Primacy of Fuzzy Relational Databases Based on Hedge Algebras

Le Ngoc Hung[1] and Vu Minh Loc[2(✉)]

[1] Sai Gon University, Ho Chi Minh City, Vietnam
lengochungsg291958@gmail.com
[2] Gia Dinh University of Information Technology, Ho Chi Minh City, Vietnam
vuminhloc@gmail.com

Abstract. Databases (DB) based on fuzzy set (FST), possibility (PT) and extended possibility theory (EPT)…which have many problems that need to be discussed in capturing, representing, storing and manipulating with fuzzy data because these approaches have difficulty implementing. Fuzzy relational databases based on hedge algebras (HA) have approach naturally. So, we will not worry about representing, storing and manipulating fuzzy data. In this paper we will investigate fuzzy relational database based on hedge algebras to clarify three primacies of which: easy to present, update and query data.

1 Introduction

About twenty years ago, prof. Nguyen Cat Ho discovered that linguistic variable domain have computing structure and after that built Hedge Algebras (HA) successfully (see [1]). HA is a new approach to implement more effective on some “hot” fields now such that fuzzy control, fuzzy reason, collect fuzzy knowledge and fuzzy databases…

DB is a field that has been applied in fact widely and deeply and so scientists are very interested in it. Many cases in fact, human have to store and handle fuzzy information. For this reason, fuzzy databases are user’s urgent requirement beside classical databases. As mentioned above, the scientists have developed many approaches to fuzzy databases because they desire to implement fuzzy application as soon as possible. In that approaches, EPT emerged as the best approach, however, it has not reached the desired results yet. Concretely, in EPT, data values of fuzzy attribute domain are possibility distributes and must associate with resemblance relations. This idea in theory seems to be optimal, but when we deploy this model that will encounter obstacles. Regardless of aspect of the "rather hard" in capturing the semantics of fuzzy data, just focus on the fuzzy data representation, we’ll see, what will we do with a database consists of many attributes and (or) tuples? The answer is that we have to build the resemblance relation table that has a lot of columns and rows (up to hundred or thousand …). It is very bulky and not factual.

Fuzzy databases based on hedge algebras having better capturing, presenting, storing and manipulating method than the others because hedge algebras capture fuzzy data naturally and it is flexible enough to represent the inherent natural meaning of

P.C. Vinh et al. (Eds.): ICTCC 2014, LNICST 144, pp. 292–305, 2015
DOI: 10.1007/978-3-319-15392-6_28

fuzzy data. Furthermore, HA is a rich math structure enough to build the tools for manipulating with fuzzy data effectively.

The rest of this paper will be organized as follows: Section 2 will represent the basis concepts of HA, Section 3 will represents the fuzzy database model based on HA, Section 4, Section 5 and Section 6 respectively will present the advantages of the approach based on HA in three aspects, representing, querying and updating data, section 7 is the conclusion of the article.

2 Some Definitions in HA

Definition 2.1 [1]

The Hedge Algebra is denoted by AX = (X, G, H, ≤), where X is a value domain of a linguistic variable.

- G is the set of generators and constants, G = (0, c^-, w, c^+, 1), where 0, w, 1 are constants expressing the smallest element, the largest element and the neutral element in X; c^- and c^+ are the negative generator and positive generator.

- H is the set of hedges that is considered as the unary operations acting on each term in X, $H = H^- \cup H^+$. $H^+ = \{h_1,..., h_p\}$ and $H^- = \{h_{-1}, ..., h_{-q}\}$, p, q >1 are the set of positive hedges and set of negative hedges respectively. They are ordered as follows $h_1 <...< h_p$ và $h_{-1} < ... < h_{-q}$.

(≤) relation is induced from semantic relations on X. We call each linguistic value x of X is a term in the hedge algebra. If the set X and H is the linear ordering, then AX = (X, G, H, ≤) called linear hedge algebra.

Example 2.1

Let's consider linguistic variable "speed", this linguistic variable can receive the linguistic values that are terms such as *fast*, *slow*, *very slow*, *rather fast*, *very fast*, *rather slow* ... and they constitute values domain of speed variable.

In here, with the order relation induced from the natural semantics as follows: very slow < rather slow < slow < rather fast < fast < very fast. Thus, we have the HA: G= {0, c^- = slow, w, c^+ = fast, 1}; H = {h^- = possible, h^+ = very}.

Fast, slow, very slow, rather fast, very fast, rather slow are terms in X.

Definition 2.2 [2]

AX = (X, G, C, H, ≤) is a HA.

A mapping f_m: X → [0, 1] is called a fuzziness measure (abbreviated fm) of the terms in X if:

1. $fm(c^-) + fm(c^+) = 1$ and $\sum_{h \in H} fm(hu) = fm(u)$, with ∀ u ∈ X; in this case, fm is called complete.

2. With constants 0, W and 1 we have fm(0) = fm(W) = fm(1) = 0;

3. With ∀ x, y ∈ X, ∀h ∈ H, $\frac{fm(hx)}{fm(x)} = \frac{fm(hy)}{fm(y)}$, this ratio does not depend on fm (x), fm(y). It is fuzzy measure of h and denoted by μ(h).

Clause 2.2 [2]

Each fuzziness measure fm on X, the following assertions are true:

1. fm(hx) = μ(h)fm(x), với ∀ x ∈ X;
2. $fm(c^-) + fm(c^+) = 1$;
3. $\sum_{-q \le i \le p, i \ne 0} fm(h_i c) = fm(c)$, $c \in \{c^-, c+\}$;
4. $\sum_{-q \le i \le p, i \ne 0} fm(h_i x) = fm(x)$;
5. $\sum_{-q \le i \le -1} \mu(h_i) = \alpha$ and $\sum_{-q \le i \le -1} \mu(h_i) = \beta$,α,β > 0 and α+β=1.

Definition 2.3 [2]

A Sign function: X → {-1, 0, 1} is a mapping defined recursively as follows: for h, h '∈ H and c ∈ {c-, c +}:

1. $Sign(c^-) = -1$, $Sign(c^+) = +1$;
2. Sign (hc) = - Sign (c), if h is negative for c, in contrast to Sign (hc) = + Sign (c);
3. Sign (h'hx) = - Sign (hx), if h'hx ≠ hx and h' is negative for h;Sign (h'hx) = + Sign (hx), if h'hx ≠ hx and h' is positive for h;
4. Sign (h'hx) = 0 if h'hx = hx.

Definition 2.4 [2]

AX = (X, G, C, He, ∑Φ, ≤) is a HA(complete linear hedge algebra).

A mapping υ: X → [0, 1] is called semantic quantitative mapping (abbreviated SQM) of AX, the following assertions are true:

1. υ is the 1 - 1 mapping from X on [0, 1] and maintain order on X. With ∀ x, y ∈ X, x <y ⇒υ(x) <υ (y) and υ (0) = 0, υ (1) = 1, with 0, 1 ∈ c;
2. ∀ x ∈ X, υ (Φ x) = infimum υ (H (x)) and υ (∑ x) = supremum υ (H (x)).

Definition 2.5 [2, 4, 5]

With fm is a fuzziness measure on X, A mapping υ: X → [0, 1] induced by fm on X that is defined as follows:

1. $\upsilon(W) = \theta = fm(c^-)$, $\upsilon(c^-) = \theta - \alpha fm(c^-) = \beta fm(c^-)$, $\upsilon(c^+) = \theta + \alpha fm(c^+)$;

2. $\upsilon(h_jx) = \upsilon(x) + Sign(h_jx)\{\sum_{i=Sign(j)}^{j} fm(h_ix) - \omega(h_jx)fm(h_jx)\}$; with j ∈ {j: -q ≤ j ≤ p and j ≠ 0}(*) and $\omega(h_jx) = \frac{1}{2}[1 + Sign(h_jx)Sign(h_ph_jx)(\beta - \alpha)] \in \{\alpha, \beta\}$;1+Sgn(hjx) j−Sign(j)1−Sgn(hj x) j−Sign(j);

3. $\upsilon(\Phi c^-) = 0$, $\upsilon(\Sigma c^-) = \theta = \upsilon(\Phi c^+)$, $\upsilon(\Sigma c^+) = 1$, ∀ j like (*),

$$\upsilon(\Phi h_jx) = \upsilon(x) + Sign(h_jx)\{\sum_{i=Sign(j)}^{j-Sign(j)\frac{1+Sgn(h_jx)}{2}} \mu(h_i)fm(x)\}$$

$$\upsilon(\Sigma h_jx) = \upsilon(x) + Sign(h_jx)\{\sum_{i=Sign(j)}^{j-Sign(j)\frac{1-Sgn(h_jx)}{2}} \mu(h_i)fm(x)\}$$

3 Fuzzy Relational Databases Approach Based on Hedge Algebra

Under this approach, a relational database schema with fuzzy data is a set DB = {U, R_1, R_2, ..., R_m, Const}, here, U = A_1, ... A_n is the universe of properties; R_i is a relational schema; Const is a set of data constraints on the DB.

Each R_i can include two attribute groups, the first group contains the common attributes (classical attributes), the second group contain the fuzzy attributes. Each fuzzy attribute can be viewed as a linguistic variable and value domain of which contains linguistic values (they constitute a hedge algebra) and real values. If A_i is the fuzzy attribute then the value domain of it is D(A_i) = FDom (A_i) ∪ DA_i in which FDom (A_i) is a set of linguistic values, DA_i is the set of normal real values.

FDom (A_i) can receive fuzzy data in common types as follows:

Type 1: fuzzy linguistic data (a very young age)
Type 2: data of interval (the age of a man in (20, 30))
Type 3: undefined data (do not know a student that has a phone number or not?)
Type 4: missing data (my boss will pay me salary but do not know exactly the figure of salary)
Type 5: data is a limited set of certain values (ages is in{31, 33, 35})
Type 6: "do not know" (unknown) data (they have been married but do not know if they have children or not)

To perform comparative operations among fuzzy terms, we have to establish a method of converting semantic representation of linguistic values to the corresponding values over the field of real numbers.

First, we will study the method of representing fuzzy data of type 1, this method will be the basis for representing other type of fuzzy data.

Suppose that attribute A is associated with a ComLin-HA AX = (X, G, C, He, ∑Φ, ≤) and FDom(A) is a finite subset of X. Set d = k (A), is the maximum length of the terms in FDom (A).With fm is the fuzziness measure given of AX. So set of J_k, k = 1, ..., d, and SQM υ induced from fm can be determined.

Based on the structure of ComLin-HAS, all x ∈ linguistic data FDom (A) can be expressed through two semantic components:

(1) a semantic value from the domain of DA;
(2) a finite set of neighbors based on fuzzy intervals.

The first semantic component is determined easily since it is just the value $\upsilon_A(x)$.

To determine the second semantic component more difficult, suppose that x is presented as follows: $x = k_{m-1} \ldots k_1c$, $c \in \boldsymbol{G}$, which means that it has m length. Second semantic component of x is a semantic neighbor systems denoted $Neig^d_{fm}(x)$, here $d = k(A) \geq m$ and fm is a fuzziness measure that was given. For each k, $1 \leq k \leq d$, neighbor of x in $Neig^d_{fm}(x)$ will be determined based on the adjacent k-intervals and is called the k-level neighbor.

To define this concept, we need some concepts as follows:

Denote $\boldsymbol{H}_1 = \{h_i, h_{-j} \in \boldsymbol{H}: 1 \leq i \leq [p/2]\ \&\ 1 \leq j \leq [q/2]\}$ includes "weak" hedges and $\boldsymbol{H}_2 = \{h_i, h_{-j} \in \boldsymbol{H}: [p/2] < i \leq p\ \&\ [q/2] < j \leq q\}$ includes "strong" hedges and $\boldsymbol{INT}_k(\boldsymbol{H}_n) = \{\mathfrak{I}_k(h_iy) \in J_k: y \in \boldsymbol{X}_{k-1}, h_i \in \boldsymbol{H}_n\}$, $n = 1, 2$. Obviously, $\boldsymbol{INT}_k(\boldsymbol{H}_1) \cap \boldsymbol{INT}_k(\boldsymbol{H}_2) = \varnothing$ và $\boldsymbol{INT}_k(\boldsymbol{H}_1) \cup \boldsymbol{INT}_k(\boldsymbol{H}_2) = J_k$ is the set of all k-intervals.

Two intervals $\mathfrak{I}_k(x)$ and $\mathfrak{I}_k(y)$ in $\boldsymbol{INT}_k(\boldsymbol{H}_n)$ are called the connected if there is a string of consecutive k-level fuzziness intervals belong $\boldsymbol{INT}_k(\boldsymbol{H}_n)$ to interconnect $\mathfrak{I}_k(x)$ and $\mathfrak{I}_k(y)$.

In this case, $\mathfrak{I}_k(x)$ is called connected in $\boldsymbol{INT}_k(\boldsymbol{H}_n)$ with every points in $\mathfrak{I}_k(y)$.

Denote k* is a positive integer number that refer to the maximum length of every values on D(A); $|x| \leq k^*$ is the maximum length of linguistic values of x, put $j = |x|$; $\mathfrak{I}(x)$ is a interval of level k contain x through υ mapping; X_k is the set of linguistic values of k-length; U is a universe of attributes in the databases.

a. If k = j: $O_{\min,\, k\,(x)} = \mathfrak{I}_{k+1}(h_{-1}x) \cup \mathfrak{I}_{k+1}(h_1x)$;

b. If $1 \leq k < j$: $O_{\min,\, k\,(x)} = \mathfrak{I}_j(x)$;

c. If $j + 1 \leq j \leq k^*$: $O_{\min,\, k\,(x)} = \mathfrak{I}_{k+1}(\mathrm{GCP}) \cup \mathfrak{I}_{k+1}(h_{l'}x)$, with $l, l' \in \{-q, p\}$.

Put H_1 is the set of "weak hedges" and H_2 is the set of "strong hedges". Concretely, $H_1 = \{h_i, h_j \mid 1 \leq i \leq [p / 2], 1 \leq j \leq [q/2]\}$, $H_2 = \{\{h_i, h_j \mid [p/2] \leq i \leq p, [q/2] \leq j \leq q\}$.

Put $I_{k+1}(H_n) = \{\mathfrak{I}_{k+1}(h_iy) \mid y \in x_k, h_i \in H_n\}$, with n = 1, 2.

Every two intervals $\mathfrak{I}_{k+1}(x)$ and $\mathfrak{I}_{k+1}(y)$ in $I_{k+1}(H_n)$ are called connected to each other if existing the $I_{k+1}(H_n)$ interrupted intervals from $\mathfrak{I}_{k+1}(x)$ to $\mathfrak{I}_{k+1}(y)$. This relation decompose $I_{k+1}(H_n)$ to connected components.

Denote C is cluster of k-level similar intervals with linguistic values x, C will be determined as follows: with $I_{k+1}(H_1) = \{\mathfrak{I}_{k+1}(h_iy) \mid y \in X_k, h_i \in H_1\}$ put $C = \{\mathfrak{I}_{k+1}(h_iy) \mid h_i \in H_1\}$.

With $I_{k+1}(H_1) = \{J_{k+1}(h_iy) \mid y \in X_k, h_i \in H_2\}$, assuming $X_k = \{x_s \mid s = 0, ..., m-1\}$ consist of m elements be arranged in a sequence so that $x_i < x_j$ if $i < j$

$H_2^- = H_2 \cap H^-$ and $H_2^+ = H_2 \cap H^+$. The clusters are generated of the fuzziness intervals of $I_{k+1}(H_2)$ with the following three types:

a. Clusters on the left x_0: put $C := \{\mathfrak{I}_{k+1}(h_ix_0) \mid h_i \in H_2^+\}$.

b. Clusters of the right x_{m-1}: put $C := \{\mathfrak{I}_{k+1}(h_ix_{m-1}) \mid h_i \in H_2^+\}$.

c. Clusters in between x_s and x_{s+1} with s = 0, ..., m-2 dependent on SGN (h_px_s) and the Sgn $(h_px_s +1)$:

C = $\{\mathfrak{I}_{k+1}(h_i x_s), \mathfrak{I}_{k+1}(h'_j x_{s+1}) \mid h_i \in H_2^+, h'_j \in H_2^-\}$, if $Sgn(h_p x_s) = +1$ and $Sign(h_p x_{s+1}) = +1$;

C = $\{\mathfrak{I}_{k+1}(h_i x_s), \mathfrak{I}_{k+1}(h'_j x_{s+1}) \mid h_i \in H_2^+, h'j \in H_2^+\}$, if $Sgn(h_p x_s) = +1$ and $Sign(h_p x_{s+1}) = +1$;

C = $\{\mathfrak{I}_{k+1}(h_i x_s), \mathfrak{I}_{k+1}(h'_j x_{s+1}) \mid h_i \in H_2^-, h'_j \in H_2^-\}$, if $Sgn(h_p x_s) = +1$ and $Sign(h_p x_{s+1}) = +1$;

C = $\{\mathfrak{I}_{k+1}(h_i x_s), \mathfrak{I}_{k+1}(h'_j x_{s+1}) \mid h_i \in H_2^-, h'j \in H_2^+\}$, if $Sgn(h_p x_s) = +1$ and $Sign(h_p x_{s+1}) = +1$.

Set of All cluster C is denoted ©.

Definition 3.1

Each C ∈ ©, we determine the interval at k-level corresponding to C as follows:

Put $S_k(C) = \cup \{\Im_{k+1} \mid \Im_{k+1} \in C\}$;

The interval representation of other fuzzy data types will be represented as follows:

Type 2: Each interval value [a, b] is represented by a set contain [a, b]. we have $\theta_{min,k}(x)$ ([a, b]) = {[a, b]} because [a, b] is not fuzzy data, with $\forall$ k ≤ k^* and $Neig^d_{fm}(x)$ = {[a, b]}.

Type 3: Each value will represent by the ∅ set, so $\theta_{i,k}$(*inapplicable*) = {∅}, with $\forall$ k ≤ k^* and $Neig^d_{fm}$(*inapplicable*) = {∅}.

Type 4: Each value of this data type can receive any value in attribute domain. For this view, $\theta_{min,k}$(*missing*) = {[a, b] | a ∈ D_A}, with $\forall$ k ≤ k^* and $Neig^d_{fm}$(*missing*) = {[a, b] | a ∈ D_A}.

Type 5: Each value of this data type can receive any value in P ⊆ D_A but do not know exactly. Similar to type 4, $\theta_{i,k}$(P) = {[a, b] | a ∈ P}, $\forall$k ≤ k^* and $Neig^d_{fm}$(P) = {[a, b] | a ∈ P}.

Type 6: Each value of this data type can be considered as combination of data type 4 and data type 5. So, $Neig^d_{fm}$(unknown) = {∅, [a, b] | a ∈ D_A}.

Clause 3.1 [2]

For AX is a linear complete hedge algebra of attribute A, H^+ and H^- have at least two hedges, the fuzziness quantitative parameters defined by definition 2.4. We have:

a. For each k, $\{S_k(u) \mid u \in X \cup C\}$ is determined uniquely and is a partition of [0, 1]

b. For all x, u ∈ X ∪ C, if φ(x) ∈ $S_k(u)$ then $O_{min,\,k}(x) \subseteq S_k(u)$.

Definition 3.2 [2]

For linear complete HA, AX and the fuzziness measure, fm. Suppose that φ_A is a quantitative semantic function on AX and for each k, where 1<k<k*, S_k is k-level similar relation on D_A. Meanwhile, with two tuples t and s on U, the two values t[A]

and s[A] on the value domain are called k-level equal that denoted by $t[A] =_{fm, k} s[A]$ or $t[A] =_k s[A]$, if existing a equivalence class $S_k(u)$ of S_k so that $O_{min, k}(t[A]) \subseteq S_k(u)$ and $O_{min, k}(s[A]) \subseteq S_k(u)$.

Definition 3.3 [2]

With t and s are two tuples on U. We write $t[A_i] =_{\upsilon, k} s[A_i]$ (k-level equal) if the following conditions are true:

1. If $t[A_i], s[A_i] \in D_{A_i}$ then $t[A_i] = s[A_i]$;
2. If only one of two $t[A_i]$ or $s[A_i]$ is linguistic data, assume that $t[A_i]$, then $s[A_i] \in S_k(t[A_i])$.

To be able to compare the two values in the domain of a linguistic attribute as well as comparing the value of the two tuples on a set of attributes we have the following two definitions:

Definition 3.4 [2]

With two tuples t, s as in definition 3.2:

1. if $S_k(t[A_i]) < S_k(s[A_i])$ then $t[A_i] <_{\upsilon, k} s[A_i]$;
2. If $S_k(t[A_i]) > S_k(s[A_i])$ then $t[A_i] >_{\upsilon, k} s[A_i]$;
3. $t[A_i] =_{\upsilon, k} s[A_i]$ or $S_k(t[A_i]) < S_k(s[A_i])$ then t[Ai] £u, ks[Ai];
if $t[A_i] =_{\upsilon, k} s[A_i]$ or $S_k(t[A_i]) > S_k(s[A_i])$ then $t[A_i] \geq_{\upsilon, k} s[A_i]$.

Example 3.1

Let's consider the schema in a fuzzy database of garment shop, R_1= {Itemcode, Brand, Importprice, Status, Saleprice }.

We have Brand, Importprice, Status, Saleprice are fuzzy attributes, itemcode was common attribute.

Table 1. The instance of R_1

Itemcode	Importprice	Status	Saleprice
A001	5	Rather old	Rather High
Q001	Very low	Old	17
A002	9	New	High
A003	8	Possible New	13

4 Primacy in Presenting Fuzzy Data

It can be said that data representation is a key factor that determine the meaning, feasibility and value of a database model because data representation will facilitate or block the construction of data manipulation operations and manipulation operations decide queries issue as well as update database.

In [11] summarize five common approaches to represent fuzzy data as follows:

Table 2. Summarize five common approaches to representing fuzzy data

Approach	Grade of membership	Values of attributes	Elements of domain
Fuzzy relation	*		
Similarity relation		*	
Possibility		* *	
Extended Possibility		* * *	* * *
Aggregation	* *	* *	

Note, the more * appearing, the more database model spreading. Thus, the database model based on extended possibility emerges as a best model.

The basic idea of the fuzzy relational database model based on extended possibility as follows: relation r on the relational schema R_i is a subset of the $\Pi(D_1)$ x $\Pi(D_2)$ x…$\Pi(D_n)$, $\Pi(D_i)$ is the possibility distributions on the value domain D_i of the attribute A_i. So every n tuple will have the form (π_{A_1}, π_{A_2} … π_{An}) with $\pi_{Ai} \in \Pi(D_i)$. Besides, each R_i is combined with a resemblance

If a relation consist a lot of tuples (hundreds of, thousands or even tens of thousands of tuples ....), it's clear which showed weaknesses of data representation problem under this model because all fuzzy values of each attribute, therefore it will be "wordy" and "downright frustrating" when to express a fuzzy value . For example, to express the age of the person belongs interval "from 30 to 40 years old" people can apply part of possibilities is {0.8/30, 0.7/31... 0.1/40}, Conspicuously, if fuzzy value interval is greater than the its express chain will be longer and more complex. Additionally, with each relational schema included m attribute will have m tables of two-dimensional (otherwise known as the two-dimensional matrix), each table used to represent close relationship between elements under range of values of a properties.

Such a data representation in scalability theoretical approaches (more general fuzzy set theory) complex which will lead to the complexity of data manipulation operations.

Two matching basic operations with fuzzy data included semantics inclusion operations and semantically equivalent operations that proposal [6] and some other documents shall be determined as follows:

$$SID_\alpha(\pi_A, \pi_B) = \sum_{i,j=1}^{n} \min_{u_i,\, u_j \in U \text{ và } Res_U(u_i,\, u_j) \geq \alpha} (\pi_B(u_i), \pi_A(u_j)) \Big/ \sum_{i=1}^{n} \pi_B(u_i) \quad (\#)$$

and

$$SED_\alpha(\pi_A, \pi_B) = \min(SID_\alpha(\pi_A, \pi_B), SID_\alpha(\pi_B, \pi_A)) \quad (\#\#)$$

In that, $SID_\alpha(\pi_A, \pi_B)$, $SED_\alpha(\pi_A, \pi_B)$respectively semantics inclusion measure and semantic equivalent measure of between the two possibility distribute π_A, π_B; Res denote closely relationship

Obviously, data representation in efficient leads to complex of operations for data matching.

The representation of fuzzy data in fuzzy databases by hedge algebra approach very natural and simple but very true to the inherent nature of the fuzzy data exist in the real world. Fuzzy data representation in this way is called "correct name" and understand the "true nature" because it was "obtained directly" from spec database when the user's observation and quantification of fuzzy data, so it may says, has not where which fuzzy data representation yet more simple and more brief.

For example, when surveying the material world consideration in any context, the observed object are evaluated as "small" or "very small"...That assessment is essentially fuzzy quantification is represented by fuzzy terms - the fuzzy data representation by hedge algebra approach - in the fuzzy database.

Such back to the above example to represents one's age ranged from 30 to 40, just use term " rather young" in that "rather" be a hedge and "young" be a generate element belong to a hedge algebra which is defined before.

Like that represents fuzzy data of simple and it's also simple when manipulating fuzzy data. By semantics quantitative mapping v(x), terms x from its fuzzy representations will be moved into fuzzy interval - semantics neighboring of x, it's as a topological included v(x) - semantics value x via mapped v(.) still ensures that the inherent semantics order. This allows us to build similar relationships level k between fuzzy terms, from which building operations "$=_k$", "$\leq_k$", "$\geq_k$", "$\neq_k$" to manipulate with fuzzy data easily available form and content like operations in relational database environment classics.

5 Primacy in Data Queries

The design goal of these databases is intended to serve for data query. Query data on the fuzzy relational database was difficult and almost cannot be done for the queries not is built according the hedge algebra approach because The design goal of these databases are intended to serve for data query. Query data on the fuzzy relational database was difficult and almost cannot be done for the queries not is built according the hedge algebra approach because it's very complex for manipulation of matching operations.

We review follow scalability theoretical approach on the example 3.1. Relation R above will be transformed into the following table:

Itemcode	Importprice	Status	Saleprice
A	{0.3/2;0.7/3;0.5/4}	{0.8/2;0.7/3;0.6/4}	{1.0/5}
Q	{1.0/9}	{0.3/5;0.7/6;0.8/7}	{0.3/6;0.6/7;0.7/6}
A	{1.0/58}	{0.4/5;0.7/6;0.5/7}	{1.0/71}

Suppose now we need to make the query "find items priced high" (Query number 2).

To perform this query, we performances a fuzzy term "priced high" as a possibility distribution, then indicates a threshold α which to the tuples satisfy the query conditions.

Next we have to browse through the tuples in the relations and to compute the SED follow the Formula (#, #......)

And in the computing process we have to reference the threshold at each table corresponds closely related......too so complex and not friendly!

At another query "Find items priced lower high" (Ex.1) with this query was almost impossible to accomplish because the comparison operations "less than" or "greater than" between two distribution capabilities are difficult to define.

On the contrary, for queries on the fuzzy relational database which follow the hedge algebra approach, things become much easier can confirm it meets most of fuzzy queries. Indeed, by the matching operations is constructed based on the "k level of close relations", a query in a fuzzy relational database follow the hedge algebra approach can be transformed into classic query (Theorem 3.2, 3.3 and 3.4).

Now we consider the database given by following table:

Example 5.1. The database as in Example 3.1 on Hedge algebra approach

Itemcode	**Importprice**	**Status**	**Saleprice**
A	Very low	Old	5
Q	9	New	High
A	58	Possible New	71

The Importprice and Saleprice properties are linguistic variables with Dom (Importprice), Dom (Saleprice) defined on the same interval [1, 100] (from $ 1 to $ 100).

The Hedge Algebra corresponding is defined with the following parameters

Elements generated: {low high | low <high}, negative hedges H^-= {possible, rather | possible < Rather}, positive hedges H^+ = {more, very | more< very}

Put fm(low) = 0.4, fm(High) = 0.6; μ(possible) = 0.15, μ(Rather) = 0.25, μ(more) = 0.2, μ(very) = 0.4.

The status property is a linguistic variable with Dom (status) is defined of over interval [0, 10]

The Hedge Algebra corresponding is defined with the following parameters

Elements generated: {Old, New | Old < New},

Negative hedges H^-= {Possible, Rather | possible < Rather}, positive hedges H^+ = {more, very | more< very}

Put fm(Old) = 0.4, fm(New) = 0.6; m(possible) = 0.15, m(Rather) = 0.25, m(more) = 0.2, m(very) = 0.4.

With query (Ex.1), we will do the following:

Suppose the query is done with the same rate k = 2;

$\upsilon_{saleprice, r}$ (high) = fm (low) + fm (high) * α = (0.4 + 0.6 * 0.4) * (100-1) = 63.36;

$\theta_{2,\text{Importprice, r}}$ (high) = $\mathfrak{I}_r$(Possible high) $\cup\mathfrak{I}_r$(rather high) = ((fm (high) - μ (possible) fm (high)), (fm (high) + μ (more) fm (high)] = (0.6 - 0:15 * 0.6, 0.6 + 0.2 * 0.6] * (100-1) = (50.49, 71.28]

see, t3 [Saleprice] = 71 ∈ $\theta_{2,\text{ Saleprice, r}}$ (high) and thus the tuple second t2, and tuple 3rd, t3 satisfy the query conditions.

With query (Ex. 2) was easily accomplished thanks to the results of the query above and theorems. The first tuple, t1, with t1 [Saleprice] = 5, $\theta_{2,\text{ Importprice, r}}$ (5) = [5, 5] < $\theta_{2,\text{Importprce , r}}$ (high) first tuple is inferred as a result of query

Through this example, we see that the queries on the fuzzy relational database which is done base on hedge algebra approach with simple manipulation but with high efficiency.

6 Primacy in Updating Data

Fuzzy database model, only really practical applications when we solve radically the problem updated. It's "depending on the way the fuzzy data semantic is represented in databases and on which concepts of the comparison between the data of different types, including fuzzy data, can be defined" [18]. Fuzzy database model approach based on hedge algebra enables unified data type in fuzzy attributes by taking into concept of level k similar relationships. It's has made the data manipulation becomes simpler very much by the alternative approaches. The unified fuzzy data on each property makes for fuzzy data manipulation similar to the traditional data manipulation. This advantage is the basis for we can build update operations.

In order to show the advantages of the approach based on hedge algebra for the updating fuzzy database we again compare it with fuzzy database update problems follows scalability theoretical approach.

In general, the update solutions approached based on scalability theory which the authors the article made, in our opinion, have not been resolved even on issue theoretically. The following shows its weakness

First, let's insert only be done with prerequisite condition the key must to be certain (the key include only certain properties), zoning such conditions, clearly it's diminish the meaning of fuzzy database.

Second, if to transfer the scheme becomes which has its normal form higher 2NF then "insertion strategy" will "There's no meaning" because it has become insert operate in the classic relational schema.

Third, the delete operations done based on the query, but as analyzed above, queries are made in this model unfavorable, as thus infer the delete operation not be smooth implementation .

Fourth, repair operation done through two of operations insertion and deletion so it will not has been well implementation.

The following examples show the superiority of the update data operations in the fuzzy database relational model which the approach based on hedge algebra.

Example 6.1

Reconsider example 3.1.

Itemcode	Importprice	Status	Saleprice
A	Very low	Old	5
Q	9	New	High
A	58	Possible new	71

Pack all the tuples in the relations r [R1], respectively t1, t2 and t3. In the relational schema exists

FFDs R1: f = { Importprice, Status } $\rightarrow_{\kappa}$Saleprice

With the updated requirements:

-To Add (insert) tuple of p = <A, 3.5, old, 7> in relations (CN1).

-To Remove (deletion) the tuples has saleprice smaller "rather high" out of relationship (CN2).

-To repair the value of saleprice of the tuples has value at importprice from "very low" to "low" (CN3)

With the request: CN1

I need to check to see p satisfied f ?

Since p [Importprice] = 3 ≠ (t2 [Importprice] and t3 [Importprice]), so we check if p [Importprice] $=_{\kappa}$t1 [very low]?.

We have $\upsilon_{\text{Imporprice, r}}$ (low) = β * fm (low) = 0.6 * 0.4 * 10 = 2.4; $\theta_{2,\text{ r}}$ (low) = $\mathfrak{I}_r$(Rather low) $\cup\mathfrak{I}_r$(Possible low) = ($\upsilon_{\text{Importprice , r}}$ (low) - fm (Rather low), $\upsilon_{\text{Importprice, r}}$ (low) + fm (Possible low)] = ((2.4 - 0.2 * 0.4, 2.4 * 0.4 + 0:15]) * 10 = (1.6, 3]. 3.5 $\notin$ (1.6, 3]

So p [Importprice $\neq_2$ t1 [very low], infer p {Importprice, Status other level 2 with t1} {Importprice , Status}, {Importprice t2, and t3} Status {Importprice, Status}

Conclusion: p satisfied FFDs f, thus p is inserted into the above system.

Result after inserted:

Itemcode	Importprice	Status	Saleprice
A	Very low	Old	5
Q	9	New	High
A	58	Possible new	71
A	3.5	Old	7

With the request CN2

The tuples are deleted, which will been satisfied with (Ex.2), that is, the first tuple in relation R will be deleted.

Result after Deletion:

Itemcode	Importprice	Status	Saleprice
Q	9	New	High
A	58	Possible new	71

With the request CN3

The tuple satisfy the repair condition was the first tuple, t1 = <Very old, old, 5>, we have 2 q, r (low) = (1.6, 3]. t1 [Importprice] = 5, so it is changed becomes the value belong to (1.6, 3], and the value is proposed u GIANHAP, r (low) = 2.4.

Result after Deletion:

Itemcode	Importprice	Status	Saleprice
A	Very low	Old	2.4
Q	9	New	High
A	58	Possible new	71

7 Conclusion

It's been several years, fuzzy databases with different approaches have tried to resolve the problems in capturing, representing and manipulating fuzzy detain order to approach practical applications, but results is seem to be hard to reach because theories is not tune to practice.

HA was built to open new approach to fuzzy databases effectively. By natural way to capture the meaning of fuzzy data – linguistic term, we can say that HA is flexible and strong enough to represent fully fuzzy data meaning.

Order to process linguistic terms, a linguistic value x can present by two semantic elements, first, semantic value of x through a sematic quantitative mapping υ, second, family of neighbors based on fuzzy intervals of x. From this base can build similar relation level k on the domain which was embedded in HA of fuzzy attribute This relation determine update operators which allow us to manipulate on fuzzy values effectively. This determines primacy of fuzzy databases based on HA considering aspects following as representing, querying, updating fuzzy data. This paper analyzed and evaluated to provide outlook onto primacy of fuzzy databases based on HA.

References

1. Ho, N.C., Wechler, W.: Hedge algebras: An algebraic approach to structures of sets of linguistic domains of linguistic truth variable. Fuzzy Sets and Systems **35**(3), 281–293 (1990)
2. Ho, N.C., Lan, V.N.: Hedge Algebras – An Oder – based Structure of Terms – Domains: An Algebraic Approach to Human Reasoning
3. Ho, N.C., Lan, V.N.: Hedge algebras: an algebraic approach to domains of linguistic variables and their applicability
4. Nguyen, C.H.: Fuzzy Relational Database with Linguistic Data – Part II: Fuzzy Functional Dependencies, Fuzzy Sets and Systems

5. Ho, N.C.: Linguistic Databases: Relational Model and Hedge-Algebra-Based Linguistic Data Semantics
6. Nakata, M., Murai, T.: Updating under integrity constraints in fuzzy databases. In: Proc. Sixth IEEE Conf. on Fuzzy Systems (FUZZ-IEEE 1997), pp. 713–719. IEEE, Barcelona (1997)
7. Raiju, K.V.S.V.N., Majumdar, A.K.: Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database system. ACM Trans. Databases Syst. **13**, 129–166 (1988)
8. Ma, Z.M., Yan, L.: Updating Extended Possibility – Based Fuzzy Relational Databases. International Journal of Intelligent Systems **22**, 237–258 (2007)
9. Bahar, O., Yazici, A.: Normalization and Lossless Join Decomposition of Similarity-Based Fuzzy Relational Databases. International Journal of Intelligent Systems **19**, 885–917 (2004)
10. Ho, N.C., Loc, V.M., Tung, H., An, N.T.: Fuzzy functional dependencies in fuzzy relational databases with linguistic data. Journal of Science and Technology (Viet Nam) **51**(2), 137–152 (2013)
11. Bouchon – Meunier, B., Thuan, H., Ha, D.T.: Fuzzy logic and application (Vietnamese book)