Journal of Intelligent & Fuzzy Systems 20 (2009) 105–117
DOI 10.3233/IFS-2009-0422
IOS Press

Uncertainty modeling for database design using intuitionistic and rough set theory

Theresa Beaubouef[a] and Frederick E. Petry[b,*]
[a]*Southeastern Louisiana University, Department of Computer Science & IT, Hammond, LA 70402, USA*
[b]*Naval Research Laboratory, Stennis Space Center, Mississippi, 39529, USA*

Abstract. This paper introduces the intuitionistic rough set and intuitionistic rough relational and rough object oriented database models. Rough set, fuzzy set, and intuitionistic set uncertainty management are discussed and compared, and the model based on intuitionistic and rough sets developed here is applied to databases. The intuitionistic rough set database models draw benefits from both the rough set and intuitionistic techniques, providing greater management of uncertainty for databases applications in a less than certain world.

1. Introduction

The only certain thing in life is uncertainty. In real world applications and models, uncertainty plays a major role, and it is up to scientists to determine suitable ways of managing uncertainty. It is significant that computers are very precise and "certain" in fundamental ways of representing data. In fact, the basic storage element of a computer is the bit: the bit is either on or off, magnetized or not magnetized, high or low. In spite of the technical advances in both computer software and hardware, the two-valued bit is still the fundamental unit of information. It is the combination of bits, however, and the "certain" representation of uncertainty that allows for the modeling of more advanced and sophisticated applications. Many theories of uncertainty management have been developed for the mathematical modeling of uncertainty. These include such theories as probability, possibility, rough sets, vague sets, fuzzy sets, and intuitionistic sets to name a few. Each of the theories has advantages and is better at modeling some type of uncertainty over another.

This paper discusses some of the methods for modeling uncertainty and imprecision, namely rough sets, fuzzy sets, and intuitionistic fuzzy sets. The first few sections provide an overview of the theories. We then introduce and formally define intuitionistic rough sets, and show how intuitionistic rough sets provide greater uncertainty management than rough set, fuzzy set, or intuitionistic set techniques alone. Moreover, we show how rough sets, fuzzy sets, intuitionistic sets, and one type of fuzzy rough set are all special cases of this intuitionistic rough set.

Later sections provide the foundation for the integration of intuitionistic rough sets into modeling of uncertainty in databases. These build upon some of our previous research [5,8] with integrating fuzzy and rough set techniques for uncertainty management in databases. We first introduce the intuitionistic rough relational database model. This database model incorporates intuitionistic and rough set uncertainty management into the underlying relational database model. Next we introduce intuitionistic rough set uncertainty management in object-oriented databases. There are significant differences between the intuitionistic rough relational and object-oriented models. However, both can take advantage of the rough and intuitionistic modeling of uncertainty.

2. Rough sets

Rough set theory, introduced by Pawlak [22] and discussed in greater detail in [18,19,24], is a technique

[*] Corresponding author. Tel.: +1 228 688 4948; Fax: +1 228 688 4853; E-mail: fpetry@nrlssc.navy.mil.

for dealing with uncertainty and for identifying cause-effect relationships in databases as a form of database learning [25]. It has also been used for improved information retrieval [26] and for uncertainty management in relational databases [7,8].

Rough sets involve the following:

U is the *universe*, which cannot be empty,

R is the *indiscernibility relation*, or equivalence relation,

$A = (U, R)$, an ordered pair, is called an *approximation space*,

$[x]_R$ denotes the equivalence class of R containing x, for any element x of U,

elementary sets in A – the equivalence classes of R,

definable set in A – any finite union of elementary sets in A.

Therefore, for any given approximation space defined on some universe U and having an equivalence relation R imposed upon it, U is partitioned into equivalence classes called elementary sets which may be used to 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:

lower approximation of X in A is the set $\underline{R}X = \{x \in U | [x]_R \subseteq X\}$

upper approximation of X in A is the set $\overline{R}X = \{x \in U | [x]_R \cap X \neq \emptyset\}$.

Another way to describe the set approximations is as follows. Given the upper and lower approximations $\overline{\mathrm{R}}$X and $\underline{\mathrm{R}}$X, of X a subset of U, the R-positive region of X is POS$_R$(X) = $\underline{\mathrm{R}}$X, the Rnegative region of X is NEG$_R$(X) = U − $\overline{\mathrm{R}}$ X, and the boundary or R-borderline region of X is BN$_R$(X) = $\overline{\mathrm{R}}$X – $\underline{\mathrm{R}}$X. X is called R-definable if and only if $\underline{\mathrm{R}}$X = $\overline{\mathrm{R}}$ X. Otherwise, $\underline{\mathrm{R}}$X $\neq$ $\overline{\mathrm{R}}$ X and X is rough with respect to R. A *rough set in A* is the group of subsets of U with the same upper and lower approximations. An illustration of rough set X is shown in Fig. 1.

The major rough set concepts of interest are the use of an indiscernibility relation to partition domains into equivalence classes and the concept of lower and upper approximation regions to allow the distinction between certain and possible, or partial, inclusion in a rough set.

The indiscernibility relation allows us to group items based on some definition of 'equivalence' as it relates to the application domain. We may use this partitioning to increase or decrease the granularity of a domain, to group items together that are considered indiscernible for a given purpose, or to "bin" ordered domains into range groups.

In order to allow *possible* results, in addition to the obvious, certain results encountered in querying an ordinary spatial database system, we may employ the use of the boundary region information in addition to that of the lower approximation region. The results in the lower approximation region are certain. These correspond to exact matches. The boundary region of the upper approximation contains those results that are possible, but not certain.

3. Fuzzy rough sets

Several researchers have studied various ways of combining fuzzy and rough set theories [13,14,20] in an effort to reap the benefits of each. Others have investigated the interrelations between the two complementary theories [11,23,27]. Fuzzy sets and rough sets are not equivalent, and because they represent different types of uncertainty, a combination of the two is often useful.

In [27] it was shown that rough sets can be expressed by a fuzzy membership function $\mu \rightarrow \{0, 0.5, 1\}$. In this model, all elements (equivalence classes) of the lower approximation, or positive region, have a membership value of one. Those elements of the boundary region are assigned a membership value of 0.5. Elements not belonging to the rough set have a membership value of zero. Rough set definitions of union and intersection were modified so that the fuzzy model would satisfy all the properties of rough sets [23], thus allowing a rough set to be expressed as a fuzzy set.

In [4] we integrated the fuzziness into the rough relational database model. However, we were interested in using the fuzzy values to quantify the level of uncertainty of the boundary region elements and therefore did not require membership values of elements of the boundary region to equal 0.5, but allowed them to take on values anywhere within the range of real numbers between zero and one, noninclusive. Additionally, database union and intersection operators for fuzzy rough sets were defined. They are comparable to those for ordinary fuzzy sets, where MIN and MAX are used to obtain membership values of redundant elements. We review those definitions here.

Let U be a *universe*, X a rough set in U.

Definition. A *fuzzy rough set* Y in U is a membership function μ_Y(x) which associates a grade of membership from the interval [0,1] with every element of U where

$$\mu_Y(\underline{R}X) = 1$$

$$\mu_Y(U - \mathrm{RX}) = 0$$

$$0 < \mu_Y(\mathrm{RX} - \underline{R}X) < 1.$$

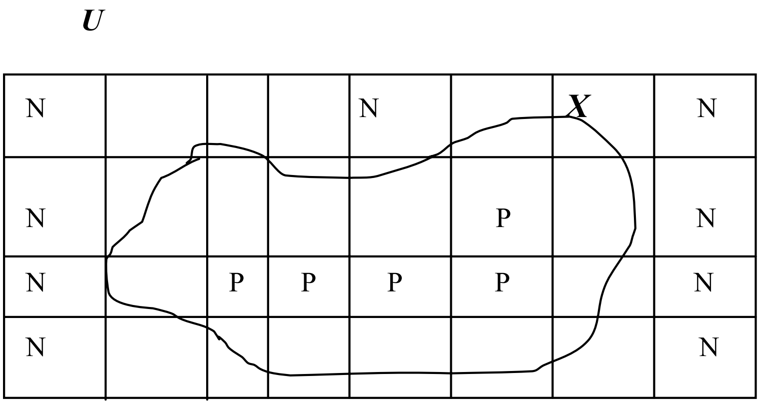

Fig. 1. Example of a Rough Set **X**.

All elements (equivalence classes) of the positive region have a membership value of one and elements of the boundary region have a membership value between zero and one.

Definition. The *union* of two fuzzy rough sets A and B is a fuzzy rough set C where

$$\mathrm{C} = \{x|x \in \mathrm{A\ OR} x \in B\}$$

$$\mu_C(x) = \mathrm{MAX}[\mu_A(x), \mu_B(x)].$$

Definition. The *intersection* of two fuzzy rough sets A and B is a fuzzy rough set C where

$$\mathrm{C} = \{x|x \in \mathrm{A\ AND} x \in B\}$$

$$\mu_C(x) = MIN[\mu_A(x), \mu_B(x)].$$

4. Intuitionistic sets

An intuitionistic set [1,2] (intuitionistic fuzzy set) is a generalization of the traditional fuzzy set introduced by Zadeh [28].

Let set X be fixed. An intuitionistic set A is defined by the following:

A $= \{< x, \mu_A(x), v_A(x) >| \ x \in X\}$, and where $\mu_A : X \to [0, 1]$, and$v_A : X \to [0, 1]$.

The degree of membership of element x $\in$ X to the set A is denoted by $\mu_A(x)$, and the degree of nonmembership of element x $\in$ X to the set A is denoted by $v_A(x)$. A is a subset of X.

Additionally, for all $x \in$ X,

$$0 \leqslant \mu_A(x) + v_A(x) \leqslant 1.$$

A hesitation margin,

$$\pi_A(x) = 1 - (\mu_A(x) + v_A(x)),$$

expresses a degree of uncertainty about whether x belongs to X or not, or uncertainty about the membership degree. This hesitancy may cater toward membership or nonmembership.

Example.

A person may be *happy* or *unhappy* in traditional logic. In the two-valued logic, there are only two choices. There is no continuum between happy and unhappy; nor is there any uncertainty involved.

In rough sets many things may be considered in the realm of happiness and unhappiness, and some of them will be grouped together in equivalence classes. Some of these classes are entirely included in the set *happy*: [overjoyed, ecstatic], [glad] or [happy], for example. Some are not in the rough set *happy* at all [upset, angry] or [furious], [sad, unhappy], for example. Lastly, there are some that involve uncertainty about the belonging to the rough set *happy*. These may include such equivalence classes as [satisfied, content], or [nonchalant, indifferent]. These would belong to the boundary, or uncertain region of the rough set.

In fuzzy sets a person could be happy to a certain degree. The degree of membership of an element to the fuzzy set of happy is represented by a membership value between zero and one. For example, one could be happy to a degree of 0.8. This implies unhappiness to a degree of 0.2. However, a person could be happy to a degree of 0.8, but not unhappy at all, or at least not to that extent. This cannot be represented in fuzzy sets.

In intuitionistic fuzzy sets, however, there are measures for both the degree of membership and the degree of nonmembership. A person could be happy to a degree of 0.8, but only unhappy to a degree of 0.05, resulting in a hesitancy of 0.15. This two sided fuzziness in the intuitionistic set provides greater management of uncertainty for many real world cases.

5. Intuitionistic rough sets

In this section we introduce the intuitionistic rough set, which incorporates the beneficial properties of both rough set and intuitionistic set techniques. Intuitionistic rough sets are generalizations of fuzzy rough sets that give more information about the uncertain, or boundary region. They follow the definitions for partitioning of the universe into equivalence classes as in traditional rough sets, but instead of having a simple boundary region, there are basically two boundaries formed from the membership and nonmembership functions.

Let U be a *universe*, Y a rough set in U, defined on some approximation space which partitions U into equivalence classes.

An *intuitionistic rough set* Y in U is $< Y, \mu_Y(x)$, $\upsilon_Y(x) \;>$, where $\mu_Y(x)$ is a membership function which associates a grade of membership from the interval [0,1] with every element (equivalence class) of U, and $\upsilon_Y(x)$ associates a degree of non membership from the interval [0,1] with every element (equivalence class) of U, where

$0 \leqslant \mu_Y(x) + \upsilon_Y(x) \leqslant 1$, where x denotes the equivalence class containing x.

A hesitation margin,

$$\pi_Y(x) = 1 - (\mu_Y(x) + \upsilon_Y(x)),$$

Consider the following special cases $< \mu, \upsilon >$ for some element of Y:

<1, 0> denotes total membership. This corresponds to elements found in $\underline{R}$Y.

<0, 1> denotes elements that do not belong to Y. Same as U – RY.

<0.5, 0.5> corresponds to traditional rough set boundary region.

<p, 1-p> corresponds to fuzzy rough set in that there is a single boundary. In this case we assume that any degree of membership has a corresponding complementary degree of non membership.

<p, 0> corresponds to fuzzy rough set. In this case there is no complement to what p shows membership in.

<0, q> This case can not be modeled by fuzzy rough sets. It denotes things that are not a member of $\underline{R}$Y or $\overline{R}$Y. It falls somewhere in the region U – $\overline{R}$Y.

<x, y> Intuitionistic set general case, uncertain double boundary, one for membership and one for nonmembership.

Let Y' denote the complement of Y. Then the intuitionistic set having $< \mu_Y(x), \mu_{Y'}(x) >$ is same as fuzzy rough set.

The last two cases above, <0, q> and <x, y>, cannot be represented by fuzzy sets, rough sets, or fuzzy rough sets. These are the situations which show that intuitionistic rough sets provide greater uncertainty management than the others alone. Note, however, that with the intuitionistic set we do not lose the information about uncertainty provided by other set theories, since from the first few cases we see that they are special cases of the intuitionistic rough set.

We may also perform operations on the intuitionistic rough sets such as union and intersection. We define these operations next. The definition of these operators is necessary for applications such as the intuitionistic rough relational database model.

Definition. The *union* of two intuitionistic rough sets A and B is an intuitionistic rough set C where

$$C = \{x \mid x \in A \text{ OR } x \in B\}$$

$$\mu_C(x) = \text{MAX}[\mu_A(x), \mu_B(x)], \upsilon_C(x)$$

$$= \text{MIN}[\upsilon_A(x), \upsilon_B(x)].$$

Definition. The *intersection* of two intuitionistic rough sets A and B is an intuitionistic rough set C where

$$C = \{x \mid x \in A \text{ AND } x \in B\}$$

$$\mu_C(x) = \text{MIN}[\mu_A(x), \mu_B(x)],\ \upsilon_C(x)$$

$$= \text{MAX}[\upsilon_A(x),\ \upsilon_B(x)].$$

In this section we defined intuitionistic rough sets and compared them with rough sets and fuzzy sets. Although there are several various way of combining rough and fuzzy sets, we focused on those fuzzy rough sets as defined in [4,5] and used for fuzzy rough databases, since our intuitionistic rough relational database model follows from this. The intuitionistic rough relational database model will have an advantage over the rough and fuzzy rough database models in that the non membership uncertainty of intuitionistic set theory will also play a role, providing even greater uncertainty management than the original models. In the remainder of the paper we present relational and object oriented database models that provide uncertainty management through the use of intuitionistic rough set techniques.

6. Intuitionistic rough relational database model

The intuitionistic rough relational database, as in the ordinary relational database, represents data as a collection of *relations* containing *tuples*. Because a relation is considered a set having the tuples as its members, the tuples are unordered. In addition, there can be no duplicate tuples in a relation. A tuple $\mathbf{t}_i$ takes the form (d_{i1}, d_{i2}, $\ldots$, d_{im}, $d_{i\mu}$ d_{iv}), where d_{ij} is a *domain value* of a particular *domain set* D_j and $d_{i\mu} \in D_\mu$, where D_μ is the interval [0,1], the domain for intuitionistic membership values, and D_v is the interval [0,1], the domain for intuitionistic nonmembership values. In the ordinary relational database, $d_{ij} \in D_j$. In the intuitionistic rough relational database, except for the intuitionistic membership and nonmembership values, however, $d_{ij} \subseteq D_j$, and although d_{ij} is not restricted to be a singleton, $d_{ij} \neq \emptyset$. Let $P(D_i)$ denote any non-null member of the powerset of D_i.

Definition. An intuitionistic *rough relation R* is a subset of the set cross product $P(D_1)\times$ $P(D_2)$ $\times \cdots \times$ $P(D_m)\times$ $D_{\mu.}\times$ D_v.

For a specific relation, R, membership is determined semantically. Given that D_1 is the set of names of nuclear/chemical plants, D_2 is the set of locations, and assuming that RIVERB is the only nuclear power plant that is located in VENTRESS,

(RIVERB, VENTRESS, 1, 0)
(RIVERB, OSCAR, 0.7, 0.3)
(RIVERB, ADDIS, 1, 0)
(CHEMO, VENTRESS, 0.3, 0.2)

are all elements of $P(D_1) \times P(D_2) \times D_\mu\times D_v$. However, only the element (RIVERB, VENTRESS, 1, 0) of those listed above is a member of the relation R (PLANT, LOCATION, μ, v), which associates each plant with the town or community in which it is located. An *intuitionistic rough tuple* **t** is any member of$R$. If $\mathbf{t}_i$ is some arbitrary tuple, then $\mathbf{t}_i = (d_{i1}, d_{i2}, \ldots, d_{im}$, $d_{i\mu}$, $d_{iv})$ where $d_{ij} \subseteq D_j$ and $d_{i\mu} \in D_\mu$, $d_{iv} \in D_v$.

Definition. An *interpretation* $\alpha = (a_1, a_2, \ldots, a_m, a_\mu$, $a_v)$ of an intuitionistic rough tuple $t_i = (d_{i1}, d_{i2}, \ldots$, d_{im}, $d_{i\mu}$, $d_{iv})$ is any value assignment such that $a_j \in d_{ij}$ for all j.

The interpretation space is the cross product $D_1\times$ $D_2 \times \cdots \times D_m\times$ $D_\mu\times$ D_v but is limited for a given relation R to the set of those tuples that 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 $\mathbf{t}_i$. Moreover, the interpretation of $\mathbf{t}_i$ is equivalent to the tuple $\mathbf{t}_i$. In the intuitionistic 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. Tuples $\mathbf{t}_i = (d_{i1}, d_{i2}, \ldots, d_{in}, d_{i\mu}, d_{iv})$ and $\mathbf{t}_k = (d_{k1}, d_{k2}, \ldots, d_{kn}, d_{k\mu}, d_{iv})$ are *redundant* if $[d_{ij}] = [d_{kj}]$ for all $j = 1, \ldots$, n.

If a relation contains only those tuples of a lower approximation, i.e., those tuples having a μ value equal to one and v equal to zero, the interpretation α of a tuple is unique. This follows immediately from the definition of redundancy. In intuitionistic rough relations, there are no redundant tuples. The merging process used in relational database operations removes duplicate tuples 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 v. As in the union of intuitionistic rough sets where the maximum membership value of an element is retained, it is the convention of the intuitionistic rough relational database to retain the tuple having the higher μ value when removing redundant tuples 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. If the μ values are equal but the v values unequal, we retain that tuple having the lower v value.

Recall that the rough relational database is in non-first normal form; there are some attribute values which are sets. 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. Two sub-tuples $X = (d_{x1}, d_{x2}, \ldots, d_{xm})$ and $Y = (d_{y1}, d_{y2}, \ldots, d_{ym})$ are *roughlyredundant*, $\approx_R$, if for some $[p] \subseteq [d_{xj}]$ and $[q] \subseteq [d_{yj}]$, $[p] = [q]$ for all $j = 1, \ldots$, m.

In order for any database to be useful, a mechanism for operating on the basic elements and retrieving specified data must be provided. The concepts of redundancy and merging play a key role in the operations defined in Section 3.3.

6.1. *Intuitionistic rough relational operators*

In [8], we defined several operators for the rough relational algebra, and in [4] demonstrated the expressive power of the fuzzy rough versions of these operators in the fuzzy rough relational database model. We now review similar operators for the intuitionistic rough relational database. Recall that for all of these operators the indiscernibility relation is used for equivalence of attribute values rather than equality of values.

6.1.1. *Difference*

The intuitionistic rough relational difference operator is very much like the ordinary difference operator in relational databases and in sets in general. It is a binary operator that returns those elements of the first argument that are not contained in the second argument.

In the intuitionistic rough relational database, the difference operator is applied to two intuitionistic 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 intuitionistic rough relations.

Definition. The *intuitionistic rough difference*, **X–Y**, between X and Y is a intuitionistic rough relation T where

$$\begin{aligned} \mathrm{T} = \{&\mathrm{t}(\mathrm{d}_1, \ldots, \mathrm{d}_n, \mu_{i,,} v_i) \in \mathrm{X} \mid \mathrm{t}(\mathrm{d}_1, \ldots, \mathrm{d}_n, \mu_{i,} v_i) \\ &\notin \mathrm{Y}\} \cup \\ &\{\mathrm{t}(\mathrm{d}_1, \ldots, \mathrm{d}_n, \mu_{i,,} v_i) \in \mathrm{X} \mid \mathrm{t}(\mathrm{d}_1, \ldots, \mathrm{d}_n, \mu_{j,} v_j) \\ &\in \mathrm{Y} \text{ and } \mu_i > \mu_j\} \cup \\ &\{\mathrm{t}(\mathrm{d}_1, \ldots, \mathrm{d}_n, \mu_{i,} v_i) \in \mathrm{X} \mid \mathrm{t}(\mathrm{d}_1, \ldots, \mathrm{d}_n, \mu_{j,} v_j) \\ &\in \mathrm{Y} \text{ and } \mu_i = \mu_j \text{ and } v_i < v_j\} \end{aligned}$$

The resulting intuitionistic 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 uncertain regions of both X and Y, but which have a higher μ value in X than in Y or equal μ values and lower v values.

6.1.2. *Union*

Database relations are sets, and as such the union operator can be applied to any two union-compatible relations to result in a third relation which has as its tuples all the tuples contained in either or both of the two original relations. This operator can be extended to apply to intuitionistic rough relations. Let X and Y be two union compatible intuitionistic rough relations.

Definition. The *intuitionistic rough union* of X and Y, **X** ∪ **Y** is a intuitionistic rough relation T where

T = {t | t ∈ X OR t ∈ Y} and μ_T(t) = MAX[μ_X(t), μ_Y(t)], and if μ_X(t) = μ_Y(t), v_T(t) = MIN[v_X(t), v_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 value, the merging process will retain only that tuple with the higher μ value. Those tuples redundant in all values except v will retain the tuple having the lower v value.

6.1.3. *Intersection*

The intuitionistic rough intersection, another binary operator on intuitionistic rough relations, can be defined similarly.

Definition. The *intuitionistic rough intersection* of X and Y, **X** ∩ **Y** is a intuitionistic rough relation T where

T = {t | t ∈ X AND t ∈ Y} and μ_T(t) = MIN[μ_X(t), μ_Y(t)], and if μ_X(t) = μ_Y(t), v_T(t) = MAX[v_X(t), v_Y(t)].

In intersection, the MIN operator is used in the merging of equivalent tuples having different μ values and the result contains all tuples that are members of both of the original intuitionistic rough relations. For like μ values in redundant tuples, the v values are compared, and the tuple having the higher will be retained.

6.1.4. *Select*

The select operator for the intuitionistic rough relational database model, σ, is a unary operator which takes a intuitionistic rough relation X as its argument and returns a intuitionistic rough relation containing a subset of the tuples of X, selected on the basis of values for a specified attribute. The operation $\sigma_{A=a}$(X), for example, returns those tuples in X where attribute A is equivalent to the class [**a**]. In general, select returns a subset of the tuples that match some selection criteria.

Let R be a relation schema, X an intuitionistic rough relation on that schema, A an attribute in R, **a** = {a$_i$} and **b** = {b$_j$}, where a$_i$,b$_j$ ∈ dom(A), and $\cup_x$ is interpreted as "the union over all x".

Definition. The *intuitionistic rough selection*, $\sigma_{A=a}$ (X), of tuples from X is an intuitionistic rough relation Y having the same schema as X and where

$$\mathrm{Y} = \{\mathrm{t} \in \mathrm{X} \mid \cup_i[\mathrm{a}_i] \subseteq \cup_j[\mathrm{b}_j]\},$$

where a$_i$ $\in$ **a**, b$_j$ $\in$ t(A) and where membership values for tuples are calculated by multiplying the original membership value μ by

card(**a**)/card(**b**)

where *card*(x) returns the cardinality, or number of elements, in x. The nonmembership value v remains the same as in the intuitionistic rough relation X, since the result of performing this operation does not give us any additional information about nonmembership.

6.1.5. Project

Project is a unary intuitionistic rough relational operator. It returns a relation that contains a subset of the columns of the original relation. Let X be a intuitionistic rough relation with schema A, and let B be a subset of A. The intuitionistic rough projection of X onto B is a intuitionistic rough relation Y obtained by omitting the columns of X which correspond to attributes in A–B, and removing redundant tuples. Recall the definition of redundancy accounts for indiscernibility, which is central to the rough sets theory, and that higher μ values have priority over lower ones. For like μ values, the v values are examined and the tuple having lower v value is retained.

Definition. The *intuitionistic rough projection* of X onto B, π_B (**X**), is an intuitionistic rough relation Y with schema Y(B) where

Y(B) = $\{$t(B) $|$ t $\in$ X$\}$.

6.1.6. Join

Join is a binary operator that takes related tuples from two relations and combines them into single tuples of the resulting relation. It uses common attributes to combine the two relations into one, usually larger, relation. Let X(A$_1$, A$_2$, $\ldots$, A$_m$) and Y(B$_1$, B$_2$, $\ldots$, B$_n$) be intuitionistic rough relations with **m** and **n** attributes, respectively, and AB $=$ C, the schema of the resulting intuitionistic rough relation T.

Definition. The *intuitionistic rough join*, **X** $_{-<\text{JOIN CONDITION}>}$ **Y**, of two relations X and Y, is a relation T(C$_1$, C$_2$, $\ldots$, C$_{m+n}$) where

T $=$ $\{$t $|\exists$ t$_X$ $\in$ X, t$_Y$ $\in$ Y for t$_X$ $=$ t(A), t$_Y$ $=$ t(B)$\}$, and where

(1) t$_X$(A $\cap$ B) $=$ t$_Y$(A$\cap$ B), $\mu = 1$

(2) t$_X$(A $\cap$ B) $\subseteq$ t$_Y$(A $\cap$ B) or t$_Y$(A $\cap$ B) $\subseteq$ t$_X$(A $\cap$ B), μ $=$ MIN(μ_X, μ_Y), if $\mu_{X=}\mu_Y$, v $=$ MAX(v_X, v_Y)

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

Only those tuples which resulted from the "joining" of tuples that were both in lower approximations in the original relations belong to the lower approximation of the resulting intuitionistic rough relation. All other "joined" tuples belong to the uncertain region, having membership values less than one. The intuitionistic membership value of the resultant tuple is simply calculated as in [6] by taking the minimum of the membership values of the original tuples. Taking the minimum value also follows the logic of [21], where in joining tuples with different levels of information uncertainty, the resultant tuple can have no greater certainty than that of its least certain component. For equal membership values, the maximum nonmembership value is retained.

This section concerned the modeling of imprecision, vagueness, and uncertainty in databases through an extension of the relational model of data: the intuitionistic rough relational database. The intuitionistic rough relational database was formally defined, along with an intuitionistic rough relational algebra for querying. Comparisons of theoretical properties of operators in this model with those in the standard relational model were discussed.

7. Intuitionistic Rough Object-Oriented Database (IROODB) model

The object-oriented programming paradigm has become quite popular in recent years, both as a modeling tool and for code development for databases and other applications. Often objects can more realistically model an enterprise, enabling developers to easily transition from a conceptual design to the actual code. The concepts of *classes* and *inheritance* allow for code reuse through specialization and generalization. A *class hierarchy* is designed such that classes at the top of the hierarchy are the most general and those nearer the bottom more specialized. A class *inherits* data and behavior from classes at higher levels in the class hierarchy. This promotes reuse of existing functionality, which can save valuable programming time. If code is already available for a task and that code has been tested, it is often better to use that code, perhaps with some slight modification, than to develop and test code from scratch. The concept of *polymorphism* allows the same name to be used for methods differing in functionality for different object types.

Essentially, an *object* is an instance of a *class* in a class hierarchy. Each class defines a particular type of object including its public and private variables and operations associated with the functionality of the object, which are called *methods*. An object method is invoked by the passing of a *message* to the object in which the method is defined. The data variables and methods are *encapsulated* in the object that defines them, which means that they are packaged in, and can only be accessed through, the object. In OODBs the values of the variables determine the state and these, along with methods determine the behavior of the object. Encapsulation enables a component of the system to be extended or modified with minimal impact on other parts of the system.

There are many advantages associated with the object-oriented database approach compared to a relational database approach. A major advantage is that very complex data structures and relationships in the data, as is often the case in spatial data can be modeled. According to Fayad and Tsai [16], object-oriented technology provides several other benefits. These include reusability, extensibility, robustness, reliability, and scalability. Object modeling helps in requirements understanding and collaboration of group members and the use of object-oriented techniques leads to high quality systems that are easy to modify and to maintain.

We next develop the intuitionistic rough object-oriented database model. We follow the formal framework and type definitions for generalized object-oriented databases proposed by [12] and extended for rough sets in [5], which conforms to the standards set forth by the Object Database Management Group [9]. We extend this framework, however, to allow for intuitionistic rough set indiscernibility and approximation regions for the representation of uncertainty as we have previously done for relational databases [4,8]. The intuitionistic rough object database scheme is formally defined by the following type system and constraints.

The type system, TS = [T, P, f^{type}_{impl}], where T can be a literal type T$_{literal}$, which can be a base type, a collection literal type, or a structured literal type. It also contains T$_{object}$, which specifies object types, T$_{reference}$, the set of specifications for reference types, and a *void* type. In the type system, each domain $dom_{ts} \in$ D$_{ts}$, the set of domains. This domain set, along with a set of operators O$_{ts}$ and a set of axioms A$_{ts}$, capture the semantics of the type specification. The type system is then defined based on these type specifications, the set of all programs P, and the implementation function mapping each type specification for a domain onto a subset of the powerset of P that contains all the implementations for the type system:

$$f^{type}_{impl}\colon \mathrm{T} \to \rho\,(\mathrm{P}) \text{ giving } ts \to \{\mathrm{p}_1, \mathrm{p}_2, \ldots \mathrm{p}_n\}.$$

We are particularly interested in object types. Following [12], we may specify a class t of object types as

$$\text{Class } id(id_1 : s_1; \ldots; id_n : s_n) \quad \text{or}$$

$$\text{Class } id\colon \overline{\overline{id}}_1, \ldots, \overline{\overline{id}}_n(id_1 : s_1; \ldots; id_n : s_n)$$

where id, an identifier, names an object type, $\{\overline{\overline{id}}_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{m}\}$ is a finite set of identifiers denoting parent types of t, and $\{id_i : s_i \mid 1 \leqslant \mathrm{i} \leqslant \mathrm{n}\}$ is the finite set of characteristics specified for object type t within its syntax. This set includes all the attributes, relationships and method signatures for the object type. The identifier for a characteristic is id_i and the specification is s_i for each of the $id_i : s_i$.

Consider a GIS which stores spatial data concerning water and land forms, structures, and other geographic information. If we have simple types defined for string, set, geo, integer, etc., we can specify an object type

```
Class Bridge (
    Location: geo;
    Name: string;
    Height: integer;
    Material: Set(string)
    WaterType:Set(string)
    WaterFlow:Set(string));
```

An example instance of object type *Bridge* might look like the following:

[oid1, Ø, *Bridge*, Struct(0289445, “Castor Creek Bridge”, 7, Set(concrete), Set(creek), Set (E,NE,N))]

following the definition of instance of an object type [12], the quadruple o = [*oid, N, t, v*] consisting of a unique object identifier, a possibly empty set of object names, the name of the object type, and for all attributes, the values (v$_i$ ∈ dom$_{si}$) for that attribute, which represent the state of the object. The object type t is an instance of the type system *ts* and is formally defined in terms of the type system and its implementation function t = [*ts*, f^{type}_{impl} (*ts*)].

Indiscernibility is the inability to distinguish between two or more values. For example, the average person describing the color of a car involved in a hit-and-run accident may say that it is “burgundy,” when it actually is maroon. As it turns out, “burgundy” is probably good enough for helping the police identify the vehicle. However, a painter who specializes in automobile paint and body work will find it easy to discern between burgundy, maroon, and several other related colors. Indiscernibility can also arise from lack of precision in measurement, limitations of computational representation, or the granularity or resolution of the sampling or observations.

In the rough set object-oriented database, indiscernibility is managed through classes. Every attribute domain is implemented as a class hierarchy, with the lowest elements of the hierarchy representing the equivalence classes based on the finest possible partitioning for the domain as it pertains to the application. Consider, for example, a GIS, where objects have an attribute called *landClass*. There are many different classifications for land area features, such as those covered by forests, pastures, urban area, or some type of water, for example.

Ignoring the non-water parts of the *landClass* domain, and focusing on the water-related parts, we see that the domain set

dom$_{landClass}$ = {creek, brook, stream, branch, river, lake, pond, waterhole, slough}

can be partitioned in several different ways. One partitioning, which represents the finest partitioning (more, but smaller, equivalence classes) is given by

R^1 = {[creek, brook, stream], [branch], [river], [lake], [slough], [pond, waterhole]}.

This is evidenced by the lowest level of the hierarchy.

An object type (domain class) for landClass may be defined as

```
Class landClass (
   numEquivClass: integer;
   name: string;
   indiscernibility: Set(Ref(equivClass)))
```

At this lowest level, each landClass object has only one reference in its attribute for indiscernibility, the object identifier for the particular equivalence class. These reference individual equivalence class objects defined by

```
Class equivClass(
   element: Set(string);
   N: integer;
   Name: string).
```

In this case, we have six separate equivalence classes shown below:

[oid56, Ø, *equivClass*, Struct(Set(“creek,” “brook,” “stream”), 3, “creek”)]
[oid57, Ø, *equivClass*, Struct(Set(“branch”), 1, “branch”)]

Note that the name of the class can be set equal to any of the values within the class.

If we want to change the partitioning, such that our application only distinguishes between flowing and standing water, for example, our equivalence classes become

R^2 = {[creek, brook, river, stream, branch], [lake, pond, waterhole, slough]}.

Using this approximation space, R^2, we would then have the landClass objects

[oid101, Ø, *landClass*, Struct(3, “Flowing water,” Set(oid56, oid57, oid58))]
[oid102, Ø, *landClass*, Struct(3, “Standing water,” Set(oid59, oid60, oid61))].

Each domain class i in the database (such as *landClass*, that has an indiscernibility relation associated with it), dom$_i$ $\in$ D$_i$, has methods for maintaining the current level of granulation, changing the partitioning, adding new domain values to the hierarchy, and for determining equivalence based on the current indiscernibility relation imposed on the domain class.

Every domain class, then, must be able to not only store the legal values for that domain, but to maintain the grouping of these values into equivalence classes. This can be achieved through the type implementation function and class methods, and can be specified through the use of generalized constraints as in [5,12] for a generalized OODB.

An ordinary (non-indiscernibility) object class in our sample database, having one of its attributes *landClass*, may be defined as follows:

Class *RuralProperty* (
   Location: geo;
   Name: string;
   Owner: string;
   landType: Set(landClass));

Particular instances of this class, for example, might include:

[oid24, Ø, *RuralProperty*, Struct(01987345, “Elm Plantation”, “Bob Owner”, Set(“waterhole,” “pasture”))],
[oid27, Ø, *RuralProperty*, Struct(01987355, Ø, “Betty Owner”, Set(“forest,” “lake”))],

See [5] for details of how rough set concepts are integrated in this OO model, and how changing the granularity of the partitioning affects query results. In that paper the OO model is extended for fuzzy and rough set uncertainty. A natural extension which we introduce here is for intuitionistic sets.

If we extend the rough OODB further to allow for intuitionistic types, the type specifications T can be generalized to a set $\check{T}$ as in [12], so that the definitions of the domains are generalized to intuitionistic sets:

For every *ts* $\in \top$, having domain *ts* being dom$_{ts}$, the type system *ts* $\in \top$ is generalized to

$$\overline{ts} \in \check{\top}$$

where domain of $\overline{ts}$ is denoted by dom $\overline{ts}$ and is defined as the set $\overline{\rho}$ (dom$_{ts}$) of intuitionistic sets on dom$_{ts}$, and O$_{ts}$ is generalized to O$\overline{ts}$, which contains the generalized version of the operators. This is consistent with the UFO database model [17] as well.

The generalized type system then is a triple

$$\text{GTS} = [\check{\top}, \text{P}, \overline{f}^{\,type}_{impl}]$$

where $\check{\top}$ is the generalized the type system, P is the set of all programs, and $\overline{f}^{\,type}_{impl}$ is a mapping which maps each $\overline{ts} \in \check{\top}$ onto that subset of P that contains the implementation for $\overline{ts}$.

An instance of this GTS is a generalized type $\overline{t} =$ [$\overline{ts}$, $\overline{f}^{\,type}_{impl}(\overline{ts})$], $\overline{ts} \in \check{\top}$.

For example,

Class *Bridge* (
  Location: geo;
  Name: string;
  Height: IntuitionisticSet(integer);
  Material: Set(string)
  WaterType:Set(string)
  WaterFlow:Set(string));

A generalized object belonging to this class is defined by

$$\overline{o} = [\text{oid, N}, \overline{t}, \overline{f}^{\,type}_{impl}(\overline{ts}), v]$$

where v draws values from the generalized domain that allows an object to contain intuitionistic membership and nonmembership values as part of the state of the object.

Both intuitionistic and rough set uncertainty management can be used in this generalized OODB model. For example, some intuitionistic rough instances of the previously defined object type *Bridge* might include:

[oid1, Ø, *Bridge*, Struct(0289445, "Castor Creek Bridge", {(5, (0.7, 0.2)), (7, (0.9, 0.1))}, Set(concrete), Set(creek), Set (E,NE,N))]

where the attribute Height is shown as an intuitionistic set, and Material, WaterType, and WaterFlow are shown as ordinary sets. We assume that each of these base objects is certain, i.e., each object fully exists and has a membership value of one. We further assume that we have defined the partitioning R^1 for the domain WaterType as discussed previously:

R^1 = {[creek, brook, stream], [branch], [river], [lake], [pond, waterhole], [slough]}.

It is easy to see the need for various types of uncertainty in spatial database from even this simplified example. There is indiscernibility in the labels for various types of water. Different users might use different names for the same water types. Or, data may have been gathered from multiple sources and currently being consolidated into a single database application. Rough sets allow us to specify this level of indiscernibility and to adjust it, when necessary to fit the application.

There is intuitionistic uncertainty in the height of the bridges. This uncertainty may be due to one of several causes. The bridge might be too high to measure accurately by a non-skilled worker, or it might be that there is uncertainty about where the top of the bridge should be marked. If there is a light on the top, do we measure to the top of the light? Uncertainty might also arise from the location at which the bridge is measured. Is it measured in height above the water? If so, the water level probably varies over time. Is it measured in height above the ground? If so, this height is likely to be different on either end of the bridge.

Direction of water flow in this example illustrates yet another type of uncertainty, which can be modeled through the approximation regions of rough sets. Often waterways twist and turn in various directions. If a river is generally flowing toward the east, but beneath the bridge it is flowing toward the northeast, we may consider including both of these directions in the database. This decision would obviously depend on the application, and whether it is necessary to have information on the direction of water flow for the entire water body, or only at the point below the bridge.

Uncertainty in spatial databases and geographic information systems becomes an even greater issue when considering topological relationships among various objects or regions, which themselves may be uncertain. However, intuitionistic and rough set uncertainty management, incorporated into an OODB, will allow for better modeling of the GIS and for additional information to be obtained from the underlying data.

In this section we extended a formal framework of object-oriented databases to allow for modeling of various types of imprecision, vagueness, and uncertainty that typically occur in spatial data. The model is based on a formal type system and specified constraints, thus preserving integrity of the database, while at the same time allowing an OODB to be generalized in such a way as to include both intuitionistic and rough set uncertainty, both well-developed methods of uncertainty management.

No information is lost in using the intuitionistic rough object-oriented data model over traditional OODBs. One can always examine the intuitionistic membership and nonmembership values of objects resulting from intuitionistic rough expressions to discover which of those are certain and which are possible. The certain results include exactly those that would be obtained in a non- intuitionistic, non-rough database. In this model, however, we are provided also with those results that are possible, along with intuitionistic membership values to aid in determining the degrees of uncertainty of each of the results. Incorporation of intuitionistic and rough set uncertainty into the OODB model is essential for representing imprecision and uncertainty in spatial data entities and in their interrelationships.

8. Spatial data applications with rough and intuitionistic sets

Many of the problems associated with data are prevalent in all types of databases systems. Spatial databases and GIS contain descriptive as well as positional data. The various forms of uncertainty may occur in both types of data, so many of the issues discussed below apply to ordinary databases as well. See [7, 8] for in-depth discussion of implementation of rough set uncertainty in (non-spatial) databases. These same techniques, including integration of data from multiple sources, time-variant data, uncertain data, imprecision in measurement, inconsistent wording of descriptive data, and the "binning" or grouping of data into fixed categories, may also be employed for spatial contexts.

Often spatial data is associated with a particular grid. The positions are set up in a regular matrix-like structure and data is affiliated with point locations on the grid. This is the case for raster data and for other types of non-vector type data such as topography or sea surface temperature data. There is a tradeoff between the resolution or the scale of the grid and the amount of system resources necessary to store and process the data. Higher resolutions provide more information, but at a cost of memory space and execution time.

If we approach the data from a rough set point of view, we can see that there is indiscernibility inherent in the process of gridding or rasterizing data. A data item at a particular grid point in essence may represent data near the point as well. This is due to the fact that often point data must be mapped to the grid using techniques such as nearest-neighbor, averaging, or statistics. We may set up our rough set indiscernibility relation so that the entire spatial area is partitioned into equivalence classes where each point on the grid belongs to an equivalence class. If we change the resolution of the grid, we are in fact, changing the granularity of the partitioning, resulting in fewer, but larger classes.

The approximation regions of rough sets come into play when information concerning sizes, lengths, and other areal properties of spatial data features are calculated or displayed. Consider an areal feature such as an airport. One can reasonably conclude that any grid point identified as "airport" that is surrounded on all sides by grid points also identified as "airport" is in fact a point represented by the feature "airport". However, consider points identified as airport that are adjacent to points identified as meadow. Is it not possible that these points represent meadow area as well as airport area but were identified as airport in the classification process? Likewise, points identified as "meadow" but adjacent to "airport" points may represent areas that contain part of the airport. This uncertainty maps naturally to the use of the approximation regions of the rough set theory, where the lower approximation region represents the certain data and the boundary region of the upper approximation represents the uncertain data.

If we force a finer granulation of the partitioning (increase the grid resolution) a smaller boundary region results. As the partitioning becomes finer and finer, eventually a point is reached where the boundary region is non-existent. In this case, the upper and lower approximation regions are the same and there is no uncertainty in the spatial data.

In spatial data, therefore, representation of indiscernibility offered by rough set theory is very useful. When paired with the intuitionistic uncertainty management, which provides information about the uncertainty of the boundary region, databases can more accurately model real world spatial data applications. A common issue for spatial data representation is the problem of indeterminate boundaries [3]. One approach is to model a spatial area as having a broad boundary [10,15] by classifying it as consisting of a core in which the classification is certain and a boundary region in which the classification is "less" certain.

Consider the application in which a planning committee is evaluating regions for the siteing of new industrial installations. Part of such a process is the development of an environmental impact statement which requires a variety of assessments of the proposed site locations to determine the effects of the new plant on the neighboring environment. One type of assessment

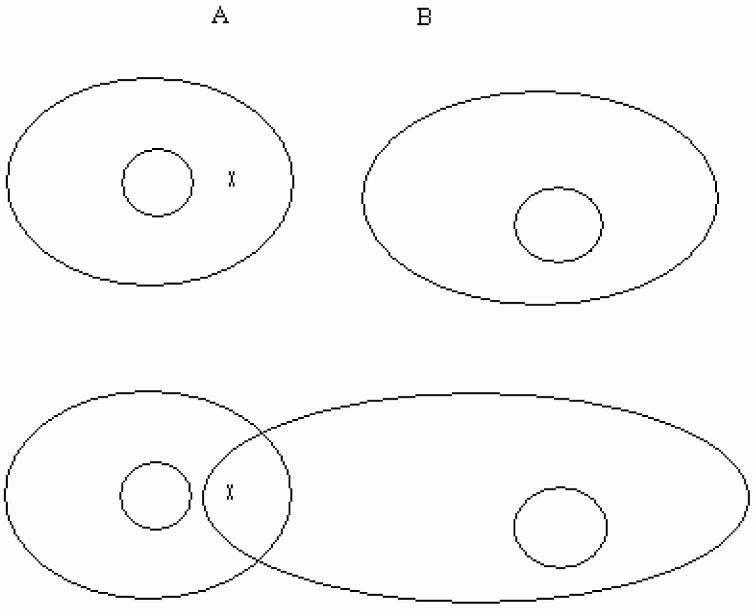

Fig. 2. Environmental Assessment Regions with Broad Boundaries.

might be classification of vegetation by an expert utilizing imagery and other data sources to determine possible vulnerability to pollution. Figure 2 illustrates such a classification.

In the upper part of Fig. 2 are two broad boundary regions labeled A and B. In the area A on the left, the $\boldsymbol{X}$ marks a potential plant location. The area A is considered satisfactory for sites because the vegetation here has been classified as a type that would not be affected by the plant's runoff and emissions. The area B however has growth that is susceptible to contamination and should not be used for a site. Note that A has points both in the certain core and in the outer boundary. The latter points then have a degree of membership $\mu_A(x) < 1.0$ that might be based, for example, on some metric such as distance from core. So we may have for the site location $\mu_A(x) =$ 0.p, but its nonmembership, $v_A(x) =$ 0.0, since there is no reason to consider the site $\boldsymbol{X}$ as unsatisfactory. However suppose now that other evaluations are made or that further information becomes available about area B and these areas now are judged to overlap in their outer boundaries as seen in lower half of Fig. 2. Since the site location $\boldsymbol{X}$ is then in the overlap, we have $v_A(x) =$ 0.r, r $>$ 0.0, since there is some doubt as to whether the site is indeed in a satisfactory area and some hesitation must be felt about this site.

This example has used intuitionistic modeling but as discussed previously in this section spatial gridding could be involved with the data and imagery for the areas A and B. This would then require a combination of the rough and intuitionistic set approaches to appropriately model these areas for the environmental impact assessment of the plant site.

9. Conclusion

In this paper we compared basic rough, fuzzy, and intuitionistic sets, and introduced the intuitionistic rough set. We then discussed how the intuitionistic rough set generalizes each of traditional rough, fuzzy, fuzzy-rough, and intuitionistic sets. These results can be used to enhance models involving uncertainty.

The intuitionistic rough relational database model and its operators were formally defined. This model allows for both rough and intuitionistic modeling of uncertainty. Because real world applications involve uncertainty, this model can more accurately represent data and relationships than traditional relational databases.

The intuitionistic rough object-oriented database model can also better model real world applications. It is especially useful in those applications involving spatial or multi-media data. We formally defined the intuitionistic rough object-oriented database and illustrated its usefulness with geographic information examples.

Databases are everywhere, and every day new uses for databases are discovered. As computers become faster and more powerful, more is expected from computerized databases, and future systems will be significantly more sophisticated. An essential feature of any database that is used for real world applications in an "uncertain" world is its ability to manage uncertainty in data. We have introduced in this paper our models for intuitionistic rough relational and object-oriented databases and shown the significance of both rough sets and intuitionistic sets for uncertainty management.

Acknowledgment

We would like to thank the Naval Research Laboratory's Base Program, Program Element No. 0602435N for sponsoring this research.

References

[1] K. Atanassov, Intuitionistic Fuzzy Sets, *Fuzzy Sets and Systems* **20** (1986), 87–96.

[2] K. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*, Physica-Verlag, A Springer-Verlag Company, Heidelberg, NY, 1999.

[3] P. Burrough and A. Frank, eds, *Geographic Objects with Indeterminate Boundaries*, Taylor & Francis, London, 1996.

[4] T. Beaubouef and F. Petry Fuzzy Rough Set Techniques for Uncertainty Processing in a Relational Database, *International Journal of Intelligent Systems* **15**(5) (2000), 389–424.

[5] T. Beaubouef and F. Petry, Fuzzy Set Uncertainty in a Rough Object Oriented Database, *Proc of North American Fuzzy Information Processing Society* (*NAFIPS-FLINT'02*), New Orleans, Louisiana, 2002, 365–370.

[6] B. Buckles and F. Petry, Uncertainty models in information and database systems, *Journal of Information Science* **11** (1985), 77–87.

[7] T. Beaubouef and F. Petry, Rough Querying of Crisp Data in Relational Databases, *Third International Workshop on Rough Sets and Soft Computing* (*RSSC'94*), San Jose, California, November 1994.

[8] T. Beaubouef, F. Petry and B. Buckles, Extension of the Relational Database and its Algebra with Rough Set Techniques, *Computational Intelligence* **11**(2) (1995), 233–245.

[9] R. Cattell, D. Barry, D. Bartels, M. Berler, J. Eastman, S. Gamerman, D. Jordan, A. Springer, H. Strickland and D. Wade, *The Object Database Standard: ODMG2.0*, Morgan Kaufmann, San Francisco, CA, 1997.

[10] E. Clementini and P. DiFelice, A Spatial Model for Complex Objects with a Broad Boundary Supporting Queries on Uncertain Data, *Data & Knowledge Engineering* **37** (2001), 285–305.

[11] S. Chanas and D. Kuchta, Further remarks on the relation between rough and fuzzy sets, *Fuzzy Sets and Systems* **47** (1992), 391–394.

[12] G. De Tré and R. De Caluwe, A Generalized Object-Oriented Database Model with Generalized Constraints, *In Proc of NAFIPS'99*, New York, 381-386, 1999.

[13] D. Dubois and H. Prade, Twofold Fuzzy Sets and Rough Sets–Some Issues in Knowledge Representation, *Fuzzy Sets and Systems* **23** (1987), 3–18.

[14] D. Dubois and H. Prade, Putting Rough Sets and Fuzzy Sets Together, *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, R. Slowinski, ed., Kluwer Academic Publishers, Boston, 1992.

[15] S. Du, Q. Qin, Q. Wang and H. Ma, Reasoning about Topological Relations Between Regions with Broad Boundaries, *International Journal of Approximate Reasoning* **47**(2) (2008), 233–246.

[16] M. Fayad and W. Tsai, Object-Oriented Experiences, *Communications of the ACM* **38** (1995), 51–53.

[17] N. Gyseghem and R. De Caluwe, Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models, *Advances in Fuzzy Systems – Applications and Theory* **13** (1997), 123–177.

[18] J. Grzymala-Busse, *Managing Uncertainty in Expert Systems*, Kluwer Academic Publishers, Boston, 1991.

[19] J. Komorowski, Z. Pawlak, L. Polkowski et al., Rough Sets: A Tutorial, in: *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, S. Pal and A. Skowron, eds, Springer-Verlag, Singapore, 1999, pp. 3–98.

[20] S. Nanda and S. Majumdar, Fuzzy rough sets, *Fuzzy Sets and Systems* **45** (1992), 157–160.

[21] A. Ola and G. Ozsoyoglu, Incomplete Relational Database Models Based on Intervals, *IEEE Transactions on Knowledge and Data Engineering* **5**(2) (April 1993), 293–308.

[22] Z. Pawlak, Rough Sets, *International Journal of Man-Machine Studies* **21** (1984),127–134.

[23] Z. Pawlak, Rough Sets and Fuzzy Sets, *Fuzzy Sets and Systems* **17** (1985), 99–102.

[24] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Norwell, MA, 1991.

[25] R. Slowinski, A Generalization of the Indiscernibility Relation for Rough Sets Analysis of Quantitative Information, *First International Workshop on Rough Sets: State of the Art and Perspectives*, Poland, September 1992.

[26] P. Srinivasan, The importance of rough approximations for information retrieval, *International Journal of Man-Machine Studies* **34** (1991), 657–671.

[27] M. Wygralak, Rough Sets and Fuzzy Sets–Some Remarks on Interrelations, *Fuzzy Sets and Systems* **29** (1989), 241–243.

[28] L. Zadeh, Fuzzy Sets, *Information and Control* **8** (1965), 338–353.