INFORMATICA, 2008, Vol. 19, No. 1, 113–134

Vague Rough Set Techniques for Uncertainty Processing in Relational Database Model

Karan SINGH

Department of Computers, Central Institute of Agricultural Engineering
Nabi Bagh, Berasia Road, Bhopal - 462 038 (M.P.), India
e-mail: singhkaran@yahoo.com

Samajh Singh THAKUR

Department of Applied Mathematics, Government Engineering College
Jabalpur (M.P.) - 482 011, India
e-mail: samajh_singh@rediffmail.com

Mangi LAL

Department of Computer Science and Mathematics
Rani Durgawati Vishwavidyalaya, Jabalpur (M.P.), India

Received: January 2006

Abstract. The study of databases began with the design of efficient storage and data sharing techniques for large amount of data. This paper concerns the processing of imprecision and indiscernibility in relational databases using vague rough technique leading to vague rough relational database model. We utilize the notion of indiscernibility and possibility from rough set theory coupled with the idea of membership and non-membership values from vague set theory to represent uncertain information in a manner that maintains the degree of uncertainty of information for each tuple of the original database and also those resulting from queries. Comparisons of theoretical properties of operators within this model with those in the standard relational database model are discussed. A simple entity-relationship type diagram for database design, a database definition language and an SQL-like query language for vague rough relational database model are described.

Key words: vague set, vague rough set, vague rough relational database model.

1. Introduction

Many real world systems and applications require information management components that provide support for managing imprecise and uncertain data. A trend in databases Beaubouef and Petry (2000), Cornelis *et al.* (2003), Date (1989), Prade and Testemale (1984), Ullman (1982) has been the usage by non-computer scientists, individuals with little or no knowledge of the technical aspects of database systems. Consequently, the external view of such systems is becoming more removed from the hardware technology and simple data models, and closer to human cognition. Although the relational database provides the necessary foundation for rigid database modeling, it does not directly support the modeling of 'human-related' concepts such as ambiguity, imprecision and uncertainty. Therefore, extensions to the relational model of data may be considered to provide the necessary mechanisms for a higher-level, more human-like model of data. In constructing a database model we always attempt to maximize usefulness. The same is closely connected with the representation and processing the information that is imprecise and uncertain in nature. In the past years the fuzzy set techniques have been used for the modeling of uncertainties in databases. In 1982, Buckles and Petry (1982) proposed fuzzy relational database for representing incorrect information in real world problems. In 1984, Prade and Testemale (1984) defined fuzzy databases using possible distribution over the attribute domains. Since 1982, significant work has been done in incorporating uncertainty management in relational databases using fuzzy set theory (Kerre *et al.* (1986), Kerre (1992)).

There are many potential areas of application of the newly proposed technique such as awareness creation, decision support systems, data warehousing and data mining and bio-informatics.

1.1. *Approaches for Processing of Uncertainty*

A single model cannot process all type of uncertaintities. Wong (1982) model can process incomplete information, Bagai and Suderraman (1995) clearly pointed out that their model can process incomplete and inconsistent information. Beaubouef and Petry (1994) model can process only indiscernibility. Our proposed model can process indiscernibility and imprecision. We found various approaches for representing and processing uncertainty in the context of different domains of applicability in the literature. The main approaches related to this study are briefly summarized in the following:

1.1.1. *Classical Set Theory*

A classical set or crisp set is a collection of well-defined objects. A crisp set A of universal set U can be defined by its characteristic function: $f_A\colon U \to \{0,1\}$, such that $f_A(a) = 1$ if $a \in A,\ f_A(a) = 0$ if $a \notin A$.

The relational model of data proposed by Codd, is based on this theory and has been described in detail in Codd (1970). This model is very popular and is able to handle the precise information only.

1.1.2. *Fuzzy Set Theory*

Fuzzy sets (Kerre, 1992; Takahasi, 1991), similar to classical sets, are capable of expressing nonspecificity. In fuzzy sets, the membership is not a matter of affirmation or denial, but rather a matter of degree. Lotfi Zadeh (1965), first defined fuzzy sets in their present form, to provide a method for constructing numerical controllers for complex electronic equipment. He summarized his motivation "as the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance become almost mutually exclusive characteristics." He extended the work to include the concept of a linguistic variable, which has the values that are words or sentences in natural language, and the concept of fuzzy logic. The concept of fuzzy set can be defined by a *membership function*: $\mu_A : U \rightarrow [0, 1]$, where $\mu_A(a)$ expresses a degree of membership of a in A, or the strength of a belief that 'a belongs to A'. Such a set A is called fuzzy set. This theory is capable to define the linguistic variables and gradual changes; hence it is more suited for control systems but can also be utilized for uncertainty management in databases. Buckles and Petry (1982), Dubois and Prade (1994), Kerre *et al.* (1986), Klir (1987), Medina *et al.* (1995), Petry (1992), Prade and Testemale (1984), and Wang and Klir (1992) used this theory for modeling and extensions of relational data model.

1.1.3. *Rough Set Theory*

Pawlak (1991) initiated the first stream in 1982, who launched rough set theory as a framework for the construction of approximations of concepts when only incomplete information is available. A rough set is an imprecise representation of a crisp set in terms of two subsets, a lower approximation and upper approximation. The available information consists of a set A of examples of a concept C, and a relation R in X. R models *indiscernibility* or *indistinguishability* and therefore generally is a tolerance relation and in most cases even an equivalence relation.

In the literature we found that "rough sets work well for classification. Rough sets appear to be well suited for data mining, which is the detection of significant relationships in data, particularly in data warehouses". Beaubouef used this theory for modeling the imprecision in databases and developed the rough relational database model which is explained in references Beaubouef and Petry (1994a, 1994b).

1.1.4. *Vague Set Theory*

Vague sets, similar to fuzzy sets, are capable of expressing nonspecificity. In addition, they are also capable of expressing *vagueness* since these sets consider both membership and non-membership function values as against only membership value, in the case of fuzzy sets. Chen (2003) and Chen and Jong (1997) pointed out that the single value tells nothing about the accuracy of belongingness of an element in the set. A vague concept has a boundary line cases, i.e., elements of the universe, which cannot be with certainty classified as elements of the concept.

In the present paper we developed an approach using both vague set and rough set theories for imprecise and uncertain data handling in a relational database model. As pointed out by Prade and Testemale (1984), rough sets capture the idea of indiscernibility among members of a set and utilize a discrete formalism of set partitions. Correspondingly, vague sets as pointed out by Gau and Buehrer (1993) can be viewed as capturing imprecision by the nature of a vaguely defined set boundary and represented as a generalization of a discrete set membership and non-membership function by a continuous function. The detail about various data models is explained in the reference Singh *et al.* (2005).

1.2. *Exact Benefits of the Proposed Model*

i. It may be used to express positive as well as negative preferences, in a logical context, with a proposition a degree of truth and one of falsity may be associated within databases, it can serve to evaluate the satisfaction as well as the violation of relational constraints.
ii. It may be used to express positive as well as negative preferences, in a voting context, where degree of truth, falsity and abstentions are to be evaluated with exactness since none of the known model is able to handle such situations so far.

1.3. *Potential Application Areas of New Data Model*

There are many potential areas of applications of new data model. Some of the prominent areas are:

1) **Awareness creation**
The awareness can be created about the issues related to a potentially hazardous nuclear or chemical plants by organizing the rallies and conducting the public meetings. The study requires the gathering and analysis of data, which incorporate uncertainty in databases.
2) **Web mining**
Essentially the data and documents on the Web are heterogeneous, uncertainty is unavoidable. Using the presentation and reasoning method of our data model, it is easier to capture uncertain information on the Web, which will provide more potentially value-added information.
3) **Bio-informatics**
There is a proliferation of data sources. Each research group and each new experimental technique seems to generate yet another source of valuable data. But these data can be uncertain and imprecise and even incomplete. So how to represent and extract useful information from these data will be a challenging problem, this model can help up to a great extent.
4) **Decision support system**
In decision support systems, we need to combine the database with the knowledge base. There will be a lot of uncertain and even inconsistent information, so we need an efficient data model to capture these information and reasoning with these information. This model can address such type of problems.

The paper is organized as follows. Section 2 of the paper deals with some of the basic definitions and concepts of various sets and relational model of data. Section 3 introduces vague rough relational data model and algebraic operators. Section 4 illustrates SQL-like query language for the proposed model. Finally, Section 5 contains some concluding remarks on the proposed model.

2. Background

2.1. *Relational Databases*

The relational database model introduced by Codd (1970) uses the mathematical concept of a relation as its data structure. In relational view, the data are stored in two-dimensional tables, which have a specific number of columns and some number of unordered rows. The tables are presented by mean of n-ary relations and are defined as ordinary subsets of a Cartesian product $D_1 \times D_2 \times \cdots \times D_n$, where each D_i is a domain for some attribute. The rows of a relational table are usually called tuples and columns are called attributes, a row has a value in each column in a table. The set of values an attribute can take on is called domain. A domain is a set of atomic values from which attributes draw their values, often specified in terms of data type and format. In his original paper on the relational model, Codd (1970) introduced ***eight*** basic operators, which could be used to manipulate data within the body parts of tables of a relational database. ***Four*** of these, union, intersection, difference and Cartesian product are traditional set operations, albeit modified to take note of the fact that their operands are relations, which, as has been seen, are special kinds of sets. The other ***four*** are the special relational operators restriction, projection, join and division. He provided the basis for a very minimal data manipulation language (DML) for information management. The basic operators are all incorporated into the standard, international, relational database language structured query language (SQL), as explained by Beaubouef and Petry (1994b), Bosc *et al.* (1988), Madina *et al.* (1995) and Takahasi (1991).

The relational algebra has the algebraic property of closure. Any operation applied to one or more relations produces a new relation. It also has the property that the operations DIFFERENCE, UNION, PROJECT, PRODUCT, and SELECT form a complete set. All other relational operations can be defined in terms of these. Therefore, these five operators are sufficient to specify data associated with any relationship in the database design, and query languages having these operations are called *relationally complete*.

2.2. *Vague Sets*

The theory of vague sets (Gau and Buehrer, 1993) is a generalization of theory of fuzzy sets because when the sum of positive and negative evidences equals to one, for all elements of the universe, the traditional fuzzy set concept is recovered.

Definition 2.2.1. Let U be the universe of discourse $U = \{u_1, u_2, \ldots u_n\}$, with a generic element of U denoted by u_i. A vague set A in U is characterized by true membership function $\mu_A\colon U \to [0,1]$ and a false membership function $\nu_A\colon U \to [0,1]$. Where $\mu_A(u_i)$ is a lower bound on the grade of membership of u_i derived from the evidence for u_i, $\nu_A(u_i)$ is a lower bound on the negation of u_i derived from the evidence against u_i, and $\mu_A(u_i) + \nu_A(u_i) \leqslant 1$.

The grade of membership of u_i in the vague set A is bounded to a sub interval $[\mu_A(u_i),\ 1 - \nu_A(u_i)]$ of [0,1]. The vague value $[\mu_A(u_i),\ 1 - \nu_A(u_i)]$ indicates that the exact grade of membership $\eta_A(u_i)$ of u_i may be unknown, but is bounded by $\mu_(u_i) \leqslant \eta_A(u_i) \leqslant 1 - \nu_A(u_i)$, where $\mu_A(u_i) + \nu_A(u_i) \leqslant 1$. The amount $1 - \mu_A(u_i) - \nu_A(u_i)$ is called hesitation part, which may cater to either true-membership value or false-membership value or both (Fig. 1).

For more clarity, the Fig. 1 may be explained as follows. The precision of our knowledge about u_i is immediately clear, with our uncertainty characterized by the difference $1 - \mu_A(u_i) - \nu_A(u_i)$. If this is small, our knowledge about u_i is relatively precise; if it is large, we know correspondingly little. If $1 - \nu_A(u_i)$ is equal to $\mu_A(u_i)$, our knowledge about u_i is exact, and the theory reverts back to that of fuzzy sets. If $1 - \nu_A(u_i)$ and $\mu_A(u_i)$ both are equal to 1 or 0 depending on whether u_i does or does not belong to A, our knowledge about u_i is very exact and the theory reverts back to that of ordinary sets. In the literature we found that the terms membership and non-membership are also being used in place of true-membership and false-membership.

When the universe of discourse U is continuous, a vague set A can be written as

$$A = \int_U [\mu_A(u_i),\ 1 - \nu_A(u_i)]/u_i.$$

When the universe of discourse U is discrete, a vague set A can be written as

$$A = \sum_{i=1}^{n} [\mu_A(u_i),\ 1 - \nu_A(u_i)]/u_i.$$

Example 1. For a vague set $[\mu_A(u_i),\ 1 - \nu_A(u_i)]/u_i$, we say that the interval $[\mu_A(u_i),\ 1 - \nu_A(u_i)]$ is the vague value to the object u_i. Consider $[\mu_A(u_i),\ 1 - \nu_A(u_i)] = [0.6,\ 0.8]$, we can see that $\mu_A(u_i) = 0.6,\ 1 - \nu_A(u_i)] = 0.8$ and $\nu_A(u_i) = 0.2$. It is interpreted as "the degree that object u_i belongs to the vague set V is 0.6, the degree that object u_i does not belong to the vague set V is 0.2." In a voting process, the vague value [0.6, 0.8] can be interpreted as "the vote for resolution is 6 in favor, 2 against, and 2 neutral."

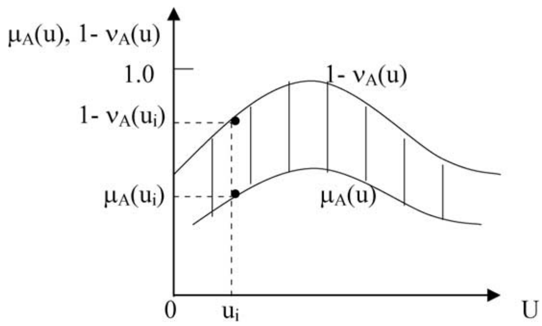

Fig. 1. Illustration of vague set.

2.2.1. *Properties of Vague Sets*
The properties of vague sets such as empty vague set, complement of a vague set, containment of vague sets, equality of vague sets, union of vague sets and intersection of vague sets are explained in detail by Gau *et al.* (1993).

Definition 2.2.1.1. A vague relation from a non-empty set X to Y is a vague subset of $X \times Y$, i.e., a vague relation R is characterized by true-membership function μ_R: $X \times Y \rightarrow [0,1]$ and a false-membership function ν_R: $X \times Y \rightarrow [0,1]$, where $\mu_R(x,y) + \nu_R(x,y) \leqslant 1$, $\forall (x,y) \in X \times Y$.

The true and false membership values of vague sets X and Y also holds true for all the subsets of $X \times Y$. The other properties of vague relations such as reflexive, symmetric etc. are defined by Gau *et al.* (1993).

2.3. *Rough Sets*

Rough set theory, introduced by Pawlak (1991) is a technique for dealing with uncertainty and for identifying cause-effect relationships in databases as a form of database learning.

Definition 2.3.1. Given the approximation space A defined on some universe U with equivalence relation R imposed upon A, U is partitioned into equivalence classes called elementary sets. The unions of combinations of these elementary sets define other sets in A. Given that $X \subseteq U$, X can be defined in terms of the definable sets in A by the following:
The *lower approximation* of X in A is the set $\underline{R}X = \{x \in U \mid [x]_R \subseteq X\}$.
The *upper approximation* of X in A is the set $\overline{R}X = \{x \in U \mid [x]_R \cap X \neq \Phi\}$.

The set approximation $\underline{R}X$ and $\overline{R}X$ may also be described as R-positive ($\underline{R}X$) region, R-negative ($U - \underline{R}X$) region and the R-borderline region ($\underline{R}X - \overline{R}X$). $[x]_R R$ denotes the equivalence class of R containing x, for an element x of U. The theory has been illustrated with example by Beaubouef and Petry (1994a).

2.4. *Vague Rough Sets*

In this section, we introduce the concept of vague rough sets by coupling both vague sets and rough sets. Along with the lower and upper approximations, rough set theory provides two kinds of membership: if an element belongs to the lower approximation of A, we are dealing with strong membership of A. It is very natural to extend this idea to fuzzy rough set theory: the strong membership function of the lower approximation of A, while the weak membership function of A is the membership function of the upper approximation of A. In vague rough set theory we also deal with two kinds of functions, namely a membership function μ and a non-membership function ν.

Note that the strong membership function of the fuzzy rough set Radzikowska and Kerre (2002) can be considered as the membership function μ of a vague rough set, while the complement of the weak membership function of the fuzzy rough set can be used as the non-membership function ν. Vague sets and rough sets model different type of uncertainty. Since both types are relevant for database applications, it is useful to combine the two concepts.

The applications of this idea are manifold. It may be used to express positive as well as negative preferences, in a logical context, with a proposition a degree of truth and one of falsity may be associated within databases, it can serve to evaluate the satisfaction as well as the violation of relational constraints.

Definition 2.4.1. Let U be a universe and X a rough set in U. A vague rough set A in U is characterized by a membership function μ_A: $U \rightarrow [0, 1]$ and a non-membership function ν_A: $U \rightarrow [0, 1]$ such that

$$\mu_A(\underline{R}X) = 1,\ 1 - \nu_A(\underline{R}X) = 1 \quad \text{OR } [\mu_A(x),\ 1 - \nu_A(x)] = [1, 1] \quad \text{if } x \in \underline{R}X$$

$$\mu_A(U - \overline{R}X) = 0,\ 1 - \nu_A(U - \overline{R}X) = 0$$

$$\text{OR } [\mu_A(x),\ 1 - \nu_A(x)] = [0, 0] \text{ if } x \in U - \overline{R}X$$

and $0 \leqslant \mu_A(\overline{R}X - \underline{R}X) + \nu_A(\overline{R}X - \underline{R}X) \leqslant 1$.

Example 2. Let U = {Child, Pre-Teen, Teen, Youth, Teenager, Young-Adult, Adult, Senior, Senior-Citizen, Elderly} be a universe.

Let the equivalence relation R be defined as follows:

$R^* =$ {[Child, Pre-Teen], [Teen, Youth, Teenager], [Young-Adult], [Adult], [Senior, Senior-Citizen, Elderly]}.

Let X ={Child, Pre-Teen, Youth, Young-Adult}.

We can define X in terms of its lower and upper approximations:

$\underline{R}X$ = {Child, Pre-Teen, Young-Adult}, and

$\overline{R}X$ ={Child, Pre-Teen, Teen, Youth, Teenager, Young-Adult}.

The membership and non-membership functions μ_A: $U \rightarrow [0, 1]$ and ν_A: $U \rightarrow [0, 1]$ can be defined at as follows:

μ_A(Child) = 1, μ_A(Pre-Teen) = 1, μ_A(Young-Adult) = 1,
ν_A(Child) = 0, ν_A(Pre-Teen) = 0, ν_A(Young-Adult) = 0,
μ_A(Adult) = 0, μ_A(Senior) = 0, μ_A(Senior-Citizen) = 0, μ_A(Elderly) = 0,
ν_A(Adult) = 1, ν_A(Senior) = 1, ν_A(Senior-Citizen) = 1, ν_A(Elderly) = 1,
μ_A(Teen) = 0.3, μ_A(Teenager) = 0.4, μ_A(Young) = 0.5,
ν_A(Teen) = 0.5, ν_A(Teenager) = 0.4, ν_A(Young) = 0.3.

Such a set A defined in U on rough set X is called vague rough set in which vague values are represented as [μ_A(Child), $1 - \nu_A$(Child)] = [1,1], [μ_A(Teenager), $1 - \nu_A$(Teenager)] = [0.4,0.6] etc.

3. Vague Rough Relational Database Model

3.1. *Introduction and Definition*

A data model is an abstract model of the data stored in a Database Management System (DBMS). This abstraction allows the user of a DBMS to focus on the context of the information, rather than the details of its physical storage. To provide, such an abstraction, a data model has two components: (1) a notation for describing data, and (2) a set of operations for manipulating the data. Vague rough relational data model is the generalization of fuzzy rough relational data model. In fact, it can be easily shown that the fuzzy rough relational data model is a special case of vague rough relational data model. We can use Fig. 2 to express the relationship among RDM, FRDM, RRDM, FRRDM and VRRDM.

The vague rough relational database is an extension of fuzzy rough relational database model proposed by Beaubouef and Petry (2000). In the proposed model a tuple t_i takes the form $(d_{i1}, d_{i2}, \ldots, d_{im}, d_{i[\mu,\nu^*]})$ where d_{ij} is a domain value of a particular domain set D_j and $d_{i[\mu,\nu^*]}$ belongs to closed interval $I = [0,1]$, the domain for true-membership and false-membership values, represented as $d_{i[\mu,\nu^*]} = [d_{i\mu}, d_{i\nu^*}],\ d_{i\mu}, d_{i\nu} \in I$ such that $d_{i\mu} + d_{i\nu} \leqslant 1$. In the vague rough relational database except for the true-membership and false-membership values $d_{ij} \subseteq D_j$ and d_{ij} is not restricted to be a singleton, $d_{ij} \neq \phi$.

Let $P(D_i)$ denote any non-null member of the power set of D_i.

Definition 3.1.1. A vague rough relation R is a subset of the set cross product $P(D_1) \times P(D_2) \times \ldots P(D_m) \times D_{[\mu,\nu^*]}$. Where $D_{[\mu,\nu]}$ denoted the family of closed sub-interval of the closed interval [0,1] such that $\mu \leqslant \nu^*$ or $\mu + \nu \leqslant 1$.

Example 3. For a specific relation, R, membership and non-membership are determined semantically. Given that D_1 is the set of names of people, D_2 is the set of city names then

(J.Singh, Sarojni Naidu,[1,1])

(Kalpna,{S.C.Bose,Gitanjali},[0.6,0.6])

(Josef, Rajiv Gandhi,[1,1])

are elements of $P(D_1) \times P(D_2) \times D_{[\mu,\nu^*]}$.

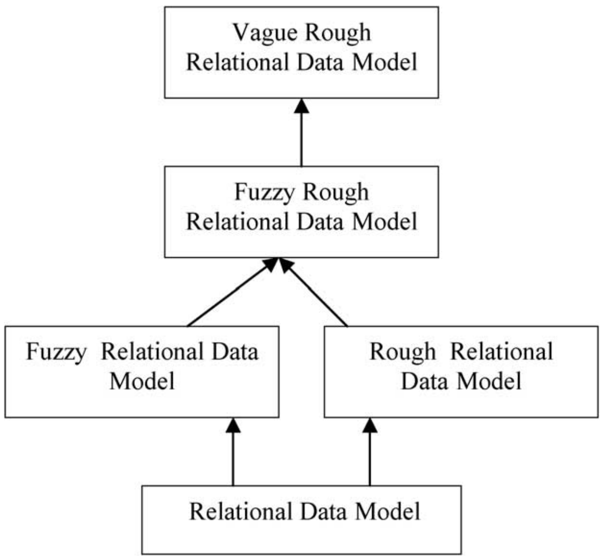

Fig. 2. Relationship among RDM, FRDM, RRDM, FRRDM and VRRDM.

A vague rough tuple t is any member of R. If t_i is some arbitrary tuple then $t_i = (d_{i1}, \ldots, d_{im}, d_{i[\mu,\nu^*]})$ where $d_{ij} \subseteq D_j$ and $d_{i[\mu,\nu^*]}) = [d_{i\mu}, d_{i\nu^*}], d_{i\mu}, d_{i\nu} \in I$ such that $d_{i\mu} + d_{i\nu} \leqslant 1$.

Definition 3.1.2. An interpretation $\alpha = (a_1, a_2, \ldots, a_m, a_{[\mu,\nu^*]})$ of a vague rough tuple $t_i = (d_{i1}, d_{i2}, \ldots, d_{im}, d_{i[\mu,\nu^*]})$ is any value assignment such that $a_j \in d_{ij}$ for all j.

The interpretation space is the cross product $D_1 \times D_2 \times \ldots D_m \times D_{[\mu,\nu^*]}$, but is limited for a given relation R to the set of those tuples which are valid according to the underlying semantics of R. In an ordinary relational database, because domain values are atomic, there is only one possible interpretation for each tuple t_i. Moreover, the interpretation of t_i is equivalent to the tuple t_i. In the vague rough relational database, this is not always the case.

Let $[d_{xy}]$ denote the equivalence class to which d_{xy} belongs. When d_{xy} is a set of values, the equivalence class is formed by taking the union of equivalence classes of members of the set, if $d_{xy} = \{c_1, c_2, \ldots, c_n\}$, then $[d_{xy}] = [c_1] \cup [c_2] \cup \ldots \cup [c_n]$.

Definition 3.1.3. Tuples $t_i = (d_{i1}, d_{i2}, \ldots, d_{in}, d_{i[\mu,\nu^*]})$ and $t_k = (d_{k1}, d_{k2}, \ldots, d_{kn}, d_{k[\mu,\nu^*]})$ are redundant if $[d_{ij}] = [d_{kj}]$ for all $j = 1 \ldots n$.

If relation contains only those tuples of a lower approximation, i.e., those tuples having true-membership value 1 and false-membership value 0, the interpretation α of a tuple is unique. This follows immediately from the definition of redundancy. In vague rough relations, there are no redundant tuples. The merging process used in relational database operations removes duplicate since duplicates are not allowed in sets, the structure upon which the relational model is based. Tuples may be redundant in all values except μ and ν, as in the union of vague rough sets where the maximum membership value and minimum non-membership values are retained. It is the convention of the vague rough relational database to retain the tuple having the higher μ and lower ν (or higher ν^*) value when removing redundant tuple during merging. If we are supplied with identical data from two sources, one certain and the other uncertain, we would want to retain the data that is certain, avoiding loss of information. So, there is need for another definition, which will be used for upper approximation tuples, is necessary for some of the alternate definitions of operators to be presented. This definition captures redundancy between elements of attribute values that are sets.

Definition 3.1.4. Two sub-tuples $X = (d_{x1}, d_{x2}, \ldots, d_{xm}, d_{x[\mu,\nu^*]})$ and $Y = (d_{y1}, d_{y2}, \ldots, d_{ym}, d_{[\mu,\nu^*]})$ are roughly redundant, R, if for some $[p] \subseteq [d_{xj}]$ and $[q] \subseteq [d_{yj}]$; $[p] = [q]$ for all $j = 1, 2, \ldots, m$.

3.2. *Application of the Model in "Awareness Creation"*

We have so far considered theoretical properties. We now present a very simple example where the necessity of incorporating the uncertainty in database is a must. An agency is studying the concerns of citizens who reside near or employed by one of potentially hazardous nuclear or chemical plants. A study is being conducted which documents these concerns and stores all the data in a database. Several public meetings and rallies were conducted to promote public involvement in the project and to gather information from the participants about their concerns. The study requires the gathering and analysis of data to determine issues related to citizens and families. Here, the indiscernibility lies in the age and height attributes since they have the values in linguistic terms such as adult, senior, medium, tall etc. Also the indiscernibility lies in the name of city since there are some people who are not able to tell the exact place or some city values are missing, which are being shown here as a CLASS. We considered 3 tables and wherever there is indiscernibility in the values of attributes, to handle it, we used rough set theory in the form of CLASS, e.g., City in People table and Age and Height in Rally and Meeting table, the values are shown as {Adult, Teen}, {Short, Medium} etc. In all the tables to handle vagueness we are using vague set theory, the evidence in favor and against in the form of true and false membership in each tuple represented as vague values, e.g., [1,1], [0.8,0.9] etc.

3.3. *Vague Rough E-R Diagram*

We first design our database using some type of semantic model and the design methodology for uncertain data as described in Cavallo and Pottareli (1987), Chaudhary *et al.*, (1994) and Medina *et al.* (1995). We use a variation of the entity-relationship diagram that we call a vague rough E-R diagram. This diagram is similar to the standard E-R diagram in that entity types depicted with rectangles, relationships with diamonds, and attributes with ovals. However, in the vague rough model, it is understood that membership and non-membership values exist for all instances of entity types and relationships. Attributes that allow values where we want to be able to define equivalences are denoted with an asterisk (*) above the oval. The vague rough E-R model is similar to fuzzy rough E-R model of second and third levels of fuzziness defined by Zvieli and Chen (1986). However, in our model, all entity and relationship occurrences are of the vague type so we do not mark a ‘v’ beside each one. We do not introduce vagueness at the attribute level of our model in this paper, only roughness, or indiscernibility, and denote those attributes with the “*”. From the fuzzy-rough E-R diagram, Beaubouef and Petry (1988) designed the structure of the fuzzy rough relational database. We have extended this diagram for vague rough relational database. If we have a priori information about the types of queries that will be involved, we can make intelligent choices that will maximize computer resources. A part of vague rough E-R diagram for the considered example appears in Fig. 3.

We introduce a vague rough data definition language (DDL) to define the vague rough relations and the indiscernibility relation. The vague rough DDL is similar to that of SQL, having such commands as CREATE, DROP, etc., but for vague rough relations.

The VRCREATE DDL command creates a base table that is vague rough in the following ways. First of all, it contains an additional attribute called TF, which draws values from the range [0,1], tuple membership and non-membership values. This attribute does not have to be specified. It is automatically included as part of all vague rough relations.

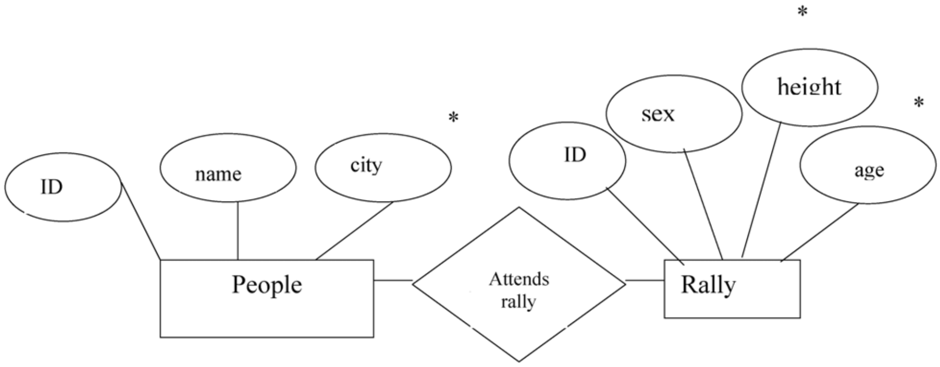

Fig. 3. Illustration of a vague rough E-R diagram.

Additionally, we can specify for each attribute whether or not we allow indiscernibility of values. This is defined by including "IND" along with the attribute line of the table definition. The VRCREATE command is used to define the vague rough relations of the example stated above. The representative tables defined and explained in this section can be found in the *Appendix*.

```
VRCREATE TABLE PEOPLE
( ID      DECIMAL(4),
NAME     CHAR(25),
CITY     CHAR(25) IND,
TF       CHAR(10),
PRIMARY KEY (ID));
```

Similarly, the structure for other tables can be created. One point is to be noted here, the attribute TF is containing the values of true and false memberships. At the time of actual implementation these values should be stored separately either in numeric form or they should get converted in numeric form using some function because we need to compare the values of μs and νs while defining the operators and making the queries from the database.

Once the database schema has been defined, we may begin to store actual data in the vague rough relations. Often database packages have utility programs to expedite the process. Alternatively, we can directly enter data into a relation with the SQL INSERT command. In the vague rough relational database, the command is similar. Data values for all attributes including the membership value and non-membership value are inserted into the specified relation. The vague rough counterpart to SQL's INSERT is VRINSERT.

```
VRINSERT
INTO PEOPLE
VALUES (5002, 'Kalpna', '{Bombay, Gorakhpur }', '[0.5,0.5]');
```

The designer may use the VRINSERT command to enter tuples in the INDISCERNIBILITY relation since it is, after all, a vague rough relation. In order to create new equivalence class, the VRCLASS command is used:

VRCLASS (TEEN, YOUTH, TEENAGER);

The vague rough relational database commands VRCLASS, VRREMOVE, and VRDELETE are special commands created to facilitate operations involving indiscernibility and for updating this special relation of equivalence classes. VRREMOVE and VRDELETE are analogous to their SQL counterparts in standard relational database. The vague rough relational database also has the usual SQL DDL and update commands for deleting or updating tuples and dropping tables. These operate on vague rough relations as DELETE, UPDATE, and DROP TABLE commands operate on ordinary relations.

Uncertainty, ambiguity, and indiscernibility are all prevalent in the considered example. In the next section we formally define the vague rough relational database operators and discuss issues relating to the data representation and modeling. Informally, however, we view indiscernibility as being modeled through the use of the indiscernibility relation, imprecision through the use of non-first normal constructs, and degree of uncertainty and vagueness through the use of tuple membership and non-membership values, which are given as the value for the TF attribute in every vague rough relation.

3.4. *Vague Rough Relational Operators*

We now define the operators for the vague rough relational database and demonstrate the expressive power of the model through its vague relational algebra.

3.4.1. *Difference Operator*

In vague rough relational database, the difference operator is applied to two vague rough relations and, as in the rough relational database, indiscernibility, rather than equality of attribute values, is used in the elimination of redundant tuples. Hence, the difference operator is somewhat more complex.

Let X and Y be two union compatible vague rough relations, the vague rough difference, $X - Y$, between X and Y is a vague rough relation T where

$$\begin{aligned}
&T = \{t(d_1, d_2, \ldots, d_n,\ [\mu_i, \nu_i^*]) \in X\colon t(d_1, d_2, \ldots, d_n, [\mu_i, \nu_i^*]) \notin Y\} \\
&\cup\{t(d_1, d_2, \ldots, d_n, [\mu_i, \nu_i^*]) \in X\colon t(d_1, d_2, \ldots, d_n, [\mu_j, \nu_j^*]) \in Y \\
&\quad \text{and } \mu_i > \mu_j \text{ and } \nu_i^* > \nu_j^*\}
\end{aligned}$$

The resulting vague rough relation contains all those tuples which are in the lower approximation of X, but not redundant with a tuple in the lower approximation of Y. It also contains those tuples belonging to upper approximation of both X and Y, but which have a higher μ value in X than in Y and higher ν^* value in X than in Y.

Consider the vague rough relations Rally and Meeting from the Appendix. The query "*Retrieve the information on individuals who attended the rally but not the meeting*" can be expressed as vague rough difference of the two relations: Rally – Meeting, which yields (Table 1).

Here, it is important to note that ID 5019 exists in both the tables Rally and Meeting but not retrieved in the resulting table because $\mu_i > \mu_j$ but ν_i^* is not $> \nu_j^*$, it does not

Table 1

Vague rough difference of tables Rally and Meeting

ID	***Sex***	***Age***	***Height***	***TF***
5014	M	Adult	Short	[0.7,0.8]
5015	M	{Adult, Teen}	{Medium, Tall}	[0.6,0.7]
5020	M	Adult	{Short, Medium}	[0.6,0.8]

satisfy the condition $\mu_i > \mu_j$ and $\nu_i^* > \nu_j^*$. On the other hand ID 5020 also exists in both the tables and retrieved in the resulting table because it satisfies the condition $\mu_i > \mu_j$ and $\nu_i^* > \nu_j^*$. This example clearly shows, the proposed technique is more powerful than the fuzzy rough set technique where only positive evidences or membership values ($\mu_i s$) are considered at the time of comparison. If we had used fuzzy rough set technique the tuple with ID 5019 would have retrieved.

3.4.2. *Union Operator*

The union operator can be applied to any two union compatible relations to result in a third relation which has its tuples, all the tuples contained in either or both of the two original relations. The union operator can be extended to vague rough relations.

The vague rough union of two vague rough relations X and Y, $X \cup Y$, is a vague rough relation T where

$$T = \{t\colon t \in X \text{ OR } t \in Y\} \text{ and } \mu_T(t) = \text{Max}\,[\mu_X(t), \mu_Y(t)]$$
$$\text{and } \nu_T(t) = \text{Max}\,[\nu_X^*(t), \nu_Y^*(t)].$$

The resulting relation T contains all tuples in either X or Y or both, merged together and having redundant tuples removed. If X contains a tuple that is redundant with a tuple in Y except for the μ and ν^* values, the merging process will retain only that tuple with the higher μ value and higher ν^* value.

The query "*List all information for individuals who participated in either the rally or the meeting or both*", the vague rough union of the relations Rally and Meeting results in the Table 2.

Again, ID 5019 is not retrieved because positive and negative evidences do not satisfy the required condition.

3.4.3. *Intersection Operator*

The vague rough intersection of X and Y, $X \cap Y$, is a vague rough relation T where

$$T = \{t\colon t \in X \text{ and } t \in Y\} \text{ and } \mu_T(t) = \text{Min}\,[\mu_X(t), \mu_Y(t)]$$
$$\text{and } \nu_T(t) = \text{Min}\,[\nu_X^*(t), \nu_Y^*(t)].$$

In intersection, the Min operator is used in merging the equivalent tuples having different μ value and ν^* value respectively, the result contains all the tuples that are members of both of the original vague rough relations.

Table 2
Vague rough union of tables Rally and Meeting

ID	***Sex***	***Age***	***Height***	***TF***
5002	F	Adult	Medium	[0.8,0.9]
5010	M	Senior	Tall	[0.7,0.8]
5014	M	Adult	Short	[0.7,0.8]
5015	M	{Adult, Teen}	{Medium, Tall}	[0.6,0.7]
5018	M	{Adult, Teen}	{Medium, Tall}	[0.5,0.6]
5020	M	Adult	{Short, Medium}	[0.6,0.8]

The query "*Retrieve all information for those individuals who have attended both the rally and meeting*" can be formulated as a vague rough intersection of Rally and Meeting (Table 3).

Once again, ID 5019 is not retrieved because positive and negative evidences do not satisfy the required condition. This makes our technique different from fuzzy rough technique.

3.4.4. *Select Operator*

The select operator for the vague rough relational database model, σ, is a unary operator which takes a vague rough relation X as its argument and returns a vague rough relation containing a subset of the tuples of X, selected on the basis of values for a specified attribute. The vague rough selection, $\sigma_A = a(x)$ of tuples from X is a vague rough relation Y having the same schema as X and where $Y = \{t \in X\colon U_i[a_i] \subseteq U_j[b_j]\}$. Where $a_i \in a,\ b_j \in t(A)$, and where membership and non-membership values for tuples are calculated by multiplying the original values by card(a)/card(b). Where: Card(x) returns the cardinality or number of elements in x.

The result of "*Select all students who are adult and attending the meeting from Meeting relation*"(Table 4).

3.4.5. *Project Operator*

Project is a unary vague rough relational operator. It returns a relation that contains a subset of columns of the original relation. Let X be a vague rough relation with schema with A, and let B be a subset of A. The vague rough projection of X onto B is a vague

Table 3
Vague rough intersection of tables Rally and Meeting

ID	***Sex***	***Age***	***Height***	***TF***
5002	F	Adult	Medium	[0.8,0.9]
5010	M	Senior	Tall	[0.7,0.8]
5020	M	Adult	{Short, Medium}	[0.5,0.6]

Table 4

All students who are adult and attended the Meeting

ID	***Sex***	***Age***	***Height***	***TF***
5002	F	Adult	Medium	[0.8,0.9]
5018	F	{Adult, Teen}	{Medium, Tall}	[0.25,0.3]
5019	F	{Adult, Senior}	Tall	[0.25,0.35]
5020	M	Adult	{Short, Medium}	[0.6,0.8]

rough relation Y obtained by omitting the columns of X which correspond to attributes in $A - B$, and removing redundant tuples and higher μ values has priority over lower ones and higher ν^* values have priority over lower ones.

The vague rough projection of X onto B, $\pi(X)$, is a vague rough relation Y with schema $Y(B)$, where $Y(B) = \{t(B) \mid t \in X\}$ and $\mu_R(t) = \text{Max}\,[\mu_X(t), \mu_Y(t)]$ and $\nu_R(t) = \text{Max}\,[\nu_X^*(t), \nu_Y^*(t)]$. Each $t(B)$ is a tuple retaining only those attributes in the requested set B.

The query "*List all ages represented at Meeting*" can be expressed as a vague rough projection on the attribute AGE of the Meeting relation. This operation projects out all other attributes and eliminates redundant tuples. Note in the result that those tuples having higher μ values and higher ν^* values retained during the merging process (Table 5).

3.4.6. *Join Operator*

Join is a binary operator that takes related tuples from two relations and combine them into a single tuple of the resulting relation. It uses common attributes to combine the two relations into one, usually larger, relation.

The vague rough join, X^θ<JOIN CONDITION>Y, of two relations X and Y, is a relation $T(C_1, C_2, \ldots, Cm + n)$ where

$$T = \{t \mid \exists t_x \in X, t_y \in Y \text{ for } t_x = t(A), t_y = t(B)\},$$

Table 5

All ages represented at Meeting

Age	***TF***
Adult	[0.8,0.9]
Senior	[0.7,0.8]
{Adult, Teen}	[0.6,0.6]
{Adult, Senior}	[0.5,0.7]

and where

$$t_x(A \cap B) = t_y(A \cap B),\ \mu = 1,\ \nu^* = 1,$$
$$t_x(A \cap B) \subseteq t_y(A \cap B) \text{ or } t_y(A \cap B) \subseteq t_x(A \cap B),\ \mu = \text{Min}\,(\mu_x, \mu_y)$$
$$\text{and } \nu^* = \text{Min}\,(\nu_x^*, \nu_y^*)$$

<JOIN CONDITION> is a conjunction of one or more conditions of the form $A = B$.

The query "*List all individuals who resides in the 'Rampur' city and attended the meeting*", can be expressed as a vague rough join on the attributes of STUDENT and MEETING relations. The resulting relation contains the attributes from both the original relations by joining them on the common attribute ID and gives result in Table 6.

4. SQL-Like Queries for Vague Rough Relational Database

SQL, is one of the most popular languages for relational databases. Bosc *et al.* (1988) and Kerre *et al.* (1986) membership from the fuzzy relational databases and Beaubouef and Petry (1994b) explained rough querying of crisp data. Though, we found many levels of SQL such as SQL1, SQL2, SQL3 but most of these are semantic extensions. We now describe the extension of SQL to VRSQL, which is powerful enough to retrieve any set of items of any degree of vagueness. As in other database query languages, there are often several ways of expressing a given query. Based on our data definition language for the vague rough relational database on SQL, we present some SQL like queries to our database.

Example 4. "*List all ages represented at the Rally*"
SELECT AGE
FROM RALLY

There are no conditions, and hence, no WHERE clause for this query since it is a simple projection of the attribute AGE from RALLY. All attribute values of the vague rough relation RALLY except AGE and TF are deleted and then redundant tuples eliminated to result in the Table 7.

Example 5. "*List all individuals who attended either rally or meeting or both*"
(SELECT ID, SEX, AGE
FROM MEETING)

Table 6
All individuals who resides in Rampur city and attended the Meeting

ID	*Name*	*City*	*Sex*	*Age*	*Height*	*TF*
5010	Josef	*Rampur*	M	Adult	Tall	[0.7,0.8]
5018	Rajiv	*Rampur*	M	{Adult, Teen}	{Medium, Tall}	[0.5,0.6]

Table 7

All ages represented at Rally

Age	*TF*
Adult	[0.8,0.9]
Senior	[0.7,0.8]
{Adult, Teen}	[0.6,0.7]
{Adult, Senior}	[0.6,0.6]

UNION
(SELECT ID, SEX, AGE
FROM RALLY)

The query illustrates a simple union of two union compatible vague rough relations (Table 8).

Example 6. "*List all the students who attended rally and are adult but short in height*".

(SELECT ID, SEX, AGE, HEIGHT FROM RALLY)
WHERE
(AGE = 'ADULT' AND HEIGHT = 'SHORT')

Since the query is having indiscernible values, the TF will have the values got through calculation by multiplying the original values by card(x)/card(y).

Table 8

Individuals who attended Rally or Meeting or Both

ID	*Sex*	*Age*	*Height*	*TF*
5002	F	Adult	Medium	[0.8,0.9]
5010	M	Senior	Tall	[0.7,0.8]
5014	M	Adult	Short	[0.7,0.8]
5015	F	{ Adult, Teen}	{Medium, Tall}	[0.6,0.7]
5018	M	{ Adult, Teen}	{Medium, Tall}	[0.5,0.6]
5020	M	Adult	{Short, Medium}	[0.6,0.8]

Table 9

Students who attended Rally and are adult but short in height

ID	*Sex*	*Age*	*Height*	*TF*
5014	M	Adult	Short	[0.7,0.8]
5020	M	Adult	{Short, Medium}	[0.3,0.4]

5. Conclusions

This paper concerns the modeling of imprecision and vagueness type of uncertainty in databases through an extension of the relational model of data: the vague rough relational database. The new model is formally defined and theoretical properties of operators and relational algebra for querying are discussed. The new vague rough E-R diagram and SQL like query languages have been described.

The various models explained in the introduction and the references are able to process specific type of uncertainty. The usefulness of new model is illustrated by a simple example. The vague rough relational database is a sound model, which incorporates the combination of techniques for uncertainty processing into the underlying data model and its algebra. The design of the databases for this model is similar to that of ordinary databases except for the user-defined indiscernibility values. The data definition and manipulation languages (DDL and DML) for the vague rough relational database are closely related to standard SQL for conventional databases. The user simply has to remember that the underlying model is based on vague rough sets, which will be used in determining results of queries, and true and false membership values must be considered when populating or updating the database.

In conclusion, the vague rough relational database model is easy to understand and to use. In addition, it more accurately models the uncertainty of real-world enterprises than do conventional databases through the use of indiscernibility and vague membership and non-membership values. Some of the potential application areas of the new data model have also been mentioned over the existing data models.

Appendix

Consider the following three tables/relations namely People, Rally and Meeting having different attributes for getting the results of various operators and queries described under the proposed model.

Table A1: PEOPLE

ID	*Name*	*City*	*TF*
5001	J.Singh	Saharanpur	[1,1]
5002	Kalpna	{Bombay, Gorakhpur}	[0.5,0.5]
5010	Josef	Rampur	[1,1]
5011	Priya	Nainital	[1,1]
5014	Rajat	{Nainital, Gorakhpur}	[0.5,0.5]
5015	Ashraf	Rampur	[1,1]
5017	John	Rampur	[1,1]
5018	Rajiv	Rampur	[1,1]
5019	Sarika	Saharanpur	[1,1]
5020	Shubham	Gorakhpur	[1,1]
5021	Chandan	Madras	[1,1]

Table A2: RALLY

ID	*Sex*	*Age*	*Height*	*TF*
5002	F	Adult	Medium	[0.8,0.9]
5010	M	Senior	Tall	[0.7,0.8]
5014	M	Adult	Short	[0.7,0.8]
5015	M	{Adult, Teen}	{Medium, Tall}	[0.6,0.7]
5019	F	{Adult, Senior}	Tall	[0.6,0.6]
5020	M	Adult	{Short, Medium}	[0.6,0.8]

Table A3: MEETING

ID	*Sex*	*Age*	*Height*	*TF*
5002	F	Adult	Medium	[0.8,0.9]
5010	M	Senior	Tall	[0.7,0.8]
5018	M	{Adult, Teen}	{Medium, Tall}	[0.5,0.6]
5019	F	{Adult, Senior}	Tall	[0.5,0.7]
5020	M	Adult	{Short, Medium}	[0.5,0.6]

References

Bagai, R., R. Sunderraman (1995). A paraconsistent relational data model. *International Journal of Computer Mathematics*, **55**.

Beaubouef, T., F.E. Petry (1994a). A rough set model for relational databases. In *2nd Intl Workshop on Rough Sets and Knowledge Discovery*. Canada.

Beaubouef, T., F.E. Petry (1994b). Rough querying of crisp data in relational databases. In *Third Intl. Workshop on Rough Sets and Soft Computing*. San Jose, California.

Beaubouef, T., F.E. Petry (2000). Fuzzy rough set techniques for uncertainty processing in a relational database. *Intl. Journal of Intelligent Systems*, **15**, 389–424.

Bosc, P., M. Galibourg, G. Hamon (1988). Fuzzy querying with SQL: extensions and implementation aspects. *Fuzzy Sets and Systems*, **28**, 333–349.

Buckles, B., F.E. Petry (1982). A fuzzy representation for relational databases, *Fuzzy Sets and Systems*, **7**, 213–226.

Cavallo, R., M. Pottareli (1987). The theory of probabilistic databases. In *Proc. of 13th Very Large Database Conference*. pp. 71–81.

Chaudhry, N.A., J.R. Moyne, E.A. Rundensteiner (1994). A design methodology for databases with uncertain data. In *Intl. Working Conference on Scientific and Statistical Database Management*. Charlottesville, Virginia.

Chen, S.M. (2003). Analyzing fuzzy system reliability using vague set theory. In *International Journal of Applied Science and Engineering*. **1**, pp. 82–88.

Chen, S.M., W.T. Jong (1997). Fuzzy query translation for relational database systems. *IEEE Transactions on Systems, Man, and Cybernetics*, **27**(4).

Codd, E.F. (1970). A relational model of data for large shared data banks. *Comm. ACM*, **13**, 377–387.

Cornelis, C., M.D. Cock, E.E. Kerre (2003). Intuitionistic fuzzy rough sets: at the crossroads of imperfect knowledge. *Expert Systems*, **20**(5), 260–270.

Date, C.J. (1989). *An Introduction to Database Systems*. 6th edition; Addison-Wesley, Reading, MA.

Dubois, D., H. Prade (1994). Fuzzy sets – a convenient fiction for modeling vagueness and Possibility. *IEEE Transactions Fuzzy Systems*, **2**(1), 16–21.

Gau, W.L., D.J. Buehrer (1993). Vague sets. *IEEE Transactions on Systems, Man, and Cybernetics*, **23**, 610–614.

Kerre, E.E., R. Zenner, R. DeCaluwe (1986). The use of fuzzy set theory in information retrieval and databases: a survey. *Journal of American Society for Information Science*, **5**, 341–345.

Kerre, E.E. (1992). The suitability of fuzzy set theory for the representation and manipulation of imprecision and uncertainty in knowledge based systems, *Communication and Cognition*, **1**, 49–79.

Klir, G.J. (1987). Where do we stand on measures of uncertainty, ambiguity, fuzziness and the like. *Fuzzy Sets and Systems*, **24**(2), 141–160.

Medina, J.M., M.A. Vila, J.C. Cubero, O. Pons (1995). Towards the implementation of a generalized fuzzy relational database model. *Fuzzy Sets and Systems*, **75**(3), 273–289.

Pawlak, Z. (1991). *Rough Sets, Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers; Norwell, MA.

Petry, F. (1992). *Fuzzy Databases: Principles and Applications*. Kluwer Academic Publishers.

Prade, H., C. Testemale (1984). Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries, *Information Sciences*, **34**, 115–143.

Radzikowska, A.A., E.E. Kerre (2002). A comparative study of fuzzy rough sets, *Fuzzy Sets and Systems*, **126**, 137–156.

Singh, K., S.S. Thakur, M. Lal (2005). Handling uncertainty in relational databases: models revisited. *Varahmihir Journal of Computer and Information Sciences*, **1**(1), 25–44.

Takahasi, Y. (1991). A fuzzy query language for relational databases, Systems. *Man and Cybernetics*, *IEEE Transactions*, **216**, 1576–1579.

Ullman, J. (1982). *Principles of Database Systems*. 2nd edition, Rockville, MD.

Umano, M. (1982). FREEDOM: A fuzzy database system. *Fuzzy Information and Decision Processes*. North-Holland. pp 339–347.

Wang, Z., G.J. Klir (1992). *Fuzzy Measure Theory*. Plenum Press, New York.

Wong, E. (1982). A statistical approach to incomplete information in database systems. *ACM Trans. on Database Systems*, **7**, 470–488.

Zadeh, L.A. (1965). Fuzzy sets, *Information and Control*, **8**, 338–353.

Zvieli, A.P., Chen (1986). In *Proceedings of International Conf. on Data Engg*. Los Angles, California. pp. 320–327.

K. Singh received the BS degree in mathematics in 1986 from University of Allahabad, India and the MCA and PhD degrees in computer science in 1993 and 2007 respectively from Rani Durgawati Vishwavidyalaya, Jabalpur, India. Presently, working as senior scientist (computer applications) at Central Institute of Agricultural Engineering, Bhopal, India. The areas of research interests are fuzzy databases, heterogeneous distributed databases, web databases and data mining.

S.S. Thakur is presently working as a professor in the Department of Applied Mathematics, Government Engineering College, Jabalpur. Dr. Thakur received MSc and PhD degrees in mathematics from Dr. H.S. Gour Vishwavidyalaya Sagar in 1977 and 1982 respectively. He has published more than 125 refered papers and supervised 11 PhD students, most in the area of topology, fuzzy topology and fuzzy databases. He has been the organizing secretary of an International Conference on Soft Computing and Intelligent Systems held in December 2007 and serving as a referee of many international journals including *Fuzzy Sets and Systems* and *Journal of Fuzzy Mathematics*. The main foci of his current work includes fuzzy topology and applications of vague and rough sets in databases.

M. Lal received the MS and PhD degrees in mathematics. Presently, working as professor at Rani Durgawati Vishwavidyalaya, Jabalpur, India in the Department of Mathematics and Computer Science. The areas of research interests are topology, functional analysis and differential & integral equations.

Neapibrėžtų šiurkščių aibių metodika neapibrėžtumo apdorojimui reliacinės duomenų bazės modelyje

Karan SINGH, Samajh Singh THAKUR, Mangi LAL

Viena pamatinių problemų, lemiančių aktyvias duomenų bazių studijas, yra didelių duomenų kiekių skirstymo būdai. Šiame straipsnyje nagrinėjamas netikslumo ir neatskiriamumo apdorojimas reliacinėse duomenų bazėse, pagal siūlomą neapibrėžtą šiurkščią metodiką sukuriant neapibrėžtą šiurkštų duomenų bazės modelį. Autoriai iš šiurkščių aibių teorijos perima neatskiriamumo ir tikimybės sąvokas, kurios apjungiamos su priklausomybės/nepriklausomybės reikšmėmis iš neapibrėžtų aibių teorijos. Tokiu būdu neapibrėžta informacija gali būti atvaizduojama išsaugant informacijos neapibrėžtumo laipsnį kiekviename duomenų bazės korteže. Teorinės siūlomame modelyje naudojamų operatorių savybės straipsnyje palygintos su analogiškomis reliacinio duomenų modelio savybėmis. Taip pat straipsnyje aprašomos neapibrėžtam šiurkščiam reliaciniam duomenų bazės modeliui skirtos duomenų bazės aprašymo ir SQL tipo užklausų kalbos.