A Generalized Decision Logic Language for Granular Computing

Y.Y. Yao
Department of Computer Science, University of Regina, Regina
Saskatchewan, Canada S4S 0A2, E-mail: yyao@cs.uregina.ca

Churn-Jung Liau
Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: liaucj@iis.sinica.edu.tw

Abstract - A generalized decision logic language *GDL* is proposed for granular computing (GrC) in the Tarski's style through the notions of a model and satisfiability. The model is an information table consisting of a finite set of objects described by a finite set of attributes. A concept or a granule is characterized by a pair consisting of the intension of the concept, a formula of the language *GDL*, and the extension of the concept, a subset of the universe. We discuss the application of *GDL* in formal concepts and decision rules. The former deals with description and interpretation of granules, and the latter deals with the relationships between granules.

I. INTRODUCTION

The concept of information granulation was first introduced by Zadeh in the context of fuzzy sets in 1979 [23]. The basic ideas of information granulation have appeared in fields, such as interval analysis, quantization, rough set theory, the theory of belief functions, divide and conquer, cluster analysis, machine learning, databases, and many others [24]. There is a fast growing and renewed interest in the study of information granulation and computations under the umbrella of *Granular Computing* (GrC) [5], [7], [10], [11], [12], [13], [15], [16], [19], [20], [25].

Basic ingredients of granular computing are subsets, classes, and clusters of a universe [16], [24]. There are many fundamental issues in granular computing, such as granulation of the universe, description of granules, relationships between granules, and computing with granules. Issues of granular computing may be studied from two related aspects, the construction of granules and computing with granules. The former deals with the formation, representation, and interpretation of granules, while the latter deals with the utilization of granules in problem solving [18]. Granular computing can be studied from both the semantic and algorithmic perspectives [18]. Semantic studies focus on the "why" type questions, and the algorithmic studies focus on the "how" type of questions. More specifically, semantics studies deal with issues such as why two objects are put into the same granule, and why different granules are related. Algorithmic studies deal with the actual processes of information granulation and computing with granules.

A general framework of granular computing was given in a recent paper by Zadeh [24]. Granules are constructed and defined based on the concept of generalized constraints. Examples of constraints are equality, possibilistic, probabilistic, fuzzy, and veristic constraints. Granules are labeled by fuzzy sets or natural language words. Relationships between granules are represented in terms of fuzzy graphs or fuzzy if-then rules. The associated computation method is known as computing with words [22]. Many more concrete models of granular computing have been studied by many authors [6], [10], [11], [12], [13], [15], [19].

The main objective of this paper is to propose a logic based framework for granular computing, which is complementary to existing studies. We adopt the decision logic language *DL* that was discussed in [9] and generalized in [3], [18]. The proposed generalized decision logic language *GDL* combines the vigorous formulation and solid foundation of decision logic and the flexibility and intuitive interpretation of generalized constraints. The language is developed in the Tarski's style through the notions of a model and satisfiability. The model is an information table (either crisp or fuzzy) consisting of a finite set of objects described by a finite set of attributes. Granular computing is formulated in terms of formal concepts, which can be used to develop a mathematical model for data mining [18]. A concept or a granule is defined as a pair consisting of the intension of the concept, a formula of *GDL*, and the extension of the concept, a subset of the universe. The intension is a formal description of the granule, and the extension is the granule itself. An object satisfies the formula of a concept if the object has the properties as specified by the formula, and the object belongs to the extension of the concepts. Rules are used to describe relationships between concepts. By interpreting a concept or a granule as a pair consisting of a logic formula and a subset of the universe, we can study granular computing in either logic or set-theoretic setting.

II. A DECISION LOGIC LANGUAGE FOR GRANULAR COMPUTING

Intuitively, a granule is "a clump of points (objects) drawn together by indistinguishability, similarity, proximity or functionality" [24]. In order to make those notions more precise, we need to design a scheme for representing objects under consideration. The notion of information tables is used for this purpose, in which each object can be described by using a finite set of attributes or features. Indistinguishability and similarity can be defined through their values on the set of attributes. More specifically, two objects are indistinguishable or similar if we can not differentiate them through their values [9].

A. **Information Tables with Added Semantics**

An information table can be expressed as [9], [19]:

$$S = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}),$$

where

U is a finite nonempty set of objects,
At is a finite nonempty set of attributes,
V_a is a nonempty set of values for $a \in At$,
$I_a : U \rightarrow V_a$ is an information function.

Each information function I_a is a total function that maps an object of U to exactly one value in V_a. An information table represents all available information and knowledge. That is, objects are only perceived, observed, or measured by using a finite number of properties. By generalize the information function I_a, we can obtain generalized information tables.

An information table defined above does not have any knowledge or information about relationship between values of an attribute. Consequently, one can only use the trivial equality relation $=$ on attribute values to describe relationships between objects [16]. Theory of rough sets and the decision logic language for an information table is developed based on the equality relation [9]. Granules of rough sets form a partition of the universe. Each granule is described by using equality conditions. For example, with respect to an attribute $a \in At$ and a value $v \in V_a$, a granule can be defined by $\{x \in U \mid I_a(x) = v\}$. The formula of the decision logic language corresponding to this granule is given by (a, v). A granule therefore consists of objects having the same values on some attributes. Other types of relationships between attribute values can be introduced to provide added semantics to an information table [16].

Relationships between attribute values lead to granulations of the attribute values. Examples of such granulations are the discretization and clustering of attribute values. For an attribute $a \in At$, let L_a be a set of labels used to name granules of V_a. In general, each member of L_a can be a subset of V_a. In the degenerated case, L_a consists of only singleton subsets of V_a. Formally, an information table with added semantics can be described by a pair:

$$S^+ = (S, \{L_a \mid a \in At\}), \tag{1}$$

where S is a standard information table. In an information table with added semantics, one can state conditions using labels in L_a. Depending on the physical meanings of labels, one may consider various types of relations. For an attribute $a \in At$ and a label $l \in L_a$, we may introduce a formula of the form (a, r, l) in a generalized decision logic language, where r denotes a particular relation between an attribute value and a label. The formula defines the granule of the universe, $\{x \in U \mid I_a(x)\ r\ l\}$. Thus, the granulation of attribute values induces a granulation of the universe.

Formulas of the form (a, r, l) are closely related to the notion of the generalized constraints proposed by Zadeh [24]. Let X be a variable taking values from U. A generalized constraint on the values of X is expressed in a general form as $X\ isr\ R$, where R is the constraining relation, isr is a variable copula and r is variable which defines the way in which R constrains X. The form of the generalized constraints allows us to describe many different types, such as equality constraints, possibilistic constraints, veristic constraints, probabilistic constraints, probability value constraints, random set constraints, and fuzzy graph constraints [24]. With a generalized constraint, the set of objects satisfying the constraint forms a granule. Decision logic languages use a similar interpretation. Formulas of the original decision logic language can be used to describe the equality constraints, and formula of the generalized decision logic language can be used to describe other types of constraints.

The notion of generalized constraints is intuitively appealing and useful. In order to use generalized constraints, one needs to precisely define and interpret various notions involved. In other words, one need to introduce a framework in which the semantics of a particular type of generalized constraints can be clearly explained.

In contrast, the decision logic language may be less general and its expressive power is limited. However, it uses an information table as a model to interpret various concepts. The advantage enables us to establish a solid basis for the study of granular computing.

In summary, the generalized decision logic language for an information table with added semantics extends the standard decision logic language and avoids some problem of the generalized constraints. In the following subsection, we present a formal description of the generalized decision logic language.

B. A Generalized Decision Logic Language

A detailed discussion of a decision logic language DL for an information table was given in the book by Pawlak [9]. Similar languages have been studied by many authors. We propose a generalized decision logic language called GDL by introducing additional vocabulary to DL.

The basic alphabet of GDL consists of the following three types of symbols:

- a finite set of attribute symbols At,
- a finite set of relation (constraint) symbols R_a for each attribute $a \in At$, and
- a set of label symbols L_a for each attribute $a \in At$.

For simplicity, we directly use the attribute names as attribute symbols, as attribute symbols will be assigned attribute names [3]. Each relation symbol represents a particular relation with the equality relation as a special case. Each label in L_a can be interpreted using V_a. In fact, it is a granule of V_a. Furthermore, we assume that there exists a type compatibility relation T_a between R_a and L_a for each attribute a. In general, $T_a(r, l)$ holds if it indeed makes sense to apply the relation symbol r to the label symbol l. The syntax of GDL is then defined as follows.

Definition 1: *Formulas of GDL are defined by the following two rules.*

- *An atomic formula of GDL is a descriptor (a, r, l), where $a \in At$, $r \in R_a$, $l \in L_a$, and $T_a(r, l)$ holds.*
- *The well-formed formulas (wff) of GDL is the smallest set containing the atomic formulas and closed under $\neg$, $\cap$, $\cup$, $\rightarrow$ and $\equiv$.*

The semantics interpretation of formulas of GDL is provided by an information table S^+. Attribute symbols are assigned attribute names, and labels symbols are assigned granule names of the attribute values. An atomic formula describes a condition on one attribute of objects. It can be verified if an object satisfies such a condition. This leads to the satisfiability relation between elements of the universe and wffs of GDL.

Definition 2: *Given a GDL and an information table S^+, the satisfiability of a formula ϕ by an object x, written $x \models_{S^+} \phi$ or in short $x \models \phi$ if S^+ is understood, is defined as follows:*

$$
\begin{array}{ll}
(1) & x \models (a, r, l) \text{ iff } I_a(x)\ r\ l, \\
(2) & x \models \neg\phi \text{ iff not } x \models \phi, \\
(3) & x \models \phi \wedge \psi \text{ iff } x \models \phi \text{ and } x \models \psi, \\
(4) & x \models \phi \vee \psi \text{ iff } x \models \phi \text{ or } x \models \psi, \\
(5) & x \models \phi \rightarrow \psi \text{ iff } x \models \neg\phi \vee \psi, \\
(6) & x \models \phi \equiv \psi \text{ iff } x \models \phi \rightarrow \psi \text{ and } x \models \psi \rightarrow \phi.
\end{array}
$$

The previous interpretation of GDL is essentially based on the classical two-valued logic. An object either satisfies a formula or does not satisfy the formula. The dichotomous notion of satisfiability can be generalized by considering degrees of satisfiability. With the notion of satisfiability, one may obtain a set-theoretic interpretation of formulas of GDL.

Definition 3: *If ϕ is a formula, the set $m_{S^+}(\phi)$ defined by:*

$$m_{S^+}(\phi) = \{x \in U \mid x \models \phi\}, \quad (2)$$

is called the meaning of the formula ϕ in S. If S^+ is understood, we simply write $m(\phi)$.

Definition 4: *A formula ϕ is said to be valid in an information table S^+, written $\models_{S^+} \phi$ or $\models \phi$ for short when S^+ is clear from the context, if and only if $m(\phi) = U$. That is, ϕ is satisfied by all objects in the universe.*

By definition, the following properties hold [9]:

$$
\begin{array}{ll}
(c1). & \models \phi \text{ iff } m(\phi) = U, \\
(c2). & \models \neg\phi \text{ iff } m(\phi) = \emptyset, \\
(c3). & \models \phi \rightarrow \psi \text{ iff } m(\phi) \subseteq m(\psi), \\
(c4). & \models \phi \equiv \psi \text{ iff } m(\phi) = m(\psi).
\end{array}
$$

Thus, we can study the relationships between concepts described by formulas of the GDL based on the relationships between their corresponding sets of objects.

C. Special Cases of the Decision Logic Language

The semantics of GDL depends on the satisfiability of the atomic formulas. However, we did not give any concrete examples of the relation symbols and label symbols. Roughly speaking, relation symbols correspond to the *isr* variable copula of Zadeh's generalized constraints, and label symbols correspond to the constraining relations. In this subsection, two concrete examples are discussed.

Decision logic language DL**:** The decision logic language DL studied by Pawlak is a very special case of GDL. For DL, we have $R_a = \{=\}$ and $L_a = V_a$. In other words, DL considers only the trivial equality relation and directly uses values from V_a as the set of label symbols. Thus, an atomic formula is of the form $(a, =, v)$, where $a \in At$ and $v \in V_a$. The satisfiability is defined by:

$$x \models (a, =, v) \text{ iff } I_a(x) = v. \quad (3)$$

Concept hierarchy: A hierarchical clustering of attribute values produces a concept hierarchy, which was widely used in data mining [2]. In a hierarchy, one typically associates a name with a cluster such that elements of the cluster are instances of the named category or concept. Such names will be used as labels of the GDL, and each label is interpreted as representing a subset of the attribute values. Let C_a be the set of all concepts in a concept hierarchy with respect to an attribute a. In this case, we have $R_a = \{=, \in\}$ and $L_a = V_a \cup C_a$ and the type compatibility relation T_a is defined such that $T_a(=, l)$ iff $l \in V_a$ and $T_a(\in, l)$ iff $l \in C_a$. Thus the atomic formulas are of the two forms, $(a, =, v)$ and $(a, \in, c)$, where $v \in V_a$, $c \in C_a$ and $c \subseteq V_a$. The satisfiability of $(a, =, v)$ is the same as in DL, and the satisfiability of $(a, \in, c)$ is defined by:

$$x \models (a, \in, c) \text{ iff } I_a(x) \in c. \quad (4)$$

In the special case where there is an order relation on V_a, one may confider intervals.

One may choose other types of relation symbols. For example, a relation s of "similar to" can be used to produce an atomic formula (a, s, v) where $v \in V_a$. The satisfiability of the formula can be similarly defined. The language GDL is very flexible due to the introduction of relation and label symbols.

III. GRANULAR COMPUTING USING GDL

To illustrate the usefulness of GDL for granular computing, we discuss two related notions, namely, formal concepts and decision rules, by summarizing the results presented in [18]. The former deals with description and interpretation of granules, and the latter deals with the relationship between granules.

A. Formal Concepts

In the study of formal concepts, every concept is understood as a unit of thoughts that consists of two parts, the intension and extension of the concept [1], [14]. The intension (comprehension) of a concept consists of all properties or attributes that are valid for all those objects to which the concept applies. The extension of a concept is the set of objects or entities which are instances of the concept. All objects in the extension have the same properties that characterize the concept. In other words, the intension of a concept is an abstract description of common features or properties shared by elements in the extension, and the extension consists of concrete examples of the concept. A concept is thus described jointly by its intension and extension. This formulation enables us to study formal concepts in a logic setting in terms of intensions and also in a set-theoretic setting in terms of extensions.

With the introduction of GDL, we have a formal description of concepts or granules [18]. A concept definable in an information table is a pair $(\phi, m(\phi))$, where ϕ is a wff. More specifically, ϕ is a description of $m(\phi)$ in S^+, the intension of concept $(\phi, m(\phi))$, and $m(\phi)$ is the set of objects satisfying ϕ, the extension of concept $(\phi, m(\phi))$. A concept $(\phi, m(\phi))$ is said to be a sub-concept of another concept $(\psi, m(\psi))$, or $(\psi, m(\psi))$ a super-concept of $(\phi, m(\phi))$, if $\models \phi \rightarrow \psi$ or $m(\phi) \subseteq m(\psi)$. A concept $(\phi, m(\phi))$ is said to be a smallest non-empty concept if there does not exist another nonempty and proper sub-concept of $(\phi, m(\phi))$. Two concepts $(\phi, m(\phi))$ and $(\psi, m(\psi))$ are disjoint if $m(\phi) \cap m(\psi) = \emptyset$. If $m(\phi) \cap m(\psi) \neq \emptyset$, we say that the two concepts have a non-empty overlap and hence are related.

The above formulation of concepts is different from the study of Wille on concept lattice [14]. Instead of using a subset of attributes to represent the intension of a concept, we use a formula from GDL. In our case, we can also form a concept lattice based on logical implication $\rightarrow$ or set inclusion $\subseteq$. More specifically, for two concepts $(\phi, m(\phi))$ and $(\psi, m(\psi))$, the meet and join are defined by:

$$\begin{aligned} (\phi, m(\phi)) \sqcap (\psi, m(\psi)) &= (\phi \wedge \psi, m(\phi) \cap m(\psi)), \\ (\phi, m(\phi)) \sqcup (\psi, m(\psi)) &= (\phi \vee \psi, m(\phi) \cup m(\psi)). \end{aligned} \quad (5)$$

In our formulation, one can easily define the extension based on the intension of a concept. However, the reverse is no longer true as in the case of formal concept lattice of Wille [14].

B. Decision Rules

Relationships between granules as represented by formal concepts can be expressed as rules. A decision rule of GDL can be expressed in the form, $\phi \Rightarrow \psi$, where ϕ and ψ are formulas in GDL, representing intensions of two concepts. By expressing rules with intensions of concepts, we may easily explain them in natural language, provided that we can explain formulas of GDL.

In many studies of machine learning and data mining, a rule is usually paraphrased by an if-then statement, "if an object satisfies ϕ, then the object satisfies ψ." The interpretation suggests a kind of cause and effect relation between ϕ and ψ. However, it is not clear if such a cause and effect relation does exist. In the context of fuzzy logic, Zadeh pointed out that although keywords such as IF and THEN are used in describing fuzzy if-then rules, one should not interpret the rules as expressing logical implications [24]. These keywords are used to simply link concepts together. Following Zadeh, we treat $\Rightarrow$ as a new connective that links two concepts together.

In the language GDL, $\Rightarrow$ can be interpreted in terms of the meaning sets of ϕ and ψ. A few probabilistic interpretations are summarized below [21]:

Generality: The generality measures the applicability of the rule. It tell the extend, namely, the portion of objects or the relative number of objects, to which the rule can be applied. The generality is defined by:

$$G(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{|U|} = \frac{|m(\phi) \cap m(\psi)|}{|U|}, \quad (6)$$

where $|\cdot|$ denotes the cardinality of a set. In data mining, G is referred to as the support of the rule, and $m(\phi \wedge \psi)$ is the supporting set of the rule.

Absolute support: The absolute support measures the correctness of the rule. It is defined by:

$$AS(\phi \Rightarrow \psi) = \frac{|m(\psi) \cap m(\phi)|}{|m(\phi)|}. \quad (7)$$

In data mining, the absolute support is also referred to as the confidence or accuracy of the rule. It may be viewed as the conditional probability of a randomly selected element satisfying ψ given that the element satisfies ϕ. In set-theoretic terms, it is the degree to which $m(\phi)$ is included in $m(\psi)$.

Change of support: Typically, a decision rule $\phi \Rightarrow \psi$ is used to draw conclusion about ψ based on ϕ. Intuitively speaking, the rule is useful if we can say something more about ψ after knowing ϕ. The change of support provides a quantitative measures to capture this aspect of a rule, which is defined by:

$$CS(\phi \Rightarrow \psi) = AS(\phi \Rightarrow \psi) - G(\psi). \quad (8)$$

One may consider $G(\psi)$ to be the prior probability of ψ and $AS(\phi \Rightarrow \psi)$ the posterior probability of ψ after knowing ϕ. The difference of posterior and prior probabilities represents the change of our confidence regarding whether ϕ is actually related to ψ. For a positive value, one may say that ϕ is positively related to ψ; for a negative value, one may say that ϕ is negatively related to ψ.

It should be pointed out that the above probabilistic interpretations of rules are different from the interpretation based on logic implication [18]. More specifically, probabilistic interpretations focus on the case where ϕ is satisfied.

Like the logic implication $\rightarrow$, the symbol $\Rightarrow$ represents a one-way relationship between concepts. In some situations, we are interested in two-way relationships [21]. For this purpose, we introduce a new symbol $\Leftrightarrow$ and a new form of decision rule $\phi \Leftrightarrow \psi$. This kind of decision rules can be used to describe the similarity, independence and overlap of two concepts. Some probabilistic interpretations are summarized below:

Mutual support: The mutual support measures the relative size of the overlap of the two concepts and is defined by:

$$MS(\phi \Leftrightarrow \psi) = \frac{|m(\phi) \cap m(\psi)|}{|m(\phi) \cup m(\psi)|}. \quad (9)$$

The mutual support can be expressed in terms of the absolute supports of the pair of rules $\phi \Rightarrow \psi$ and $\psi \Rightarrow \phi$ as [21]:

$$MS(\phi \Leftrightarrow \psi) = \frac{1}{\frac{1}{AS(\phi \Rightarrow \psi)} + \frac{1}{AS(\psi \Rightarrow \phi)} - 1}. \quad (10)$$

The mutual support can be re-expressed by:

$$MS(\phi \Leftrightarrow \psi) = 1 - \frac{|m(\phi) \Delta m(\psi)|}{|m(\phi) \cup m(\psi)|}, \quad (11)$$

where $A \Delta B = (A \cup B) - (A \cap B)$ is the symmetric difference between two sets A and B. The measure $|A \Delta B| / |A \cup B|$ is commonly known as the MZ metric for measuring distance between two sets [8]. Thus, MS is a similarity measure of ϕ and ψ.

Probabilistic independence (ratio): The degree of probabilistic independence of ϕ and ψ is defined by:

$$IND(\phi \Leftrightarrow \psi) = \frac{G(\phi \wedge \psi)}{G(\phi)G(\psi)}. \quad (12)$$

It is the ratio of the joint probability of $\phi \wedge \psi$ and the probability obtained if ϕ and ψ are assumed to be probabilistically independent. Alternatively, independence can be expressed as [4]:

$$IND(\phi \Leftrightarrow \psi) = \frac{AS(\phi \Rightarrow \psi)}{G(\psi)}. \quad (13)$$

It is related to the change of support in the sense that IND is the ratio of the posterior probability of ψ given ϕ and the prior probability of ψ.

Probabilistic independence (difference): The difference between $G(\phi \wedge \psi)$ and $G(\phi)G(\psi)$ may be used, instead of the ratio, to measure probabilistic independence:

$$D(\phi \Leftrightarrow \psi) = G(\phi \wedge \psi) - G(\phi)G(\psi). \quad (14)$$

The change of support CS is related to D by:

$$CS(\phi \Rightarrow \psi) = \frac{D(\phi \Leftrightarrow \psi)}{G(\psi)}. \quad (15)$$

Thus, the measure $CS(\phi \Rightarrow \psi)$ may be viewed as a relative difference.

Each of the probabilistic interpretations of decision rules discussed so far captures a particular aspect of decision rules. In practice, one may choose any of them depending on the intended use of decision rules in an application. Additional measures should also be studied to cover other aspects of decision rules.

IV. CONCLUSION

In this paper, we propose a generalized decision logic language GDL for granular computing. With GDL, basic issues of granular computing can be studied in either logical or set-theoretic setting. Our preliminary investigation uses a two-valued notion of satisfiability. For future research, we will study many-valued notion of satisfiability. It is expected that this will increase the expressive power of GDL.

References

[1] S. Demri and E. Orlowska, Logical analysis of indiscernibility, in: *Incomplete Information: Rough Set Analysis*, Physica-Verlag, Heidelberg, 347-380, 1998.

[2] J.W. Han, Y. Cai and N. Cercone, Data-driven discovery of quantitative rules in relational databases, *IEEE Transactions on Knowledge and Data Engineering*, **5**, 29-40, 1993.

[3] T.-F. Fan, W.-C. Hu, and C.-J. Liau, Decision logic for knowledge representation in data mining, *Proceedings of the 25th Annual International Computer Software and Applications Conference*, pp. 626-631, 2001.

[4] Gray, B. and Orlowska, M.E. CCAIIA: clustering categorical attributes into interesting association rules, *Proceedings of PAKDD'98*, pp. 132-143, 1998.

[5] S. Hirano, M. Inuiguchi, and S. Tsumoto (Eds.), *Proceedings of International Workshop on Rough Set Theory and Granular Computing*, a special issue of the *Bulletin of International Rough Set Society*, Vol. 5, No. 1-2, 2001.

[6] T.Y. Lin, Granular computing on binary relations I: data mining and neighborhood systems, in: L. Polkowski and A. Skowron (Eds.), *Rough Sets in Knowledge Discovery 1, Methodology and Applications*, Physica-Verlag, Heidelberg, pp. 286-318, 1998.

[7] T.Y. Lin, Y.Y. Yao, and L.A. Zadeh (Eds), *Data Mining and Granular Computing*, Physica-Verlag, Heidelberg, in press.

[8] E. Marczewski and H. Steinhaus, On a certain distance of sets and the corresponding distance of functions, *Colloquium Mathemmaticum*, **6**, 319-327, 1958.

[9] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[10] Z. Pawlak, Granularity of knowledge, indiscernibility and rough sets, *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, pp. 106-110, 1998.

[11] W. Pedrycz, *Granular Computing: An Emerging Paradigm*, Springer-Verlag, Berlin, 2001.

[12] L. Polkowski and A. Skowron, Towards adaptive calculus of granules, *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, pp. 111-116, 1998.

[13] A. Skowron and J. Stepaniuk, Information granules and approximation spaces, *Proceedings of Seventh International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 354-361, 1998.

[14] R. Wille, Concept lattices and conceptual knowledge systems, *Computers Mathematics with Applications*, **23**, 493-515, 1992.

[15] R.R. Yager and D. Filev, Operations for granular computing: mixing words with numbers, *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, pp. 123-128, 1998.

[16] Y.Y. Yao, Information tables with neighborhood semantics, *Data Mining and Knowledge Discovery: Theory, Tools, and Technology II*, B.V. Dasarathy (Ed.), The International Society for Optical Engineering, Bellingham, Washington, pp. 108-116, 2000.

[17] Y.Y. Yao, Granular computing: basic issues and possible solutions, *Proceedings of the 5th Joint Conference on Information Sciences*, Volume I, Association for Intelligent Machinery, pp. 186-189, 2000.

[18] Y.Y. Yao, Modeling data mining with granular computing, *Proceedings of the 25th Annual International Computer Software and Applications Conference*, pp. 638-643, 2001.

[19] Y.Y. Yao and N. Zhong, Granular computing with information table, manuscript, 1998, to appear in [7].

[20] Y.Y. Yao and N. Zhong, Potential applications of granular computing in knowledge discovery and data mining, *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics, Volume 5, Computer Science and Engineering*, pp. 573-580, 1999.

[21] Y.Y. Yao and N. Zhong, An analysis of quantitative measures associated with rules, *Proceedings of PAKDD'99*, 479-488, 1999.

[22] L.A. Zadeh, Fuzzy logic = computing with words, *IEEE Transactions on Fuzzy Systems*, **4**, 103-111, 1996.

[23] L.A. Zadeh, Fuzzy sets and information granularity, in: *Advances in Fuzzy Set Theory and Applications*, Gupta, N., Ragade, R. and Yager, R. (Eds.), North-Holland, Amsterdam, pp. 3-18, 1979.

[24] L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, **19**, 111-127, 1997.

[25] N. Zhong, A. Skowron, and S. Ohsuga S. (Eds.), *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, Springer-Verlag, Berlin, 1999.