Updating Relational Databases with Linguistic Data Based on Hedge Algebras

Le Ngoc Hung[1], Vu Minh Loc[2(✉)], and Hoang Tung[3]

[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
[3] Dong Nai University, Bien Hoa, Vietnam
tungaptechbd@gmail.com

Abstract. Relational Databases (DB) with linguistic data based on hedge algebras (HA) were introduced, following this approach, data manipulation (include linguistic data) is simpler and more efficient, practical than the other one. On this basis, in this paper, we will present the update operations on relational databases with linguistic data based on HA. Update operations are built by mean of semantically quantifying mapping (SQM) and similarity relation of depth k, where k is the length of a linguistic value that belongs to the values domain of an attribute.

Keywords: Hedge algebras · Relational databases with linguistic data · Semantically quantifying mapping · Similarity relation of depth k · Clear key · Mixture key · Fuzzy key

1 Introduction

Updating and querying are major issues in databases. Continuing success in building theory database models following approaches such as: fuzzy set theory, possibility theory, extended possibility theory ... data updating problem has been studied. However, the results of these studies have not been reached practical requirements. In the fuzzy relational database model with linguistic attributes based on HA, universe U of its attributes is a set that includes two type of subsets, the first subset type contains classical attributes and the second subset contains attributes that are considered as linguistic variables. Linguistic and real values are adopted by linguistic variables.

In HA we have notions: semantically quantifying mapping, smallest neighboring of depth k and similarity interval of depth k. By these notions, we can unify data type of real and linguistic value to manipulate with fuzzy data becoming easy. This is facility that enables us to build update operations on relation databases with linguistic data.

The paper is organized as follows: in section 2, some basic concepts about HA will be introduced. Section 3 deals with relation databases with linguistic data based on HA. In section 4 update operations, the major problem in this paper, will be studied. Some conclusions will be given in Section 5.

P.C. Vinh et al. (Eds.): ICTCC 2014, LNICST 144, pp. 274–291, 2015.
DOI: 10.1007/978-3-319-15392-6_27

2 Some Basic Concepts

Definition 2.1 [1]

Let A**X** = (X, G, C, H, ≤) be a linear complete hedge algebras (ComLin-HA), a mapping fm: X → [0, 1] is called a fuzziness measure (abbreviated fm) of terms belong to X if:

1. $fm(c^-) + fm(c^+) = 1$ and $\sum_{h \in H} fm(hu) = fm(u)$, with ∀ u ∈ X, in this case fm called complete.
2. With the constants 0, W and 1: 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 any fm(x), fm(y). and it is the fuzziness measure of hedge h, denoted by μ(h).

Clause 2.1 [1]

For each fuzziness measure on X fm, the following statements are true:

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

Definition 2.2 [1]

A sign function: X → {-1, 0, 1} is a mapping which is defined recursively as follows: with h, h' ∈ H and c ∈ $\{c^-, c^+\}$ then

1. Sign(c-) = -1, Sign(c+) = +1,
2. Sign(hc) = - Sign(c) if h is negative w.r.t c, where as Sign(hc) = + Sign(c);
3. Sign(h'hx) = - Sign(hx), if h'hx ≠ hx and h' is negative w.r.t h; Sign(h'hx) = + Sign(hx) if if h'hx ≠ hx and h' is positive w.r.t h

   Sign (h'hx) = + Sign (hx), if h'hx ≠ hx and h' is negative w.r.t h;
4. Sign (h'hx) = 0 if h'hx = hx.

Definition 2.3 [1]

Let A**X** = (X, G, C, He, ∑, Φ, ≤) be a ComLin-HA

A mapping φ: X $\rightarrow$ [0, 1] is called semantically quantifying mapping (abbreviated as SQM) of AX, the following affirms are true:

1. φ is mapped 1-1 from X on [0, 1] and maintain order on the X. With $\forall$ x, y $\in$ X, $x < y \Rightarrow \varphi(x) < \varphi(y)$ and $\varphi(0) = 0$, $\varphi(1) = 1$, với 0, 1 $\in$ c;

2. $\forall$ x $\in$ X, $\varphi(\Phi x)$ = infimum $\varphi(H(x))$ and $\varphi(\Sigma x)$ = supremum $\varphi(H(x))$.

Definition 2.4 [1, 3- 4]

fm is the fuzziness measure on X. a mapping φ: X $\rightarrow$ [0, 1], induced by fm on X, is defined as follows:

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

2. $\varphi(h_jx) = \varphi(x) + Sign(h_jx)\{\sum_{i=Sign(j)}^{j} fm(h_ix) - \omega(h_jx) fm(h_jx)\}$; with j $\in$ {j: $-q \le j \le p$ và $j \ne 0$} (*) and $\omega(h_jx) = \frac{1}{2}[1 + Sign(h_jx)Sign(h_ph_jx)(\beta - \alpha)] \in \{\alpha, \beta\}$;

3. $\varphi(\Phi c^-) = 0$, $\varphi(\Sigma c^-) = \theta = \varphi(\Phi c^+)$, $\varphi(\Sigma c^+) = 1$, with every j like (*), $\varphi(\Phi h_jx) = \varphi(x) + Sign(h_jx)\{\sum_{i=Sign(j)}^{j-Sign(j)\frac{1+Sgn(h_jx)}{2}} \mu(h_i) fm(x)\}$

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

3 Relational Database with Linguistic Data Based on Hedge Algebra

3.1 The Basic Ideas for Building the Databases with Linguistic Data Based on Hedge Algebras

Authors in [1, 3-4] have built a relational database model with language data based on HA as follows:

Relational database schema with linguistic data DB = {U, R_1, R_2, ..., R_m, Const}, U = {A_1, ... A_n} is attribute universe; R_i are relational schemas; Const is a set of data constraint on DB. Each R_i may contain two attribute groups, first group is normal attributes (classical attributes), the remaining groups is linguistic attributes.

Each linguistic attribute can be viewed as a linguistic variable that its value domain are linguistic values constitutes an HA mixed with set of real values. If A_i is a linguistic attribute then its value domain is $D(A_i)$ = LDom $(A_i) \cup DA_i$, in which, LDom (A_i) is a set of linguistic values and the D_{Ai} is a set of real values.

In addition, according to [4] the value domain of linguistic attribute can also receive value types such as interval values, undefined values, missing values, uncertain values, unknown values. These values can be transformed to unify with linguistic data in one data type. In this paper, we do not deal with these data types mentioned above.

Linguistic and real data type can be unified by mean of semantically quantifying mapping and similar relation of depth k. Based on this, a linguistic value x belong to linguistic values domain of a linguistic attribute, can be expressed through two semantic components:

- *The first one* is a semantic value which belong to the real domain DA, it is just the value of $\upsilon(x)$ (υ is a semantically quantifying mapping).

- *The second one* is a finite set of fuzziness-intervals-based neighborhoods.

Along with the concept of similar interval of depth k, S_k, we can build equal and matching operation of depth k to compare not only between two linguistic values also between linguistic value and real value.

Similar relation of depth k based on equivalence classes, S_k, composed from D(A) permitting us to build matching operation on the databases. With x, y in D(A), we call "x similar to y at depth k or $x =_k y$" if smallest neighborhoods of them located into same equivalence class of depth k.

We can construct equivalence classes, S_k, as follows:

Denote: k* is a positive integer that is maximum length of each value in D(A).

$|x| \leq k^*$ is the length of linguistic values x, put $j = | x |$, $T_k(x)$ is fuzziness interval of depth k that contain x by mean of mapping φ.

X_k is the set of linguistic values of length k, U is the universe of attributes belong to the database.

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

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

c. if $j + 1 \leq k \leq k^*$: $O_{min, k}(x) = T_{k+1}(h_lx) \cup T_{k+1}(h_{l'}x)$, with $l, l' \in \{-q, p\}$.

Put H_1 is subset of strong hedges , H_2 is subset of weak hedges, $H_1 = \{h_i, h_{-j} \mid 1 \leq i \leq [p/2], 1 \leq j \leq [q/2]\}$, $H2 = \{h_i, h_{-j} \mid [p/2] \leq i \leq p, [q/2] \leq j \leq q\}$.

Put $I_{k+1}(H_n) = \{T_{k+1}(h_iy) \mid y \in X_k, h_i \in H_n\}$, with n = 1, 2. Two intervals $T_{k+1}(x)$ and $T_{k+1}(y)$ in $I_{k+1}(H_n)$ are called interconnected exist intervals belong to $Ik_{+1}(H_n)$ consecutive ranging from $T_{k+1}(x)$ to $Tk_{+1}(y)$. This relationship will compose $I_{k+1}(H_n)$ into interconnected components.

Denote C be the set of similarity intervals of depth k of linguistic value x, C is defined as follows:

With $I_{k+1}(H_1) = \{T_{k+1}(h_iy) \mid y \in X_k, h_i \in H_1\}$, $C = \{T_{k+1}(h_iy) \mid h_i \in H_1\}$

With $I_{k+1}(H_1) = \{T_{k+1}(h_iy) \mid y \in X_k, h_i \in H_2\}$, Suppose that $X_k = \{x_s \mid s = 0, \ldots, m-1\}$ of m elements are arranged in the sequence so that $x_i \leq x_j$ if and only if $i \leq j$. Denote $H_2^- = H_2 \cap H^-$ and $H_2^+ = H_2 \cap H^+$. Clusters generated from fuzziness intervals $Ik_{+1}(H_2)$ has the following three categories:

a. Cluster on the left x_0: $\{T_{k+1}(h_ix_0) \mid h_i \in H_2^+\}$.

b. Cluster on the right x_{m-1}: $\{T_{k+1}(h_ix_{m-1}) \mid h_i \in H_2^+\}$.

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

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

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

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

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

Set the all clusters C is denoted ©.

Definition 3.1 [5]

Each C ∈ ©, similarity interval of depth k that correspond to C is:

$$S_k(C) = \cup \{T_{k+1} \mid T_{k+1} \in C\}$$

Clause 3.1 [5]

Let AX be a ComLin-HA of the attribute A, H^+ and H^- have at least two element, the fuzziness quantifying parameters are determined following the definition 2.4. We have:

a. For each k, $\{S_k(u) \mid u \in X \cup C\}$ are uniquely identified and it's a partition of interval [0, 1]

b. For each x, $u \in X \cup C$, if $\varphi(x) \in S_k(u)$ then $O_{min,\,k}(x) \subseteq S_k(u)$

Definition 3.2 [1]

Let AX be a ComLin-HA and fm is the fuzziness measurer. Suppose that φ_A is SQM on AX with each k that $1 \le k \le k^*$, S_k is similarity relationship of depth k on DA. Then, with two arbitrary tuples t, s on U, t[A] and s[A] on the value domain has been called the equal level k, denoted by $t[A] =_{fm,\,k} s[A]$ or $t[A] =_k s[A]$, if there exists a equivalence class $S_k(u)$ of S_k so that $O_{min,\,k}(t[A]) \subseteq S_k(u)$ and $O_{min,\,k}(s[A]) \subseteq S_k(u)$.

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

Definition 3.3 [1]

Suppose that t and s are two tuples in the U. We write $t[A_i] =_{\varphi,\,k} s[A_i]$ and they are called equal in depth k, 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 the two tuples t[Ai] or s [Ai] is the linguistic data, assume that t [Ai] then $s[A_i] \in S_k(t[A_i])$;

Definition 3.4 [1]

Assume t, s the same as in definition 3.2, then

1. We write $t[A_i] <_{\varphi, k} s[A_i]$, if $S_k(t[A_i]) < S_k(s[A_i])$;
2. We write $t[A_i] >_{\varphi, k} s[A_i]$, if $S_k(t[A_i]) > S_k(s[A_i])$;
3. We write $t[A_i] \leq_{\varphi, k} s[A_i]$, if $t[A_i] =_{\varphi, k} s[A_i]$ or $S_k(t[A_i]) < S_k(s[A_i])$ and $t[A_i] \geq_{\varphi, k} s[A_i]$, if $t[A_i] =_{\varphi, k} s[A_i]$ or $S_k(t[A_i]) > S_k(s[A_i])$.

Thus, a relational database with linguistic data, will be built with above ideas , they allow us to deploy this type of databases by following reasons:

- The way to build models of a relational database with linguistic data based on hedge algebras very simple, but the ability to capture, as well as the performed actions with linguistic information is effective;
- Data in the linguistic attributes of the database has been unified into one data type that should be very favorable for manipulation;
- Linguistic data in real applications usually only the maximum length is 3 and the number of these linguistic values are commonly used is not greater, therefore it's not too complex to build a series of elements of a Linguistic attribute;
- It is not difficult to construct a sequence of similarity intervals of depth k (Sk) to the linguistic values, based on a sequence of this intervals that manipulation with data become simple.

3.2 Fuzzy Functional Dependencies (FFD)

Authors in [3] presented general issues and complete information about FFD, we recall some of the concepts, definitions important about FFD:

Let A is a linguistic attribute of the relational database with linguistic data, it will be combined with a set of similarity relationships k_A, this relationship is to define a concept of the fuzziness uncertain equal in level k_A and the denoted $=_{k(A)}$, $0 \leq k_A \leq k_A$, k_A is the maximum length of terms over A.

$K : U \rightarrow N$ (N is the set of positive integers) is a function of parts, it is defined on the set $X \subseteq U$ and assigned to each linguistic attribute A is a positive integer K(A) satisfies conditions $k_A \geq K(A) = k_A > 0$.

As so $K = \{k_A : A \in X\}$; if exists $K = \{k_A: A \in X\}$ and exists $K' = \{k'_A: A \in X$ and write $K_X \geq K'_X$ if $K_A \geq K'_A$ for all $A \in X$.

With $X \subseteq U$, we say that two tuples of t, s on U are equal with the similarity level K, and write $t[X] =_K s[X]$, if we have $t[A] =_{K(A)} s[A]$, for all $A \in X$.

Definition 3.3 [3]

With DB is a relational database with linguistic data and R(U) is a relational schema of DB. With any expression $f = X \rightarrow_K Y$ format called a level K fuzziness dependencies K (K-FFD), $X, Y \subseteq R$ and K is a similarity level to the previous definition $XY = X \cup Y$, and its semantics are interpreted as follows:

a relation any r(R), f is called satisfies r if
$(\forall\ t, s \in r)\ (t[X] =_K s[X]) \Rightarrow t[Y] =_K s[Y])$

In this case we also say that the relationship r satisfied $X \rightarrow_K Y$ or $X \rightarrow_K Y$ be true on r.

Offers by [3] we have axiomatic system for case fuzziness function depends as following:

K1(Reflexivity): if $Y \subseteq X$ then $X \rightarrow_K Y$

K2(Subsumption): if $X \rightarrow_K Y$ then $X \rightarrow_{K^*} Y$, with every K* on XY so that $K^*_X \geq K_X$ and $K^*_Y \leq K_Y$.

K3(Augmentation): if $X \rightarrow_K Y$ then $XZ \rightarrow_{K \vee K^*(Z)} YZ$, with all $Z \subseteq U$ and with all K*on Z so that $K^*_{Y \cap Z} \leq K_{Y \cap Z}$, where $XZ = X \cup Z$. and $YZ = Y \cup Z$.

K4 (TransitivIty): if $X \rightarrow_K Y$, $Y \rightarrow_{K^*} Z$ then $X \rightarrow_{K \vee K^*} Z$, with $K^*_Y \leq K_Y$ with $X \subseteq U$ and t, s are two tuples in U, we write $t[X] \leq_K s[X]$, if with any $\forall A \in X$ we always have $t[A] \leq_{K_A} s[A]$.

Definition 3.4 [10]

Let R(U) be a relational shema, relation r on R. X, $Y \subseteq U$ are two set of attributes. We can say r satisfy monotonically increasing fuzzy function dependencies X determine Y at depth k, abbreviated $X^+ \rightarrow_K Y$ in r, if we have: $\forall$ t, s $\in$ r, $t[X] \leq_K s[X] \Rightarrow t[Y] \leq_K s[X]$.

Definition 3.5 [10]

Let R(U) be a relational shema, relation r on R. X, $Y \subseteq U$ are two set of attributes. We can say r satisfy monotonically decreasing fuzzy function dependencies X determine Y at depth k, abbreviated $X^+ \rightarrow_K Y$ in r, if we have: $\forall$ t, s $\in$ r, $t[X] \leq_K s[X] \Rightarrow t[Y] \geq_K s[X]$.

Definition 3.6

Let R(U) is a relational schema, F be FFD on U, K are called key of R(U) if and only if the two following conditions are simultaneously satisfied:

1. $K \rightarrow_K U$
2. Do not exists $K' \subset K$ so that $K' \rightarrow_K U$.

4 Update Operations

If we resolve the problem of updating on fuzzy databases successfully, we can build significant factual applications. Fuzzy databases with other approaches such as similar relation, possibility theory, extended possibility theory, … show many limits in capturing, presenting and storing fuzzy data (see [5], [7]). So, the ability to deploy applications of these model are low because of this reason. With HA, we have concept of semantically quantifying mapping, smallest neighboring of depth k and similarity interval of depth k. We can use these concepts to build matching operation, based on this operation, we will build updating operations on databases with linguistic.

As stated above, a relational databases with linguistic data includes two attribute groups, the first group are the classical attributes, the second group are linguistic attributes as linguistic variables.

In fact, the value of linguistic data in linguistic attributes do not usually have greater than 3 of length, for instance, we consider a linguistic attribute to store information describing the new or old status of a product. The values of this attribute can be "very new", "very very new " ... or "old", "very old", "very very old". The values like "very very very very new" … that is not factual. Thus, we suppose that linguistic attribute values that has the length is always less than or equal to 3.

We distinguish three types of relational schema with linguistic data, including: relational schema with linguistic data has clear key (the key includes only classical attributes), mixture key (the key includes classical attributes and linguistic attributes) and fuzzy key (the key only includes linguistic attributes).

As we known, the update operations that include insert, modify and delete operations. Now, we'll study these operations on databases with linguistic data.

Let R (U, F) is a relational schema, in which, U is the universe of attributes, F = F = $F_1 \cup F_2$. With F_1 is the set of FFD by definition 3.3, F_2 is the set of monotonically increasing (decreasing) fuzzy function dependencies by the definitions 3.4 and 3.5. Let $U = A_1 \ldots A_n$, $U = U_1 \cup U_2$, $U_1 = A_1 \ldots A_m$ are classical attributes and $U_2 = A_{m+1} \ldots A_n$ are the linguistic attributes.

4.1 Insert Operation

Insert operation is understood as executed by adding tuple t into a relation r(R). Tuple t will be inserted into r, if t satisfies the data constraint on r, concretely, t must satisfy the FFD in F. These FFD in F are divided into two groups, first group, F_1 and second group, F2, as mentioned above.

Tuple t will be inserted into r(R) if t can be passed two checks: check t if satisfies F_1 and check if t satisfies F_2 ? and another problem of insert operation to consider: tuple t as mentioned above, before it is inserted into r(R), first, we needs to check t satisfies F_1? For each $s \in r(R)$, this check is actually check to see t and s have the same key at depth k or not. Thus, when we check to see whether there's the same key between t and s, if we do not specify clearly which of k that is matching, we will have to make even a lot of operations to insert t in the database. This case will become very complicated when r(R) has the large of tuples. So, it's necessary to specify what is the depth of k clearly. With the things that we discussed above, insert operation can be done as follows:

Insertion can be divided into three cases corresponding to three types of relational schema:

- In the first case: insertion in the relation scheme that has the clear key
- In the second case: insertion in the relation scheme has the mixture key
- In the third case: insertion in the relation scheme has The fuzzy key

4.1.1 Insert Operation in the Relational Schema that Has the Clear Key

Check Data Constraint with F_1

This check is tested to verify that tuple t be duplicated the key with any tuple in r or not. It is done the same as in the classical relational schema. If the tuple t satisfied key constraint then it will continue to be tested with data dependencies F_2 with depth k, otherwise tuple t will not be inserted into r (R).

Check Data Constraint with F_2

If F_2 exist, we will use them to check if the tuple t satisfy the condition in definition 3.4 or 3.5, if tuple t satisfy these conditions then t will be inserted into r.

4.1.2 Insert Operation in the Relational Schema that Has the Mixed Key

The examination of data constraint in this case more complicated than the first case. The key of relational schema in this case = group of classical attributes (X) $\cup$ group of fuzzy attributes (Y).

Check Data Constraint with F_1

For each $s \in r$ if s(key) = t(key) $\Leftrightarrow$ s(X) = t(X) (1) and s(Y) $=_K$ t(Y) (2).

The examination (1) is simple because of the comparison between two real values. Suppose (1) is correct, the remaining problem is to check (2).

To be able to check (2) we must perform the following steps:

- Build similarity intervals of depth k_{Ai} of the values $\in$ Dom (Ai) with $A_i \in Y$;
- If with $\forall A_i \in Y$ that $t[A_i] \in S_{k_{Ai}}(s[A_i])$ then testing (2) is correct, that mean tuple t will do not be inserted onto r (R) (because the same key), in contrast, tuple t will be checked with the group of F_2 (if available).

Check Data Constraint with F2

This check is done the same as the first case.

To facilitate tracking of data values in a relation with mixtures key or fuzzy key, each relation need to be supplemented attribute of depth k that contains the set of values of matching of depth k_{Ai} . Each value corresponds to a tuple in relation database to indicate the participating of this tuple in relation databases following the certain matching of depth k.

For example, we have the following relation:

K	A	B
3, 2	a_1	b_2
2, 2	a_2	b_2

In the above relation, we can see the first tuple, $t_1<a_1, b_2>$ is inserted into relation by matching of depth k = {3, 2}.

4.1.3 Insert Operation in the Relational Schema that Has the Fuzzy Key

Check Data Constraint with F_1

Verifying duplicate key in this case is the same as case 2, because the relational schema's key do not include classical attributes.

Check with F_2 Data Constraint with F_2

It's implemented as two above cases.

4.2 The Delete Operation

Executing this operation is accompanied by the delete condition to identify the tuples should be deleted, keep in mind if this condition is not accompanied by any conditions then all of the tuples in the relation will be deleted. Delete condition is actually a classical query, fuzzy query or both of all; With a fuzzy query, based on HA, we can convert to a classical query of depth k. We can distinguish three case of delete conditions:

Case 1:
The delete condition do not include linguistic attributes (classical query). We can handle this case same as in classical databases.

Case 2:
The delete condition that has includes linguistic attributes (include both fuzzy query and classical query)

This case, the delete condition has the form: $\forall$ t $\in$ r, t (delete condition) = true $\Leftrightarrow$ (t[X_1] ∂ $value_1$) θ (t[X_2] ∂ $value_2$) … θ (t[X_u] ∂ $value_u$) θ t[Y_1] ∂_{k_1} $fvalue_1$) θ (t[Y_2] ∂_{k_2} $fvalue_2$)…θ(t[Y_v] ∂_{k_v} $fvalue_v$) is true; in which $X_i \in U_1$ (i=1…u), $Y_j \in U_2$ (j = 1 … v); θ is the AND or OR operations; ∂ is one equation =, $\le$, $\ge$, $\ne$, > and <.

The tuple t satisfies two condition groups simultaneously, the first ones, tuple t must be satisfied on set of $X_i \in U_1$ (i=1…u), the second ones, tuple t must be satisfied on set of $Y_j \in U_2$ (j = 1 … v).

The first condition group was processed same as the classical databases, second condition group we will use methods (*) below to process.

(1). Build list V_i of level k_{Yi} similarity intervals, S_{kYi} in Dom (Yi) with $Y_i \in U_2$

(2). For each t $\in$ r:

- Calculate similarity intervals $S_{k_{Y_i}}(t[Y_i])$;

- Calculate $O_{min, k_i}(fvalue_i)$, fvaluei is a linguistic values;

- Verify whether value of logical expression t[Y_1] ∂_{k_1} $fvalue_1$) θ (t[Y_2] ∂_{k_2} $fvalue_2$)…θ(t[Y_v] ∂_{k_v} $fvalue_v$) is true or not?

4.3 The Modify Operation

Modify operation to be made through the processing of the two conditions, first condition is used to determine tuples which be modified with matching of depth k in r (denoted X, $X \subseteq r$, assume X has m elements), the second condition is the condition that $\forall t \in X$ after modified data be satisfied.

So, the modify operation in nature is to delete tuples that it satisfies the condition 1 (in X) and insert new tuples that it satisfies the condition 2 into r. We will study two methods for handling this condition.

Processing Conditions 1

Condition 1 of modify operation is the same as delete condition, so we can apply again the way of condition processing of the delete operation above.

Processing Conditions 2

The result after condition 1 processed is understood as extracting X from r, further work can be described as follows: extract tuple t_i (i = 1, …, m) form X and edit the values on some attributes of the t_i so that it satisfies conditions 2 and finally insert t_i into r.

The problem is that how do we modify the value of some attributes of t_i? we would classify the attributes of t_i that its values be modified into two groups:

The first group: comprises the classical attributes group

The modifying the value of this group is the same as in the classic.

The second group*:* consists of linguistic attributes

Modifying the value of this group is not simple, it's usually classified into the following cases:

Case 1

A real value will be modified to another real value equal to a linguistic value of depth k.

For example: in a employee salary management database, we have the request: "Look for employees with relatively *rather young age* and their contributions at same level to raise their salary up to *quite high* ".

Suppose that with matching operation of depth k = 2, an employee's salary level at 2.0 belong to *rather low* level, now, we need to modify this salary level become to linguistic value at *rather high*. This modify operation is called modifying a real value become to another real value other that it is similarity of depth k with a linguistic value.

In general, we will process this case as follows:

real c value is converted to real b value, that it is similarity of depth k with x linguistic values.

if $a' = \varphi(x)$, b is similarity to with x of depth k ($\forall k \leq k^*$). Thus, in this case the c value will be changed to $a' = \varphi(x)$.

Case 2

A linguistic value will be replaced by a purely linguistic values, such as "rather good" replaced by "good"

To proceed this case, a linguistic value x will be modified to become a linguistic value y, easily, we replace string represented x by string represented y.

Case 3

Linguistic values x will be modified to a linguistic values y, with condition: $y = x \partial z$, ∂ is a operation of arithmetic and z is a numeric value.

This case occurs when the condition 2 (increasing or decreasing value) that require the values of a specific attribute of tuples to satisfy the condition. Some linguistic values of attributes (remained values) will also have to change its value to the corresponding to the numeric value.

For example, suppose that we have the condition 2 on an attribute A of a database as follows:

"Increase values (for tuples that satisfy the first condition) of attribute A up to 15%" (#). How can we solve this query if the values of A do not include linguistic values but also include real values ? We cannot perform this operation $y = x \partial z$ because x and y are linguistic values.

To solve this problem, we propose approximate solution for this case as follows:

We will modify the "core" of linguistic values x, $\varphi(x)$ become to fvalue so that fvalue = $\varphi(x) \partial z$. Next, we will review a series of similar intervals at level k for any $k \le k^*$ of values of attribute domain which we are considering to determine what similarity intervals of depth k fvalue belong to, if fvalue $\in S_k(x')$ then x will be modified become to y.

4.4 Some Examples about Databases with Linguistic Values

Example 1

Let's consider relational shema R_1(SffCode, Fullname, Recowork, Reward) store information about bonus for staffs in a company. Sffcode: Staff code; Reworkco: review work completion.

r(R1)			
SffCode	**Fullname**	**Recowork**	**Reward**
A001	Nguyen Van Phu	More Good	More High
A002	Truong Phi Qua	Poor	Rather Low
A003	Huynh Phu Hao	8.5	More High
A004	Bang Quan	Very Good	300
A005	Banh Tien Len	2	More Low
A006	Bui The Gian	Very very good	Very High

Recowork and Reward are two linguistic attributes with agreement Dom(Recowork) = [0, 10] (review work completion get values from 0 to 10 points) with generated elements of {Poor, Good }, $\mathbf{H}^-$= {Rather, Possible }, $\mathbf{H}^+$ = {More, Very} the Dom (Reward) = [0, 500] (Reward get the values from 0 to 500 million) are linguistic variables with generated elements of {Low, High}, $\mathbf{H}^-$ = {Rather, Possible}, $\mathbf{H}^+$ = {More, Very}

For attribute Recowork: Put fm(Poor) = 0.35 fm(Good) = 0.65; μ(Possible) = 0:15, μ(Rather) = 0.25 μ(More) = 0.2, μ(Very) = 0.4.

For attribute Reward: Put fm(Low) = 0:55, fm(High) = 0:45; μ(Possible) = 0:15, μ(Rather) = 0.25 μ(More) = 0.2, μ(Very) = 0.4.

On R we identify set F of FFD as follows:

[StffCode]$\rightarrow_K$[Funame], two attributes StffCode and Funame are classical one, so FFD fuzzy return the common dependencies:

[SffCode] → [Funame] (1);

[StffCode] $\rightarrow_K$[Recowork] (2);

[StffCode] $\rightarrow_K$[Reword] (3);

These FFD are valid with k > 0.

StffCode is the key of R_1

With every attributes belong to R3, suppose k* = 3.

Example 2

R_2 (Antiqes, Techpater, Seprice) store information about the stock character of antique shops. Antiqes: Antiques Name; Techpater: Technical Parameter; Seprice: Sale Price.

r (R_2)			
Depth of K	**Antiqes**	**Techpater**	**Seprice**
2, 2	Bowl	Rather Good	More High
1, 2	Bowl	Good	Low
3, 2	Plate	Very Good	High
3, 2	Big jar	Possible Good	Rather High
3, 2	Vase	Rather Poor	Rather High
3, 3	Cup	Very Poor	Rather High
3, 2	Big jar	4.5	Very Low

Attribute Techpater is a linguistic variable with Dom(Techpater) = [0, 10] and two generated elements of {Good, Poor}

H^- = {Rather, Possible}, H^+ = {more very}. Put fm (Poor) = 0.45, fm(Good) = 0.55; μ(Possible) = 0.15, μ(Rather) = 0.25, μ(more) = 0.2, μ(very) = 0.4.

- Attribute SaPrice is a linguistic variable with Dom(SaPrice) = [500, 100000] (from 500 USD to 100000 USD)

With two generated elements of {Low , High}. H^- = {Rather, Possible } , H^+ = {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.

Review set F of FFD on R_2 include:

[Antiqes] [Techpater]$\rightarrow_K$SaPrice (5);

The Key on R_2 be [Antiqes] [Techpater];

With every attributes in R_3. Suppose k* = 3

Example 3

R_3(Brd, Impri , Stus, Sapri) of a database about sale the old and new garments. Brd: Brand; Impri: Import price; Stus: Status, Sapri: Sale Price.

r(R3)				
Depth of K	**Brd**	**Impri**	**Stus**	**Sapri**
2, 2, 2	Good	7000	Rather Old	Rather High
1, 1, 1	Rather Poor	Very Low	Old	17000

- Attribute Brd is a linguistic variable with Dom (Brd) = [0, 10] and two generated elements of {Good, Poor}, H^- = {Rather, Possible}, H^+ = {More, Very}.

Put fm (Poor) = 0.45, fm (Good) = 0.55; μ(Possible) = 0.15, μ(Rather) = 0.25, μ(More) = 0.2, μ(Very) = 0.4.

- Attribute Impri is a linguistic variable Dom (Impri) = [5000, 150000] (from 150000 VND to 5000 VND). With two generated elements of {Low , High}, H^- = {Rather, Possible} , H^+ = {More, Very}.

Put fm (Low) = 0.4, fm (High) = 0.6; μ(Possible) = 0.15, μ(Rather) = 0.25, μ(More) = 0.2, μVery) = 0.4.

- Attribute Stus is a linguistic variable with Dom (Tinhtrang) = [0, 10] and two generated elements {Old, New},

H^- = {Rather, Possible}, H^+ = {More, Very}.

Put fm (Old) = 0.4, fm (New) = 0.6; μ(Possible) = 0.15, μ(Rather) = 0.25, μ(More) = 0.2, μ(Very) = 0.4.

- Attribute Sapri is a linguistic variable Dom (Sapri) = [10000, 500000] (from 10000 VND to 500000 VND). With two generated elements of {Low, High}, H^- = {Rather, Possible} H^+ = {More,Very}.

Put fm(Low) = 0.4, fm (High) = 0.6; μ(Possible) = 015, μ(Rather) = 0.25, μ(More) = 0.2, μ(Very) = 0.4.

Review:

F is a set of FFD on R_3 include:

[Brd] [Impri] [Stus] $\rightarrow_K$[Sapri] (6).

The Key of R_3 is [Brd] [Impri] [Stus].

With every attributes in R_3, suppose k* = 3

Next, we will present the update on three schemas R_1, R_2, R_3. schemes are distinguished by their nature of key. The key of R_1 only include clear attributes, the key of R_2 include mixed attributes (clear and fuzzy); the key of R_3 only include fuzzy attributes.

Insert Operation

Suppose we have the following requirements:

① Inserting tuple t = <"A008", "Phuong Nam Ngang", "Poor", "Rather Low"> on relations r (R_1);

② Inserting tuple p = <"Vase", 5.0, "Rather high"> on relations r (R_2) with matching level between p and the tuples in relation is $K_{\mathbf{Techpater,Sapri}}$ = {1, 1};

③ Inserting tuple q = <"Rather Good", 150000, " Rather Old", "230000"> with level matching between p and tuples of the relationship is $K_{\mathbf{Impri, Stus}}$ = {1, 1}.

With the Request *①*

This case a tuple is inserted into the relational schema with its key only include classical attributes. Tuple t is inserted into r(R_1) if t satisfy the FFDs:

t satisfied FFDs: (1), (2), (3) and also satisfied monotonically increasing FFD (4).

Conclusion: t is inserted r(R1)

r(R1) after tuple t is inserted as follows:

SffCode	Fullname	Recowork	Reward
A001	Nguyen Van Phu	More Good	More High
A002	Truong Phi Qua	Poor	Rather Low
A003	Huynh Phu Hao	8.5	More High
A004	Bang Quan	Very Good	300
A005	Banh Tien Len	2	More Low
A006	Bui The Gian	Very very good	Very High
A008	***Phuong Nam Ngang***	***Good***	***Rather Low***

With the Request *②*

This is insert operation on relational schema that its key contains mixed between fuzzy attribute and classical ones.

Check p satisfies for FFD (5)?

For each s ∈ r(R2), we need to check p and s having same value ? that mean p and s simultaneously satisfy FFD (5)?

Case p[Antiqes] = s[Antiqes], we need to check p [Techpater] $=_1$ s[Techpater]?

If p[Antiqes] ≠ s[Antiqes], we conclude p and s satisfy with (5).

If ∃ s ∈ r so that key(p) $=_k$ key(s), we will conclude p does not satisfy (5) and obviously p can not be inserted on r(R_2).

Concretely, with the p as above, p[Antiqes] = "Big jar", this value is different from all value in attribute Antiqes of tuples in r(R_2) except tuple 4 (s_4) and tuple 7 (S_7). So, we just check if p[Techpater] $=_1$ $\mathbf{s_4}$ [Techpater] then p[Saprice] $=_1$ s_4 [Saprice] ? (~)

And if p [Techpater] $=_1$ s_7[Techpater] then p[Saprice] $=_1$ s_7 [Saprice]? (~ ~).

Consider (~): With the matching of depth k = 1, S_1(possible good) = T_1(possible good) = ((φ(Fine) + α.fm(Fine)) - fm (Possible good), φ(Fine) + α.fm(Fine))] = (0:45

+ 0.4 * 0:55 to 0:15 * 0:55, 0:45 + 0:55 * 0.4] = (0:45, 0.67], it mean interval (4.5, 6.7] on the reference value domain. S_4 [Techpater] = 5.0 ∈ S_1[Possible good) so p[Techpater] $=_1$ s_4[Techpater]. Now we consider p[Sapri] $=_1$ s_4[Sapri]?

p[Sapri] = "high" = s_4[Sapri] so (~) is correct, mean p correct with the s_4, so p will not be inserted on r(R_2) and we do not need to consider (~ ~).

Conclusion: The tuple p is not inserted into relations r(R_2) at matching of depth $K_{Techpater, Sapri}$ = {1, 1}. Relations r(R_2) remain status.

With the Request ③

This is case that relational schema has only fuzzy key.

Check for each s ∈ r(R3) if s and q satisfy (6)?

With first tuple (s_1): we have s_1[Impri] ≠ q[Impri], so the value of key are different on s_1 and q, that mean they satisfies (6)

With second tuple (s2): We have S_1(very low) = [0 +5, fm (very low) * (150000 - 5000) +5] = [5000, 0.4 * 0.4 * 145000 +5000] = [5000, 28200], to replace q [Impri] = 150000 ∉ S_1(very low), so q[Impri] $\neq_1$ s_1[Impri], from this, we have value of key on s_2 and q are different, that mean they satisfy (6).

Conclusion: q will be inserted into r(R_3) and r(R_3) after insert tuple q as follows:

Level K	Brd	Impri	Stus	Sapri
2, 2, 2	Good	7000	Rather Old	Rather High
1, 1, 1	Rather Poor	Very Low	Old	17000
1, 1, 1	***Rather Good***	***150000***	***Old***	***23000***

Delete Operation

Suppose that we have the following requirements:

- "Delete from r(R1) tuples which have "poor" Brd and the depth of k at $k_{Recowork}$ =1 (when we apply delete condition" ④;

Deletion condition includes linguistic attribute which can be formulated as t[Recowork]$=_1$"Poor". We have: S_1(Poor) = (1.4, 2.6] and easily see that on relation r(R_1) tuple 2 (s_2) and tuple 5 (s_5) will be deleted, because s_2[Recowork] and s_5[Recowork] belong to S_1(Poor).

Relation r(R_1) after delete:

SffCode	Fullname	Recowork	Reward
A001	Nguyen Van Phu	More Good	More High
A003	Huynh Phu Hao	8.5	More High
A004	Bang Quan	Very Good	300
A006	Bui The Gian	Very very good	Very High

Modify Operation

Suppose we have the following requirements:

- " With relations r(R1) find all persons who have “very good” recowork, then modified their Reward level become "High" (the matching operation of depth k=2) (5)

Consider the Requirements (5)

Condition 1: All tuples t satisfying this condition, they must have t [Recowork] $=_2$"Very Good". we have S_3(Very Good) = ((1 - fm (Very Good) + fm (Rather Very Good), 1-fm (Very Very Good)] * 10 = (8.1, 9.0]; we have tuples 3 and 4 of $r(R_1)$ will be modified values.

Condition 2: With tuple 3 (s_3), value of attribute Reward satisfied. With tuple 4th (s_4), we have s_4[Reward] = 300 $\notin$ S_2(More High). We have φ(More High) = 1 - fm (Very High) - bfm(More High) = 0.71, so, φ(300) will be replaced by φj(More High), corresponding to the value of reference domain is 0.71 * 500 = 355.

Relation $r(R_1)$ after modified as required (5):

SffCode	Fullname	Recowork	Reward
A001	Nguyen Van Phu	More Good	More High
A002	Truong Phi Qua	Poor	Rather Low
A003	Huynh Phu Hao	8.5	More High
A004	Bang Quan	Very Good	355
A005	Banh Tien Len	2	More Low
A006	Bui The Gian	Very very good	Very High

5 Conclusion

In this paper, we present updating operations on relational databases model with linguistic data based on hedge algebra, included the operations insert, delete and modify. Insert operation is proposed for the three relational schema types, including relational schema with clear key, with fuzzy key and key including clear attributes and fuzzy one; delete operation is done entirely due the delete condition is determined based on the idea converting a fuzzy query become to a clear query with the similar level k; modify operation to be carried out through the delete and insert operation.

With hedge algebras we have some concepts: SQM mapping, fuzziness-intervals-based neighborhoods of a point, k-equality "$=_k$ " which enable us to build updating operations on relational databases based on hedge algebras more conveniently than on the other one.

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 algebraic approach to domains of linguistic variables and their applicability
3. Ho, N.C.: Fuzzy Relational Database with Linguistic Data – Part II: Fuzzy Functional Dependencies, Fuzzy Sets and Systems
4. Ho, N.C.: Linguistic Databases: Relational Model and Hedge-Algebra-Based Linguistic Data Semantics
5. Ho, N.C., Vinh, L.X., Hao, N.C.: Unifying and building similar relation in linguistic databases by Hedge algebras. Journal of Computer and Cybernetics, T.25, S.4, 314–332
6. Nakata, M., Murai, T.: Updating under integrity constraints in fuzzy databases. In: Proc. Sixth IEEE Conf. on Fuzzy Systems (FUZZ-IEEE 1997), Barcelona, pp. 713–719. IEEE (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, Ozgun, Yazici, Adnan: Normalization and Lossless Join Decomposition of Similarity-Based Fuzzy Relational Databases. Internationnal Journal of Intelligent Systems **19**, 885–917 (2004)
10. Ho, N.C., Hao, N.C.: Monotonically functional independencies in fuzzy databases based on hedge algebras. Journal of Computer and Cybernetics, T.24, S.1 (2008)