International Journal of Computer Applications (0975 – 8887)
Volume 184– No.29, September 2022

Satisfying Essential Normalization Properties of Neutrosophic Decomposed Relations

Soumitra De
Department of Computer Science & Engineering
College of Engineering and Management,
Kolaghat, West Bengal, India, 721171

Jaydev Mishra
Department of Computer Science & Engineerin
College of Engineering and Management,
Kolaghat, West Bengal, India, 721171

ABSTRACT
Neutrosophic database is an extension of classical database model with the framework of neutrosophic data and similarity measure formula to process uncertain data. Designing good neutrosophic database, normalization plays an important role in which a relation is decomposed into smaller relations with respect to neutrosophic functional dependencies to obtain desired neutrosophic normal forms. The objective of this paper is to investigate lossless join and dependency preservation properties of normalization in neutrosophic database. Two algorithms are designed to maintain lossless join and dependency preservation of decomposition of a relation in neutrosophic database. The algorithms have been tested and validated with examples.

Keywords
Neutrosophic Data, Neutrosophic functional dependency, Neutrosophic Lossless Join and Dependency Preservation

1. INTRODUCTION
In relational database model normalization technique plays an important role in designing good database. Normalization as introduced by Codd [1] is a technique by which authors can decompose a relation into minimum 3NF relations to reduce different anomalies. These decomposed 3NF and BCNF relation in traditional database always satisfy lossless join and dependency preservation properties of normalization. Present day since the communication is not restricted only in certain data so, fuzzy database model has been developed to process uncertain data. These database theories are also developed using fuzzy set [2] and vague set [3] while processing uncertain data. Authors have studied fuzzy and vague normalization to process lossless and dependency preservation in the literatures [4-10]. Researches in designing neutrosophic database which manages an inconsistent data related problem has started after the invention of new concept of neutrosophic set by Smarandache [11] in 2001. A few research have been reported in the literature [12-16] to defined neutrosophic functional dependency (α-nfd), neutrosophic closure of attribute set, neutrosophic key and neutrosophic normal forms. No such work is reported on neutrosophic dependency preservation and lossless join decomposition into desire normal forms in neutrosophic database.

This paper is focused on these two important issues such as dependency preserving and lossless join decomposition to design a neutrosophic relational database. In this work authors have designed two algorithms that guarantee neutrosophic lossless join and dependency preservation properties while decomposing a relation into smaller relations to achieve desired neutrosophic normal form.

The paper is organized as follows: In **section 2**, all previous work which is useful to explain present work is revisited. Algorithms for testing neutrosophic lossless join and dependency preservations of decomposition into neutrosophic third normal form or Boyce Codd normal form have been presented with examples in **section 3**. The final conclusion of the paper is drawn in **section 4**.

2. BASIC DEFINITIONS
Definition 1: Neutrosophic Set
Let U_1 is the universe set and m is the one element of U_1.

N is a neutrosophic set on U_1, represented by the membership functions such as:

i) Membership of truthness function $t_N : U_1 \to [0,1]$,

ii) Membership of falseness function $f_N : U_1 \to [0,1]$,

iii) Membership of indeterminate function $i_z : U_1 \to [0,1]$ with $t_N(m) + f_N(m) \le 1$ and

$t_N(m) + f_N(m) + i_N(m) \le 2$, written as

$N = \{\langle m, [t_N(m), i_N(m), f_N(m)]\rangle, m \in U_1\}$.

Definition 2: Similarity Measure between two neutrosophic data
Let two neutrosophic values of p and q represented as $p = [t_p, i_p, f_p]$ and $q = [t_q, i_q, f_q]$ where $0 \le t_p \le 1$, $0 \le i_p \le 1$, $0 \le f_p \le 1$ and $0 \le t_q \le 1, 0 \le i_q \le 1, 0 \le f_q \le 1$ with
$0 \le t_p + f_p \le 1$, $0 \le t_q + f_q \le 1$,
$0 \le t_p + i_p + f_p \le 2$,
$0 \le t_q + i_q + f_q \le 2$.

Now the similarity measure between two neutrosophic data denoted by $SE(p,q)$ is defined as follows

$$SE(p,q) = \sqrt{1 - \frac{|(t_p - t_q) - (i_p - i_q) - (f_p - f_q)|}{3}\left(1 - |(t_p - t_q) + (i_p - i_q) + (f_p - f_q)|\right)}$$

Definition 3: Neutrosophic α-equality of $t_1[X]$ *and* $t_2[X]$
Consider $r(R)$ be a neutrosophic relation with schema of R (A_1, $A_{2\ldots}$ A_n). Let $X_1 \subset R$ and t_1, t_2 are any two tuples in r. Then t_1 and t_2 are said to be α-equal on X_1 if SM ($t_1[A_i]$, $t_2[A_i]$) $\ge \alpha$ $\forall i$ =1 to k. This equality is represented as $t_1[X_1](NE)_\alpha t_2[X_1]$.

Definition 4: Neutrosophic Functional Dependency (α -nfd)

Let *R* (A_1, $A_{2\ldots}$ A_n) be a neutrosophic relation schema and X_1, $Y_1 \subset R$. Now X_1 neutrosophic functionally determines Y_1 at the level of $\alpha \in [0,1]$ is denoted by $X_1 \xrightarrow[\alpha]{nfd} Y_1$ and is defined as for any two tuples t_1 and t_2 if $t_1[X_1](NE)_\alpha t_2[X_1]$ is true then $t_1[Y_1](NE)_\alpha t_2[Y_1]$ is also true.

The above α -nfd can also be read as, "Y_1 is neutrosophic functionally determined by X_1 at α level".

For α -nfd the following propositions are straightforward.

The following proposition is straightforward from the above definition.

Proposition 1

If $0 \le \alpha_2 \le \alpha_1 \le 1$, then

$t_1[X](NE)_{\alpha_1} t_2[X] \Rightarrow t_1[X](NE)_{\alpha_2} t_2[X]$

Proposition 2

If $0 \le \alpha_2 \le \alpha_1 \le 1$, then $X_1 \xrightarrow[\alpha_1]{nfd} Y_1 \Rightarrow X_1 \xrightarrow[\alpha_2]{nfd} Y_1$

Proposition 3 α -nfd union rule

If $\{X_1 \xrightarrow[\alpha_1]{nfd} A_1, \ldots\ldots, X_1 \xrightarrow[\alpha_n]{nfd} A_n\}$ then

$X_1 \xrightarrow[\min(\alpha_1,\alpha_2,\ldots\ldots,\alpha_n)]{nfd} A_1, \ldots, A_n$

2.1. Neutrosophic closure of attribute set

Neutrosophic closure helps to obtain key of any relation in neutrosophic database. Following algorithms is used to find neutrosophic closure of any attribute set.

Algorithm 1: Find the Neutrosophic Closure of any attribute set

Input:
Schema of neutrosophic relation i.e. R_1, a set of **nfds** *N* on R_1 and attributes set X_1.
Output:
Attributes set X_1^+, the closure neutrosophic set of X_1.
Method:
Let $X_1^+ = X_1$.
i.e., $X_1^+ = (X_1, \alpha_1)$ where α_1 =1 [Q $X_1 \xrightarrow[1]{nfd} X_1$]
repeat
$X_1^* = X_1^+$.
for each nfd $Y_1 \xrightarrow[\alpha_2]{nfd} Z_1$ in N
do
if $Y_1 \subseteq X_1^+$ then $X_1^+ = (X_1^+ \cup Z_1, \alpha_3)$ where α_3 = min(α_1 , α_2) .
end for
until ($X_1^* = X_1^+$).

2.3. Neutrosophic Key

Neutrosophic key of neutrosophic relation R_1 is defined as follows:

Let K_1 a subset of attribute set of R_1 and *N* be a set of nfds for R_1. Now K_1 is a neutrosophic key of R_1 at α -level of choice with $\alpha \in [0,1]$ if K_1 is minimal subset of R_1 and $K_1 \xrightarrow[\alpha]{nfd} R_1$ holds in *N*.

2.4. Neutrosophic Normalization

Inconsistent data may lead to different database anomalies during database operation. For designing anomaly free information retrieval database, it must satisfy minimum third normal form. In this section the definition of different neutrosophic normal forms i.e., neutrosophic first (1NNF), neutrosophic second (2NNF), neutrosophic third (3NNF) and neutrosophic BCNF (NBCNF) normal forms is also revisited.

Definition 5: Neutrosophic First Normal Form

Let D_i has the attributes domain A_i, R_1 is the schema of a relation which is called the first neutrosophic normal form (1NNF) for any relation $r_1$in R_1, no one attribute is multi-valued and α -cut similarity based relation is in 1NNF**.**

Definition 6: Neutrosophic Second Normal Form

Let *N* is the nfds set for the schema of R_1 relation and *S* is a neutrosophic key at α –level. $R_1$is called the second neutrosophic normal form (2NNF), for no one of attribute in nonprime manner is partly dependent on the neutrosophic key.

Definition 7: Neutrosophic Third Normal Form

Let consider relation schema R_1 with the nfds set and the neutrosophic key is *S* at α –level. R_1 is to be in third neutrosophic normal form(3NNF), only if $R_1$is in 2NNF and R_1should not contain any nonprime attribute with nfd , for any non-trivial nfd $X \xrightarrow[\alpha]{nfd} A$ in *N* either *A* is neutrosophic –prime or the neutrosophic key is present in *X*.

Definition 8: Neutrosophic Boyce Codd Normal Form

In relation R_1 schema consists of neutrosophic key *S* and the set of nfds noted by *N* at the level of α . R_1 is BCNF neutrosophic normal form (NBCNF), only if R_1 is in N3NF and for any nfd which is non-trivial $X \xrightarrow[\alpha]{nfd} A$ in *N*, *X* is a neutrosophic key of R_1 means $X \supseteq S$.

3. NEUTROSOPHIC LOSSLESS JOIN AND DEPENDENCY PRESERVATION PROPERTIES TESTING

Normalization is a process in which a relation is decomposed into smaller relations. For good database designing, any decomposition always satisfies lossless join and dependency preservation properties. Lossless join means there should not be any loss of information between the original relation and the relation obtaining after joining of decomposed relations. Dependency preservation means all the dependencies in the original relation must be incurred from the closure of nfd set obtained after taking union of nfd sets of the decomposed relations.

Objective of the present work is to design neutrosophic database model in such a manner that during decomposition of a relation it also satisfies both neutrosophic lossless join and dependency preservation properties. Authors also explained two algorithms which achieve these properties while decomposing a relation of neutrosophic database into desire normal forms with respect to a given neutrosophic functional dependency (nfd) set. The first step of algorithm is to find canonical cover of the given nfd set. Steps for obtaining canonical cover of any neutrosophic functional dependency set is stated below.

3.1. Steps to obtain canonical cover of a given neutrosophic functional dependency set

Canonical cover of a given neutrosophic functional dependency set means to eliminate all extraneous attributes and redundant nfd from the neutrosophic functional dependency set.
Let F be the given set of nfds. Initially assign F to G i.e., $G \coloneqq F$

1st step: Make the right hand side atomic

Replacing each nfd $X \xrightarrow[\alpha_1]{nfd} \{A_1, A_2, ..., A_n\}$ in G by n nfds as stated below
$X \xrightarrow[\alpha_1]{nfd} A_1, X \xrightarrow[\alpha_1]{nfd} A_2, X \xrightarrow[\alpha_1]{nfd} A_n.$

2nd step: Remove extraneous attribute from left hand side

For any nfd $X \xrightarrow[\alpha_1]{nfd} A_k$ in G if we get an attribute $B \in X$ such that $(X\text{-}\{B\}) \xrightarrow[\alpha_2]{nfd} A_k$ where α_2 α_1 the nfd $X \xrightarrow[\alpha_1]{nfd} A_k$ contains extraneous attribute and is replaced by $(X\text{-}\{B\}) \xrightarrow[\alpha_2]{nfd} A_k$ in G.

3rd step: Removal of redundant nfd

An **nfd** $X \xrightarrow[\alpha_1]{nfd} A_k$ in G is redundant if { $(G - \{X \xrightarrow[\alpha_1]{nfd} A_k\})$ is equivalent to G. Redundant nfd must be eliminated from G.

Example 3.1: Let $R = (A_2, B_2, C_2, D_2, E_2)$ and a set of **nfd**s
$NF = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2B_2E_2, A_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$
Find minimal cover of NF.

Solution:
Minimal cover **algorithm 3.1** is applied to get the minimal cover of *NF*. G is initialized to the set of n**fd**s *NF* i.e.,
$G = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2B_2E_2, A_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$.

Step1: Make right hand side atomic
$G = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2, C_2D_2 \xrightarrow[0.7]{nfd} B_2, C_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$

Step2: Remove any redundant left hand side attribute
From $A_2 \xrightarrow[0.8]{nfd} B_2$ and $B_2 \xrightarrow[0.9]{nfd} E_2$, using α **-nfd-transitive rule,** we get $A_2 \xrightarrow[0.8]{nfd} E_2$ which implies $A_2 \xrightarrow[0.7]{nfd} E_2$ using **Proposition 2**. Hence in $A_2D_2 \xrightarrow[0.7]{nfd} E_2$, D is a redundant attribute. So $A_2D_2 \xrightarrow[0.7]{nfd} E_2$ is replaced by $A_2 \xrightarrow[0.7]{nfd} E_2$ in G.
$\therefore G = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2, C_2D_2 \xrightarrow[0.7]{nfd} B_2, C_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$

Step3: Remove any redundant **nfd**
The **ffd** $A_2 \xrightarrow[0.7]{nfd} E_2$ is now redundant in G, since $A_2 \xrightarrow[0.7]{nfd} E_2$ is obtained from $A_2 \xrightarrow[0.8]{nfd} B_2$ and $B_2 \xrightarrow[0.9]{nfd} E_2$ of G by using α **-ffd-transitive rule** and **Proposition 2**. So $A_2 \xrightarrow[0.7]{nfd} E_2$ is removed from G.
$\therefore G = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2, C_2D_2 \xrightarrow[0.7]{nfd} B_2, C_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$
is the minimal cover of *NF*.

3.2. Dependency Preservation with Lossless Decompose into Third Neutrosophic Normal Form

Following algorithm is designed to decompose a relation into neutrosophic third normal form satisfying both dependency preservation and lossless join properties.

Algorithm 1: Decomposition into 3NNF satisfying dependency preservation and lossless join properties

Input: A neutrosophic relation schema $R_n(A_1, \ldots, A_n)$ satisfying a set of neutrosophic functional dependencies F_n.

Procedure:

1. Find the canonical cover (M_c) of nfd set F_n

2. Create a new relation schema considering all attributes that are not included in any nfd of M_c and exclude them from R_n

3. Convert all nfd in M_c with same LHS into a single nfd using ***Proposition 3*** i.e., if $\{X_1 \xrightarrow[\alpha_1]{nfd} NA_1, \ldots.., X_1 \xrightarrow[\alpha_n]{nfd} NA_n\}$ exists in M_c convert these into $X_1 \xrightarrow[\min(\alpha_1, \alpha_2, \ldots., \alpha_n)]{nfd} NA_1, \ldots, NA_n$

4. For every nfd $X_1 \xrightarrow[\alpha]{nfd} NA_1, \ldots., NA_n$ in M_c, create a new schema R_s with attributes $\{X_1 \cup NA_1 \ldots.. \cup NA_n\}$ which guarantee neutrosophic dependency preservation properties

5. If decomposed relations do not acquire the neutrosophic key of R_n, then create another relation containing all attributes of neutrosophic key of R_n. This will guarantee the property of neutrosophic lossless join.

Output: R_n is decomposed into $R_{n1}, R_{n2}, \ldots, R_{nk}$ be satisfying nfds NF_1, $NF_2, \ldots, NF_k$ respectively, such that $M_c = \{NF_1 \cup NF_2 \cup \ldots.. \cup NF_k\}$ and
$R_n = R_{n1} >< R_{n2} >< \ldots\ldots. >< R_{nk}$

Example 1

Consider neutrosophic relation $R_d = (A_2, B_2, C_2, D_2, E_2)$ with set of nfds

$NF = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2B_2E_2, A_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$

Solution: neutrosophic key of R_d is C_2D_2 at α = 0.7 level of tolerance.

1st step: Find out M_c for *NF*. Here minimal cover is

$$\therefore M_C = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2, C_2D_2 \xrightarrow[0.7]{nfd} B_2, C_2D_2 \xrightarrow[0.7]{nfd} E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$$

as calculated in the section **3.1**.

2nd step: all attributes of R_d are incorporated in the nfds of the minimal cover.

3rd step: Using α -nfd union based rule,

Once again M_c can be written as $M_c = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2B_2E_2, A_2 \xrightarrow[0.8]{nfd} B_2, B_2 \xrightarrow[0.9]{nfd} E_2\}$

4th step: For nfd $C_2D_2 \xrightarrow[0.7]{nfd} A_2B_2E_2$ in M_c, we get $R_{11}=(C_2,D_2,A_2,B_2,E_2)$ with nfds $NF_1 = \{C_2D_2 \xrightarrow[0.7]{nfd} A_2B_2E_2\}$ and neutrosophic key is C_2D_2 at 0.7 level.

Similarly, for $A_2 \xrightarrow[0.8]{nfd} B_2$, we get $R_{12}=(A_2, B_2)$ with nfds $NF_2 = \{A_2 \xrightarrow[0.8]{nfd} B_2\}$ and neutrosophic key is A_2 at 0.8-level.

For $B_2 \xrightarrow[0.9]{nfd} E_2$, we get $R_{13}=(B_2, E_2)$ with nfds $NF_3 = \{B_2 \xrightarrow[0.9]{nfd} E_2\}$ and neutrosophic key is B_2 at 0.9-level. Here we guaranteed the dependency preservation, is represented as $M_c = NF_1 \cup ...... \cup NF_3$.

5th step: It has been confirmed the lossless join, is denoted by $R_d = R_{11} >< R_{12} >< R_{13}$.

After decompose R_d into 3NNF with the fulfillment of dependency and join in lossless way , we got three new relations $R_{11} =(C_2,D_2,A_2,B_2,E_2)$, $R_{12}=(A_2, B_2)$ and $R_{13}=(B_2, E_2)$

3.3. Lossless join decomposition into NBCNF

Here we have designed an algorithm which decomposes a relation into neutrosophic Boyce Codd normal form (NBCNF) satisfying lossless join properties.

Algorithm 2: Decomposition into NBCNF satisfying lossless join properties

Input: $R_1(A_1, A_2,\ldots,A_n)$ is a relation schema and a set of nfds *N* of R_1.

Procedure:

1. Set $\rho_1 := \{R_1\}$
2. For any relation schema R_1 in ρ which is not in NBCNF

do

{

Find neutrosophic functional dependency $X_1 \xrightarrow[\alpha]{nfd} Y_1$ in R_1 that violates the rules of NBCNF.

Decompose R_1 into R_{11} and R_{12}.

R_{11} contain attributes $X_1 \cup Y_1$ and R_{12} all attributes of R_1 except the attributes in Y_1.

i.e., $R_{11} = \{X_1 \cup Y_1\}$, $R_{12} = \{R_1 - Y_1\}$.

} until all decomposed relations are in NBCNF

Output: A set of decomposed neutrosophic Boyce Codd normal form (NBCNF) relations $R_1, R_2, ..., R_k$ satisfying lossless join property i.e., $R_1 = R_{11} >< R_{12} >< .... >< R_{1k}$

Example 2

$Person = \{Pfname, Pstate, Psstatus, Pincome, P\exp\}$ relation and nfd set

$$N_d = \{Pstate \xrightarrow[0.99]{nfd} Psstatus, P\exp \xrightarrow[0.9]{nfd} Pincome, PfnamePstate \xrightarrow[1]{nfd} P\exp\}$$

Find a decompose join in lossless way of Person relation into NBCNF.

Solution:

Neutrosophic key of *Person* is *(Pfname, Pstate)* at 0.9-level as per neutrosophic closure property.

Step1: $\rho := \{Person(Pfname, Pstate, Psstatus, Pincome, P\exp)\}$

Step2: Here *Person* relation is not in NBCNF, since in the nfd $Pstate \xrightarrow[0.99]{nfd} Psstatus$, *Pstate* is not a neutrosophic key. Therefore, *Person* relation is decomposed as:

P_1(Pstate, Psstatus) ;
$P_1 = \{Pstate \xrightarrow[0.99]{nfd} Psstatus\}$;
neutrosophic key is *Pstate* at 0.99-level and

P_2(Pfname,Pstate,Pexp,Pincome);
$P_2 = \{P\exp \xrightarrow[0.9]{nfd} Pincome, PfnamePstate \xrightarrow[1]{nfd} P\exp\}$;
neutrosophic key *(Pfname, Pstate)* at 0.9-level.

Here P_1 is in NBCNF, but P_2 is not in NBCNF since $P\exp \xrightarrow[0.9]{nfd} Pincome$ not satisfy the rules. Now, further decompose is required for P_2 as follows:

P_A(Pexp, Pincome);
$P_A = \{P\exp \xrightarrow[0.9]{nfd} Pincome\}$;
neutrosophic key is *Pexp* at 0.9-level and

P_B(Pname,Pcity,Pexp);
$P_B = \{PfnamePstate \xrightarrow[1]{nfd} P\exp\}$;
neutrosophic key is *(Pfname, Pstate)* at 1-level .

Here both P_A and P_B are satisfying NBCNF.

Finally by applying algorithm 2, the *Person* relation is decomposed into three NBCNF relations as given below:

P_1(Pstate, Psstatus) with nfd $P_1 = \{Pstate \xrightarrow[0.99]{nfd} Psstatus\}$
P_A(Pexp, Pincome) with nfd
$P_A = \{P\exp \xrightarrow[0.9]{nfd} Pincome\}$
P_B(Pfname, Pstate, Pexp) with nfd
$P_B = \{PfnamePstate \xrightarrow[1]{nfd} P\exp\}$.

Also we get, $Person = \{P_1 >< P_A >< P_B\}$ which satisfies lossless property.

4. CONCLUSION

Neutrosophic database processing imprecise data may face problem of different anomalies and data redundancy like as classical relational data model if the design part is not proper. Normalization of neutrosophic representation with α -nfd acts as an important role in designing good neutrosophic database. Normalization always decomposes a relation into smaller relations to achieve desire neutrosophic normal forms. After decomposing a relation into smaller relations confirmation of the dependency preservation and lossless join properties must be validated. In this paper authors have focused on these two properties and have developed two algorithms that guarantee that the dependency preservation and lossless join properties are achieved during decomposition.

Conflicts of interest

The authors have no conflicts of interest to declare.

7. REFERENCES

[1] Codd E. A Relational Model for Large Shared Data Banks. Comm. of ACM 13. 1970; 377–387.

[2] Zadeh A. L. Fuzzy Sets. Information and Controls 8. 1965; 338–353.

[3] Gau W. L. and Buehrer D. J., 1993, "Vague Sets", IEEE Trans. Syst. Man, Cybernetics, Vol. 23, No. 2, pp. 610-614.

[4] Mishra J, Ghosh S. A Multivalued Integrity Constraint in Fuzzy Relational Database. International Journal of Computational Cognition 9. 2011; 72–78.

[5] Chen G. , Kerre E. E., Vandenbulcke, J. Normalization Based on ffd in a Fuzzy Relational Data Model. Information Systems 21.1996; 299–310.

[6] Raju N.V.S.V.K., Majumdar K.A. Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database system. ACM Transactions on Database Systems 13. 1988: 129–166.

[7] Yazici A, Sozat, I. M. The Integrity Constraints for Similarity-Based Fuzzy Relational Databases. International Journal of Intelligent Systems 13. 1998; 641–659.

[8] Zhao A, Ma L, Yan M. Z. A Vague Relational Model and Algebra. Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD2007). 2007; 81-85.

[9] Zhao F, Ma M. Z. Vague Query Based on Vague Relational Model. Springer- Verlag Berlin Heidelberg, AISC 61. 2009; 229-238.

[10] Mishra J, Ghosh S. A Vague Multivalued Data Dependency. International Journal of Fuzzy Information and Engineering, Springer Vol. 5. 2013; 459-473.

[11] Smarandache F. First International Conference on Neutrosophy, Neutrosophic Probability, Set, and Logic. University of New Mexico. 2001; 1(3).

[12] De S, Mishra J. A New Approach of Functional Dependency in a Neutrosophic Relational Database Model. Asian Journal of Computer Science and Technology.2019; 8(2):44-48.

[13] De S, Mishra J. Query Processing of Inconsistent Data using Neutrosophic Set. IEEE International Conference on Computing Communication and Automation.2016; 10(6):120-124.

[14] De S, Mishra J. Compare Different Similarity Measure Formula Based Imprecise Query on Neutrosophic Data. International Conference on Advanced Computing and Intelligent Engineering, 2016; 5(12): 330-334.

[15] De S, Mishra J. Inconsistent Data Processing Using Vague Set and Neutrosophic Set for Justifying Better Outcome. IEEE International Conference on Inventive Communication and Computational Technologies,2017;.7(4):26-30.

[16] De S, Mishra J. Processing of Inconsistent Neutrosophic Data and Selecting Primary Key from the Relation. International Conference on Inventive Computing and Informatics. 2017;.6(7): 245-250.

8. AUTHOR'S PROFILE

Educational Qualification

Pursuing Ph. D (Computer Science & Engineering from Maulana Abul Kalam Azad University of Technology, West Bengal, India)

M. Tech (Computer Science & Engineering, 2007)

Soumitra De was born in West Bengal, India. Received M. Tech degree in Computer Science & engineering from West Bengal University of Technology , (presently MAKAUT, West Bengal), India in 2007 and pursuing PhD degree in Computer Science and Engineering from West Bengal University of Technology, Kolkata, (presently Maulana Abul Kalam Azad University, West Bengal), India. Major field of study: Fuzzy, Vague and Neutrosophic Database, Object Oriented Database, Artificial Intelligence. Published 20 International Journal and Conference papers. Currently working as an Assistant Professor in the Department of Computer Science and Engineering, College of Engineering & Management, Kolaghat, West Bengal, INDIA.

Educational Qualification

Ph. D (Computer Science & Engineering)

M. Tech (Computer Applications)

Jaydev Mishra was born in West Bengal, India. Received M. Tech degree in Computer Applications from Indian School of Mines University, Dhanbad (presently IIT-ISM, Dhanbad), India in 1998. Also obtained PhD degree in Computer Science and Engineering from West Bengal University of Technology, Kolkata, (presently Maulana Abul Kalam Azad University, West Bengal), India in 2014. Major field of study: Fuzzy, Vague and Neutrosophic Database, Object Oriented Database. Published 30 International Journal and Conference papers. Currently working as an Assistant Professor in the Department of Computer Science and Engineering, College of Engineering & Management, Kolaghat, West Bengal, INDIA.