Modeling of High Quality Granules

Andrzej Skowron[1] and Jaroslaw Stepaniuk[2]

[1] Institute of Mathematics
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[2] Department of Computer Science
Bialystok University of Technology
Wiejska 45A, 15-351 Bialystok, Poland
jstepan@wi.pb.edu.pl

Abstract. In Granular Computing (GC) we search for granules satisfying some criteria. These criteria can be based on the minimal length principle, can express acceptable risk degrees of granules, or can use some utility functions. We discuss the role of approximation spaces in modeling granules satisfying such criteria.

1 Introduction

Information granulation can be viewed as a human way of achieving data compression and it plays a key role in implementing the divide-and-conquer strategy in human problem-solving [22]. Granules are obtained in the process of information granulation. Granular computing (GC) is based on processing of complex information entities called granules. Generally speaking, granules are collection of entities, that are arranged together due to their similarity, functional adjacency or indistinguishability [22].

One of the main branch of GC is Computing with Words and Perceptions (CWP). GC "derives from the fact that it opens the door to computation and reasoning with information which is perception - rather than measurement-based. Perceptions play a key role in human cognition, and underlie the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are driving a car in city traffic, playing tennis and summarizing a story" [22].

We consider the optimization tasks in which we are searching for optimal solutions satisfying some constraints. These constraints are often vague, imprecise, and/or specifications of concepts and dependencies between them involved in the constraints are incomplete. Decision tables [11] are examples of such constraints. Another example of constraints can be found, e.g., in [4,15] where a specification is given by a domain knowledge and data sets. Domain knowledge is represented by ontology of vague concepts and dependencies between them. In a more general case, the constraints can be specified in a simplified fragment of a natural language [22].

M. Kryszkiewicz et al. (Eds.): RSEISP 2007, LNAI 4585, pp. 300–309, 2007.

Granules are constructed in computations aiming at solving the mentioned above optimization tasks. In our approach, we use the general optimization criterion based on the minimal length principle. In searching for (sub-)optimal solutions it is necessary to construct many compound granules using some specific operations such as generalization, specification or fusion. Granules are labeled by parameters. By tuning these parameters we optimize the granules relative to their description size and the quality of data description, i.e., two basic components on which the optimization measures are defined.

From this general description of tasks in GC it follows that together with specification of elementary granules and operation on them it is necessary to define measures of granule quality (e.g., measures of their inclusion, covering or closeness) and tools for measuring the size of granules. Very important are also optimization strategies of already constructed (parameterized) granules.

We discuss the searching process for relevant (for concept approximation) neighborhoods in approximation spaces based on modeling relevant relational and syntactical structures build from partial information about objects and concepts.

The importance in GC of risk measures defined on granules is emphasized. The values of such measures are indicating how properties of granules are changing when some of their parameters were changed.

We present an example showing how utility functions defined on granules can be used in GC. In general, utility functions are helping to relax the binary constraints by making it possible to work with constraints which should be satisfied to a degree expressed by utility functions.

This paper is structured as follows. In Section 2 we discuss definitions of approximation spaces and approximations. In Section 3 we discuss constraints that must be satisfied during the information granulation process. In Section 4 we present some remarks about risk in construction of granules.

2 Approximation Spaces and Approximations

In this section, we discuss the definition of an approximation space from [13,19]. Approximation spaces can be treated as granules used for concept approximation. They are some special parameterized relational structures. Tuning of parameters is making it possible to search for relevant approximation spaces relative to given concepts.

Definition 1. *A parameterized approximation space is a system* $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$*, where*

- U *is a non-empty set of objects,*
- $I_{\#} : U \to P(U)$ *is an uncertainty function, where* $P(U)$ *denotes the power set of* U*,*
- $\nu_{\$} : P(U) \times P(U) \to [0,1]$ *is a rough inclusion function,*

and $\#, \$$ *denote vectors of parameters (the indexes* $\#, \$$ *will be omitted if it does not lead to misunderstanding).*

The uncertainty function defines for every object x, a set of objects described similarly to x. The set $I(x)$ is called the neighborhood of x (see, e.g., [11,13]).

The rough inclusion function $\nu_{\$} : P(U) \times P(U) \to [0,1]$ defines the degree of inclusion of X in Y, where $X, Y \subseteq U$.

In the simplest case it can be defined by (see, e.g., [13,11]):

$$\nu_{SRI}(X,Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)} & \text{if } X \neq \emptyset \\ 1 & \text{if } X = \emptyset. \end{cases}$$

The lower and the upper approximations of subsets of U are defined as follows.

Definition 2. *For any approximation space $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ and any subset $X \subseteq U$, the lower and upper approximations are defined by*
$LOW\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}(x), X\right) = 1\}$,
$UPP\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}(x), X\right) > 0\}$, *respectively.*

The lower approximation of a set X wit respect to the approximation space $AS_{\#,\$}$ is the set of all objects, which can be classified with certainty as objects of X with respect to $AS_{\#,\$}$. The upper approximation of a set X with respect to the approximation space $AS_{\#,\$}$ is the set of all objects which can be possibly classified as objects of X with respect to $AS_{\#,\$}$.

Several known approaches to concept approximations can be covered using the discussed here approximation spaces, e.g., (see, e.g., references in [13]).

One can use yet another approach to approximation based on a fusion of inclusion degree of neighborhoods in concepts and their complements in definition of approximations. Let $f : [0,1] \longrightarrow [0,1]$ denote such a fusion function. For any subset $X \subseteq U$, the lower and upper approximations are defined by

$$LOW(AS_{\#,\$}, X) = \{x \in U : f(\{\nu_{\$}(I_{\#}(y), X) : x \in I_{\#}(y)\}) = \{1\}\},$$

$$UPP(AS_{\#,\$}, X) = \{x \in U : f(\{\nu_{\$}(I_{\#}(y), X) : x \in I_{\#}(y)\}) \neq \{0\}\}.$$

The classification methods for concept approximation developed in machine learning and pattern recognition make it possible to decide for a given object if it belongs to the approximated concept or not. The classification methods yield the decisions using only partial information about approximated concepts. This fact is reflected in the rough set approach by assumption that concept approximations should be defined using only partial information about approximation spaces. To decide if a given object belongs to the (lower or upper) approximation of a given concept the rough inclusion function values are needed. In the next section, we show how such values necessary for classification making are estimated on the basis of available partial information about approximation spaces.

3 Quality of Approximation Space

A key task in granular computing is the information granulation process, which is responsible in the formation of information aggregates (patterns) from a set

of available data. A methodological and algorithmic issue is the formation of transparent (understandable) information granules, meaning that they should provide a clear and understandable description of patterns held in data. Such fundamental property can be formalized by a set of constraints that must be satisfied during the information granulation process. Usefulness of these constraints is measured by quality of approximation space:

$$Quality_1 : Set\_AS \times P(U) \to [0,1]$$

where U is a non-empty set of objects and Set_AS is a set of possible approximation spaces with the universe U.

Example 1. If $UPP(AS,X)) \neq \emptyset$ for $AS \in Set\_AS$ and $X \subseteq U$ then

$$Quality_1(AS,X) = \nu_{SRI}(UPP(AS,X), LOW(AS,X)) = \frac{card(LOW(AS,X))}{card(UPP(AS,X))}$$

The value $1 - Quality_1(AS,X)$ expresses the degree of completeness of our knowledge about X, given the approximation space AS.

Example 2. In applications we usually use another quality measures based on the minimal length principle [12,21] where also the description length of approximation is included. Let us denote by $description(AS,X)$ the description length of approximation of X in AS. the description length may be measured, e.g., by the sum of description lengths of algorithms testing membership for neighborhoods used in construction of the lower approximation, the upper approximation, and the boundary region of the set X. Then the quality $Quality_2(AS,X)$ can be defined by

$$Quality_2(AS,X) = g(Quality_1(AS,X), description(AS,X))$$

where g is a relevant function used for fusion of values $Quality_1(AS,X)$ and $description(AS,X)$.

One can consider different optimization problems relative to a given class Set_AS of approximation spaces. For example, for given $X \subseteq U$ and a threshold $t \in [0,1]$ one can search for an approximation space AS satisfying the constraint $Quality(AS,X) \geq t$. Another example can be related to searching for an approximation space satisfying additionally the constraint $Cost(AS) < c$ where $Cost(AS)$ denotes the cost of approximation space AS (e.g. measured by the number of attributes used to define neighborhoods in AS) and c is a given threshold.

In the process of searching for (sub-)optimal approximation spaces different strategies are used. Let us consider one illustrative example. Let $DT = (U,A,d)$ be a decision system (a given sample of data) where U is a set of objects, A is a set of attributes and d is a decision. We assume that for any object x is accessible only a partial information equal to the A-signature of x (object signature, for short), i.e., $Inf_A(x) = \{(a,a(x)) : a \in A\}$ and analogously for any concept there

is only given a partial information about this concept by a sample of objects, e.g., in the form of decision table. One can use object signatures as new objects in a new relational structure $\mathcal{R}$. In this relational structure $\mathcal{R}$ are also modeled some relations between object signatures, e.g., defined by the similarities of these object signatures. Discovery of relevant relations on object signatures is an important step in the searching process for relevant approximation spaces. In the next step, we select a language $\mathcal{L}$ of formulas expressing properties over the defined relational structure $\mathcal{R}$ and we search for relevant formulas in $\mathcal{L}$. The semantics of formulas (e.g., with one free variable) from $\mathcal{L}$ are subsets of object signatures. Observe that each object signature defines a neighborhood of objects from a given sample (e.g., decision table DT) and another set on the whole universe of objects being an extension of U. In this way, each formula from $\mathcal{L}$ defines a family of sets of objects over the sample and also another family of sets over the universe of all objects. Such families can be used to define new neighborhoods of a new approximation space, e.g., by taking unions of the described above families. In the searching process for relevant neighborhoods, we use information encoded in the given sample. More relevant neighborhoods are making it possible to define relevant approximation spaces (from the point of view of the optimization criterion). It is worth to mention that often this searching process is even more compound. For example, one can discover several relational structures (not only one, e.g., $\mathcal{R}$ as it was presented before) and formulas over such structures defining different families of neighborhoods from the original approximation space and next fuse them for obtaining one family of neighborhoods or one neighborhood in a new approximation space. Such kind of modeling is typical for hierarchical modeling [4], e.g., when we search for relevant approximation space for objects composed from parts for which some relevant approximation spaces have been already found.

Let us consider some illustrative examples of granule modeling (see Figure 1). Any object $x \in U$, in a given information system $IS_1 = (U, A)$, is perceived by means of its signature $Inf_A(x) = \{(a, a(x)) : a \in A\}$. On the first level, we consider objects with signatures represented by the information system $IS_1 = (U, A)$. Objects with the same signature are indiscernible. On the next level of modeling we consider as objects some relational structures over signatures of objects from the first level. For example, for any signature u one can consider as a relational structure a neighborhood defined by a similarity relation τ between signatures of objects from the first level (see Figure 1). Attributes of objects on the second level describe properties of relational structures. Hence, indiscernibility classes defined by such attributes are sets of relational structures; in our example sets of neighborhoods. We can continue this process of hierarchical modeling by considering as objects on the third level signatures of objects from the second level. In our example, the third level of modeling represents modeling of clusters of neighborhoods defined by the similarity relation τ. Observe that it is possible to link objects from a higher level with objects from a lower level. In our example, any object from the second level is a neighborhood or τ. Any element u' of this neighborhood defines on the first level an elementary gran-

ule (indiscernibility class) $\{x \in U : Inf_A(x) = u'\}$. Hence, any neighborhood $\tau(u)$ defines on the first level a family of elementary granules corresponding to signatures from the neighborhood. Now, one can consider as a quality measure for the similarity τ a function assigning to τ a degree to which the union of the elementary granules mentioned above is included into a given concept.

In the second example, we assume that the information system on the first level has a bit more general structure. Namely, on any attribute value set V_a there is defined a relational structure $\mathcal{R}_a$ and a language $\mathcal{L}_a$ of formulas for expressing properties over V_a. For example, one can consider an attribute *time* with values in the set $\mathcal{N}$ of natural numbers, i.e., $V_a \subseteq \mathcal{N}$. The value $time(x)$ is interpreted as a time at which the object x was perceived. The relational structure $\mathcal{R}_{time}$ is defined by (V_a, S), where S is the successor relation in $\mathcal{N}$, i.e., xSy if and only if $y = x + 1$. Then relational structures on the second layer can correspond to windows of a given length T, i.e., structures of the form $(\{u_1, \ldots, u_T\}, S)$ where for some $x_1, \ldots, x_T$ we have $u_i = Inf_A(x_i)$ and $time(x_{i+1}) = time(x_i) + 1$ for $i = 1, \ldots, T$. Hence, the attributes on the second layer of modeling correspond to properties of windows while attributes on the third level could correspond to clusters of windows. Again in looking for relevant clusters we should consider links of the higher levels with lower levels. Another possibility will be to consider some relational structures on the attributes values sets on the second layer. They could allow us to model relations between windows such as overlapping, earlier than. Then, attributes on this level could describe properties of sequences of windows. Such attributes can correspond to some models of processes. Yet another possibility is to use additionally some spatial relations (e.g., nearness) between the successive elements of windows.

For structural objects, it is often used a decomposition method for modeling relational structures on the second level. The object signatures are decomposed into parts and some relations between such parts are considered which are defined over relational structures with the universe $\times_{a \in A} V_a$. One of the methods is based on searching for (i) a decomposition of the object signatures; (ii) tolerance relations defined on parts of object signatures received by decomposition; and (iii) relations over tolerance classes of such tolerance relations (e.g., expressing closeness of classes of parts corresponding to tolerance classes). This method aims to discover relational structures such that it is possible to define over such structures relevant clusters (granules, patterns) of objects for the considered task (e.g., approximation of concepts). The relations over tolerance classes are used for filtering relevant compositions of parts of object signatures defined by tolerance classes. This approach is closely related to constrained sums of information systems [16]. For example, any object of the constrained sum $+_R(IS_1, IS_2)$ of information systems IS_1, IS_2 consists of pairs (x_1, x_2) of objects from IS_1 and IS_2 satisfying some constraints described by $R \subseteq U_1 \times U_2$, i.e., $U = R \cap (U_1 \times U_2)$. The attributes of $+(IS_1, IS_2)$ consist of the attributes of IS_1 and IS_2, except that if there are any attributes in common, then we make their distinct copies, to avoid confusion.

It is worthwhile mentioning that in searching under uncertainty for relevant granules it is also necessary to use methods for estimation if the discovered patterns on (training) samples of objects are relevant on the whole universe of objects.

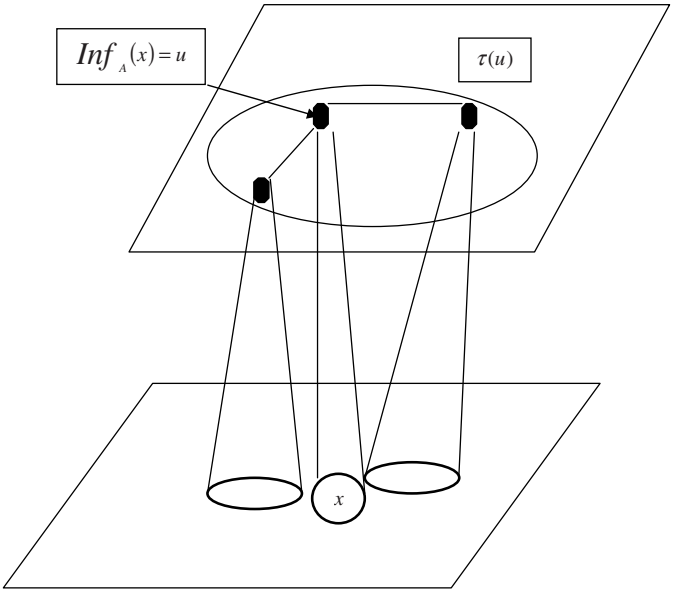

Fig. 1. Modeling of granules

The above examples are typical for granular computing where for a given task it is necessary to search for granules in a given granular system which are satisfying some optimization criteria. The discussed methods are used in spatio-temporal reasoning (see, e.g, [17]), in behavioral pattern identification and planning (see, e.g., [4,3]). There are some other basic concepts which should be considered in granular computing. One of them is related to risk. In the following section we present some remarks about risk in construction of granules.

4 Risk and Utility Functions in Construction of Granules

There is a large literature on relationships between decision making and risk. In this section, we discuss some problems related to risk in granular computing. An example of risk analysis (based on rough sets) for medical data the reader can find in [5].

First we recall the definition of *granule system*. Any such system GS consists of a set of granules G. Moreover, a family of relations with the intended meaning *to be a part to a degree* between granules is distinguished. The degree structure

is described by a relation *to be an exact part*. More formally, a granule system is any tuple

$$GS = (G, H, <, \{\nu_p\}_{p \in H}, size) \tag{1}$$

where G is a non-empty set of granules. H is a non-empty set of granule inclusion degrees with a binary relation $<$ (usually a strict partial order) which defines on H a structure used to compare the degrees. $\nu_p \subseteq G \times G$ is a binary relation *to be a part to a degree at least p* between granules from G, called *rough inclusion*. $size : G \longrightarrow R_+$ is the granule size function, where R_+ is the set of nonnegative reals.

In constructing of granule systems it is necessary to give a constructive definition of all their components. In particular, one should specify how more compound granules are defined from already defined granules or given elementary granules. Usually, the set of granules is defined as the least set generated from distinguished elementary granules (e.g., defined by indiscernibility classes) by some operations on the granules. These operations are making it possible to fuse elementary granules for obtaining new granules relevant for the task to be solved. In the literature many different operations on granules are reported (see, e.g., [15]) from those defined by boolean combination of descriptors to compound classifiers or networks of classifiers.

Let us consider, a task of searching in the set of granules of a granule system GS for a granule g satisfying a given constraint to a satisfactory degree, e.g., $\nu_{tr}(g, g_0)$, where $\nu : G \times G \longrightarrow [0, 1]$ is the inclusion function, $\nu_{tr}(g, g_0)$ means that $\nu(g, g_0) \geq tr$, g_0 is a given granule and tr is a given threshold. Let g^* be a solution, i.e., g^* satisfies the condition

$$\nu(g^*, g_0) > tr. \tag{2}$$

Risk analysis is a well established notion in decision theory [6]. We would like illustrate the importance of risk analysis in GC.

A typical risk analysis task in GC can be described as follows. For a granule g^* is constructed a granule $N(g^*)$, i.e. representing a cluster of granules defined by g^* received by changing some parameters of g^* such as attribute values used in the g^* description. We would like to estimate how this changes influence the condition (2).

First, let us assume that $\nu(g^*, g_0) = \nu_{SRI}(\|g^*\|, \|g_0\|)$, where $\|\cdot\|$ denotes the semantic of granule, i.e., a function $\|\cdot\| : G \longrightarrow P(U)$ for a given universe of objects U and ν_{SRI} is the standard rough inclusion function. Then, one can take $\delta^* = \arg\min_{\delta \in [0,tr]}(\nu(N(g^*), g_0) \geq tr - \delta)$. The value δ^* can be treated as a *risk degree* of changing the inclusion degree in g_0 when the granule g^* is substituted by $N(g^*)$.

One can consider a hierarchy of granules over g^* defined by an ascending sequence $N_1(g^*), \ldots, N_k(g^*)$, i.e., $\|N_1(g^*)\| \subseteq \ldots \subseteq \|N_k(g^*)\|$ and corresponding risk degrees $\delta_1^* \leq \ldots \delta_k^*$. For example, if δ_1^* is sufficiently small than g^* is called *robust* with respect to deviations caused by taking $N_1(g^*)$ instead of g^*. However, when i is increasing then taking $N_i(g^*)$ instead of g^* gradually increases the

risk degree. The above example illustrates importance of risk analysis in GC. Information maps introduced in [18] can be used for risk analysis.

Let us now move to the concept of *utility function* over granules. The concept of utility function has been intensively studied in decision theory or game theory [8,7]. We would like to present an illustrative example showing that such functions are important for granule systems.

We assume two granule systems GS and GS_0 with granule sets G and G_0 are given. We consider two properties of granules in this systems, i.e., $P \subseteq G$ and $P_0 \subseteq G_0$ Moreover, we assume that checking the membersip for P is much simpler than for P_0 (e.g., because granules from G_0 are much simpler than granules from G). This means that there are given algorithms $\mathcal{A}$, $\mathcal{A}_0$ for checking the membership in P and P_0, respectively, and the complexity of $\mathcal{A}_0$ is much lower than the complexity of the algorithm $\mathcal{A}$. Under the above assumptions it is useful to search for a *utility function* $Utility : G \longrightarrow G_0$ reducing the membership problem for P to the membership problem for P_0, i.e., a function with the following property: $g \in P$ if and only if $Utility(g) \in P_0$. Construction of the utility function satisfying the above condition may be not feasible. However, it becomes often feasible when we relax the binary membership relation $\in$ to the membership at least to a given degree (see, e.g., [20]). This example illustrates, the important property of utility functions. Usually, G_0 is a set of scalar values or it is assumed that some preference relation over G_0 is given.

Finally, we would like to add that in GC it is necessary to develop methods searching for approximation of risk degrees and utility function from data and domain knowledge analogously to approximation of complex concepts (see, e.g., [4]).

5 Conclusions

We have discussed the role of approximation spaces in construction of granules satisfying criteria expressed by the minimal length principle. The role of risk measures and utility functions in GC was illustrated. In our system searching for adaptive approximation of complex concepts, we plan to implement strategies based on the minimal length principle in GC, risk measures in GC, and utility functions in GC. This will also require developing methods for approximation of risk measures and utility functions.

Acknowledgements. The research of Andrzej Skowron was supported by the grant from Ministry of Scientific Research and Information Technology of the Republic of Poland. The research of Jaroslaw Stepaniuk was supported by the Białystok University of Technology Rector's Grant.

References

1. Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.): RSCTC 2002. LNCS (LNAI), vol. 2475, pp. 14–16. Springer, Heidelberg (2002)
2. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)

3. Bazan, J.G., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 418–427. Springer, Heidelberg (2006)
4. Bazan, J., Peters, J.F., Skowron, A.: Behavioral pattern identification through rough set modelling. In: Ślęzak, D., Wang, G., Szczuka, M., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 688–697. Springer, Heidelberg (2005)
5. Bazan, J., Osmólski, A., Skowron, A., Ślęzak, D., Szczuka, M., Wróblewski, J.: Rough set approach to the survival analysis. In: [1], pp. 522–529
6. Byrd, D.M., Cothern, C.R.: Introduction to Risk Analysis: A Systematic Approach to Science-Based Decision Making. ABS Group, Rockville, MD (2000)
7. Fishburn, P.C.: Utility Theory for Decision Making, Robert E. Krieger Publishing Co, Huntington, NY (1970)
8. Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton, NJ, 1944 sec.ed (1947)
9. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1) 3–27 (2007), Rough sets: Some extensions. Information Sciences 177(1) 28–40 (2007), Rough sets and Boolean reasoning. Information Sciences 177(1) 41–73 (2007)
10. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Springer, Berlin (2004)
11. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht (1991)
12. Rissanen, J.: Minimum-description-length principle. In: Kotz, S., Johnson, N. (eds.) Encyclopedia of Statistical Sciences, pp. 523–527. John Wiley & Sons, New York (1985)
13. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
14. Skowron, A., Stepaniuk, J.: Information granules: Towards foundations of granular omputing. International Journal of Intelligent Systems 16(1), 57–86 (2001)
15. Skowron, A., Stepaniuk, J.: Information granules and rough-neural computing. In: [15], pp. 43–84.
16. Skowron, A., Stepaniuk, J.: Constrained sums of information systems. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 300–309. Springer, Heidelberg (2004)
17. Skowron, A., Synak, P.: Complex patterns. Fundamenta Informaticae 60(1-4), 351–366 (2004)
18. Skowron, A., Synak, P.: Reasoning in information maps. Fundamenta Informaticae 59(2-3), 241–259 (2004)
19. Stepaniuk, J.: Knowledge discovery by application of rough set models. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems, pp. 137–233. Physica–Verlag, Heidelberg (2000)
20. Stepaniuk, J., Skowron, A., Peters, J., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)
21. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae 53(3-4), 365–390 (2002)
22. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)