

## Research Article

# Lossless Join Decomposition for Extended Possibility-Based Fuzzy Relational Databases

Julie Yu-Chih Liu<sup>1,2</sup>

<sup>1</sup> Department of Information Management, Yuan Ze University, Chung Li, Taoyuan 320, Taiwan

<sup>2</sup> Innovation Center for Big Data & Digital Convergence, Yuan Ze University, Chung Li, Taoyuan 320, Taiwan

Correspondence should be addressed to Julie Yu-Chih Liu; [imyuchih@saturn.yzu.edu.tw](mailto:imyuchih@saturn.yzu.edu.tw)

Received 16 January 2014; Revised 5 August 2014; Accepted 7 August 2014; Published 21 August 2014

Academic Editor: Hui-Shen Shen

Copyright © 2014 Julie Yu-Chih Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Functional dependency is the basis of database normalization. Various types of fuzzy functional dependencies have been proposed for fuzzy relational database and applied to the process of database normalization. However, the problem of achieving lossless join decomposition occurs when employing the fuzzy functional dependencies to database normalization in an extended possibility-based fuzzy data models. To resolve the problem, this study defined fuzzy functional dependency based on a notion of approximate equality for extended possibility-based fuzzy relational databases. Examples show that the notion is more applicable than other similarity concept to the research related to the extended possibility-based data model. We provide a decomposition method of using the proposed fuzzy functional dependency for database normalization and prove the lossless join property of the decomposition method.

## 1. Introduction

Database normalization plays a crucial role in the design theory of relational database to avoid insertion and deletion and update anomalies in a database. The database normalization involves decomposition of a relation schema (table) into several smaller ones. The essential requirement of the decomposition is *lossless join* property, which ensures that the original relation can be obtained from its decomposed results via combination operations [1]. Several methods have been proposed to design normalized relation schemes based on the keys and functional dependencies of a relation to achieve *lossless join decomposition* [2, 3]. The design theory has been applied to fuzzy databases, in which uncertain and imprecise information can be represented and manipulated. The fuzzy databases are extended from the classical databases based on fuzzy sets and possibility theory [4], and they can be resemblance-based fuzzy model [5, 6] and possibility-based fuzzy model [7, 8]. In the context of fuzzy databases, fuzzy functional dependency (FFD) has emerged to extend the classical functional dependency to represent functional relationships between classes/attributes of objects for fuzzy database

models. Various FFD definitions have been proposed in some fuzzy data models for database normalization [9, 10].

However, very few research methods discuss lossless join property for the normalization in possibility-based fuzzy databases. To achieve lossless join decomposition by using FFDs for the possibility-based fuzzy databases is more difficult than for the resemblance-based fuzzy databases, especially for extended possibility-based fuzzy database. The extended possibility-based fuzzy database [7] is an extension of possibility-based fuzzy database [8] by including a resemblance-based fuzzy model [6]. In the fuzzy database, attribute values could be the possibility distributions of the attribute on its domain. Additionally, the elements in a domain have some degree of resemblance. Previous work has applied FFDs on the decomposition for the fuzzy database [10–12]. Informally, these FFDs are based on a certain degree of similarity between two attribute values. Namely, two tuples that are similar but not identical might be regarded as redundant. Applying the similarity-based FFDs on relation decomposition prompts the difficulty for lossless join decomposition on two facets: (i) redundancy removal: how to eliminate redundant tuples that are not identical from the

decomposed results so that the results can be, later on, used to produce the original relation without losing information and (ii) tuple merging: how to combine two relations via merging their tuples of which attribute values are similar but not identical.

Complicating this problem further, most similarity measures [7, 13, 14] of values in the form of possibility distribution are not *transitive*. When tuple redundancy is determined by the similarity measures of nontransitivity, the result of eliminating redundant tuples from a decomposed relation might be not unique (or order sensitive<sup>1</sup>). An inconsistent data redundancy removal not only leads to unstable results of data integration, as described in [15], but also causes decomposition results not lossless. When the decomposition result of a relation is not unique, the combination of the result will have many different outcomes, at least one of which is different from the original relation. Accordingly, the decomposition inevitably violates the lossless join property. Moreover, the nonunique results occur for relation combination when the attribute values to be joined/merged have similarity relation of nontransitivity.

To avoid nontransitivity, Chen et al. provided FFD with embedded classical FD [11, 16], where redundancy removal is restricted to duplicate tuples. But, this restriction draws the normalization process back to the traditional operations of crisp data. To obtain transitive relationship among tuples, some research applied the max-min transitive closure on the relationship matrix of similarity degree between tuples [17]. The max-min transitive closure of a relationship matrix must be a matrix with max-min transitivity [18]. By referring to the transitive closure of the relationship matrix, the tuples which have similarity higher than a given threshold can be grouped into disjointed sets. The tuples in the same set were regarded as redundant. However, this approach cannot determine the similarity of two tuples by merely examining these two, and the similarity is changed by inserting or deleting other tuples. The nondeterministic and dynamic characteristic is not applicable to the practice of databases.

To our knowledge, very few studies provide a complete guideline to perform normalization that ensures lossless join decomposition in the fuzzy databases. Therefore, the purpose of this study is to fill up this gap. This study first proposes a notion of approximate equality which represents the transitive equivalent relation among tuples. Then, it provides new definition of FFD and lossless join decomposition based on approximate equality for the fuzzy databases. Both functional dependencies and lossless join decomposition in a traditional database are special cases in this proposal. Examples show that the notion is more applicable than other similarity concepts to the research related to the fuzzy databases. This work also provides the method of achieving the lossless join decomposition for the fuzzy databases.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to database normalization and fuzzy database and the survey of the similarity measures related to the fuzzy database. Section 3 demonstrates the problem of using nontransitive similarity measures for determining tuple redundancy and provides a notion of approximate equality for it. The FFD is then defined based

on the approximate equality in Section 4. Besides, the lossless join decomposition is proposed for the fuzzy databases, and its property is proven as well. Section 5 draws the conclusion of this paper.

## 2. Preliminaries

This section first briefly reviews the essential operations for lossless join decomposition in traditional databases. Then, it introduces the fuzzy databases considered in this work and the similarity measures of values in form of possibility distribution.

**2.1. Essential Operations for Lossless Join Decomposition.** In traditional relational database, a row is called a tuple; a column header is called an attribute; and the table is called a relation. Given an  $m$ -ary relation schema  $R(A_1, A_2, \dots, A_m)$ , an instance of  $R$  denoted by  $r(R)$  is the set of all tuples in  $R$ . Let  $\mathbf{A}$  denote a set of attributes  $A_1, A_2, \dots, A_m$ . A functional dependency FD  $A_i \rightarrow A_j$  existing in  $R(\mathbf{A})$  represents the tuples having the same values on attribute  $A_i$  that must be identical on  $A_j$ , where  $A_i, A_j \in \mathbf{A}$ . Two operations are related to the lossless join decomposition: *projection* and *natural join*. The operation *projection* generates a result by selecting certain attributes from given relation and removing redundant tuples. Let  $\Theta$  denote a set of attributes in  $R(\mathbf{A})$ ; that is,  $\Theta \subset \mathbf{A}$ . The result of projection  $R$  over attributes  $\Theta \subseteq \mathbf{A}$  is  $\Pi_{\Theta}(R) = \{t[\Theta] \mid t \in r(R)\}$ , where  $t[\Theta]$  represents the composite of values on  $\Theta$  in tuple  $t$ . The *natural join* (denoted by  $*$ ) of  $R'(\mathbf{XY})$  and  $R''(\mathbf{YZ})$  is obtained by removing duplicate attribute from the results of *equal join* on joined attribute  $\mathbf{Y}$  and is denoted as shown below:

$$\begin{aligned} R' * R'' &= \{(t'[\mathbf{X}], t'[\mathbf{Y}], t''[\mathbf{Z}]) : t'[\mathbf{Y}] = t''[\mathbf{Y}] \mid t' \in r(R'), \\ &\quad t'' \in r(R'')\}. \end{aligned} \quad (1)$$

The *natural join* and *projection* operations are, respectively, used to combine and decompose relations.

Formally, decomposition  $\{R_1, R_2, \dots, R_k\}$  of  $R$  is *lossless join* if equation  $r(R) = \Pi_{R_1}(R) * \Pi_{R_2}(R) * \dots * \Pi_{R_k}(R)$  holds. In other words, the *lossless join* decomposition ensures that the combination of the decomposed results of a relation has no spurious tuple or missing tuple to the relation via natural join operation [1].

For example, given a relation  $R$ , the results of  $R' = \Pi_{A_1, A_2}(R)$  and  $R'' = \Pi_{A_2, A_3}(R)$  are shown in Table 1.

In this case, the decomposition  $\{R', R''\}$  of  $R$  has lossless join property because the *natural join* result of  $R'$  and  $R''$  is exactly the same as  $R$  (as shown in Table 2).

**2.2. The Fuzzy Databases.** In last two decades, fuzzy concepts have been incorporated in traditional databases [5, 8, 19] and applied to measure the relation between data [20–22]. The fuzzy databases enable dealing with imprecision and uncertainty in the real world based on the theory of

TABLE 1

	R			R'		R''	
	$\underline{A}_1$	$A_2$	$A_3$	$\underline{A}_1$	$A_2$	$A_2$	$A_3$
$t_1$ :	$x$	$p$	$m$	$x$	$p$	$p$	$m$
$t_2$ :	$y$	$p$	$m$	$y$	$p$	$q$	$n$
$t_3$ :	$z$	$q$	$n$	$z$	$q$		

TABLE 2

	equal join of $R'$ and $R''$ on $A_2$				$R' * R''$		
	$\underline{A}_1$	$A_2$	$A_2$	$A_3$	$\underline{A}_1$	$A_2$	$A_3$
$t_1$ :	$x$	$p$	$p$	$m$	$x$	$p$	$m$
$t_2$ :	$y$	$p$	$p$	$m$	$y$	$p$	$m$
$t_3$ :	$z$	$q$	$q$	$n$	$z$	$q$	$n$

fuzzy sets and possibility distribution theory. The possibility-based fuzzy theory has been widely applied in environmental management, such as flood-diversion planning [23], water resources management [24], and air quality management [25]. This work considers the extended possibility-based databases proposed by Chen et al. [7] because it can capture both the possibility-based fuzzy model and the resemblance-based fuzzy concept. The fuzzy database has drawn much attention of research on semantic measures, information processing, update operation, and UML class diagram therewith [20, 26, 27]. The data model of the fuzzy databases is a hybrid of a possibility-based data model in [8] and a resemblance-based data model in [6]. The possibility-based model derives from Zadeh's fuzzy theory. In the theory [4], a fuzzy set  $F$  on a universe of discourse  $U$  is described by  $\{\mu_F(u)/u \mid u \in U\}$ , where  $\mu_F : U \rightarrow [0, 1]$  is a membership function for the fuzzy set  $F$  and  $\mu_F(u)$  denotes the degree of membership of  $u$  in  $F$ . In a possibility-based database [8], the value of an attribute  $A$  on a domain  $D$  is a possibility distribution  $\pi_A = \{\pi_A(u)/u \mid u \in D\}$ , where  $\pi_A(u)$  denotes the possibility that  $u$  is the actual value of  $A$ . For example,  $D = \{u_1, u_2, u_3\}$  and  $\pi_A = \{0.8/u_1, 0.5/u_2, 0.6/u_3\}$ . An example of applying the possibility-based fuzzy theory<sup>2</sup> in real world is shown below. Consider a domain of attribute "eye color" is black, brown, blue, green and a possibility distribution is given below:

$$\text{Asia-Color} = \left\{ \frac{1.0}{\text{black}}, \frac{0.8}{\text{brown}}, \frac{0.3}{\text{blue}}, \frac{0.1}{\text{green}} \right\}. \quad (2)$$

Suppose that John's eye color is an "Asia color." Then, according to the interpretation for possibility-based fuzzy theory, one concludes that the possibility of John's eye being brown blue color is 0.3.

In the extended possibility-based database, attribute values are represented by possibility distributions of an attribute on its domain, and a domain is associated with a similarity relation of domain elements. Formally, an  $m$ -ary relation instance  $r$  on a schema  $R(A_1, A_2, \dots, A_m)$  in the fuzzy database is a subset of *Cartesian product* of  $\Phi(A_1) \times \Phi(A_2) \times \dots \times \Phi(A_m)$ , where  $\Phi(A_i)$  represents a set of all possibility distributions of attribute  $A_i$  on its domain. For a domain  $D_i$ , a proximity relation is given to describe the resemblance

between domain elements in  $D_i$ . A proximity is a mapping  $s_i : D_i \times D_i \rightarrow [0, 1]$  with reflexivity and symmetry; that is,  $s_i(u, u) = 1$  and  $s_i(u, v) = s_i(v, u)$ . The elements in a domain cannot directly be partitioned into disjoint equivalent classes by a threshold cutting on the proximity relation for the domain elements.

To acquire equivalent classes of a proximity relation on a domain, Sheno et al. [18] proposed  $\alpha$ -proximate relation. Two elements  $u, v \in D$  are  $\alpha$ -proximate (denoted by  $uS_\alpha^+v$ ) if  $s(u, v) > \alpha$  or there exists a sequence  $w_1, w_2, \dots, w_r \in D$ , Such that  $\min\{s(u, w_1), s(w_1, w_2), \dots, s(w_{r-1}, w_r), s(w_r, v)\} > \alpha$ . Given a proximity relation  $s$  and a threshold  $\alpha$  for domain  $D$ , the domain can be partitioned into disjoint subsets (called  $\alpha$ -proximate equivalent classes) such that the elements in a partition are  $\alpha$ -proximate. The equivalent classes are regarded as basic concepts for the methods being reviewed or proposed hereinafter.

By extending traditional functional dependency, research has proposed variety of fuzzy functional dependencies (FFDs) for fuzzy databases [14, 21, 28]. The FFDs are determined by the degree of similarity of attribute values rather than by the identity. Several similarity measures of attribute values are proposed for the extended possibility-based frameworks [7, 16, 20, 29]. Most of them provide the estimates within the interval  $[0, 1]$ . The similarity measures are briefly restated hereinafter, in which  $s$  and  $\alpha \in [0, 1]$ , respectively, denote the proximity relation and a threshold defined on a given domain  $D = \{u_1, u_2, \dots, u_n\}$ ;  $\pi_A$  and  $\pi_B$  represent two possibility distributions on  $D$ . The degree of closeness between  $\pi_A$  and  $\pi_B$ , denoted by  $\xi_1(\pi_A, \pi_B)$ , is defined as follows [29]:

$$\begin{aligned} \xi_1(\pi_A, \pi_B) &= \begin{cases} 0 & \min_{u_i, u_j \in \widehat{D}} s(u_i, u_j) < \alpha \\ \min_{u_i \in D} (1 - |\pi_A(u_i) - \pi_B(u_i)|) & \text{otherwise,} \end{cases} \end{aligned} \quad (3)$$

where  $\widehat{D} = \{u_i \in D : \pi_A(u_i) > 0\} \cup \{u_i \in D : \pi_B(u_i) > 0\}$ .

The measure of  $\xi_1$  may give low degree of similarity for two values that are very similar to each other, for example,  $\xi_1(\{0.9/\text{excellent}, 1/\text{good}\}, \{1/\text{good}\}) = 0.1$ . To prevent some counter-intuitive estimates of  $\xi_1$ , Chen et al. defined the possibility that  $\pi_A = \pi_B$  is true as shown below [7] (here  $\wedge$  denotes minimum):

$$\xi_2(\pi_A, \pi_B) = \sup_{s(u_i, u_j) \geq \alpha, u_i, u_j \in D} (\pi_A(u_i) \wedge \pi_B(u_j)). \quad (4)$$

This assessment is widely adopted in the extended possibility-based databases and is adoptable for the application with subnormal distribution (i.e.,  $\xi_2(\pi_A, \pi_A) < 1$ , or see [4] for details). For normal distribution, Chen et al. [16] included *identity* relation (denoted by  $=_{id}$ ) into (4) as follows:

$$\xi_3(\pi_A, \pi_B) = \begin{cases} 1 & \pi_A =_{id} \pi_B \\ \sup_{s(u_i, u_j) \geq \alpha, u_i, u_j \in D} \pi_A(u_i) \wedge \pi_B(u_j) & \text{o.w.} \end{cases} \quad (5)$$

Ma et al. defined the similarity measure from the perspective of the semantic closeness between two attribute values [20] as shown below:

$$\xi_4(\pi_A, \pi_B) = \delta(\pi_A, \pi_B) \wedge \delta(\pi_B, \pi_A), \quad (6)$$

where  $\delta$  denotes a semantic inclusion degree. Consider the following:

$$\delta(\pi_A, \pi_B) = \frac{\sum_{u_i, u_j \in D, s(u_i, u_j) \geq \alpha} (\pi_A(u_i) \wedge \pi_B(u_j))}{\sum_{j=1}^n \pi_B(u_j)}. \quad (7)$$

The notion of  $\xi_4$  may violate the convention that the similarity degree of two values lies within  $[0, 1]$ ;  $\xi_4(\pi_A, \pi_B) = 1.52$  for the case that  $\pi_A = \{0.9/\text{excellence}, 0.8/\text{good}\}$  and  $\pi_B = \{0.7/\text{excellence}, 0.6/\text{good}\}$  when the similarity of “excellence” and “good” is larger than the given threshold; that is,  $s(\text{excellence}, \text{good}) \geq \alpha$ . It is difficult to set up a proper threshold for estimates that range out of  $[0, 1]$ , having an unpredictable upper bound.

Liu et al. [13] extended the semantic equivalence to ensure that the result of similarity measure lies within  $[0, 1]$ . The measurement adjusts the possibility distributions of values based on  $\alpha$ -proximate equivalent classes of the domain before measuring their similarity. Let  $C = \{C_1, C_2, \dots, C_r\}$  be the  $\alpha$ -proximate equivalent classes of domain  $D$ . The adjusted value of possibility distribution  $\pi_*$  is defined as follows:

$$\tilde{\pi}_* = \bigcup_{k=1}^r \left\{ \frac{\bar{u}_{*k}}{u'} : u' \in C_k \right\}, \quad (8)$$

where  $\bar{u}_{*k} = \sum_{j=1, u_j \in C_k}^{|\bar{C}_k|} \pi_*(u_j) / |\bar{C}_k|$  and  $\bar{C}_k = \{u_j \in C_k : \pi_*(u_j) > 0\}$ ,  $k = 1, \dots, r$ . Then,

$$\xi_5(\pi_A, \pi_B) = \delta(\pi_A, \pi_B) \wedge \delta(\pi_B, \pi_A),$$

$$\text{where } \delta(\pi_A, \pi_B) = \frac{\sum_{i=1}^n (\tilde{\pi}_A(u_i) \wedge \tilde{\pi}_B(u_i))}{\sum_{j=1}^n \tilde{\pi}_B(u_j)}. \quad (9)$$

Although the methods mentioned above differ from each other on measuring similarity of attribute values, most of the methods of measuring the similarity of tuples are the same. The methods adopt the minimum of the similarity of each pair of attribute values. Given tuples  $t = (\pi_{A_1}, \pi_{A_2}, \dots, \pi_{A_m})$  and  $t' = (\pi'_{A_1}, \pi'_{A_2}, \dots, \pi'_{A_m})$ , the resemblance of tuples  $t$  and  $t'$ , denoted by  $\eta(t, t')$ , is given by

$$\eta(t, t') = \min_{i=1, \dots, m} \xi_*(\pi_{A_i}, \pi'_{A_i}), \quad (10)$$

where  $\xi_*$  could be either  $\xi_1, \xi_2, \xi_3, \xi_4$ , or  $\xi_5$  in (3)–(9). Tuples  $t$  and  $t'$  are redundant to each other if  $\eta(t, t') \geq \alpha$ , where  $\alpha$  is a given threshold. The similarity measure of tuples has been applied to extract representative tuples for reducing information redundancy [17].

Fuzzy functional dependency (FFD) is a concept derived from traditional FD. Both FFD and FD have several applications on databases, for example, redundancy elimination

[30], missing data prediction, fuzzy data compression [17, 31], and lossless join decomposition [10, 28, 32]. In literature, various FFDs are defined for different fuzzy data model. For some fuzzy data representation, FFDs are defined based on the equivalence classes of tuples, such as the similarity-based fuzzy data model [33]. In the extended possibility-based databases, the definition of FFD is also of variety, such as literature [10, 14, 34, 35]. One example among the FFD definitions in the literature is listed below.

**Definition 1** (see [10], fuzzy functional dependency). Let  $X \sim Y$  denote that attribute  $X$  is fuzzy functional which depends on attribute  $Y$  in a relation  $R$ . The FFD:  $X \sim Y$  holds in the instance  $r(R)$  if and only if  $\eta(t[X], t'[X]) \leq \eta(t[Y], t'[Y])$  for every  $t, t' \in r(R)$ .

The example helps in understanding the problem of applying the FFDs on relation decomposition in the fuzzy databases illustrated in Section 3.

### 3. Redundancy Removal and Tuple Merging

Several factors determine whether the relation decomposition possesses the lossless join property. They are the ways to decompose a relation, to remove redundant tuples, and to combine the decomposed results. Redundancy removal is to eliminate redundant tuples. If the similarity measures used to measure tuple redundancy are not *transitive*, the result of redundancy removal could be nonunique. An example of *nontransitivity* is that tuples  $t'$  and  $t''$  are redundant to each other, and  $t'$  and  $t''$  are redundant as well, but  $t$  and  $t''$  are not redundant. In this case, the result of redundancy removal will be  $\{t, t''\}$  if  $t'$  is deleted first, which differs from the one-tuple result (either  $\{t\}$  or  $\{t''\}$ ) when first deleting the tuples other than  $t'$ . The nontransitivity makes the result of redundancy removal order sensitive and hinders the lossless join decomposition.

Nevertheless, most well-defined similarity measures [7, 10, 20, 29] for the values of possibility distribution are reflexive and symmetric but not transitive. For example, consider adopting (4) to measure the similarity of tuples. Given three values  $\pi_A = \{0.6/u_1, 1.0/u_2, 0.6/u_3\}$ ,  $\pi'_A = \{1.0/u_1, 1.0/u_2, 0.6/u_3\}$ , and  $\pi''_A = \{1.0/u_1, 0.6/u_2, 0.6/u_3\}$ , then we have  $\xi_2(\pi_A, \pi'_A) = 1$ ,  $\xi_2(\pi'_A, \pi''_A) = 1$ , and  $\xi_2(\pi_A, \pi''_A) = 0.6$ . Considering tuples  $t = (\pi_A)$ ,  $t' = (\pi'_A)$ , and  $t'' = (\pi''_A)$ , we have  $\eta(t, t') \geq \alpha$  and  $\eta(t', t'') \geq \alpha$  but  $\eta(t, t'') < \alpha$  for any  $\alpha > 0.6$  according to (10). Thus, the similarity measure of tuples is not transitive.

In generalizing *projection* and *equal join* operations of traditional database to fuzzy databases, when the redundancy removal is order sensitive, it is hard to obtain lossless join decomposition. Consider the case that  $Y \sim Z$  holds in the instance  $r(R)$  of relation  $R(X, Y, Z)$  based on Definition 1; namely,  $\eta(t[Y], t'[Y]) \leq \eta(t[Z], t'[Z])$  for every  $t, t' \in r(R)$ . Assuming that  $r(R)$  consists of three tuples  $\langle x_1, y, z \rangle$ ,  $\langle x_2, y', z' \rangle$ ,  $\langle x_3, y'', z'' \rangle$  and  $X$  is a key attribute, it is possible that the two values in each of pairs  $(y, y')$ ,  $(y, y'')$ ,  $(z, z')$ , and  $(z, z'')$  are redundant to each other, but  $(y', y'')$  and  $(z', z'')$



are not. Since  $Y \sim Z$ ,  $R$  should be decomposed to avoid redundancy. After decomposing  $R(X, Y, Z)$  to  $R'(X, Y)$  and  $R''(Y, Z)$ , if tuple  $\langle y, z \rangle$  is first removed because it is redundant to tuple  $\langle y', z' \rangle$ , the result of  $R''(Y, Z)$  contains two tuples  $\langle y', z' \rangle$  and  $\langle y'', z'' \rangle$ . The natural join of  $R'(X, Y)$  and  $R''(Y, Z)$  generates a four-tuple result  $\langle x_1, y, z' \rangle, \langle x_1, y, z'' \rangle, \langle x_2, y', z' \rangle, \langle x_3, y'', z'' \rangle$ , which contains spurious tuple.

To resolve this problem, this study proposes the operations of *projection* and *equal join* for the fuzzy databases, which involves evaluation of redundancy and tuple merging. Since the decomposition of relations is based on FFD, it depends on the similarity of tuples. For the data in the fuzzy model, (3)–(9) can be used to measure the similarity of tuples and define FFDs in the fuzzy databases. However, (5) restricts redundant tuples to those duplicate. Equations (3), (4), and (6) lack transitivity. Therefore, this work adopts (9) and (10) to define *approximate equality* for the tuples that might not be identical but have high similarity degree. The *approximate equality* enables obtaining a unique result of redundancy removal.

**Definition 2** (approximately equal tuples). Two tuples  $t = (\pi_1, \pi_2, \dots, \pi_m)$  and  $t' = (\pi'_1, \pi'_2, \dots, \pi'_m)$  are approximately equal, denoted by  $t \cong t'$ , if it is satisfied that  $\min_{j=1, \dots, m} \{\xi_5(\pi_j, \pi'_j)\} = 1$ .

In other words, tuples  $t$  and  $t'$  are approximately equal if their similarity  $\eta(t, t') = 1$ .

**Lemma 3.** The approximate equality of tuples (or attribute values) is transitive.

*Proof.* Based on (9), it is obvious that if  $\xi_5(\pi_A, \pi_B) = 1$  and  $\xi_5(\pi_B, \pi_G) = 1$ , then  $\xi_5(\pi_A, \pi_G) = 1$ . Thus, if  $t \cong t'$  and  $t' \cong t''$ , then  $t \cong t''$  based on (10).  $\square$

The tuples of *approximate equality* are considered to be redundant to each other. The notion of approximate equality can be applied to query processing with the predicate containing fuzzy concept [36] for fuzzy databases in different models. For simplicity, we let  $\pi_A \cong \pi_B$  denote  $\xi_5(\pi_A, \pi_B) = 1$  hereafter.

**Example 4.** Given values  $\pi_A = \{0.75/\text{pretty}, 0.65/\text{cuteness}\}$  and  $\pi_B = \{0.6/\text{pretty}, 0.7/\text{charm}, 0.8/\text{cuteness}\}$  on domain  $D$  and equivalent classes  $C_1 = \{\text{pretty}\}$  and  $C_2 = \{\text{charm}, \text{cuteness}\}$  for  $D$ , the average possibilities of  $\pi_B$  are  $\bar{u}_{B1} = 0.6/1 = 0.6$  and  $\bar{u}_{B2} = (0.7 + 0.9)/2 = 0.8$ , yielding  $\bar{\pi}_B = \{0.6/\text{pretty}, 0.8/\text{charm}, 0.8/\text{cuteness}\}$ . Likewise,  $\bar{u}_{A1} = 0.8$ ,  $\bar{u}_{A2} = 0.65$ , and  $\bar{\pi}_A = \{0.75/\text{pretty}, 0.65/\text{charm}, 0.65/\text{cuteness}\}$ . We have  $\delta(\pi_A, \pi_B) = (0.6 + 0.65 + 0.65)/(0.6 + 0.8 + 0.8) = 0.86$  and  $\delta(\pi_B, \pi_A) = 1.9/2.05 = 0.92$ . Thus,  $\xi_5(\pi_A, \pi_B) = \min\{0.86, 0.92\} = 0.86$ . Given  $\pi_G = \{0.6/\text{pretty}, 0.65/\text{charm}, 0.85/\text{cuteness}\}$  on  $D$ , we have  $\pi_G \cong \pi_B$  even though  $\pi_G$  is not identical to  $\pi_B$ .

**Proposition 5.** The approximate equality can be used to classify values of the fuzzy database into disjoint sets (equivalence classes).

*Proof.* Based on the definition of (9), it is obvious that  $\xi_5$  is *reflective* and *symmetric*; that is,  $\xi_5(\pi_A, \pi_A) = 1$  and  $\xi_5(\pi_A, \pi_B) = \xi_5(\pi_B, \pi_A)$  for values  $\pi_A$  and  $\pi_B$ . Besides, approximate equality is *transitive* according to Lemma 3. Therefore, two different sets of approximately equal values are either disjoint sets or same class sets, where any two of the values are approximately equal to each other.  $\square$

The transitivity of similarity measure is important to any operation involving redundancy removal or tuple merging. Besides, the measure of transitivity can be applied to clustering methods or data groupings, such as the ones in [36, 37].

**Proposition 6.** Given  $\pi_A$  and its adjusted value  $\bar{\pi}_A$  following (8),  $\pi_A \cong \bar{\pi}_A$ .

*Proof.* It is obvious by the definition of (9).  $\square$

Buckles and Petry first proposed the way of tuple merging and applied it to remove redundant tuples in a fuzzy database [5]. Tuple merging can also be used at join operation. This study extends the tuple merging of Chen et al. [16] to be (11) for relation combination as well as redundancy removal. Given tuples  $t = (\pi_{A_1}, \pi_{A_2}, \dots, \pi_{A_m})$  and  $t' = (\pi'_{A_1}, \pi'_{A_2}, \dots, \pi'_{A_m})$ , tuple merging of  $t$  and  $t'$ , denoted by  $t \circ t'$ , is given by

$$t \circ t' = (\bar{\pi}_{A_1} \cup_F \bar{\pi}'_{A_1}, \bar{\pi}_{A_2} \cup_F \bar{\pi}'_{A_2}, \dots, \bar{\pi}_{A_m} \cup_F \bar{\pi}'_{A_m}), \quad (11)$$

where each  $\bar{\pi}_\bullet$  (or  $\bar{\pi}'_\bullet$ ) is the adjusted value of  $\pi_\bullet$  (or  $\pi'_\bullet$ ) according to (8) and  $\cup_F$  denotes fuzzy union. For single-value tuples  $t = (\pi_{A_1})$  and  $t' = (\pi'_{A_1})$ , tuple merging is alternatively denoted by  $\pi_{A_1} \circ \pi'_{A_1}$ .

**Lemma 7.** Let  $\pi_A$  and  $\pi'_A$  be two possibility distributions on the same domain. If  $\pi_A \cong \pi'_A$ , then  $\pi_A \cong \pi_A \circ \pi'_A \cong \pi'_A$ .

*Proof.* Based on (9) and (11), it is obvious that if  $\xi_5(\pi_A, \pi'_A) = 1$ , then  $\xi_5(\pi_A, \pi_A \circ \pi'_A) = 1$  and  $\xi_5(\pi'_A, \pi_A \circ \pi'_A) = 1$ .  $\square$

Based on the literature review and Lemma 7, we summarize the property of different similarity measures with threshold  $\alpha = 1$  in Table 3 to show the merit of (9) adopted in this work.

## 4. Approximate Lossless Join Decomposition

This section first offers the operations for relation decomposition and combination. Then, it proposes a notion of *approximate lossless join decomposition* (ALJD), which incorporates fuzzy concepts into lossless join decomposition. It also provides the method to achieve the ALJD.

Similar to the works in [37], this study generalizes the *projection* and *natural join* operations in traditional database to the fuzzy databases, as below. Here, given a relation  $R$ ,  $\Theta$  denotes a set of attributes in  $R$  (i.e.,  $\Theta \subset R$ ), and  $t[\Theta]$  denotes the composite of values in tuple  $t$  over attribute  $\Theta$ .

TABLE 3: Property of different similarity measures with threshold  $\alpha = 1$ .

Properties\measures	Equation (5) [16]	Equation (6) [20]	Equation (9) [13]
Transitivity	Yes	No	Yes
Result after merging	Lossless	Non-lossless	Approximate lossless
Predicates for lossless join	Identical values	Different values	Different values

For example, given  $t \in r(R)$ ,  $t = (\pi_{A_1}, \pi_{A_2}, \dots, \pi_{A_m})$ , and  $\Theta = (A_2, A_3, A_4)$ , then  $t[\Theta] = (\pi_{A_2}, \pi_{A_3}, \pi_{A_4})$ .

*Projection.* Projecting the instance of relation  $R$  on attributes  $\Theta \subset R$ , denoted by  $\Pi_{\Theta}(R)$ , is given by

$$\Pi_{\Theta}(R) = \{t[\Theta] \circ t'[\Theta] : t[\Theta] \cong t'[\Theta] \mid t, t' \in r(R)\}. \quad (12)$$

*Natural Join.* Natural join instances of  $R(X, Y)$  and  $R'(Y, Z)$ , denoted by  $R \otimes R'$ , are defined as follows:

$$\begin{aligned} R \otimes R' \\ = \{(t[X], t[Y] \circ t'[Y], t'[Z]) : t[Y] \cong t'[Y] \mid t \in r(R), \\ t' \in r(R')\}. \end{aligned} \quad (13)$$

In (12) and (13), tuple redundancy is determined by approximate equality (e.g.,  $t[Y] \cong t'[Y]$ ; see Definition 2), and both redundancy removal and tuple combination use tuple merging in (11).

**Proposition 8.** *The projection result of a relation based on (12) must be unique.*

*Proof.* It can be directly derived from Proposition 5.  $\square$

Based on the operations (12) and (13), the ALJD is formally defined following the extension of approximate equality from tuple level to relation level in Definition 9.

**Definition 9** (approximately equal relation instances). Two relation instances  $r(R)$  and  $r'(R)$  in the fuzzy database are approximately equal, denoted by  $r(R) \approx r'(R)$ , if for every tuple  $t \in r(R)$ , there must exist a tuple  $t' \in r'(R)$  such that  $t \cong t'$  and vice versa.

**Definition 10** (approximate lossless join). A composition  $\{R_1, R_2, \dots, R_k\}$  of a relation  $R$  in the fuzzy database is approximate lossless join if  $r(R) \approx (\Pi_{R_1}(R) \otimes \Pi_{R_2}(R) \otimes \dots \otimes \Pi_{R_k}(R))$ .

The approximate lossless join decomposition means the natural join of all decomposed results of a relation instance is approximately equal to the original relation instance. More specifically, every tuple in the original relation is *approximately equal* to one of tuples in the combination result.

**Proposition 11.** *Consider the following:*

$$\Pi_{\Theta}(R) \approx \{t[\Theta] \mid t \in r(R)\}. \quad (14)$$

*Proof.* It can be derived from (11) and (12).  $\square$

**Corollary 12.** *Consider the following:*

$$\Pi_R(R) \approx r(R). \quad (15)$$

*Proof.* It can be derived directly from Proposition 11.  $\square$

The projection of a relation over the same schema, as shown in Corollary 12, represents no operations other than removing redundant tuples from the instance of the relation via tuple merging. Corollary 12 shows that the result of redundancy removal of a relation instance is approximately equal to the original instance. This property is essential for obtaining the combination result that is approximately equal to the original instance after relation decomposition.

This study proposes FFD for the decomposition in the fuzzy database as shown below.

**Definition 13.** The FFD  $X \rightsquigarrow Y$  holds in the relation instance  $r(R)$  if  $r(R)$  satisfies that, for every  $t, t' \in r(R)$ ; if  $t[X] \cong t'[X]$ , then  $t[Y] \cong t'[Y]$ .

**Remark 14.** An FD in a traditional database is a special case of the FFD. If a FD  $X \rightarrow Y$  holds in  $r(R)$ , then  $X \rightsquigarrow Y$  holds as well. It is because  $t[\Theta] \cong t'[\Theta]$  must be true for any  $\Theta \subset R$  if  $t[\Theta] = t'[\Theta]$ .

**Lemma 15.** *Given relations  $R$  and  $R'$  and  $r(R') \approx r(R)$ , if  $r(R)$  satisfies a set  $F$  of FFDs, then  $r(R')$  satisfies  $F$ .*

*Proof.* Proof by contradiction: we assumed that  $r(R') \approx r(R)$  and there exists an FFD  $X \rightsquigarrow Y$  such that  $R$  satisfies  $X \rightsquigarrow Y$  and  $R'$  does not. Because  $X \rightsquigarrow Y$  exists in  $r(R)$ ,

$$\begin{aligned} \text{for every } t_1, t_2 \in r(R) \text{ if } t_1[X] \cong t_2[X], \\ \text{then } t_1[Y] \cong t_2[Y]. \end{aligned} \quad (16)$$

Since  $X \rightsquigarrow Y$  does not exist in  $r(R')$ , there exists  $t'_1, t'_2 \in r(R')$ , such that  $t'_1[X] \cong t'_2[X]$  and  $\eta(t'_1[Y], t'_2[Y]) \neq 1$ . Since  $r(R') \approx r(R)$ , there exists  $t'' \in r(R)$  such that  $t \cong t'_1$  and  $t'' \cong t'_2$ . Then, we have  $t, t'' \in r(R)$  and  $t[X] \cong t''[X]$  but  $\eta(t[Y], t''[Y]) \neq 1$ , which contradicts (16).  $\square$

It is noted that the FFD in Definition 13 satisfies Armstrong's axioms (inference rules), including reflexive rule, augmentation rule, and transitive rule<sup>3</sup>. This property enables the result of *lossless join decomposition* that has dependency preservation property<sup>4</sup> [1].

Inputs  $R$  and  $F$ , where  $R$  is a relation and  $F$  is the set of FFDs exists in  $r(R)$ .

Step 1. Let  $\mathfrak{R} = \{R\}$  and  $F'$  be the set of all  $f \in F$  that are not trivial.

Step 2. For a  $f': X \rightsquigarrow Y$  in  $F'$ , Let  $R'$  be the relation chosen from  $\mathfrak{R}$  such that both  $X$  and  $Y$  are in  $R'$ .

Step 3. If  $X$  is not a key attribute in  $R'$ , do followings:

(1) decompose  $R'$  into  $R_1$  and  $R_2$ , such that  $R_1 = \Pi_{(X,Y)}(R')$  and  $R_2 = \Pi_{R'(\bar{X})}(R')$

(2) let  $F' = F' - \{f'\}$  (remove  $f'$  from  $F'$ )

(3) let  $\mathfrak{R} = \mathfrak{R} \cup \{R_1, R_2\} - \{R'\}$ .

Step 4. Go to Step 2 if  $F'$  is not empty.

Output is  $\mathfrak{R}$ , the set of relations decomposed from  $R$ .

Note:  $R'(\bar{X})$  represents the list of all attributes in  $R'$  other than  $X$ .

#### ALGORITHM 1: ALJD Algorithm.

**Lemma 16.** Let  $R(A, X, Y)$  be a relation and  $X \rightsquigarrow Y$  be an FFD in  $r(R)$ . If  $R_1 = \Pi_{(A,X)}(R)$  and  $R_2 = \Pi_{(X,Y)}(R)$ , then the decomposition  $\{R_1, R_2\}$  of  $R$  has approximate lossless join property.

*Proof.* We proved that  $r(R) \approx \Pi_{R_1}(R) \otimes \Pi_{R_2}(R)$  based on Definition 10. Let  $R'$  be a relation such that  $r(R') = \Pi_{R_1}(R) \otimes \Pi_{R_2}(R)$ . We first prove that, for all  $t \in r(R)$ , there exist  $t' \in r(R')$  and  $t' \cong t$  and then prove that, for all  $t \in r(R')$ , there must exist  $t' \in r(R)$  and  $t' \cong t$ . Proof by contradiction: let us assume that  $t \in r(R)$  is the tuple such that no  $t' \in r(R')$  satisfies  $t' \cong t$ . Let  $t_1$  and  $t_2$  be tuples such that  $t_1 = t[A, X]$  and  $t_2 = t[X, Y]$ . Based on Proposition 11, there must be  $\hat{t}_1 \in r(R_1)$  and  $\hat{t}_2 \in r(R_2)$  such that  $\hat{t}_1 \cong t_1$  and  $\hat{t}_2 \cong t_2$  because  $R_1 = \Pi_{(A,X)}(R)$  and  $R_2 = \Pi_{(X,Y)}(R)$ . Since  $r(R') \approx \Pi_{R_1}(R) \otimes \Pi_{R_2}(R)$  and  $t_1[X] \cong t[X] \cong t_2[X]$ , there must exist  $t' \in r(R')$  such that  $t' \cong (t_1[A], t_1[X] \circ t_2[X], t_2[Y])$  according to (13). Also, since  $t_1 = t[A, X]$  and  $t_2 = t[X, Y]$ , we have  $(t_1[A], t_1[X] \circ t_2[X], t_2[Y]) \cong (t[A], t[X], t[Y])$  by Lemma 7. Thus,  $t' \cong t$ , which contradicts the assumption.

Proof by contradiction for second part with renewed symbols: assume that  $t' \in r(R')$  is the tuple such that no  $t \in r(R)$  satisfies  $t' \cong t$ . Since  $r(R') \approx \Pi_{R_1}(R) \otimes \Pi_{R_2}(R)$ , there exist  $t_1 \in r(R_1)$  and  $t_2 \in r(R_2)$  such that  $t_1 \cong t'[A, X]$ ,  $t_2 \cong t'[X, Y]$ , and  $t_1[X] \cong t_2[X]$  based on (13). Also, we let  $\hat{t}_1$  and  $\hat{t}_2$  be the tuples such that

$$\begin{aligned} \hat{t}_1 &\in r(R_1), & \hat{t}_2 &\in r(R_2), \\ \hat{t}_1 &\cong t'[A, X], & \hat{t}_2 &\cong t'[X, Y]. \end{aligned} \quad (17)$$

Since  $R_1 = \Pi_{(A,X)}(R)$ , based on Proposition 11, there must exist  $t \in r(R)$  such that

$$t[A, X] \cong \hat{t}_1. \quad (18)$$

Likewise, there must also exist  $t'' \in r(R)$

$$\text{such that } t''[X, Y] \cong \hat{t}_2. \quad (19)$$

Based on (17) and (18), we have  $t[X] \cong t''[X]$ . Because  $X \rightsquigarrow Y$ ,  $t[Y] \cong t''[Y]$  holds based on Definition 13. Thus,  $t[Y] \cong \hat{t}_2[Y] \cong t'[Y]$  based on (19), and  $t' \cong t$ , which contradicts the assumption.  $\square$

In Lemma 16, each one of  $A$ ,  $X$ , and  $Y$  could be a single attribute or a set of attributes.

**Definition 17.** Let  $F$  be the set of FFDs. An FFD  $f: X \rightsquigarrow Y$  in  $F$  is trivial if there exists an FFD  $f': X \rightsquigarrow Y'$  in  $F$  such that  $Y \subset Y'$ .

Based on Armstrong's inference rules, IR1 and IR3 (see endnote 2), if a set  $F$  of FFD contains  $X \rightsquigarrow YZ$ , the closure of  $F$  will also contain  $X \rightsquigarrow Y$  and  $X \rightsquigarrow Z$ , which is trivial. When a relation is decomposed into more relations, it takes more join operations to obtain the original data for query process. Considering the cost of join operations, it is not efficient to decompose a relation that has already been in the third normal form. A relation is in the third normal form if there is no functional dependency between nonkey attributes in the relation [1]. Accordingly, the relation decomposition has two prerequisites as follows.

- (i) It needs to avoid decomposing a relation based on trivial FFDs.
- (ii) It needs to make sure that the decomposed result preserves the closure of FFDs in the original relation.

For example, if  $X$  is not a key in  $r(R)$ , then  $R$  will be decomposed based on  $X \rightsquigarrow YZ$  rather than on trivial FFD  $X \rightsquigarrow Y$  or  $X \rightsquigarrow Z$ . Based on Lemma 16 and Definition 17, we propose an algorithm for ALJD (see Algorithm 1).

In the ALJD algorithm, an FFD containing key attributes is excluded from the decomposition process at Step 3. This follows the concept of the normalization of traditional databases, where only the FD of nonkey attributes is considered. To have a consistent presentation of data, this work generalizes the definition of key attributes for the fuzzy databases; namely, an attribute  $A$  is a *key attribute* in  $R$  if there does not exist two tuples  $t$  and  $t'$  in  $r(R)$  such that  $t[A] \cong t'[A]$ . The exclusion of processing FFDs containing key attributes can prevent unnecessary decomposing on the relations which have no update anomaly problem. Although the decomposition without the key exclusion is still an ALJD, it increases the cost of the join operations of query process.

**Proposition 18.** Let  $R(A, X, Y)$  be a relation and let  $X \rightsquigarrow Y$  be an FFD in  $r(R)$ . If  $R_1 = \Pi_{(A,X)}(R)$  and  $R_2 = \Pi_{(X,Y)}(R)$ , then (i) each FFD existing in  $r(R_1)$  or  $r(R_2)$  must exist in  $r(R)$

and (ii) every FFD existing in  $r(R)$  must either exist in  $r(R_1)$  or  $r(R_2)$  or be derived via FFDs in  $r(R_1)$  and  $r(R_2)$ .

*Proof.* Statement (i) can be derived by Proposition 11 and Lemma 15. Statement (ii) can be derived by Lemma 16 and the property of FFD (namely, Armstrong's axiom IRs 1, 2, and 3, described at endnote 2).  $\square$

The above statements show that the ALJD also preserves the closure of FFDs in the original relation, which is important to the issues related to the application of FFDs.

## 5. Conclusion

The contribution of this work is threefold. First, it highlights the problem of relation decomposition when tuple elimination is order sensitive. To overcome the problem, it proposes the notion of approximate equality for the tuples or relations in the fuzzy databases and provides the measure of the approximate equality. The measurement is reflexive, symmetric, and transitive. It enables classifying tuples into disjoint sets and ensures that a decomposed relation has unique result after redundancy removal or tuple merging. Therefore, the notion of approximate equality is important for data operations in the fuzzy databases. Second, it proposes approximate lossless join decomposition for the fuzzy databases and defines two operations *projection* and *equal join* for the decomposition, all of which are based on the approximate equality. The data operations and ALJD can be applied to the issue on data compression in the fuzzy databases. Third, this work defines FFDs and proposes an algorithm to decompose relations in the fuzzy databases based on the FFDs. The decomposition by the algorithm ensures the approximate lossless join property. The FFD and ALJD proposed for the fuzzy databases are, respectively, the general cases of the traditional FD and lossless join decomposition. The general property is important for dealing with the databases containing crisp data and fuzzy data. Forth, similar to the existing approaches of database normalization on resemblance-based fuzzy databases, this study provides several propositions to prove that the proposed approach of decomposition satisfies a degree of lossless join property. Compared to the normalization approaches for resemblance-based fuzzy databases, achieving lossless join decomposition for the extended possibility-based fuzzy databases is more difficult because of having more complex data.

There are some directions of future work. Future study can adopt the notion of approximate equality to define data operations for the query processing in the fuzzy databases. Research can apply the notion on the research related to data compression, fuzzy association rules, missing value prediction, relation compactness, and the integrity constraint in the fuzzy databases. Study aims to incorporate the fuzzy concept into clustering methods or data groupings for decision-making in marketing, healthcare applications, or business operations that can adopt the approximate equality for the similarity measures. Since the fuzzy concept has been incorporated into object-oriented databases in literature, future

work can provide the approximate equality specifically for the data in the fuzzy object-oriented data models.

## Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The author would like to acknowledge the support of National Science Council NSC102-2410-H-155-036-MY2 and Innovation Center for Big Data & Digital Convergence.

## Endnotes

1. Different orders on removing redundant tuples could lead to different results.
2. For further details on possibility distribution and on the difference between possibility and probability measures, the reader is referred to [38].
3. IR1 (reflexive rule):  $X \rightarrow Y$  if  $Y \subseteq X$ ; IR2 (augmentation rule): if  $X \rightarrow Y$ , then  $XZ \rightarrow YZ$ ; IR3 (transitive rule): if  $X \rightarrow Y$  and  $Y \rightarrow Z$ , then  $X \rightarrow Z$  (see [1]).
4. Each FFD in  $r(R)$  either directly exists in some individual relations that decomposed from  $R$  or can be represented via Armstrong's inference rules of the FFDs in these relations.

## References

- [1] S. B. Navathe and R. Elmasri, *Fundamentals of Database Systems*, Addison-Wesley, New York, NY, USA, 2010.
- [2] P. A. Bernstein, J. R. Swenson, and D. C. Tsichritzis, "A unified approach to functional dependencies and relations," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 237–245, 1975.
- [3] E. F. Codd, "A relational model of data for large shared data banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, 1970.
- [4] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978.
- [5] B. P. Buckles and F. E. Petry, "A fuzzy representation of data for relational databases," *Fuzzy Sets and Systems*, vol. 7, no. 3, pp. 213–226, 1982.
- [6] S. Sheno and A. Melton, "Proximity relations in the fuzzy relational database model," *Fuzzy Sets and Systems*, vol. 31, no. 3, pp. 285–296, 1989.
- [7] G. Chen, J. Vandenbulcke, and E. E. Kerre, "A general treatment of data redundancy in a fuzzy relational data model," *Journal of the American Society for Information Science*, vol. 43, no. 4, pp. 304–311, 1992.
- [8] H. Prade and C. Testemale, "Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries," *Information Sciences*, vol. 34, no. 2, pp. 115–143, 1984.
- [9] J. C. Cubero and M. A. Vila, "New definition of fuzzy functional dependency in fuzzy relational databases," *International Journal of Intelligent Systems*, vol. 9, no. 5, pp. 441–448, 1994.



- [10] K. Raju and A. K. Majumdar, "Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems," *ACM Transactions on Database Systems*, vol. 13, no. 2, pp. 129–166, 1988.
- [11] G. Chen, E. 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.
- [12] S. Jyothi and M. S. Babu, "Multivalued dependencies in fuzzy relational databases and lossless join decomposition," *Fuzzy Sets and Systems*, vol. 88, no. 3, pp. 315–332, 1997.
- [13] J. Y. Liu, P. Chang, and C. P. C. Yeh, "Consistent data operations for multi-databases in extended possibility-based data models," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6174–6180, 2009.
- [14] Z. M. Ma, W. J. Zhang, W. Y. Ma, and F. Mili, "Data dependencies in extended possibility-based fuzzy relational databases," *International Journal of Intelligent Systems*, vol. 17, no. 3, pp. 321–332, 2002.
- [15] J. Y. Liu, "Data integration constraints for consistent data redundancy in fuzzy databases," *International Journal of Intelligent Systems*, vol. 23, no. 6, pp. 635–653, 2008.
- [16] G. Chen, J. Vandenbulcke, and E. E. Kerre, "On the lossless-join decomposition of relation scheme(s) in a fuzzy relational data model," in *Proceedings of the 2nd International Symposium on Uncertainty Modeling and Analysis*, pp. 440–446, IEEE, College Park, Md, USA, April 1993.
- [17] J. Zhang, G. Chen, and X. Tang, "Extracting representative information to enhance flexible data queries," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 928–941, 2012.
- [18] S. Sheno, A. Melton, and L. T. Fan, "An equivalence classes model of fuzzy relational databases," *Fuzzy Sets and Systems*, vol. 38, no. 2, pp. 153–170, 1990.
- [19] M. Koyuncu and A. Yazici, "IFOOD: An intelligent fuzzy object-oriented database architecture," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 5, pp. 1137–1154, 2003.
- [20] Z. Ma, W. Zhang, and W. Ma, "Semantic measure of fuzzy data in extended possibility-based fuzzy relational databases," *International Journal of Intelligent Systems*, vol. 15, no. 8, pp. 705–716, 2000.
- [21] V. V. Cross, "Fuzzy extensions for relationships in a generalized object model," *International Journal of Intelligent Systems*, vol. 16, no. 7, pp. 843–861, 2001.
- [22] V. V. Cross, "Defining fuzzy relationships in object models: abstraction and interpretation," *Fuzzy Sets and Systems*, vol. 140, no. 1, pp. 5–27, 2003.
- [23] S. Wang and G. H. Huang, "A two-stage mixed-integer fuzzy programming with interval-valued membership functions approach for flood-diversion planning," *Journal of Environmental Management*, vol. 117, pp. 208–218, 2013.
- [24] S. Wang and G. H. Huang, "An interval-parameter two-stage stochastic fuzzy program with type-2 membership functions: An application to water resources management," *Stochastic Environmental Research and Risk Assessment*, vol. 27, no. 6, pp. 1493–1506, 2013.
- [25] S. Wang and G. H. Huang, "Interactive fuzzy boundary interval programming for air quality management under uncertainty," *Water, Air, and Soil Pollution*, vol. 224, article 1574, 2013.
- [26] Z. M. Ma and F. Mili, "Handling fuzzy information in extended possibility-based fuzzy relational databases," *International Journal of Intelligent Systems*, vol. 17, no. 10, pp. 925–942, 2002.
- [27] Z. M. Ma and L. Yan, "Updating extended possibility-based fuzzy relational databases," *International Journal of Intelligent Systems*, vol. 22, no. 3, pp. 237–258, 2007.
- [28] P. Bosc, D. Dubois, and H. Prade, "Fuzzy functional dependenciesan overview and a critical discussion," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 325–330, 1994.
- [29] P. Bosc and O. Pivert, "On the impact of regular functional dependencies when moving to a possibilistic database framework," *Fuzzy Sets and Systems*, vol. 140, no. 1, pp. 207–227, 2003.
- [30] E. A. Rundensteiner, L. W. Hawkes, and W. Bandler, "On nearness measures in fuzzy relational data models," *International Journal of Approximate Reasoning*, vol. 3, no. 3, pp. 267–298, 1989.
- [31] P. Bosc, D. Dubois, and H. Prade, "Fuzzy functional dependencies and redundancy elimination," *Journal of the American Society for Information Science*, vol. 49, no. 3, pp. 217–235, 1998.
- [32] Z. M. Ma, W. J. Zhang, and F. Mili, "Fuzzy data compression based on data dependencies," *International Journal of Intelligent Systems*, vol. 17, no. 4, pp. 409–426, 2002.
- [33] B. Bhuniya and P. Niyogi, "Lossless join property in fuzzy relational databases," *Data and Knowledge Engineering*, vol. 11, no. 2, pp. 109–124, 1993.
- [34] Ö. 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.
- [35] T. K. Bhattacharjee and A. K. Mazumdar, "Axiomatisation of fuzzy multivalued dependencies in a fuzzy relational data model," *Fuzzy Sets and Systems*, vol. 96, no. 3, pp. 343–352, 1998.
- [36] Z. M. Ma and L. Yan, "Generalization of strategies for fuzzy query translation in classical relational databases," *Information and Software Technology*, vol. 49, no. 2, pp. 172–180, 2007.
- [37] J. Zhang, Q. Wei, and G. Chen, "An efficient incremental method for generating equivalence groups of search results in information retrieval and queries," *Knowledge-Based Systems*, vol. 32, pp. 91–100, 2012.
- [38] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, vol. 144 of *Mathematics in Science and Engineering*, Academic Press, New York, NY, USA, 1980.