
Foreword

It is my great pleasure to welcome a new book on rough sets edited by the prominent scientists Sankar K. Pal, Lech Polkowski, and Andrzej Skowron. The objective of the book is to explore a new paradigm in soft computing: granulation of knowledge. This paradigm has a long history that is related not only to computing and information but also to its sources in the long-lasting controversy between the continuous versus the discrete — of utmost importance to the foundation of mathematics and physics. The idea has been revived recently by Prof. Zadeh in the context of computation, under the name “computing with words.”

Granulation of information is inherently associated with the concept of rough sets and is derived from the indiscernibility relation generated by data. Thus, the paradigm has its natural mathematical model in rough set theory that can be used theoretically and practically to support this approach.

The book stresses a newly emerging approach to the problem of knowledge granulation, called rough-neurocomputing. The approach combines rough and fuzzy set theory, neural networks, and evolutionary computing and is meant to be used to study and explore the granular structure of complex information. The book covers a very wide spectrum of topics related to its main paradigm and gives fair state-of-the-art information about the subject. It can be recommended to all those working in this fascinating area of research.

The authors and the editors deserve the highest appreciation for their outstanding work.

Warsaw, November 25, 2002

*Zdzisław Pawlak**

*Member of the Polish Academy of Sciences, Professor in the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice, Poland and in the University of Information Technology and Management, Warsaw, Poland.

Foreword

Edited by Profs. S.K. Pal, L. Polkowski and A. Skowron, "Rough-Neurocomputing: Techniques for Computing with Words," or RNC for short, is a work whose importance is hard to exaggerate. A collection of contributions by prominent experts in their fields, RNC addresses some of the most basic issues centering on the concept of information granularity.

In essence, information granularity is a concomitant of the bounded ability of sensory organs, and ultimately the brain, to resolve detail and store information. As a consequence, human perceptions are, for the most part, intrinsically imprecise. More specifically, perceptions are f-granular in the sense that: (a) the boundaries of perceived classes are fuzzy; and (b) the values of perceived attributes are granular, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality. In effect, information granulation may be viewed as a human way of achieving data compression. Furthermore, it plays a key role in implementation of the strategy of divide-and-conquer in human problem-solving.

Information granules may be crisp or fuzzy. For example, crisp granules of age are the intervals $[0, 1]$, $[1, 2]$, $[2, 3]$, ... , while the fuzzy intervals labeled young, middle-aged and old are fuzzy granules. Conventionally, information granules are assumed to be crisp. In fuzzy set theory and fuzzy logic, information granules are assumed to be fuzzy. Clearly, crisp granularity may be viewed as a special case of fuzzy granularity, just as a crisp set may be viewed as a special case of a fuzzy set.

Information granularity plays a key role in Prof. Pawlak's pioneering theory of rough sets (1982). At first glance, it may appear that the concepts of rough set and fuzzy set are closely related. In fact, this is not the case. A fuzzy set is a class with unsharp boundaries, whereas a rough set is a crisp set with imprecisely described boundaries. In recent years, however, the theory of rough sets has been extended in many directions, leading to the concepts of fuzzy rough set and rough fuzzy set. I should like to suggest that rough sets be called Pawlak sets in honor of Prof. Pawlak.

An extension of rough set theory which is developed in RNC relates to what is referred to as rough-neurocomputing – an important extension which generalizes neurocomputing, opening the door to many new applications that are described in

VIII

Parts III and IV of the book, which deal, respectively, with application areas and case studies.

More broadly, RNC extends the theory of Pawlak sets to granular computing and, more particularly, to computing with words and perceptions. In what follows, I will take the liberty of focusing my comments on the extension to computing with words and perceptions (CWP).

In contrast to computing with numbers (CN), the objects of computation in CWP are words, propositions and perceptions described in a natural language. In science, there is a deep-seated tradition of striving for the ultimate in rigor and precision. Words are less precise than numbers. Why and where, then, should words be used in preference to numbers?

There are three principal rationales. First, when the available information is not precise enough to justify the use of numbers. Second, when there is a tolerance for imprecision which can be exploited to achieve tractability, robustness and low solution cost. And third, when the expressive power of words is higher than the expressive power of numbers. In CWP, what is employed in this instance is Precisiated Natural Language (PNL).

In large measure, the importance of CWP 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. There is an enormous literature on perceptions, spanning psychology, linguistics, philosophy and other fields. But what cannot be found in this literature is a theory in which perceptions are objects of computation.

A key idea in CWP is that of dealing with perceptions not directly but through their descriptions in a natural language. In this way, computation with perceptions is reduced to computation with words and propositions drawn from a natural language. This is where Precisiated Natural Language (PNL) enters into the picture.

More specifically, as was noted earlier, perceptions are intrinsically imprecise. More concretely, perceptions are f-granular. F-granularity of perceptions is passed on to their descriptions in a natural language. The implication is that to be able to compute with perceptions it is necessary to be able to compute with f-granular propositions. This is what conventional predicate-logic-based meaning-representation methods cannot do.

The ability of PNL to deal with f-granularity derives from a key idea. Specifically

in PNL, the meaning of a proposition, p , drawn from a natural language, NL, is represented as a generalized constraint on a variable which is implicit in p . More concretely, the meaning of p is precisiated through translation of p into a precisiation language. In the case of PNL, the precisiation language is the Generalized Constraint Language (GCL). Thus, the elements of GCL are generalized constraints. One of the generalized constraints is the Pawlak set constraint. This is the link between PNL and Prof. Pawlak's theory of rough sets.

In RNC, PNL is not an object of explicit exposition or discussion. But the concepts and techniques described in RNC have a direct bearing on computing with words. In this perspective, RNC lays the groundwork for PNL and PNL-based methods in computing with words and perceptions.

The concepts and techniques described in RNC point to a new direction in information processing and decision analysis—a direction in which the concept of information granularity plays a pivotal role. One cannot but be greatly impressed by the variety of issues and problems which are addressed in RNC. Clearly, rough-neurocomputing in combination with computing with words provides new and powerful tools for the conception, design and utilization of intelligent systems. The editors, the contributors and the publisher, Springer-Verlag, have produced a book that excels in all respects. They deserve our thanks and congratulations.

Berkeley, November 25, 2002

*Lotfi A. Zadeh**

*Professor in the Graduate School and Director, Berkeley Initiative in Soft Computing (BISC), Computer Science Division and the Electronics Research Laboratory, Department of EECS, University of California, Berkeley, CA 94720-1776, USA.

Preface

Lotfi Zadeh has recently pioneered a research area known as “computing with words” (CW) and explained the computational theory of perception (CTP). The objective of this new research is to build foundations for future intelligent machines and information systems that perform computations on words (names of concepts in natural language) rather than on numbers. The main notion of this approach is related to information granulation. Information granules are understood as clumps of objects (points) that are drawn together by indistinguishability, similarity, or functionality. Information granulation and methods for constructing relevant information granules are crucial for exploiting tolerance of imprecision and uncertainty to achieve tractability, robustness, and low production costs for future intelligent systems. Several approaches to the formulation of information granule concepts have been proposed so far. Granular computing (GC) is one such computing paradigm based on information granule calculi.

This book is dedicated to a newly emerging approach to CW and CTP, called rough-neurocomputing (RNC). In RNC, computations are usually performed on information granules. The foundations of RNC are built in soft computing frameworks comprising synergistic hybridization of rough set theory, rough mereology, fuzzy set theory, neural networks, and evolutionary computing.

Any approach to information granulation should make it possible to define complex information granules (e.g., granules relevant to spatial and temporal reasoning in autonomous systems). The following facts are especially important in the process of complex information granule construction:

- (i) Any concept of an information granule considered should reflect and describe its inherent vagueness in formal terms.
- (ii) Target granules cannot be constructed (induced) directly from input granules but rather are constructed in a many-stage process.
- (iii) The schemes of new granule construction also interpreted as approximate reasoning schemes (AR schemes) should be robust with respect to input granule deviations.
- (iv) For real-life applications, adaptive AR schemes become a necessity.

To deal with vagueness, one can adopt soft computing methods developed by fuzzy or rough set theoretical approaches and their different integrations. Information

granules are represented in this book by parameterized formulas over signatures of relational systems, called information granule systems. In such systems, we emphasize the role of inclusion and closeness of information granules, to a degree, on the basis of a rough mereological approach. Information granule systems are relational systems over which information granules can be interpreted. Target information granule systems represent approximation spaces. These spaces are generalizations of approximation spaces used in rough set theory. The second aspect in the above list is related to several issues, such as reasoning from measurements to perception, multilayered learning of concept approximations, and fusion of information coming from different sources. Methods of searching for AR schemes are investigated using rough mereological tools. In general, they return hierarchical schemes for constructing information granules.

Among important topics discussed in this book are different methods for specifying operations on information granules. Such operations are used for constructing relevant information granules from experimental data and background knowledge. These are the basic components of methods aimed at constructing hierarchical schemes of information granules. In the more general case, we deal with network-like structures, transforming and exchanging information granules or information about them. Such networks are called rough-neural networks (RNN), and they are a generalization of AR schemes represented by trees. One of the important aspects of the approach to information granule calculi, as described in this book, is its strong connection with multiagent systems. For example, constructing AR schemes is closely related to ideas of cooperation and conflict resolution in multiagent systems. Moreover, agents exchanging information granules and information about them perform operations on information granules represented in languages they “understand.” Hence, granules received in argument ports by a given agent as arguments of his/her operation should be approximated by properly tuning the approximation spaces. These spaces create interfaces between agents. Rough-neural networks are analogous to neural networks. They perform computations on information granules. The process of tuning parameters in rough-neural networks corresponds to adjusting of weights in neural networks. Parameters of rough-neural networks are related, in particular, to approximation spaces used in their construction. By tuning these parameters, one can expect to induce relevant target information granules. The relevance of target information granules can be measured by carefully selected criteria. For example, one can use measures based on inclusion (closeness) degrees of granules and/or some other measures related to their sizes.

The methods that induce rough-neural networks using rough sets in combination with other soft computing tools create a core for RNC. These methods show that connectionist and symbolic approaches can work complementarily and not competitively. For example, derived AR schemes are aimed at representing patterns sufficiently included in target complex concepts. The structures of AR schemes are derived using both approaches. Symbolic reasoning is used to get the structure of

schemes from data and background knowledge. A connectionist approach is used for tuning the parameters of such structures.

One of the most important research directions of RNC concerns relationships between information granules and words or phrases in a natural language. Investigating such relationships leads to methods for inducing (from data and background knowledge) rough-neural networks approximating reasoning schemes in natural language. This creates a strong link between the approach presented and the approaches directed toward operating and reasoning with perception-based information, such as CW and CTP. In particular, one can interpret rough-neural networks that approximate reasoning schemes in natural language as schemes of approximate reasoning from measurement to perception.

Formally, the robustness of a rough-neural network means that such a network produces a higher-order information granule, which is a clump (e.g., a set) of information granules rather than a single granule. The inputs for such networks are clumps of deviations (up to an acceptable degree) from some standard input information granules to networks. In general, rough-neural networks should be robust with respect to different sources of noise. One of the basic research directions in RNNs that we have in mind in this book is developing strategies for adaptive rough-neural network construction.

This book presents recent advances made in RNC by researchers from different countries. In the first part, the foundations of information granule calculus are discussed. Such a calculus based on a rough mereological approach creates a basis for synthesizing and analyzing rough-neural networks. Recent results on the foundations of RNC are included. The reader can find an introduction to a rough set theoretical approach together with an explanation of why a generalization of approximation spaces, used so far in rough set theory, has been introduced. Close relationships of rough set approaches with multivalued (especially with three valued) logic are also presented. The second part shows how different integrations of soft computing tools can help to induce information granules. Special emphasis is given to methods based on hybridization of rough sets with neural techniques. Such techniques are crucial for developing RNC methods in synthesizing complex information granules. The reader can find how different approaches to constructing information granules based on fuzzy sets, rough sets, rough fuzzy sets, and other soft computing paradigms can work in synergy. Moreover, the way different approaches can be combined with symbolic approaches like nonmonotonic reasoning, deductive databases, and logic programming is presented. Methods for constructing interfaces between experimental knowledge and symbolic knowledge are discussed. The necessity of using statistical tools in information granule construction is underlined. Selected application areas for RNC and CW are discussed in the third part. Modeling methods for complex sociological situations or sociological games and semantic models for biomedical reasoning are included. Finally, the last part of the book con-

sists of several case studies illustrating recent developments based on RNC. This includes problems in signal analysis, medical data analysis, and pattern recognition.

It is worthwhile mentioning that from a logical point of view, research in RNC is closely related to the pragmatic aspects of natural language. As an example of such a pragmatic aspect investigated in RNC, one can consider the attempts made to understand concepts by means of experimental data and reasoning schemes in natural language. Another example would be communication between agents using different languages for information granule representation.

We do hope that this self-contained book will encourage students and researchers to join a fascinating journey toward building intelligent systems.

The book has been very much enriched thanks to forewords written by Prof. Zdzisław Pawlak, founder of rough set theory and Prof. Lotfi A. Zadeh, founder of fuzzy set theory and more recently of new paradigms CW and CTP. We are honored to have their contributions and we would like to express our special gratitude to both of them.

We are grateful to our colleagues who prepared excellent chapters for this volume. Our special thanks go to: Prof. Leonard Bolc, a member of advisory board of the series; to Ms. Ingeborg Mayer and Ms. Vette-Guillaume of Springer for their help with the production of this volume; to Dr. Sinh Hoa Nguyen, Dr. Hung Son Nguyen, and Ms. Grażyna Domańska for preparing the L^AT_EX version of the book. We are grateful to the copy editors of Springer for their devoted help in producing this volume, and especially to Mr. Ronan Nugent.

Our work was helped by grants from the State Committee for Scientific Research (KBN) of the Republic of Poland, Nos. 8T11C 02519 (A. Skowron) and 8T11C02417 (L. Polkowski). A. Skowron was, moreover, supported by a grant from the Wallenberg Foundation.

Warsaw, November 2002

*Sankar K. Pal
Lech Polkowski
Andrzej Skowron*

Chapter 1

Elementary Rough Set Granules: Toward a Rough Set Processor

Zdzisław Pawlak

Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Bałtycka 5, 44-100 Gliwice, Poland

and

University of Information Technology and Management,
Newelska 6, 01-477 Warsaw, Poland

zpw@ii.pw.edu.pl

Summary. In this chapter, the basics of the rough set approach are presented, and an outline of an exemplary processor structure is given. The organization of a simple processor is based on elementary rough set granules and dependencies between them. The rough set processor (RSP) is meant to be used as an additional fast classification unit in ordinary computers or as an autonomous learning machine. In the latter case, the RSP can be regarded as an alternative to neural networks.

1 Introduction

Rough set theory [4] has proved its effectiveness in drawing conclusions from data [6]. However, to take full advantage of the theory in data analysis, adequate processor organization is necessary. The architecture of such processors was proposed first in [4]. In this chapter, another proposal for rough set processor organization is presented.

Rough-set-based data analysis starts from a decision table, which is a data table. The columns of a decision table are labeled with attributes; the rows are labeled with objects of interest; and attribute values are entered in the data cells of the table. Attributes of the decision table are divided into two disjoint groups called condition and decision attributes, respectively. Each row of a decision table induces a decision rule, which specifies the decision (action, results, outcome, etc.) if some conditions are satisfied. If a decision rule uniquely determines a decision in terms of conditions, the decision rule is certain. Otherwise the decision rule is uncertain. Decision rules are closely connected with approximations, which are basic concepts of rough set theory. Roughly speaking, certain decision rules describe the lower approximation of decisions in terms of conditions, whereas uncertain decision rules refer to the upper approximation of decisions.

Two conditional probabilities, called the certainty and the coverage coefficient, are

associated with every decision rule. The certainty coefficient expresses the conditional probability that an object belongs to the decision class specified by the decision rule, given that it satisfies the conditions of the rule. The coverage coefficient gives the conditional probability of the reasons for a given decision.

It turns out that the certainty and coverage coefficients satisfy Bayes' theorem. This gives new insight into the interpretation of Bayes' theorem, showing that Bayes' theorem can be used differently for drawing conclusions from data than the use offered by classical Bayesian inference philosophy [5].

This idea is at the foundation of rough set processor organization. In this chapter, the basics of rough set theory are presented, and an outline of an exemplary processor structure is given. The rough set processor is meant to be used as a "rough" classifier, or as a learning machine, and can be regarded as an alternative to neural networks.

2 Information Systems and Decision Tables

In this section we define the basic concept of rough set theory: information systems. The rudiments of rough set theory can be found in [4, 6]. An information system is a data table whose columns are labeled with attributes, rows are labeled with objects of interest, and attribute values are entered in the data cells of the table.

Formally, the *information system* is a pair $S = (U, A)$, where U and A are nonempty finite sets called the *universe* of objects and the set of *attributes*, respectively, such that $a : U \rightarrow V_a$, where V_a is the set of all *values* of a , called the *domain* of a , for each $a \in A$. Any subset B of A determines a binary relation $I(B)$ on U , which will be called an *indiscernibility relation*, and is defined as follows:

$$(x, y) \in I(B) \quad \text{if and only if} \quad a(x) = a(y) \quad \text{for every} \quad a \in B,$$

where $a(x)$ denotes the value of the attribute a for the element x . Obviously $I(B)$ is an equivalence relation. The family of all equivalence classes of $I(B)$, i.e., a partition determined by B , will be denoted by $U/I(B)$, or simply by U/B ; an equivalence class of $I(B)$, i.e., the block of the partition U/B containing x will be denoted by $B(x)$.

If (x, y) belongs to $I(B)$, we will say that x and y are *B-indiscernible objects* (*indiscernible with respect to B*). Equivalence classes of the relation $I(B)$ (or blocks of the partition U/B) are referred to as *B-elementary sets* or *B-elementary granules*.

If we distinguish in the information system two disjoint classes of attributes, called *condition* and *attribute decision*, respectively, then the system will be called a *decision table* and will be denoted by $S = (U, C, D)$, where C and D are disjoint sets of condition and decision attributes, respectively, and $C \cup D = A$. $C(x)$ and $D(x)$ will

be referred to as the condition class and the decision class induced by x , respectively.

Thus the decision table describes decisions (actions, results etc.) taken when some conditions are satisfied. In other words, each row of the decision table specifies a decision rule that determines decisions in terms of conditions.

An example of a simple decision table is shown in Table 1. In the table, *age*, *sex*, and *profession* are condition attributes, whereas *disease* is the decision attribute.

Table 1. Decision table

<i>Decision rule</i>	<i>Age</i>	<i>Sex</i>	<i>Profession</i>	<i>Disease</i>
1	<i>Old</i>	<i>Male</i>	<i>Yes</i>	<i>No</i>
2	<i>Med.</i>	<i>Female</i>	<i>No</i>	<i>Yes</i>
3	<i>Med.</i>	<i>Male</i>	<i>Yes</i>	<i>No</i>
4	<i>Old</i>	<i>Male</i>	<i>Yes</i>	<i>Yes</i>
5	<i>Young</i>	<i>Male</i>	<i>No</i>	<i>No</i>
6	<i>Med.</i>	<i>Female</i>	<i>No</i>	<i>No</i>

The table contains data on the relationship among age, sex, and profession and a certain vocational disease. Decision tables can be simplified by removing superfluous attributes and attribute values, but we will not consider this issue in this chapter.

3 Decision Rules

In what follows, we will describe decision rules more exactly. Let $S = (U, C, D)$ be a decision table. Every $x \in U$ determines a sequence

$$c_1(x), \dots, c_n(x), d_1(x), \dots, d_m(x),$$

where $\{c_1, \dots, c_n\} = C$ and $\{d_1, \dots, d_m\} = D$. The sequence will be called a *decision rule induced by x* (in S) and denoted by

$$c_1(x), \dots, c_n(x) \rightarrow d_1(x), \dots, d_m(x),$$

or in short, $C \xrightarrow{x} D$. The number $\text{supp}_x(C, D) = |A(x)| = |C(x) \cap D(x)|$ will be called the *support* of the decision rule $C \xrightarrow{x} D$, and the number,

$$\sigma_x(C, D) = \frac{\text{supp}_x(C, D)}{|U|},$$

will be referred to as the *strength* of the decision rule $C \xrightarrow{x} D$, where $|X|$ denotes the cardinality of X . Another decision table is shown in Table 2. This decision table

can be understood as an abbreviation of a bigger decision table containing 1100 rows. Support of the decision rule means the number of identical decision rules in the original decision table.

Table 2. Support and strength

Decision rule	Age	Sex	Profession	Disease	Support	Strength
1	Old	Male	Yes	No	200	0.18
2	Med.	Female	No	Yes	70	0.06
3	Med.	Male	Yes	No	250	0.23
4	Old	Male	Yes	Yes	450	0.41
5	Young	Male	No	No	30	0.03
6	Med.	Female	No	No	100	0.09

With every decision rule $C \xrightarrow{x} D$, we associate the *certainty factor* of the decision rule, denoted $cer_x(C, D)$ and defined as follows:

$$cer_x(C, D) = \frac{|C(x) \cap D(x)|}{|C(x)|} = \frac{supp_x(C, D)}{|C(x)|} = \frac{\sigma_x(C, D)}{\pi[C(x)]},$$

where $\pi[C(x)] = \frac{|C(x)|}{|U|}$.

The certainty factor may be interpreted as a conditional probability that y belongs to $D(x)$, given y belongs to $C(x)$, symbolically $\pi_x(D|C)$. If $cer_x(C, D) = 1$, then $C \xrightarrow{x} D$ will be called a *certain decision rule*; if $0 < cer_x(C, D) < 1$, the decision rule will be referred to as an *uncertain decision rule*. We will also use a *coverage factor* of the decision rule, denoted $cov_x(C, D)$ [7] defined as

$$cov_x(C, D) = \frac{|C(x) \cap D(x)|}{|D(x)|} = \frac{supp_x(C, D)}{|D(x)|} = \frac{\sigma_x(C, D)}{\pi[D(x)]},$$

where $D(x) \neq \emptyset$ and $\pi[D(x)] = \frac{|D(x)|}{|U|}$. Similarly,

$$cov_x(C, D) = \pi_x(C|D).$$

If $C \xrightarrow{x} D$ is a decision rule, then $D \xrightarrow{x} C$ will be called an *inverse decision rule*. Inverse decision rules can be used to give *explanations (reasons)* for a decision.

4 Approximation of Sets

Suppose we are given an information system $S = (U, A)$, $X \subseteq U$, and $B \subseteq A$. Our task is to describe the set X in terms of attribute values from B . To this end, we

define two operations assigning to every $X \subseteq U$ two sets $B_*(X)$ and $B^*(X)$ called the *B-lower* and the *B-upper approximation* of X , respectively, and defined as

$$B_*(X) = \bigcup_{x \in U} \{B(x) : B(x) \subseteq X\} \text{ and}$$

$$B^*(X) = \bigcup_{x \in U} \{B(x) : B(x) \cap X \neq \emptyset\}.$$

Hence, the *B-lower* approximation of a set is the union of all *B-granules* that are included in the set, whereas the *B-upper* approximation of a set is the union of all *B-granules* that have a nonempty intersection with the set. The set

$$BN_B(X) = B^*(X) - B_*(X),$$

will be referred to as the *B-boundary region* of X .

If the boundary region of X is the empty set, i.e., $BN_B(X) = \emptyset$, then X is *crisp* (*exact*) with respect to B ; in the opposite case, i.e., if $BN_B(X) \neq \emptyset$, X is referred to as *rough* (*inexact*) with respect to B .

There is an interesting relationship between approximations and decision rules. Let $C \xrightarrow{x} D$ be a decision rule. The set,

$$\bigcup_{y \in D(x)} \{C(y) : C(y) \subseteq D(x)\},$$

is equal to the lower approximation of the decision class $D(x)$, by condition classes $C(y)$, whereas the set,

$$\bigcup_{y \in D(x)} \{C(y) : C(y) \cap D(x) \neq \emptyset\},$$

is equal to the upper approximation of the decision class by condition classes $C(y)$.

That means that approximations and decision rules are two different methods for expressing imprecision. Approximations are better suited to expressing topological properties of decision tables, whereas rules describe hidden patterns in data in a simple way.

5 Probabilistic Properties of Decision Tables

Decision tables have important probabilistic properties that are discussed next.

Let $C \xrightarrow{x} D$ be a decision rule, and let $\Gamma = C(x)$ and $\Delta = D(x)$. Then the following properties are valid:

$$\sum_{y \in \Gamma} cer_y(C, D) = 1, \tag{1}$$

$$\sum_{y \in \Delta} cov_y(C, D) = 1, \quad (2)$$

$$\pi[D(x)] = \sum_{y \in \Gamma} cer_y(C, D) \cdot \pi[C(y)] = \sum_{y \in \Gamma} \sigma_y(C, D), \quad (3)$$

$$\pi[C(x)] = \sum_{y \in \Delta} cov_y(C, D) \cdot \pi[D(y)] = \sum_{y \in \Delta} \sigma_y(C, D), \quad (4)$$

$$cer_x(C, D) = \frac{cov_x(C, D) \cdot \pi[D(x)]}{\sum_{y \in \Delta} cov_y(C, D) \cdot \pi[D(y)]} = \frac{\sigma_x(C, D)}{\pi[C(x)]}, \quad (5)$$

$$cov_x(C, D) = \frac{cer_x(C, D) \cdot \pi[C(x)]}{\sum_{y \in \Gamma} cer_y(C, D) \cdot \pi[C(y)]} = \frac{\sigma_x(C, D)}{\pi[D(x)]}, \quad (6)$$

that is, any decision table satisfies (1) – (6). Observe that (3) and (4) refer to the well-known *total probability theorem*, whereas (5) and (6) refer to *Bayes' theorem*. Thus, to compute the certainty and coverage factors of decision rules according to formula (5) and (6), it is enough to know only the strength (support) of all decision rules. The strength of decision rules can be computed from data or can be a subjective assessment.

These properties will be used as a basis for the rough set processor organization. The certainty and coverage factors for the decision table presented in Table 2 are shown in Table 3.

Table 3. Certainty and coverage factors

<i>Decision rule</i>	<i>Strength</i>	<i>Certainty</i>	<i>Coverage</i>
1	0.18	0.31	0.34
2	0.06	0.40	0.13
3	0.23	1.00	0.43
4	0.41	0.69	0.87
5	0.03	1.00	0.06
6	0.09	0.60	0.17

Let us observe that, according to formulas (5) and (6), the certainty and coverage factors can be computed employing only the strength of decision rules. In Table 2, decision rules 3 and 5 are certain, whereas the remaining decision rules are uncertain. This means that middle-aged males having a profession and young males not having a profession are certainly healthy. Old males having a profession are most probably ill (probability = .69), and middle-aged females not having a profession

are most probably healthy (probability = .60).

The inverse decision rules say that healthy persons are most probably middle-aged males having a profession (probability = .43) and ill persons are most probably old males having a profession (probability = .87).

6 Decision Tables and Flow Graphs

With every decision table, we associate a *flow graph*, i.e., a directed acyclic graph defined as follows: to every decision rule $C \xrightarrow{x} D$, we assign a *directed branch* x connecting the *input node* $C(x)$ and the *output node* $D(x)$. The strength of the decision rule represents a *throughflow* of the corresponding branch. The throughflow of the graph is governed by formulas (1) – (6).

Formulas (1) and (2) say that the outflow of an input node or an output node is equal to their respective inflows. Formula (3) states that the outflow of the output node amounts to the sum of its inflows; whereas formula (4) says that the sum of the outflows of the input node equals its inflow. Finally, formulas (5) and (6) reveal how throughflow in the flow graph is distributed between its inputs and outputs. The flow graph associated with the decision table presented in Table 2 is shown in Fig. 1.

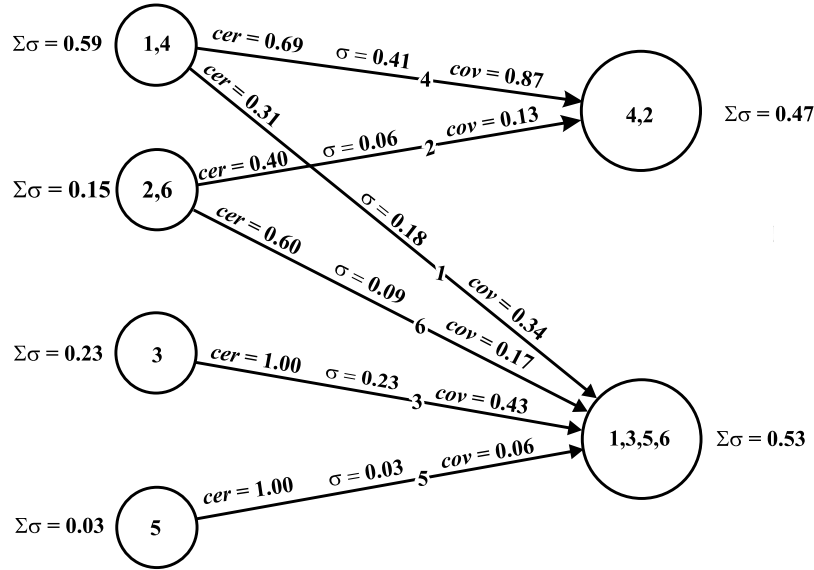


Fig. 1. Flow graph

The application of flow graphs to represent decision tables gives very clear insight into the decision process. The classification of objects in this representation boils

down to finding the maximal output flow in the flow graph; whereas, the explanation of the decisions is connected to the maximal input flow associated with the given decision (see also [1] and [6]).

7 Rough Set Processor

To make the most of rough set theory in data analysis, a special microprocessor, the RSP, is necessary, to speed up the classification process. The RSP should perform operations pointed out by the flow graph of a decision table, that is, first it should compute its strengths from the supports of decision rules and afterward compute the certainty and coverage factors of all decision rules. Finally, the maximal certainty (coverage) factor should be computed, pointing out the most probable decision class (reason) for the classified object.

Many hardware implementations of this idea are possible. An example of a simplified RSP structure is depicted in Fig. 2.

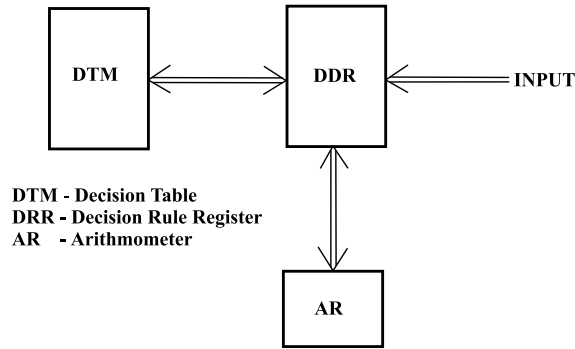


Fig. 2. RSP structure

The RSP consists of decision table memory (DTM), a decision rule register (DDR) and an arithmometer (AR). Decision rules are stored in the DTM. The word structure of the decision table memory is shown in Fig. 3.

<i>Condition</i>	<i>Decision</i>	<i>Support</i>	<i>Strength</i>	<i>Certainty</i>	<i>Coverage</i>
------------------	-----------------	----------------	-----------------	------------------	-----------------

Fig. 3. Word structure

At the initial state, only conditions, decisions, and support of each decision rule are given. Next, the strength of each decision rule is computed. Afterward, certainty and coverage factors are computed. Finally, the maximal certainty (coverage) factors are

ascertained. They will be used to define the most probable decision rules (inverse decision rules) induced by data. Let us also observe that the flow graph can easily be implemented as an analogue electrical circuit.

8 Conclusion

Rough-set-based data analysis consists of discovering hidden patterns in decision tables. It is shown that decision tables display basic probabilistic features; particularly, they satisfy the total probability theorem and Bayes' theorem. Moreover, rough set theory allows us to represent the above theorems in a very simple way using only the strengths of the decision rules. This property allows us to represent decision tables in the form of a flow graph and interpret the decision structure of the decision table as throughflow in the graph. The flow graph interpretation of decision tables can be employed as a basis for rough set processor organization.

References

1. M. Berthold, D.J. Hand. *Intelligent Data Analysis: An Introduction*. Springer, Heidelberg, 1999.
2. J. Łukasiewicz. Die Logischen Grundlagen der Wahrscheinlichkeitsrechnung. Kraków, 1913. In L. Borkowski, editor, *Jan Łukasiewicz: Selected Works*, North-Holland, Amsterdam and Polish Scientific Publishers, Warsaw, 16–63, 1970.
3. A. Kanasugi. A design of architecture for rough set processor. In *Proceedings of the International Workshop on Rough Set Theory and Granular Computing (RSTGC 2001)*, volume 5(1/2) of the *Bulletin of International Rough Set Society*, 201–204, 2001.
4. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.
5. Z. Pawlak. Rough sets and decision algorithms. In *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC 2000)*, LNAI 2005, 30–45, Springer, Berlin, 2001.
6. A. Skowron. Rough sets in KDD (plenary talk at IFIP 2000). In *Proceedings of the Conference on Intelligent Information Processing (IIP 2000)*, 1–17, Publishing House of Electronic Industry, Beijing, 2001.
7. D. Ślęzak. Data model based on approximate bayesian network. In B. Bouchon-Meunier, J. Gutierrez-Rios, L. Magdalena, R.R. Yager, editors, *Technologies for Constructing Intelligent Systems 2: Tools*, 313–326, Physica, Heidelberg, 2002.
8. S. Tsumoto, H. Tanaka. Discovery of functional components of proteins based on PRIMEROSE and domain knowledge hierarchy. In *Proceedings of the 3rd Workshop on Rough Sets and Soft Computing (RSSC'94)*, 280–285, San Jose, CA, 1995.



<http://www.springer.com/978-3-540-43059-9>

Rough-Neural Computing
Techniques for Computing with Words
Pal, S.K.; Polkowski, L. (Eds.)
2004, XXV, 736 p., Hardcover
ISBN: 978-3-540-43059-9