A Note on Granular Sets and Their Relation to Rough Sets

Antoni Ligęza and Marcin Szpyrka

Institute of Automatics
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
ali@agh.edu.pl, mszpyrka@agh.edu.pl

Abstract. The paper discusses mathematical concept of granular model for data and knowledge manipulation. In order to overcome the difficulties caused by extensive data representation a new model based on *granular sets* and *granular relations* is put forward. The key idea is that the notion of set may consist of basic elements grouped into bigger granules. A granular set is formed from a universe set and a semi-partition defining granules of its elements. Formal definition of granular sets and some basic algebraic operations on granular sets are introduced in the paper. Further, the concept of granular relation is also defined and some possibilities of application of granular sets and relations to knowledge representation are put forward.

1 Introduction

Representation of data and knowledge with *adaptable granularity* of details seems to be an interesting issue for efficient dealing with large data sets. The paper presents a relatively new concept of a *granular set* and *granular relation* [5], [6]. A granular set is a structure composed of a set and a number of disjoint subsets embedded in it (the so-called semi-partition). An algebra of such sets can be constructed. Granular relation can be defined as a subset of Cartesian product of granular sets. A *granular relational algebra* can be defined as a tool for knowledge manipulation. It can be applied for verification and analysis of tabular knowledge-based systems [7] and for direct knowledge manipulation.

One reason for using granular representations can be the need for efficient dealing with large data sets. In such a case numerous detailed data are grouped into a single *granule* which can be regarded as more abstract knowledge representation. The number of detailed data items is drastically reduced and simultaneously some unimportant characteristics are hidden. In this way *knowledge extraction* from data can go on.

Another reason for using granular knowledge representation consists in the need for structuring knowledge into smaller, separate, easily manipulable chunks of knowledge. Such “knowledge granules” can be easier interpreted and understood, selected and manipulated, analyzed and verified. Granularity of knowledge seems to be an intrinsic issue in the domain of *knowledge management*.

There are a number of conceptual approaches aimed at dealing with impreciseness and knowledge abstraction. Some most important ones include *Fuzzy Sets* [13], [3],

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 251–260, 2007.

Rough Sets [9], *interval algebra*, as well as selected basic purely mathematical approaches such as ones based on *equivalence relation* and *equivalence classes*. Each of these approaches incorporates some philosophical interpretation of impreciseness. The granular set approach can be considered as an extension of interval algebra towards unordered sets and lattices or type hierarchy trees. Somewhat similar approaches are also presented in [1] and [12].

The main aim of the paper is to present a concept of granularity in sets and relational tables. The main ideas, initially presented in [5] and [6], are recapitulated in a slightly changed framework and the relationship with rough sets is discussed. Introducing granularity in the sets of data items is aimed at a more general knowledge representation, and knowledge manipulation is moved to higher abstraction level. The analysis is moved to a more abstract level of granularity, which improves efficiency – instead of atomic values of attribute domains one considers now a set or granular values. Algebraic operations on semi-partitions, granular sets and granular relations are defined. The level of granularity is adaptable – it changes according to details of knowledge representation and operations performed.

2 Granular Sets and Their Properties

A *granular set* is a structure composed of a set and a number of its disjoint subsets. It allows to consider arbitrary *granules* of the elements of the base set instead of too numerous and too detailed atomic elements. A granular set with finite number of granules can be constructed even for continuous infinite sets. Moreover, in contrast to discretization methods (where the original set is replaced with a new discrete one), it is still possible to manipulate the atomic elements or to change the partitioning of the base set.

Let a set V and some subsets $V_1, V_2, \ldots V_k$ of V be given.

Definition 1. *The sets $V_1, V_2, \ldots V_k$ form a* partition *of V, iff:*

(1) $V_1 \cup V_2 \cup \ldots \cup V_k = V$ (i.e. partition satisfies the completeness condition*),*
(2) $V_i \cap V_j = \emptyset$ for any $i \neq j$ (i.e. partition satisfies the separation condition*).*

A partition is usually induced by an equivalence relation defined over V. The sets $V_1, V_2, \ldots, V_k$ are equivalence classes; here they are called *blocks*. Note that in practice, we often do not have the possibility to consider all the subsets necessary to form a partition. In such a case the completeness condition is not satisfied. The separation condition is also not necessary however we will often expect that a semi-partition satisfies it.

Definition 2. *A* semi-partition *of V is any collection of its subsets $V_1, V_2, \ldots, V_k$. A semi-partition is* normalized *(in normal form) iff $V_i \cap V_j = \emptyset$ for all $i \neq j$.*

A semi-partition will be also called an *incomplete partition*, or an s-partition for short. An s-partition of V will be denoted as $\sigma(V)$. If not stated explicitly, all the considerations will concern normalized s-partitions. Examples of Fig. 1 show such s-partitions for a nominal and an ordered set.

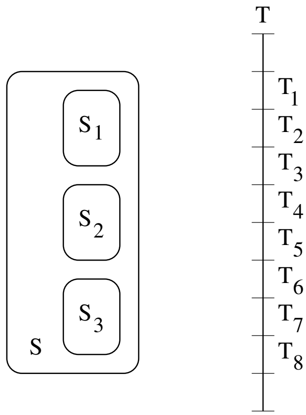

Fig. 1. Two examples of granular sets

If $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ is an s-partition, then the set of all the elements of V occurring in the s-partition $\sigma(V)$ will be called the *support* of it and it will be denoted as $[\![\sigma(V)]\!]$, and determined as $[\![\sigma(V)]\!] = V_1 \cup V_2 \cup \ldots \cup V_k$.

Note that any family of subsets of some set V can be transformed into a normalized s-partition of V having the same support. Let us consider an arbitrary collection of subsets of V, say $V_1', V_2', \ldots, V_m'$ (not necessarily disjoint ones). By subsequent replacing any two sets V_i' and V_j' $(i \neq j)$ such that $V_i' \cap V_j' \neq \emptyset$, with three sets: $V_i' \setminus V_j'$, $V_j' \setminus V_i'$ and $V_i' \cap V_j'$ one can generate an s-partition $\sigma(V)$.

For a given set V a granular set over V is defined as follows.

Definition 3. *A* granular set *G is a pair $G = \{V, \sigma(V)\}$, where $\sigma(V)$ is any s-partition defined on V. If the s-partition $\sigma(V)$ is unnormalized, then the granular set is also said to be an* unnormalized *one.*

The set V is called the domain *of the granular set, while the s-partition $\sigma(V)$ defines the so-called* signature of granularity.

Consider some two granular sets $G = (V, \sigma(V))$ and $G' = (V, \sigma'(V))$, where $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ and $\sigma'(V) = \{V_1', V_2', \ldots, V_m'\}$.

Definition 4. *The support of granular set G is bigger (smaller) than the support of granular set G' iff $[\![\sigma'(V)]\!] \subseteq [\![\sigma(V)]\!]$ ($[\![\sigma(V)]\!] \subseteq [\![\sigma'(V)]\!]$).*

Again, compare two granular sets with the same domain but different signatures. A granular set can provide finer or more rough signature of granularity.

Definition 5. *An s-partition $\sigma'(V) = \{V_1', V_2', \ldots, V_m'\}$ is* finer *than an s-partition $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ iff any set $V_i \in \sigma(V)$ can be expressed as $V_i = V_{i_1}' \cup V_{i_2}' \cup \ldots \cup V_{i_n}'$, where $V_{i_1}', V_{i_2}', \ldots V_{i_n}' \in \sigma'(V)$.*

In other words, a finer granular set (or s-partition) is build from smaller blocks and can be used to re-build the more rough one. In general, it can also contain some additional blocks not used for reconstructing the ones of the more rough s-partition.

Now, we will introduce a partial order relation among granular sets and s-partitions. For intuition, a more general granular set (its signature) covers a less general one iff any block of the latter is covered by some block of the former one.

Definition 6. *A set G is* more general *than a set G' ($G \geq G'$) iff $\sigma(V) \geq \sigma'(V)$, where the latter condition means that $\forall V_i' \in \sigma'(V)\ \exists V_j \in \sigma(V)\colon V_i' \subseteq V_j$.*

In fact Def. 6 introduces the Hoare order. The s-partition $\sigma(V)$ will be called *more general* as one operating at more abstract level of granularity. As straightforward consequences of Def. 6 we have the following propositions.

Proposition 1. *The relation of being* more general *defined by Def. 6 is an ordering relation.*

Proposition 2. *If $G \geq G'$ then also $[\![\sigma'(V)]\!] \subseteq [\![\sigma(V)]\!]$.*

Obviously, the inverse proposition is not true in general case. However, the following one holds.

Proposition 3. *Assume that $[\![\sigma'(V)]\!] \subseteq [\![\sigma(V)]\!]$. Then there exists an s-partition $\sigma^0(V)$ such that $[\![\sigma^0(V)]\!] = [\![\sigma(V)]\!]$ and $\sigma^0(V) \geq \sigma'(V)$. Simultaneously, there exists an s-partition $\sigma''(V)$ such that $[\![\sigma'(V)]\!] = [\![\sigma''(V)]\!]$ and $\sigma(V) \geq \sigma''(V)$.*

The meaning of the above proposition is simple: a bigger s-partition (i.e. one having bigger support) can always be transformed into one being also *more general* than the smaller one, but with the same support (intuitively: by gluing together some of its granules; this process is also called reduction). Further to that, a smaller s-partition can always be transformed into one having the same support but also *less general* one (the operation is based on split of the granules).

For granular sets (s-partitions) we can define typical algebraic operations. The *product* of such two s-partitions $\sigma(V)$, $\sigma'(V)$ is defined as:

$$\sigma(V) \cdot \sigma'(V) = \{V_{ij}\colon V_{ij} = V_i \cap V_j \wedge V_i \in \sigma(V) \wedge V_j \in \sigma'(V) \wedge V_{ij} \neq \emptyset\}. \quad (1)$$

Obviously, the product of two s-partitions is an s-partition. Roughly speaking, the product of two s-partitions is the s-partition composed of all nonempty intersections of their blocks. The product of two s-partitions is less general than any of them, i.e. $\sigma(V) \cdot \sigma'(V) \leq \sigma(V)$ and $\sigma(V) \cdot \sigma'(V) \leq \sigma'(V)$.

In a similar way a *composition* of s-partitions can be defined. Let a semi-partition $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ be given. For any two sets $V_i, V_j \in \sigma(V)$ we define the following partition generation operation as $V_i \sqcap V_j = \{V_i \setminus V_j, V_j \setminus V_i, V_i \cap V_j\}$. The operation can be extended to the whole semi-partition. The semi partition $\bigsqcap(\sigma(V))$ is evaluated as follows:

1. $\bigsqcap(\sigma(V))_0 = \sigma(V)$
2. If there exist sets $V_i, V_j \in \bigsqcap(\sigma(V))_n$ such that $V_i \cap V_j \neq \emptyset$ then $\bigsqcap(\sigma(V))_{n+1} = \bigsqcap(\sigma(V))_n - \{V_i, V_j\} \cup V_i \sqcap V_j$, else $\bigsqcap(\sigma(V))_{n+1} = \bigsqcap(\sigma(V))_n$.
3. If $\bigsqcap(\sigma(V))_{n+1} = \bigsqcap(\sigma(V))_n$ then $\bigsqcap(\sigma(V)) = \bigsqcap(\sigma(V))_n$.

The semi-partition $\bigsqcap(\sigma(V))$ is a normalized one and the result is independent on the order of applying of the $\sqcap$ operator. Let $x \in [\![\sigma(V)]\!]$ and let $V_x \in \bigsqcap(\sigma(V))$ denote the set containing x. The semi-partition can be divided into two disjoint subsets: $\sigma_x(V) = \{V_i \in \sigma(V) \colon x \in V_i\}$ and $\sigma_{\bar{x}}(V) = \sigma(V) - \sigma_x(V)$. Hence, the following equality holds:

$$V_x = \bigcap \sigma_x(V) - \bigcup \sigma_{\bar{x}}(V). \tag{2}$$

The *composition* of s-partitions $\sigma(V)$ and $\sigma(V)$ is defined as follows:

$$\sigma(V) \circ \sigma'(V) = \bigsqcap(\sigma(V) \cup \sigma'(V)). \tag{3}$$

For any two s-partitions $\sigma(V)$ and $\sigma'(V)$ we define also a *cover* of them, i.e. an s-partition covering all the elements of V belonging to some component set of at least one of them. For a semi-partition $\sigma(V) = \{V_1, V_2, \ldots, V_k\}$ the following sum operation is introduced:

1. $\bigsqcup(\sigma(V))_0 = \sigma(V)$
2. If there exist sets $V_i, V_j \in \bigsqcup(\sigma(V))_n$ such that $V_i \cap V_j \neq \emptyset$ then $\bigsqcup(\sigma(V))_{n+1} = \bigsqcup(\sigma(V))_n - \{V_i, V_j\} \cup \{V_i \cup V_j\}$, else $\bigsqcup(\sigma(V))_{n+1} = \bigsqcup(\sigma(V))_n$.
3. If $\bigsqcup(\sigma(V))_{n+1} = \bigsqcup(\sigma(V))_n$ then $\bigsqcup(\sigma(V)) = \bigsqcup(\sigma(V))_n$.

The semi-partition $\bigsqcup(\sigma(V))$ is a normalized one and the result is independent on the order of applying of the $\sqcup$ operator. Let $x \in [\![\sigma(V)]\!]$ and let $V_x \in \bigsqcup(\sigma(V))$ denotes the set containing x. Let the set $\sigma_x(V)$ be defined as follows:

1. $\sigma_x(V)_0 = \{V_i\}$, where $V_i \in \sigma(V)$ and $x \in V_i$.
2. $\sigma_x(V)_{n+1} = \sigma_x(V)_n \cup \{V_i \in \sigma(V) - \sigma_x(V)_n \colon V_i \cap (\bigcup \sigma_x(V)_n) \neq \emptyset\}$.
3. If $\sigma_x(V)_{n+1} = \sigma_x(V)_n$ then $\sigma_x(V) = \sigma_x(V)_n$.

Hence, $V_x = \bigcup \sigma_x(V)$.

The *cover* of s-partitions $\sigma(V)$ and $\sigma'(V)$ is defined as follows:

$$\sigma(V) + \sigma'(V) = \bigsqcup(\sigma(V) \cup \sigma'(V)). \tag{4}$$

For intuition, both s-partitioning and generating a cover are kinds of operations preserving covering of the same elements of V which are covered by the initial family of subsets (the support). However, in case of s-partitioning one preserves also the definition of initial signatures (structuring) (e.g. the boundaries of intervals of characteristic subsets of V), while in the case of cover generation a kind of maximal reduction of the subsets is performed. There is also $\sigma(V) + \sigma'(V) \geq \sigma(V)$ and $\sigma(V) + \sigma'(V) \geq \sigma'(V)$.

Consider a *reduction* operation of transforming an s-partition into another, more general one, by *gluing* some of its elements (non-overlapping ones). The reduction of an s-partition consists in replacing several blocks with an equivalent single block. The generated output is aimed to be a normalized s-partition, so in the case of intervals, gluing is allowed only for intervals which meet or the so-called non-convex intervals must be admitted. The generated s-partition is equivalent with regard to the elements covered, but simultaneously it is more general than the input one.

Finally, consider the so-called *split* operation. The operation consists in replacing each element V_i of the initial s-partition with a family of its subsets such that the sum of them is equal to V_i. The result of the split operation is less general than the initial s-partition. In general, the split operation gives no unique result. For this reason, it may be useful to define the so-called *induced split*, where the result depends on another s-partition which is compared with the one under interest.

3 Granular Sets and Rough Sets

Granular sets as introduced in Section 2 may constitute a tool for defining rough sets. However, since the assumptions of the presented approach are weaker than in case of a partition induced by an equivalence relation, it is not always possible to define the upper approximation.

Consider a set V and an s-partition $\sigma(V) = \{V_1, V_2, \ldots, V_n\}$. Let X denote some subset of V, i.e. $X \subseteq V$. The lower approximation of X with s-partition $\sigma(V)$ is defined as

$$\underline{R}X = \{V_i \in \sigma(V) \colon V_i \subseteq X\}.$$

The lower approximation of X always exists; in some cases it can be the empty set. Also $\underline{R}X \subseteq X$ in the sense $\llbracket \underline{R}X \rrbracket \subseteq X$.

Contrary to classical rough set theory, the upper approximation defined with a particular s-partition may not exist, i.e. it may be empty. We define an approximation of set X in the following way:

$$RX = \{V_i \in \sigma(V) \colon V_i \cap X \neq \emptyset\}.$$

Note that contrary to the case of partitions based on equivalence relation, it can be the empty set. Further, in some cases the basic property that $X \subseteq \overline{R}X$ (in the sense that $X \subseteq \llbracket \overline{R}X \rrbracket$) may be violated.

For practical reasons, to obtain the upper approximation covering X (e.g. when verifying completeness of systems) it may be of interest to look for the uncovered cases, i.e. the completion of an approximation to an upper approximation – such that all elements of X will be covered. In order to do that we first define the completion of an approximation as

$$\overline{RX} = X \setminus RX.$$

The upper approximation can be defined now as

$$\overline{R}X = RX \cup \overline{RX}.$$

4 Granular Relations

Using the presented idea of granular set, a granular relation can be defined in a straightforward way. Consider some collection of sets $D_1, D_2, \ldots, D_n$. Let there be defined some granular sets on them, i.e. $G_1 = (D_1, \sigma_1(D_1)), G_2 = (D_2, \sigma_2(D_2)), \ldots, G_n = (D_n, \sigma_n(D_n))$.

Definition 7. *A* granular relation $R(G_1, G_2, \ldots, G_n)$ *is any set* $R_G \subseteq U_G$ *where*

$$U_G = \sigma_1(D_1) \times \sigma_2(D_2) \times \ldots \times \sigma_n(D_n). \tag{5}$$

The set U_G *will be referred to as* granular universe *or* granular space. *If at least one of the granular sets was unnormalized, the relation is also said to be unnormalized one.*

The elements (rows) of a granular relation will be called *boxes*. Note that in fact a granular relation defines a kind of meta-relation, i.e. one based on sets instead of single elements. In fact, if R is a relation defined as $R \subseteq D_1 \times D_2 \times \ldots \times D_n$, then any tuple of R is like a thread in comparison to elements of R_G which are like a cord or a pipe.

Consider an example concerned with time-table development for a university or a school. First, there is certainly a finite set of students, say S. Instead of specifying for each student his personal schedule, the university authorities consider "granules" of them, i.e. years, groups, etc. If S_1, S_2, and S_3 are the groups of the first year, then a granular structure $G(S) = (S, \{S_1, S_2, S_3\})$ can be considered useful when assigning classes to the students of the first year. Further, time is also considered granular – instead of precise exact time one would rather consider traditional intervals, such as lessons (e.g. 45 or 55 minutes each) or periods of the length 1h30min which form a frame for constructing the schedule. Let T be the discrete set of time values from 7:00 to 21:00, and let T_1=[8:00,9:30], T_2=[9:30,11:00], T_3=[11:00,12:30], T_4=[12:30,14:00], T_5=[14:00,15:30], T_6=[15:30,17:00], T_7=[17:00,18:30] and T_8=[18:30,20:00]. Some other sets, such as the set of professors P, the set B of rooms or the set of classes (subjects) C are considered here at the level of single items. For simplicity, we focus on the schedule for some specific day, so the problem is to assign each group a professor, room and subject for any legal time interval. If $P = \{p_1, p_2, p_3\}$ is the set of professors, $B = \{b_1, b_2\}$ is the set of rooms and $C = \{c_1, c_2, c_3\}$ is the set of subjects, the relation representing the schedule can be as shown in Fig. 2.

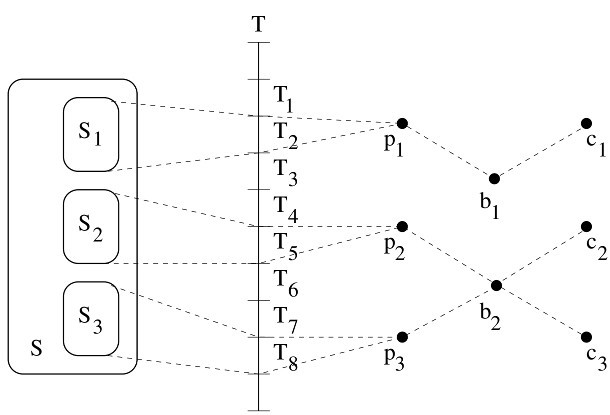

Fig. 2. Example of a granular relation

The relation shown in Fig. 2 is defined by the following tuples: $\{(S_1, T_2, p_1, b_1, c_1), (S_2, T_6, p_2, b_2, c_2), (S_3, T_8, p_3, b_2, c_3)\}$. In the tuples of this relation only the first two elements of each tuple are granules; the other three are basic items.

5 Granular Knowledge Representation Systems

Granular sets and granular relations can be applied to develop *granular knowledge representation systems* (also called *extended tabular systems*) [7]. In comparison with knowledge representation systems considered in [9], nonatomic values of attributes are admissible. In similar way granular decision tables can be introduced.

Table 1. Optician Decision Table

Number	Age	Spectacle	Astigmatic	Tear p.r.	Decision
1	y	m	y	n	h
2	y	h	y	n	h
3	p	m	y	n	h
4	q	m	y	n	h
5	y	m	n	n	s
6	y	h	n	n	s
7	p	m	n	n	s
8	p	h	n	n	s
9	q	h	n	n	s
10	y	m	n	r	n
11	y	m	y	r	n
12	y	h	n	r	n
13	y	h	y	r	n
14	p	m	n	r	n
15	p	m	y	r	n
16	p	h	n	r	n
17	p	h	y	r	n
18	p	h	y	n	n
19	q	m	n	r	n
20	q	m	n	n	n
21	q	m	y	r	n
22	q	h	n	r	n
23	q	h	y	r	n
24	q	h	y	n	n

After Pawlak [9] let us consider the following decision table (see Tab. 1). The attributes and their domains are as follows:

- A_1 := age; $D_1 = \{y, p, q\}$, where: y – young, p – pre-presbyotic, q – presbyotic,
- A_2 := spectacle; $D_2 = \{m, h\}$, where: m – myope, h – hypermyope,
- A_3 := astigmatic; $D_3 = \{n, y\}$, where: n – no, y – yes,
- A_4 := tear production rate; $D_4 = \{r, n\}$, where: r – reduced, n – normal,

– D := type of contact lenses (decision attribute); $D_D = \{h, s, n\}$, where: h – hard contact lenses, s – soft contact lenses, n – no contact lenses.

The considered table is complete and deterministic. Methods based on the rough set theory can be used to reduce such a decision table. The reduction algorithm consists in the elimination of conditions from a decision table, which are unnecessary to make decisions specified in the table. Finally, a table with only nine decision rules can be received (see Tab.2).

Table 2. The reduced form of the Optician Decision Table (using rough set approach)

Number	Age	Spectacle	Astigmatic	Tear p.r.	Decision
1	y	–	y	n	h
2	–	m	y	n	h
3	y	–	n	n	s
4	p	–	n	n	s
5	–	h	n	n	s
6	–	–	–	r	n
7	p	h	y	–	n
8	q	h	y	–	n
9	q	m	n	–	n

If granular sets and relations are considered further reduction is possible. The most reduced form of the considered decision table is presented in Tab. 3. The third and sixth row contains non-atomic values of the attribute Age.

Table 3. The reduced form of the Optician Decision Table

Number	Age	Spectacle	Astigmatic	Tear p.r.	Decision
1	y	–	y	n	h
2	–	m	y	n	h
3	$\{y, p\}$	–	n	n	s
4	–	h	n	n	s
5	–	–	–	r	n
6	$\{p, q\}$	h	y	–	n
7	q	m	n	–	n

Row 3 of table 3 is the result of gluing rows 3 and 4 of table 2. Similarly, row 6 of table 3 is the result of gluing rows 7 and 8 of table 2.

6 Summary

The paper presents a concept of granular knowledge representation and manipulation. The key notions discussed here are the one of granular set and granular relation, both of them base on the idea of s-partition of a set. It has been shown that granular sets and relations can be applied to develop granular knowledge representation systems, which enable to represent knowledge in more condense form.

The granular approach presented in the paper can be used for efficient knowledge representation in rule-based systems [7]. Some directions of possible future works include development of efficient algorithms for verification of theoretical properties, such as subsumption among rules, completeness of sets of rules, possibility of reduction, etc. Moreover, further extensions of granular attributive logic are also explored [8].

References

1. Järvinen, J., Kortelainen, J.: A Note on Definability in Rough Set Theory. In: De Baets, B., De Caluwe, R., De Tré, G., Fodor, J., Kacprzyk, J., Zadrożny, S. (eds.): Current Issues in Data and Knowledge Engineering, Akademicka Oficyna Wydawnicza EXIT, Warsaw, Poland (2004)
2. Peters, J.F., Pawlak, Z., Skowron, A.: A rough set approach to measuring information granules. In: Proc. of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment (COMPSAC'02), Oxford, pp. 1135–1139 (2002)
3. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic. In: Theory and Applications, Prentice-Hall, Englewood Cliffs (1995)
4. Ligęza, A.: Intelligent data and knowledge analysis and verification; towards a taxonomy of specific problems. In: Vermesan, A., Coenen, F. (eds.) Validation and Verification of Knowledge Based Systems: Theory, Tools and Practice, pp. 313–325. Kluwer Academic Publishers, Boston (1999)
5. Ligęza, A.: Granular sets and granular relations: Towards a higher abstraction level in knowledge representation. In: Intelligent Information Systems 2002, pp. 331–340. Physica-Verlag, Heidelberg (2002)
6. Ligęza, A.: Granular sets and granular relations for algebraic knowledge management. In: Smart engineering system design: neural networks, fuzzy logic, evolutionary programming, complex systems and artificial life: Proc. of the Artificial Neural Networks in Engineering Conference (ANNiE 2003). ASME Press Series on Intelligent Engineering Systems Through Artificial Neural Networks, St. Louis, Missouri, USA, vol. 13, pp. 169–174 (2003)
7. Ligęza, A.: Logical foundations of rule-based systems. In: Studies in Computational Intelligence, vol. 11, Springer, Berlin (2006)
8. Ligęza, A., Nalepa, G.J.: Knowledge Representation with Granular Attributive Logic for XTT-based Expert Systems. In: Prof. of the 20th International FLAIRS Conference, Key West, Florida (to appear)
9. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, The Netherlands (1991)
10. Peters, J.F., Skowron, A., Suraj, Z., Rząsa, W., Borkowski, M.: Clustering: A rough set approach to constructing information granules. In: Proceedings of the 6th International Conference on Soft Computing and Distributed Processing (SCDP'02), Rzeszów, pp. 57–61 (2002)
11. Peters, J.F., Pawlak, Z., Skowron, A.: A rough set approach to measuring information granules. In: Proc. of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, Oxford, 2002, pp. 1135–1139 (2002)
12. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
13. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)