International Conference on Computing, Communication and Automation (ICCCA2015)

Dependency Preservation and Lossless Join Decomposition in Fuzzy Normalization

Jaydev Mishra and Soumitra De
Computer Science & Engineering
College of Engineering and Management, Kolaghat
Kolaghat, India
jsm03@cemk.ac.in
soumitra@cemk.ac.in

Sharmistha Ghosh
Mathematics
Galgotias University
Greater Noida, India
sharmistha.ghosh@galgotiasuniversity.edu.in

***Abstract*- The traditional relational database model may be extended into a fuzzy database model based on the mathematical framework of fuzzy set theory to process imprecise or uncertain information. While designing such a fuzzy relational database model that does not suffer from data redundancy and anomalies, the present authors have defined several fuzzy normal forms in ref [1]. However, as one decomposes an unnormalized relation into a desired normal form, it should also satisfy the essential properties of dependency preservation and lossless join of relation schemes which take a significant role in the design theory of a relational database. We have concentrated on these two important issues in this paper and have designed algorithms that confirm dependency preservation and lossless join decomposition of an unnormalized relation into the fuzzy third normal form or fuzzy Boyce Codd normal form. The algorithms have been tested and validated with examples.**

Keywords- fuzzy set; fuzzy functional dependency; fuzzy normal forms; fuzzy dependency preservation; fuzzy lossless join decomposition.

I. INTRODUCTION

Objectives of any good database design are to decrease redundancy and to provide consistency in data. Data redundancies as well as insertion, deletion and updation of anomalies are important features in the design theory of a relational database. Codd [2] had introduced a number of normal forms. The designer of a good relational database should take care that any relation should satisfy at least the third or Boyce Codd normal form so that data redundancy and anomalies may be minimized. Thus as one extends the traditional database to deal with uncertain information, it is important to study the above mentioned concepts in the fuzzy paradigm. Such studies related to normalization process for fuzzy relational models have been accounted in references [3, 4, 5, 6, 7]. The present authors have also introduced a number of fuzzy normal forms, namely, fuzzy first (**F1NF**), fuzzy second (**F2NF**), fuzzy third (**F3NF**) and fuzzy Boyce Codd (**FBCNF**) normal forms [1] that use the concept of fuzzy functional dependency (**ffd**) based on the idea of α-equality of tuples [8, 9].

It may be noted that the process of normalization actually uses the idea of decomposition to obtain the desired normal form. However, one should not decompose a relation scheme into an arbitrary set of smaller schemes. Decomposition should be done in a way such that the original relation can be recovered from its projections. This concept is known lossless join decomposition. Without the concept of lossless join, the components may not always represent the original relation. Further, while decomposing the relations it is also important to preserve the data dependencies in each individual relation obtained after decomposition. Thus, it is essential to confirm the dependency preservation as well as lossless join properties of decomposition for designing a good database model.

The present paper is devoted to these two important issues, i.e, lossless join decomposition and dependency preservation so that we can design a fuzzy relational database model that is free from data redundancies and different kinds of anomalies. In this work we have designed two algorithms that guarantee that our synthesized schemes satisfy the property of lossless join and preserve data dependencies in each individual relation as one decomposes an unnormalized relation into the desired fuzzy normal form.

The paper is organized in different sections as follows: The references of definitions of fuzzy set, fuzzy functional dependency, fuzzy closure of attribute set and different fuzzy normal forms have been mentioned in section **II**. In section **III**, two testing algorithms have been presented with examples for the dependency preservation and lossless join properties of decomposition into fuzzy third normal form and Boyce Codd normal form. The final conclusions appear in section **IV**.

II. BASIC DEFINITIONS

Basic definitions of fuzzy set [10]; fuzzy functional dependency (**ffd**) and the basic propositions related to **ffd** [8, 9]**;** fuzzy closure of an attribute set, fuzzy prime attributes, fuzzy non prime attributes and various fuzzy normal forms as defined in [1] will be used throughout the paper as we would deal with the properties of dependency preservation and lossless join in the fuzzy paradigm using the concept of α-**ffd.**

III. DEPENDENCY PRESERVATION AND LOSSLESS JOIN DECOMPOSITION

Normalization based on **ffd** is a process of decomposing a relation into smaller relations to obtain the desired normal form. But normal forms do not always guarantee a good database design. Generally, it is not only sufficient to check that each relation scheme in the database is in one of the fuzzy normal forms. Existence of two additional and desirable properties called lossless join and dependency preservation in fuzzy database should also be confirmed by the normalization process. Below we have designed algorithms that ensure that the dependency preservation and lossless join properties are achieved during decomposition of a relation into the desired fuzzy normal form.

A. Minimal Cover

As we proceed to present the algorithms for the dependency preservation and lossless join properties, it would be essential to introduce the concept of minimal cover.

Definition 9

A minimal cover of a set of **ffd**s F, is a set of **ffd**s that is equivalent to F and it does not contain any redundant attributes and redundant **ffd**s. A set of **ffd**s F is minimal if it satisfies the following criterions:

i) Right hand side of every **ffd** in F has a single attribute.
ii) F does not contain any redundant attributes in the LHS of any **ffd**. i.e., for any **ffd** $X \xrightarrow[\alpha_1]{} A$ there does not exists any **ffd** with $Y \xrightarrow[\alpha_2]{} A$ where $Y \subset X$ and $\alpha_2 \geq \alpha_1$ holds.
iii) F does not contain any redundant **ffd**.

The algorithm below finds the minimal cover of a given **ffd** set.

Algorithm 1: Minimal Cover Algorithm
Let F be the set of **ffd**s, and assign F to G, i.e., $G := F$.

Step1 Make the right hand side atomic
Replace each **ffd** $X \xrightarrow[\alpha_1]{} \{A_1, A_2, \ldots\ldots, A_n\}$ in G by n **ffd**s
$X \xrightarrow[\alpha_1]{} A_1, X \xrightarrow[\alpha_1]{} A_2, \ldots\ldots, X \xrightarrow[\alpha_1]{} A_n$.
Step2 Remove any redundant left hand side attribute
For each **ffd** $X \xrightarrow[\alpha_1]{} A_k$ in G and
for each attribute $B \in X$
if $((G - \{X \xrightarrow[\alpha_1]{} A_k\}) \cup ((X - \{B\}) \xrightarrow[\alpha_2]{} A_k))$
where $\alpha_2 \geq \alpha_1$ is equivalent to G, then replace $X \xrightarrow[\alpha_1]{} A_k$ with $(X - \{B\}) \xrightarrow[\alpha_2]{} A_k$ in G.
Step3 Remove any redundant **ffd**
For each remaining **ffd** $X \xrightarrow[\alpha_1]{} A_k$ in G
If $(G - \{X \xrightarrow[\alpha_1]{} A_k\})$ is equivalent to G, then remove $X \xrightarrow[\alpha_1]{} A_k$ from G.

Example 1

Let $R = (A, B, C, D, E)$ and a set of **ffd**s
$F = CD \xrightarrow[0.7]{} ABE, AD \xrightarrow[0.7]{} E, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E$. Find minimal cover of F.

Solution

Minimal cover algorithm 1 is applied to get the minimal cover of F. G is initialized to the set of **ffd**s F i.e.,
$G = CD \xrightarrow[0.7]{} ABE, AD \xrightarrow[0.7]{} E, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E$.

Step1 Make right hand side atomic
$G = CD \xrightarrow[0.7]{} A, CD \xrightarrow[0.7]{} B, CD \xrightarrow[0.7]{} E, AD \xrightarrow[0.7]{} E, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E$

Step2 Remove any redundant left hand side attribute
From $A \xrightarrow[0.8]{} B$ and $B \xrightarrow[0.9]{} E$, using α**-ffd-transitive rule,** we get $A \xrightarrow[0.8]{} E$ which implies $A \xrightarrow[0.7]{} E$ using Proposition 4. Hence in $AD \xrightarrow[0.7]{} E$, D is a redundant attribute. So $AD \xrightarrow[0.7]{} E$ is replaced by $A \xrightarrow[0.7]{} E$ in G.
$\therefore G = CD \xrightarrow[0.7]{} A, CD \xrightarrow[0.7]{} B, CD \xrightarrow[0.7]{} E, A \xrightarrow[0.7]{} E, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E$

Step3 Remove any redundant **ffd**
The **ffd** $A \xrightarrow[0.7]{} E$ is now redundant in G, since $A \xrightarrow[0.7]{} E$ is obtained from $A \xrightarrow[0.8]{} B$ and $B \xrightarrow[0.9]{} E$ of G by using α**-ffd-transitive rule** and Proposition 2.2.5 in [1]. So $A \xrightarrow[0.7]{} E$ is removed from G.
$\therefore G = CD \xrightarrow[0.7]{} A, CD \xrightarrow[0.7]{} B, CD \xrightarrow[0.7]{} E, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E$
is the minimal cover.

B. Dependency Preserving and Lossless Decomposition into Fuzzy Third Normal Form

While decomposing the relations in a fuzzy database, it is important to preserve the data dependencies like their classical counterpart, because each dependency in the fuzzy database represents a constraint of the database. If one of the dependencies is not represented in some individual relation R_i, we have to join two or more relations in the decomposition and then proceed, which is inefficient and impractical. The dependency preservation property ensures that each **ffd** of any relation should present directly or logically in any decomposed relation. Another desired property for decomposition is the lossless join property. If decomposition does not have the lossless join property, then we may get some spurious tuples after joining decomposed relations. These fake tuples may contain incorrect information. Therefore, this property is critical and must certainly be achieved. Lossless join property guarantees that spurious tuple generation problem does not occur with respect to the relation schemas created after decomposition.

Now, we present an algorithm that decomposes a relation R based on a set of **ffd**s F into smaller **F3NF** relations which satisfy both the dependency preservation and lossless join propreties of decomposition.

Algorithm 2: Dependency Preserving and Lossless Join Decomposition into F3NF

Input: A relation schema $R(A_1, A_2, ..., A_n)$ and a set of **ffd**s F of R.

Method:

Step1 Find the minimal cover G for F.

Step2 Place any attribute that have not been included in any of the **ffd**s of G in a separate relation schema, and eliminate it from R.

Step3 Use α **-ffd-union rule** to convert any set of **ffd**s $\{X \xrightarrow[\alpha_1]{} A_1, X \xrightarrow[\alpha_2]{} A_2, ..., X \xrightarrow[\alpha_k]{} A_k\}$ in G into a single **ffd** $X \xrightarrow[\min(\alpha_1, \alpha_2, ..., \alpha_k)]{} A_1 A_2 ....A_k$.

Step4 For any **ffd** $X \xrightarrow[\alpha]{} A_1 A_2 ....A_k$ in G, create a new relation schema R_i with attributes $\{X \cup A_1 \cup A_2 \cup .... \cup A_k\}$. $X \xrightarrow[\alpha]{} A_1 A_2 ....A_k$ is the only **ffd** of R_i and X is the fuzzy key of R_i at α -level of choice. This guarantees preservation of data dependency.

Step5 If none of the decomposed relation schemas contain the fuzzy key of the relation R, create one more relation schema that contains attributes that form the fuzzy key of R. This guarantees the lossless join property.

Output: A set of decomposed relation schemas $R_1, R_2, ..., R_k$ with **ffd**s $F_1, F_2, ..., F_k$ respectively, such that $G = \{F_1 \cup F_2 \cup ..... \cup F_k\}$ and $R = R_1 \rhd\lhd R_2 \rhd\lhd ..... \rhd\lhd R_k$.

Example 2

Find a dependency preservation and lossless join decomposition of $R = (A, B, C, D, E)$ into **F3NF** with set of **ffd**s

$F = \{CD \xrightarrow[0.7]{} ABE, AD \xrightarrow[0.7]{} E, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E\}$

Solution

Fuzzy key of R is CD at 0.7-level of choice (using algorithm 3.1 in **[1]**).

Step1 Find the minimal cover G for F using minimal cover algorithm. Minimal cover is $G = \{CD \xrightarrow[0.7]{} A, CD \xrightarrow[0.7]{} B, CD \xrightarrow[0.7]{} E, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E\}$ as calculated in example 1.

Step2 All attributes are included in any one of the **ffd**s.

Step3 Using α **-ffd-union rule** [9], G can be re-written as

$G = CD \xrightarrow[0.7]{} ABE, A \xrightarrow[0.8]{} B, B \xrightarrow[0.9]{} E$

Step4 For **ffd** $CD \xrightarrow[0.7]{} ABE$ in G, we get $R_1 = (C, D, A, B, E)$ with **ffd**s $F_1 = \{CD \xrightarrow[0.7]{} ABE\}$ and **fuzzy key**: CD at 0.7 level of choice.

Similarly, for $A \xrightarrow[0.8]{} B$, we get $R_2 = (A, B)$ with **ffd**s $F_2 = \{A \xrightarrow[0.8]{} B\}$ and **fuzzy key**: A at 0.8-level of choice.

For $B \xrightarrow[0.9]{} E$, we get $R_3 = (B, E)$ with **ffd**s $F_3 = \{B \xrightarrow[0.9]{} E\}$ and **fuzzy key**: B at 0.9-level of choice. This step guarantees the dependency preservation property i.e., $G = \{F_1 \cup F_2 \cup F_3\}$.

Step5 Since fuzzy key of R is already contained in the decomposed relation schema R_1, hence there is no need to create a new relation for this example. This step confirms the lossless join property i.e., $R = (R_1 \rhd\lhd R_2 \rhd\lhd R_3)$.

Therefore, after a dependency preservation and lossless join decomposition of R into **F3NF**, we have following three relations $R_1 = (C, D, A, B, E)$, $R_2 = (A, B)$ and $R_3 = (B, E)$.

C. Lossless Join Decomposition into Fuzzy Boyce Codd Normal Form (FBCNF)

Now we present an algorithm that provides lossless join decomposition of a relation schema into **FBCNF**. Such lossless join property guarantees that the problem of spurious tuple generation does not occur with respect to the relation schemas created after decomposition.

Algorithm 3: Lossless Join Decomposition into FBCNF

Input: A relation schema $R(A_1, A_2, ..., A_n)$ and a set of **ffd**s F of R.

Method:
Step1 Set $\rho := \{R\}$
Step2 While there is a relational schema S in ρ that is not in **FBCNF** do
{
Find a fuzzy functional dependency $X \xrightarrow[\alpha]{} Y$ in S that violates **FBCNF**. i.e., $X \xrightarrow[\alpha]{} Y$ violates **FBCNF** if X is not a fuzzy key of R.

Replace S in ρ by S_1 and S_2, where S_1 contains the attributes in $X \cup Y$ and S_2 will contain the attributes in S except those in Y. i.e., $S_1 = \{X \cup Y\}$ and $S_2 = \{S - Y\}$.
}

Output: A set of decomposed relation schemas $R_1, R_2, ..., R_k$ with **ffd** sets $F_1, F_2, ..., F_k$ respectively, satisfying the desired **Fuzzy Boyce Codd Normal form (FBCNF)** and lossless join property i.e., $R = R_1 \rhd\lhd R_2 \rhd\lhd ..... \rhd\lhd R_k$.

Example 3 Let us consider the $EMPDetail(Name, City, City\_Status, Experience, Salary)$ relation and **ffd** set $F = \{City \xrightarrow[0.99]{} CityStatus, Experience \xrightarrow[0.9]{} Salary, Name\ City \xrightarrow[1]{} Experience\}$
Find a lossless join decomposition of the relational schema $EMPDetail$ into **FBCNF**.

Solution
Fuzzy key of $EMPDetail$ is $(Name\ City)$ at 0.9-level of choice (using algorithm 3.1 in [**1**]).

Step1
Set $\rho := \{EMPDetail(Name, City, CityStatus, Experience, Salary)\}$
Step2 Here $EMPDetail$ is not in **FBCNF**, **s**ince in the **ffd** $City \xrightarrow[0.99]{} CityStatus$, $City$ is not a fuzzy key. Therefore, $EMPDetail$ is decomposed into the following two relations:
$E_1(City, CityStatus)$; $F_1 = \{City \xrightarrow[0.99]{} CityStatus\}$;
fuzzy key: $City$ at 0.99-level of choice and
$E_2(Name, City, Experience, Salary)$;
$F_2 = \{Experience \xrightarrow[0.9]{} Salary, Name\ City \rightarrow Experience\}$;
fuzzy key: $(Name\ City)$ at 0.9-level of choice.

Here E_1 is in **FBCNF**, but E_2 is not in **FBCNF** since $Experience \xrightarrow[0.9]{} Salary$ violates the rule. So, we again decompose E_2 into the following two relations:
$E_{21}(Experience, Salary)$;
$F_{21} = \{Experience \xrightarrow[0.9]{} Salary\}$;
fuzzy key $Experience$ at 0.9-level of choice and
$E_{22}(Name, City, Experience)$;
$F_{22} = \{Name\ City \xrightarrow[1]{} Experience\}$;
fuzzy key $(Name\ City)$ at 1-level of choice.
Here both E_{21} and E_{22} are in **FBCNF**.
Therefore, finally $EMPDetail$ is decomposed into following three relation schemas $E_1(City, CityStatus)$, $E_{21}(Experience, Salary)$ and $E_{22}(Name, City, Experience)$ with the **ffd** set $F_1 = \{City \xrightarrow[0.99]{} CityStatus\}$, $F_{21} = \{Experience \xrightarrow[0.9]{} Salary\}$ and $F_{22} = \{Name\ City \xrightarrow[1]{} Experience\}$ respectively.
Also we get, $EMPDetail = (E_1 \rhd\lhd E_{21} \rhd\lhd E_{22})$.
Hence the lossless join property has been achieved in the above decomposed relations that satisfy the fuzzy Boyce Codd normal form. It may be noted that the above decomposition also satisfies the dependency preservation property since $F = \{F_1 \cup F_{21} \cup F_{22}\}$.

IV. CONCLUSION

Fuzzy normalization based on α-**ffd** [1] plays an important role in designing a good fuzzy relational database. However, it is not only sufficient to check that each relation scheme in the database is in one of the fuzzy normal forms. Existence of two additional and desirable properties called lossless join and dependency preservation which are of immense importance in the design theory of a fuzzy relational database should also be confirmed by the normalization process.

In this work we have focused on these issues and have devised two algorithms that ensure that the two desirable properties are achieved during decomposition. Finally, it has been illustrated with examples how an un-normalized fuzzy relation can be decomposed into a set of normalized relations that satisfy both the lossless join and dependency preservation properties.

REFERENCES

[1] J. Mishra and S. Ghosh, "*Normalization in Fuzzy*

Relational Database Model", International Journal of Computer Engineering & Technology, Vol. 3, No. 2, pp. 506-517, 2012. (IF 3.9580).

[2] E. F. Codd, "A Relational Model for Large Shared Data Banks", Comm. of ACM, Vol. 13, No. 6, pp. 377-387, 1970.

[3] O. Bahar and A. Yazici , "Normalization and Lossless Join Decomposition of Similarity-Based Fuzzy Relational Databases", International Journal of Intelligent Systems, Vol. 19, No. 10, pp. 885-917, 2004.

[4] B. Bhunia and P. Niyogi , "Lossless join property in fuzzy relational databases", Data of Knowledge Engineering, Vol.11, No. 2, pp. 109-124, 1993.

[5] G. Chen , E. Kerre and J. Vandenbulcke , "Normalization Based on fuzzy functional dependency in a Fuzzy Relational Data Model", Information Systems, Vol. 21, No. 3, pp. 299-310, 1996.

[6] K.V.S.V.N. Raju and A.K Majumdar, "Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database system", ACM Transactions on Database Systems, Vol. 13, No. 2, pp. 129-166, 1988.

[7] S. Shenoi and A. Melton, "Functional dependencies and normal forms in the fuzzy relational database model", Information Sciences, Vol. 60, No. 1-2, pp. 1-28, 1992.

[8] S. Al-Hamouz, R. Biswas ,"Fuzzy Functional Dependencies in Relational Databases", International Journal of Computational Cognition, Vol. 4, No. 1, pp. 39-43, 2006.

[9] J. Mishra and S. Ghosh, "A *Study of Fuzzy Relational Database*" International Journal of Computational Cognition, Vol. 6, No. 4, pp. 45-50, 2008.

[10] A. L. Zadeh , "Fuzzy Sets", Information and Control, Vol. 8, No. 3, pp. 338-353, 1965.

[11] E. F. Codd , "The relational Model for Database Management", Addison Wesley, 1990.

[12] J.C. Date , "An introduction to Data Base Systems", 8/E, Addison Wesley, 2004.

[13] R. Elmasri , S. B. Navathe , "Fundamentals of Database Systems", 6/E, Pearson.