more low	0
oid3	Minh	Technical	Manager	very high	very high	very high
oid4	Tuan	Technical	Engineer	about 26	possibly high	high
oid5	Van	Accounting	Accountancy	5.5	5	0

Table 4.1. Class *"Employee"*

Suppose we have the required test class *"Employee "* satisfy fuzzy functional dependency: Department, Job, Experience $\sim>_2$ Salary?

Experience, Salary and Income tax are fuzzy attributes, those values are number, interval and linguistic values, so first, we will build a hedge algebra for each fuzzy attribute:

– For the attribute Experience, we have:
G = $\{low, high\}$, $H^- = \{little, possibly\}$ and $H^+ = \{more, very\}$. The fuzzy parameters: $fm(low) = 0.35$; $fm(high) = 0.65$; $\mu(very) = 0.3$; $\mu(more) = 0.25$; μ $(possibly) = 0.2$; $\mu(little) = 0.25$. cdom(Experience) = [0, 40], we use coefficient $r = 40$ to convert from [0, 1] to [0, 40].

+ With $k = 1$, we have: $|I_r(low)| = fm(low) \times 40 = 14$. $Ir(low) = [0.0, 14]$, $fm_r(very\ low) = \mu(very) \times fm(low) \times 40= 4.2$, $fm_r(more\ low) = \mu(more) \times fm(low) \times 40 = 3.5$. Because $\{I_r(very\ low), I_r(more\ low), I_r(possiblylow), I_r(little\ low)\}$ are a partition of $I_r(low)$, we deduce $I_r(very\ low) = [0.0, 4.2]$, $I_r(more\ low) = (4.2, 7.7]$. So, $O_{min,1}(more\ low) = I_r(more\ low) = (4.2, 7.7]$. Similar, we have: $O_{min,1}(very\ high) = I_r(very\ high) = (32.2, 40]$.

+ With $k = 2$:
$O_{min,2}(more\ low) = I_r(more\ more\ low) \cup I_r(possibly\ more\ low) = (5.25, 6.825]$, $O_{min,2}$ $(very\ high) = I_r(possibly\ very\ high) \cup I_r(more\ very\ high) = = (34.15, 37.66]$.

For number and interval values, we have: $O_{min,k}(25) = [25, 25]$,
$O_{min,k}(5.5) = [5.5, 5.5]$, $O_{min,k}$ $(about\ 26) = [25, 27]$, $\forall 1 \leq k \leq k^*$.

– For the attribute Salary, we have: $G = \{\ low,\ high\ \}$,
$H^- = \{little, possibly\}$ and $H^+ = \{\ more, very\}$. The fuzzy parameters: $fm(low) = 0.25$; $fm(high) = 0.75$; $\mu(very) = 0.3$; $\mu(more) = 0.3$;
μ $(possibly) = 0.25$; $\mu(little) = 0.15$. cdom($Salary$) = [2, 30].

+ With $k = 1$, we have: $O_{min,1}(very\ high) = Ir\ (very\ high) = (23.7, 30]$,
$O_{min,1}(possibly\ high) = Ir\ (possibly\ high) = (12.15, 17.4]$,
$O_{min,1}(more\ low) = Ir(more\ low) = (4.1, 6.2]$.

+ With $k = 2$, we have: $O_{min,2}(very\ high) = I_r(possibly\ very\ high) \cup$ $\cup\ I_r(more\ very\ high) = (24.645, 28.11]$, $O_{min,2}(possibly\ high) =$ $= Ir(possibly\ possibly\ high) \cup I_r(more\ possibly\ high) = (12.9375, 15.825]$,

$O_{min,2}(more\ low) = I_r\ (more\ more\ low) \cup I_r(possibly\ more\ low) =$
$= (4.73,\ 5.885]$.

We found class *"Employee"* satisfy fuzzy functional dependency: Department, Job, Experience $\sim>_2$ Salary for:

Department and *Job* are two crisp attributes, so objects of the class either equal or different with all levels above that properties. Therefore, we only need to consider the objects are equal on these crisp attributes:

– At the first object (oid_1) and fourth (oid_4), we have:
$oid_1(Experience) =_2\ oid_4\ (Experience)$ because $\exists\ S_2(high) = I_r(very\ possibly\ high) \cup I_r(little\ more\ high) = (24.14,\ 25.7] \cup (25.7,\ 27.325] = (24.14,\ 27.325]$:
$O_{min,1}\ (oid_1(Experience)) = [25,\ 25] \subseteq \exists\ S_1(high)$ and
$O_{min,1}(oid_4(Experience)) = [25,\ 27] \subseteq \exists\ S_1(high)$. And we have:
$oid_1(Salary) =_2 oid_4(Salary)$ because $\exists\ S_2(possibly\ high) = I_r(possibly\ possibly\ high) \cup I_r(more\ possibly\ high) = (12.9375,\ 14.25] \cup (14.25,\ 15.825] = (12.9375,\ 15.825]$: $O_{min,2}(oid_1(Salary)) = [15,\ 15] \subseteq S_2(possibly\ high)$ and
$O_{min,2}(oid_4(Salary)) = (12.9375,\ 15.825] \subseteq S_2(possibly\ high)$.

– At the second object (oid_2) and fifth (oid_5), we have:
$oid_2\ (Experience) =_2 oid_5(Experience)$ because
$\exists\ S_2(more\ low) = I_r(more\ more\ low) \cup I_r(possibly\ more\ low) = (5.25,\ 6.825]$:
$O_{min,2}(oid_2(Experience)) = (5.25,\ 6.825] \subseteq S_2(more\ low)$ and
$O_{min,2}(oid_4(Experience)) = [5.5,\ 5.5] \subseteq S_2(more\ low)$. And we have:
$oid_2(Salary) =_2 oid_5(Salary)$ because $\exists\ S_2(more\ low) = I_r(more\ more\ low) \cup$
$\cup I_r(possibly\ more\ low) = (4.73,\ 5.885]$: $O_{min,2}(oid_2\ (Salary)) = (4.73,\ 5.885] \subseteq$
$\subseteq S_2(more\ low)$ and $O_{min,2}(oid_5\ (Salary)) = [5,\ 5] \subseteq S_2\ (more\ low)$.

Proposition 4.1. *For fuzzy object class C with set of attributes U, and $X, Y, Z \subseteq U$. We have inference axioms based on fuzzy functional dependencies:*

- *Reflexivity: If $X \supseteq Y$, then $X \sim>_k Y$.*
- *Augmentation: If $X \sim>_k Y$, then $XZ \sim>_k YZ$.*
- *Transitivity: If $X \sim>_k Y$ and $Y \sim>_k Z$, then $X \sim>_k Z$.*
- *Pseudotransitivity: If $X \sim>_{k1} Y$ and $Y \sim>_{k2} Z$, then $X \sim>_{min(k1,k2)} Z$*

These axioms are soundness.

Proof.

- Reflexivity: Since $X \supseteq Y$, so with any two objects o_1, o_2 of class C, if $o_1(X) =_k o_2(X)$ then $o_1(Y) =_k o_2(Y)$. So by definition fuzzy functional dependencies 4.1, we have $X \sim>_k Y$.

- Augmentation: Since class C satisfies fuzzy functional dependency $X \sim>_k Y$, so we have $o_1(X) =_k o_2(X)$ and $o_1(Y) =_k o_2(Y)$, with all $o_1, o_1 \forall C(1)$. Otherwise, we have $o_1(XZ) =_k o_2(XZ)$ so $o_1(X) =_k o_2(X)$ and $o_1(Z) =_k o_2(Z)$ (2). Since (1) and (2) we have $o_1(YZ) =_k o_2(YZ)$. Thus $XZ \sim>_k YZ$.

- Pseudotransitivity:

  - By suppose $X \sim>_{k1} Y$: $o_1(X) =_{k1} o_2(X)$ then $o_1(Y) =_{k1} o_2(Y)$, and $X \sim>_{k1} Y : o_1(Y) =_{k2} o_2(Y)$ then $o_1(Z) =_{k2} o_2(Z)$ (1).
  - Consider the case $k1 > k2$: $o_1(X) =_{k1} o_2(X) \implies o_1(X) =_{k2}$ $=_{k2} o_2(X)$, and $o_1(Y) =_{k1} o_2(Y) \implies o_1(Y) =_{k2} o_2(Y)$ (by Corollary 3.1) (2). Since (1) and (2) we have: $o_1(X) =_{k2} o_2(X) \implies$ $\implies o_1(Z) =_{k2} o_2(Z)$ (3).
  - Consider the case $k1 < k2$: By Corollary 3.1 too, if $o_1(Y) =_{k2}$ $=_{k2} o_2(Y) \implies o_1(Y) =_{k1} o_2(Y)$ and $o_1(Z) =_{k2} o_2(Z) \implies$ $\implies o_1(Z) =_{k1} o_2(Z)$ (4). From (1) and (4) we have: $o_1(X) =_{k1}$ $=_{k1} o_2(X) \implies o_1(Z) =_{k1} o_2(Z)$ (5). Since (3) and (5): $o_1(X) =_{min(k1,k2)} o_2(X) \implies o_1(Z) =_{min(k1,k2)} o_2(Z)$. Thus if $X \sim>_{k1} Y$ and $Y \sim>_{k2} Z$ then $X \sim>_{min(k1,k2)} Z$.

4.2. Fuzzy functional dependency with linguistic quantifiers

The use of the linguistic quantifiers such as *several, most, much, many, few* ... in fuzzy functional dependencies makes describe the data dependencies are flexible and close to reality, such as: "*Most* employees in the same parts have similar job and experience must have approximately equal salary". Zadel [11] divided by quantifiers into two type namely: absolute quantifier and proportion quantifier. Call Q is quantifier of fuzzy functional dependency, O is the original set of objects of the class C, $||O|| = m$ is the number objects of set O, domain $DC = [0..m]$. We can divide the quantifier Q into two cases:

1. Where Q is the absolute quantifier: We denote $||Q||$ is the determined amount of quantifier Q, if Q is increasingly linear quantifier: We construct a function $f_Q^A : D_C \rightarrow \{0, 1\}$ such that $\forall x \in D_C, f_Q^A(x) = 1$ if $x \leq |Q|$ and $f_Q^A(x) = 0$ otherwise. If Q is decreasingly linear quantifier: I construct a function $f_Q^D : D_C \rightarrow \{0, 1\}$ such that $\forall x \in D_C, f_Q^D(x) = 1$ if $x \geq |Q|$ and $f_Q^D(x) = 0$ otherwise.

2. Where Q is the proportion quantifier: Since $Dc = [0, m]$ is continuous so we using linear transformation convert to interval $[0, 1]$. Then we build two fuzzy interval of the two primitive concept *small* and *large*,

denoted by I(*small*) and I(*large*) with corresponding length *fm(small)* and *fm(large)* so that they form a partition of [0, 1]. Next, we build equivalence classes S(1), S(*large*), S(W), S(*small*), S(0). From there, we can assert that the total number of *objects* of class C satisfy fuzzy conditions with the quantifier Q if the total number of objects Q belongs to one of the intervals: $S(1) \times m$, S(*large*) $\times m$, S(W) $\times m$, S(*small*) $\times$ m or S(0) $\times m$.

Definition 4.2. For fuzzy object class C with set of fuzzy attributes U, and X, $Y \subseteq U$. We call $O_{satisfy}$ the set of objects of class C satisfy X and Y with level set k and is defined as follows: $O_{satisfy} = \{o_i \in C : (\exists j \neq i, o_i(X)_k = = o_j(X) \wedge o_i(Y) =_k o_j(Y)) \vee (\forall j \neq i, o_i(X)_k \neq o_j(X))\}$; $O_{dissatisfy}$ is a set of objects of class C satisfy X, but not satisfy Y with level k and are defined: $O_{dissatisfy} = \{o_i \in C : \exists j \neq i, o_i(X)_k = o_j(X) \wedge o_i(Y) \neq_k o_j(Y)\}$.

Definition 4.3. For fuzzy object class C with set of attributes U, and $X, Y \subseteq \subseteq U$. We say that the class C satisfies fuzzy functional dependencies X determines Y with level k and linguistic quantifier Q, denoted by $X \sim>_{kQ} Y$, if:

1. Q is the increasingly linear absolute quantifier then $f_Q^A(||O_{\text{satisfy}}||) = 1$;
2. Q is the decreasingly linear absolute quantifier then $f_Q^D(||O_{\text{satisfy}}||) = 1$;
3. Q is the quantifier "few", then $||Osatisfy||/m \in S(0)$;
4. Q is the quantifier "about a half" then $|Osatisfy||/m \in S(W)$;
5. Q is the quantifier "most" then $||Osatisfy||/m \in S(1)$;
6. Q is the quantifier "all" the $||Osatisfy||/m = 1$.

Algorithm 4.2. Check the class C satisfies fuzzy functional dependency $X \sim>_k QY$?

Input: Class C with set of attributes X, Y, the set of objects $o_i, i = 1, ..., m$, level k and quantifier Q.

Output: True if C satisfies $X \sim>_k QY$, else False.

Method:

1. Construct hedge algebras for fuzzy attributes of X and Y.
2. Construct minimum neighborhood level k for values of objects of X and Y.
3. $O_{satisfy} = O$.

4. Browse in turn objects of class C to detect objects satisfies X but do not satisfy Y with level k:

   For (each object $o_i \in C, i = 1, ..., m-1$)

   if ($\exists j \neq i : (o_i(X) =_k o_j(X))$ and $(o_i(Y) \neq_k o_j(Y))$)

   $O_{satisfy} = O_{satisfy} o_i$;

5. result = *False*;

6. if (Q is absolute quantifier)

7. if ($f_Q^A(||O_{\text{satisfy}}||) = 1$ or $f_Q^D(||O_{\text{satisfy}}||) = 1$) then result = *True*;

8. if (Q is proportion quantifier)

9. Construct intervals S(1), S(*large*), S(W), S(*small*), S(0)

10. Case Q of

11. "few": if $||O_{satisfy}||/m \in S(0)$ then result = *True*;

12. "about a half' : if $||O_{satisfy}||/m \in S(W)$ then result = *True*;

13. "most" : if $||O_{satisfy}||/m \in S(1)$ then result = *True*;

14. "all" : if $||O_{satisfy}||/m = 1$ then result = *True*;

15. Return result.

Algorithm 4.2 ensures stop because number of attributes (n) and number of objects (m) of class C are finite and complexity of the algorithm is $O(m^2 * n)$.

Example 4.2. We consider the class *"Employee"* has structure such as 4.1 and the object as follows:

Oid	Name	Department	Job	Experience	Salary	Income tax
oid1	Binh	Technical	Engineer	25	15	0.36
oid2	Lan	Accounting	Accountancy	more low	more low	0
oid3	Minh	Technical	Manager	very high	very high	high
oid4	Tuan	Technical	Engineer	about 26	possibly high	low
oid5	Van	Accounting	Accountancy	5.5	5	0
oid6	Hai	Technical	Manager	35	about 26	little low
oid7	Nhan	Accounting	Accountancy	about 10	about 8	0
oid8	Danh	Technical	Engineer	8	10	0.05
oid9	Suong	Accounting	Accountancy	10	7	0

Table 4.2. Class "Employee"

Let consider the constraint that use of quantifier: *"Most of the employees in the same parts have similar job and experience must have nearly equal salary "*. This constraint corresponding fuzzy functional dependency *Department, job, experience* $\sim>_{2,Most}$ salary. We use algorithm 4.2 to check class C satisfies fuzzy functional dependency with linguistic quantifier as above or not. Based on the example 4.1, we have the results:

– For the attribute Experience:
$O_{min,k}(oid_1(\text{Experience})) = O_{min,k}(\ 25)=[25,\ 25], \forall 1 \leq k \leq k^*$,
$O_{min,2}(oid_2\ (\text{Experience})) = O_{min,2}(more\ low) = (5.25,\ 6.825]$,
$O_{min,2}(oid_3\ (\text{Experience})) = O_{min,2}\ (very\ high) = (34.15,\ 37.66]$,
$O_{min,k}(oid_4(\text{Experience})) = O_{min,k}(\text{about } 26) = [25,\ 27],\ \forall 1 \leq k \leq k^*$,
$O_{min,k}(oid_5(\text{Experience})) = O_{min,k}(5.5) = [5.5,\ 5.5],\ \forall 1 \leq k \leq k^*$,
$O_{min,2}(oid_6(\ \text{Experience})) = O_{min,k}(\text{high}) = (24.14,\ 27.325]$,
$O_{min,k}(oid_7(\text{Experience})) = O_{min,k}(8) = [8,\ 8],\ \forall 1 \leq k \leq k^*$,
$O_{min,k}(oid_8(\text{Experience})) = O_{min,k}(\text{about } 10) = [9,\ 11],\ \forall 1 \leq k \leq k^*$,
$O_{min,k}(oid_9(\text{Experience})) = O_{min,k}(10) = [10,\ 10], \forall 1 \leq k \leq k^*$.

– For the attribute Salary:
$O_{min,k}(oid_1(Salary)) = O_{min,k}(15) = [15{,}15],\ \forall 1 \leq k \leq k^*$,
$O_{min,2}(oid_2(Salary)) = O_{min,2}(more\ low) = (4.73,\ 5.885]$,
$O_{min,2}(oid_3\ (Salary)) = O_{min,2}(very\ hig\text{h}) = (24.645,\ 28.11]$,
$O_{min,2}(oid_4(Salary)) = O_{min,2}(possibly\ high) = (12.9375,\ 15.825]$,
$O_{min,k}(oid_5(Salary)) = O_{min,k}(5) = [5,\ 5],\ \forall 1 \leq k \leq k^*$,
$O_{min,k}(oid_6(Salary)) = O_{min,k}(\text{about } 26) = [25,\ 27],\ \forall 1 \leq k \leq k^*$,
$O_{min,k}\ (oid_7(Salary)) = O_{min,k}(10) = [10,\ 10],\ \forall 1 \leq k \leq k^*$,
$O_{min,k}(oid_8(Salary)) = O_{min,k}(\text{about } 8) = [7,\ 9],\ \forall 1 \leq k \leq k^*$,
$O_{min,k}(oid_9(Salary)) = O_{min,k}(7) = [7,\ 7], \forall 1 \leq k \leq k^*$.

According to steps 3 and 4 in algorithm 4.2, we built a set $O_{satisfy} = \{oid_1, oid_2\ , oid_3, oid_4, oid_5, oid_6, oid_8\}$. Because "most" is proportion quantifier so we built fuzzy interval S(1). Select *fm(big)* = 0.6, *fm(small)* = 0.4, $\mu(very)$ = 0.4, $\mu(more)$ = 0.15, $\mu(possibly)$= 0.25, $\mu(little)$=0.2. We have *fm(very big)* = 0.4 × 0.6 = 0.24, so S(1) = (0.76, 1]. Moreover, we have $||Osatisfy||/m = 7/9 = 0.777 \in S(1)$, So by definition, the class *"Employee"* satisfies fuzzy functional dependency *Department, job, experience* $\sim>_{2,most}$ salary.

4.3. The forms of fuzzy functional dependencies and the normal forms of fuzzy object class

The main purpose of construction a data model is to create an exact representation of the data, the data relationships and constraints. To achieve this goal, we have identified a set of appropriate relations. One approach to determine the relations is called normalization. Normalization helps to database designer checking the relations were normalized or not to avoid the occurrence of anomalies when update data. First, we study some concepts of k-key, fully fuzzy functional dependency, partial fuzzy functional dependency and transitive fuzzy functional dependency, as the basis for the construction of the normal forms of fuzzy object-class.

4.3.1. The forms of fuzzy functional dependencies

Definition 4.4. Let fuzzy object class C with set of attributes $U, X, Y \subseteq U$, $A \in U$, for each integer k and $1 \leq k \leq k^*$. Then:

1. A is a fully fuzzy functional dependency on X with level k, denoted by $X \sim>_{kF} A$, if $X \sim>_k A$ and not exist $Y \subseteq X$ to let $Y \sim>_k A$.
2. A is a partial fuzzy functional dependency on X with level k, denoted by $X \sim>_{kP} A$, if $X \sim>_k A$ and exist $Y \subseteq X$ to let $Y \sim>_k A$.
3. A is a transitive fuzzy functional dependency on X with level k, denoted by $X \sim>_{kT} A$, if exist $Y \subseteq U$ to let $X \sim>_k Y$, $Y! \sim>_k X$, $Y \sim>_k A, A \notin XY$.

Example 4.3. We consider the class *"Student"* includes attributes {IdStudent, Name, IdClass, NameClass, Subject, scores} and set of fuzzy functional dependencies F={ IdStudent $\sim>_1$ Name, IdClass; IdClass $\sim>_1$ NameClass; IdStudent, Subject $\sim>_1$ scores }.

We have IdStudent, Subject $\sim>_1$ scores is fully fuzzy functional dependency with level 1 because IdStudent $!\sim>_1$ scores and Subject $!\sim>_1$ scores l. We have IdStudent, IdClass $\sim>_1$ NameClass is partial fuzzy functional dependency with level 1 because IdClass $\sim>_1$ NameClass. We have IdStudent $\sim>_1$ NameClass is partial fuzzy functional dependency with level 1 because IdStudent $\sim>_1$ IdClass; IdClass $\sim>_1$ NameClass; NameClass $\notin${ IdStudent, IdClass }.

In object oriented database model, each object has an independent existence of its value by the object identifiers and the identifiers are invisible to user. This can lead to instances of existing objects are equal on all the value of the attributes, this is a form of data redundant in object oriented databases. To overcome this problem, when design object oriented database, it is common to use a set of attributes of the object that its value is used to uniquely identify an object in the class. According to the concept of fuzzy functional dependency with level k, the concept key of fuzzy class can be stated as follows:

Definition 4.5. Let fuzzy object class C with set of attributes U, and $K \subseteq U$. K is called *k-key* of fuzzy object class C if satisfy simultaneously two conditions as follows:

1. $K \sim>_k U$;
2. Not exist $K' \subset K$ to let $K' \sim>_k U$.

4.3.2. The normal forms of fuzzy object class

Definition 4.6. Let fuzzy class C with set of attribute U.

1. A fuzzy class C is in the first fuzzy object normal form with level k, denote by $k-1$FONF, if for every attribute $A \in U$, the A is not type of collections and gets set value, or the A is type of collections and does not get set value.

2. A fuzzy class C is in the second fuzzy object normal form with level k, denote by $k-2$FONF, if C is in $k-1$FONF and all attributes non k-key are fully fuzzy functional dependency level k on all k-key.

3. A fuzzy class C is in the third fuzzy object normal form with level k, denote by $k-3$FONF, if C is in $k-1$FONF and not exist attribute non-k-key is transitive fuzzy functional dependency level k on all k-key.

Example 4.5. Let a class *"Student"* is in normal form $1-1$FONF includes attributes U={ IdStudent, NameStudent, IdClass, NameClass, Subject, scores} and set of fuzzy functional dependencies F={ IdStudent $\sim>_1$ Name, IdClass; IdClass $\sim>_1$ NameClass; IdStudent, Subject $\sim>_1$ scores}.

The set of attributes (IdStudent, Subject) is one $1-$key of class C because IdStudent, Subject $\sim>_1$ U and IdStudent $!\sim>_1$U; Subject $!\sim>_1$U. We see that class C is not in $1-2$FONF because NameStudent, NameClass are partial fuzzy functional dependency on $k-$key (IdStudent, Subject). This is the cause of the abnormal data in class. We can reconstruct the class *" Student "* into classes as *" Student-Class "* and *" Student-Score "* in the following normal form $1-2$FONF:

– Class *"Student-Class"* includes attributes {IdStudent, NameStudent, IdClass, NameClass} and set of fuzzy functional dependencies F_1={ IdStudent $\sim>_1$ NameStudent, IdClass; IdClass $\sim>_1$ NameClass }.

– Class *"Student-Score"* includes attributes { IdStudent, Subject, scores} and set of fuzzy functional dependencies F_2={IdStudent, Subject $\sim>_1$ scores}. but not all the anomalies, due to dependencies exist transitive fuzzy.

The class *"Student-Class"* is in normal form $1-2$FONF, but it still has abnormals because exist transitive fuzzy functional dependency IdStudent $\sim>_{1T}$ NameClass. And the class *"Student-Score"* is in normal form $1-3$FONF.

5. Conclusion

On the basic of fuzzy object-oriented databases model with hedge algebra has been studied in [2], we study and construct some forms of data dependencies in this model. The paper also propose fuzzy functional dependencies in fuzzy object-class, some inference rules based on fuzzy functional dependencies and prove their soundness. Besides that, some other forms of fuzzy functional dependencies are also mentioned as basis for construction of the normal forms of fuzzy object oriented database model. Issue about linguistic quantifiers is also mentioned according own approach and take into fuzzy dependencies, making functional dependencies are flexible and close to reality. Some forms of special data dependencies in fuzzy object-class will be study in following articles.

References

[1] **Nguyen Cat Ho, Le Xuan Vinh and Nguyen Cong Hao,** Unify data and establish similarity relation in linguistic databases base on hedge algebra, *Journal of Computer Science & Cybernetics*, **25(4)** (2009), 314–332.

[2] **Nguyen Cong Hao and Truong Thi My Le,** Fuzzy object-oriented database model base on hedge algebra, *Journal of Computer Science & Cybernetics*, **20(3)** (2012), 129–140.

[3] **Nguyen Cong Hao,** Fuzzy functional dependency with linguistic quantifiers base on hedge algebra, *Journal on Information and Communications Technology*, **22(2)** (2009), 87–93.

[4] **Nguyen Cong Hao,** Fuzzy normal forms an approach hedge algebras, *Journal on Information and Communications Technology*, **17** (2008), 101–107.

[5] **Nguyen Kim Anh,** Normalizing object-oriented database schema, *Journal of Computer Science & Cybernetics*, **19(2)** (2003), 125–130.

[6] **Thuan, H. and T.T. Thanh,** Fuzzy functional dependencies with linguistic quantifiers, *Journal of Computer Science & Cybernetics*, **18(2)** (2002), 97–108.

[7] **Ho Cam Ha and Vu Duc Quang,** Fuzzy function dependencies in fuzzy object-oriented databases, *Journal of HNUE*, **7** (2011), 23–31.

[8] **Raju, H.V.S.N and A.K. Mazumdar,** Fuzzy function dependencies and lossless join decomposition of fuzzy relational database systems, *ACM Transaction of Database Systems*, **13(2)**, (1988), 129–166.

[9] **Cristina-Maria Vladarean,** Extending object-oriented databases for fuzzy information modeling, *S.C. WATERS Romania S.R.L, Romai J.*, **9(1)** (2006), 225–237.

[10] **Zadeh, L.A.,** Fuzzy sets, *Information and Control*, **8** (1965), 338–353.

[11] **Zadeh, L.A.,** A computational approach to fuzzy quantifiers in natural languages, *Computers and Mathematics with Applications*, **9(1)** (1983), 149–184.

[12] **Sadok Ben Yahia, Habib Ounalli and Ali Jaoua,** An extension of classical functional dependency, *Information Science*, **119(3-4)** (1999), 219–234.

[13] **Ma, Z.M.,** *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications*, Idea Group Publishing, 2004.

Nguyen Cong Hao
Hue University
Vietnam
nchao@hueuni.edu.vn

Truong Thi My Le
Quang Trung University
Quy Nhon
Vietnam
Le_truongthi@yahoo.com