

Interpreting Low and High Order Rules: A Granular Computing Approach

Yiyu Yao, Bing Zhou and Yaohua Chen

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {yyao, zhou200b, chen115y}@cs.uregina.ca

Abstract. The main objective of this paper is to provide a granular computing based interpretation of rules representing two levels of knowledge. This is done by adopting and adapting the decision logic language for granular computing. The language provides a formal method for describing and interpreting conditions in rules as granules and rules as relationships between granules. An information table is used to construct a concrete granular computing model. Two types of granules are constructed from an information table. They lead to two types of rules called low order and high order rules. As examples, we examine rules in the standard rough set analysis and dominance-based rough set analysis.

Keywords. Low order rules, high order rules, granular computing, dominance-based rough set analysis

1 Introduction

Rules are a commonly used form for representing knowledge. Two levels of knowledge may be expressed in terms of low and high order rules, respectively [16]. A low order rule expresses connections between attribute values of the same object. Classification rules are a typical example of low order rules. For example, a classification rule may state that “if the Hair color is blond and Eye color is blue, then the Class is +.” A high order rule expresses connections of different objects in terms of their attribute values. Functional dependencies are a typical example of high order rules. For example, a functional dependency rule may state that “if two persons have the same Hair color, then they will have the same Eye color.” The notion of a high order rule is also related to relational learning, in which a k -ary predicate is used to define a relation between k objects [10]. For simplicity, in this paper we only consider binary relations.

In rough set analysis [6–8], a decision logic language DL is used to build conditions in low order rules and to interpret these conditions as subsets (i.e., granules) of objects. This language is referred to as \mathcal{L}_0 [16]. In order to describe high order rules, an extended language \mathcal{L}_1 is introduced [16]. In \mathcal{L}_1 , conditions are interpreted in terms of a set of object pairs. The two languages share the same syntactic rules, with two semantic interpretations. That is, their main differences lie in the different interpretations of atomic formulas. It is therefore possible to introduce a common language.

In this paper, we propose a decision logic language \mathcal{L} for granular computing. Instead of expressing the atomic formulas by a particular concrete type of conditions, we treat them as primitive notions that can be interpreted differently. This flexibility enables us to describe different types of rules. The language is interpreted in the Tarski's style through the notion of a model and satisfiability. The model is a non-empty domain consisting of a set of individuals. An individual satisfies a formula if the individual has the properties as specified by the formula. A concept is therefore jointly defined by a pair consisting of the intension of the concept, a formula of the language, and the extension of the concept, a subset of the model.

As illustrative examples to show the usefulness of the proposed language, we analyze rules in the standard rough set analysis [6–8] and dominance-based rough set analysis [2–4, 11, 15].

2 A Decision Logic Language for Granular Computing

By extracting the high-level similarities of the decision logic languages DL , \mathcal{L}_0 , and \mathcal{L}_1 , we propose a logic language \mathcal{L} for granular computing.

The language \mathcal{L} is constructed from a finite set of atomic formulas, denoted by $\mathcal{A} = \{p, q, \dots\}$. Each atomic formula may be interpreted as representing one piece of basic knowledge. The physical meaning of atomic formulas becomes clearer in a particular application. In general, an atomic formula corresponds to one particular property of an individual under discussion. The construction of atomic formulas is an essential step of knowledge representation. The set of atomic formulas provides a basis on which more complex knowledge can be represented. Compound formulas can be built recursively from atomic formulas by using logic connectives. If ϕ and ψ are formulas, then so are $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$.

The semantics of the language \mathcal{L} can be defined in the Tarski's style through the notion of a model and satisfiability. The model is a nonempty domain consisting of a set of individuals, denoted by $M = \{x, y, \dots\}$. The meaning of formulas can be given recursively. For an atomic formula p , we assume that an individual $x \in M$ either satisfies p or does not satisfy p , but not both. For an individual $x \in M$, if it satisfies an atomic formula p , we write $x \models p$, otherwise, we write $x \not\models p$. The satisfiability of an atomic formula by individuals of M is viewed to be the basic knowledge describable by the language \mathcal{L} . An individual satisfies a formula if the individual has the properties as specified by the formula. Let ϕ and ψ be two formulas, the satisfiability of compound formulas is defined as follows:

- (1). $x \models \neg\phi$ iff $x \not\models \phi$,
- (2). $x \models \phi \wedge \psi$ iff $x \models \phi$ and $x \models \psi$,
- (3). $x \models \phi \vee \psi$ iff $x \models \phi$ or $x \models \psi$,
- (4). $x \models \phi \rightarrow \psi$ iff $x \models \neg\phi \vee \psi$,
- (5). $x \models \phi \leftrightarrow \psi$ iff $x \models \phi \rightarrow \psi$ and $x \models \psi \rightarrow \phi$.

In order to emphasize the roles played by the set of atomic formulas \mathcal{A} and the set of individuals M , we also rewrite the language \mathcal{L} as $\mathcal{L}(\mathcal{A}, M)$.

The construction of the set of atomic formulas and the model M depends on a particular application. For modeling different problems, we may choose different sets of atomic formulas and models. The language \mathcal{L} therefore is flexible and enables us to describe a variety of problems.

With the notion of satisfiability, one can introduce a set-theoretic interpretation of formulas of the language \mathcal{L} . If ϕ is a formula, the meaning of ϕ in the model M is the set of individuals defined by [7]:

$$m(\phi) = \{x \in M \mid x \models \phi\}. \quad (1)$$

That is, $m(\phi)$ is the set of individuals satisfying a formula ϕ . This enables us to establish a correspondence between logic connectives and set-theoretic operators. Specifically, the following properties hold:

- (C1). $m(\neg\phi) = -m(\phi)$,
- (C2). $m(\phi \wedge \psi) = m(\phi) \cap m(\psi)$,
- (C3). $m(\phi \vee \psi) = m(\phi) \cup m(\psi)$,
- (C4). $m(\phi \rightarrow \psi) = -m(\phi) \cup m(\psi)$,
- (C5). $m(\phi \leftrightarrow \psi) = (m(\phi) \cap m(\psi)) \cup (-m(\phi) \cap -m(\psi))$,

where $-m(\phi) = M - m(\phi)$ denotes the set complement of $m(\phi)$.

In the study of concepts, many interpretations have been proposed and examined. The classical view regards a concept as a unit of thought consisting of two parts, i.e., the intension and extension of the concept [1, 13]. By using the language \mathcal{L} , we obtain a simple and precise representation of a concept in terms of its intension and extension. That is, a concept is defined by a pair $(m(\phi), \phi)$. The formula ϕ is the description of properties shared by individuals in $m(\phi)$, and $m(\phi)$ is the set of individuals satisfying ϕ . A concept is thus described jointly by its intension and extension. This formulation enables us to study concepts in a logic setting in terms of intensions and in a set-theoretic setting in terms of extensions.

The language \mathcal{L} provides a formal method for describing granules. Elements of a granule may be interpreted as instances of a concept, i.e., the extension of the concept. The formula is a formal description of the granule. In this way, the language \mathcal{L} only enables us to define certain subsets of M . For an arbitrary subset of M , we may not be able to construct a formula for it. In other words, depending on the set of atomic formulas, the language \mathcal{L} can only describe a restricted family of subsets of M .

3 Interpretation of Low and High Order Rules Using the Language \mathcal{L}

We interpret different types of rules of an information table as concrete applications to show the usefulness of the language \mathcal{L} .

3.1 Information Table

An information table provides a convenient way to describe a finite set of objects by a finite set of attributes [7]. Formally, an information table can be expressed as:

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

where

U is a finite nonempty set of objects called universe,
 At is a finite nonempty set of attributes,
 V_a is a nonempty set of values for $a \in At$,
 $\{R_a\}$ is a family of binary relations on V_a ,
 $I_a : U \rightarrow V_a$ is an information function.

Each information function I_a maps an object in U to a value of V_a for an attribute $a \in At$.

Our definition of an information table considers more knowledge and information about relationships between values of attributes. Each relation R_a can represent similarity, dissimilarity, or ordering of values in V_a [1]. The equality relation $=$ is only a special case of R_a . The rough set theory and the *DL* language use the trivial equality relation on attribute values [7].

Pawlak and Skowron [8] consider a more generalized notion of an information table. For each attribute $a \in At$, a relational structure \mathfrak{R}_a over V_a is introduced. Furthermore, a language can be defined based on the relational structures. A binary relation is a special case of relational structures. Thus, the discussion of this paper may be viewed as a special case of Pawlak and Skowron's formulation.

3.2 Low Order Rules

For interpreting low order rules, we construct a language by using U as the model M . That is, individuals of M are objects in the universe U . The set of atomic formulas are constructed as follows. With respect to an attribute $a \in At$ and an attribute value $v \in V_a$, an atomic formula of the language \mathcal{L} is denoted by (a, R_a, v) . An object $x \in U$ satisfies an atomic formula (a, R_a, v) if the value of x on attribute a is R_a -related to the value v , that is $I_a(x) R_a v$, we write:

$$x \models (a, R_a, v) \text{ iff } I_a(x) R_a v.$$

We denote the language for interpreting low order rules as $\mathcal{L}(\{(a, R_a, v)\}, U)$. The granule corresponding to the atomic formula (a, R_a, v) , namely, its meaning set, is defined as:

$$m(a, R_a, v) = \{x \in U \mid I_a(x) R_a v\}.$$

Granules corresponding to compound formulas are defined by Equation (1).

A low order rule can be derived according to the relationships between these granules. We can express rules in the form $\phi \Rightarrow \psi$ by using formulas of the language \mathcal{L} . For easy understanding, we reexpress the formula (a, R_a, v) in another form based on the definition of satisfiability of the atomic formulas. An example of a low order rule is given as:

$$\text{Low Order rule: } \bigwedge_{i=1}^n (I_{a_i}(x) R_{a_i} v_{a_i}) \Rightarrow \bigwedge_{j=1}^m (I_{d_j}(x) R_{d_j} v_{d_j}),$$

where $x \in U$, $v_{a_i} \in V_{a_i}$, $v_{d_j} \in V_{d_j}$, $a_i \in At$, and $d_j \in At$. For simplicity, we only use conjunction \wedge in the rule.

3.3 High Order Rules

For interpreting high order rules, we construct a language by using $U \times U$ as the model M . That is, individuals of M are object pairs in $U \times U$. The set of atomic formulas are constructed as follows. With respect to an attribute $a \in At$, an atomic formula of the language \mathcal{L} is denoted by (a, R_a) . A pair of objects $(x, y) \in U \times U$ satisfies an atomic formula (a, R_a) if the value of x is R_a -related to the value of y on the attribute a , that is, $I_a(x) R_a I_a(y)$. We write:

$$(x, y) \models (a, R_a) \text{ iff } I_a(x) R_a I_a(y).$$

For clarity, we denote the language as $\mathcal{L}(\{(a, R_a)\}, U \times U)$. The granule corresponding to the atomic formula (a, R_a) , i.e., the meaning set, is defined as:

$$m(a, R_a) = \{(x, y) \in U \times U \mid I_a(x) R_a I_a(y)\}.$$

Granules corresponding to the compound formulas are defined by Equation (1).

A high order rule expresses the relationships between these granules. An example of a high order rule is given as:

$$\text{High Order rule: } \bigwedge_{i=1}^n (I_{a_i}(x) R_{a_i} I_{a_i}(y)) \Rightarrow \bigwedge_{j=1}^m (I_{d_j}(x) R_{d_j} I_{d_j}(y)),$$

where $(x, y) \in U \times U$, $a_i \in At$, $d_j \in At$.

3.4 Quantitative Measures of Rules

The meanings and interpretations of a rule $\phi \Rightarrow \psi$ can be further clarified by using the extensions $m(\phi)$ and $m(\psi)$ of the two concepts. More specifically, we can define quantitative measures indicating the strength of a rule. A systematic analysis of probabilistic quantitative measures can be found in [14]. Two examples of probabilistic quantitative measures are [12]:

$$\text{accuracy}(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{|m(\phi)|}, \quad \text{coverage}(\phi \Rightarrow \psi) = \frac{|m(\phi \wedge \psi)|}{|m(\psi)|}, \quad (2)$$

Table 1. Rough Set Approaches for Studying Low and High Order Rules

Relation	Low Order Rule	High Order Rule	Method
R	$I_a(x)R_a v_a \Rightarrow I_d(x)R_d v_d$	$I_a(x)R_a I_a(y) \Rightarrow I_d(x)R_d I_d(y)$	Generalized Rough Set Analysis
$=$	$I_a(x) = v_a \Rightarrow I_d(x) = v_d$	$I_a(x) = I_a(y) \Rightarrow I_d(x) = I_d(y)$	Standard Rough Set Analysis
\succeq	$I_a(x) \succeq_a v_a \Rightarrow I_d(x) \succeq_d v_d$	$I_a(x) \succeq_a I_a(y) \Rightarrow I_d(x) \succeq_d I_d(y)$	Dominance-based Rough Set Analysis

where $|\cdot|$ denotes the cardinality of a set. The two measures are applicable to both low and high order rules. This demonstrates the flexibility and power of the language \mathcal{L} .

While the accuracy reflects the correctness of the rule, the coverage reflects the applicability of the rule. If $accuracy(\phi \Rightarrow \psi) = 1$, we have a strong association between ϕ and ψ . A smaller value of accuracy indicates a weak association. A higher coverage suggests that the relationships of more individuals can be derived from the rule. The accuracy and coverage are not independent of each other, one may observe a trade-off between accuracy and coverage. A rule with higher coverage may have a lower accuracy, while a rule with higher accuracy may have a lower coverage.

4 Rough Set Approaches on Rules

In this section, we use two rough set approaches [2–4, 6–8, 11, 15] as examples to illustrate the usefulness of the language \mathcal{L} . The basic results are summarized in Table 1. For comparison, we also include the results of generalized rough set analysis based on an arbitrary binary relation R_a on attribute values.

4.1 Standard Rough Set Analysis

The standard rough set analysis is based on the trivial equality relation on attribute values [6–8]. It is used for the extraction of rules for classification and attribute dependency analysis. By using the language \mathcal{L} , the standard rough set approach can be formulated as follows.

For low order rules, the language is given by $\mathcal{L}(\{(a, =, v)\}, U)$ with atomic formulas of the form of $(a, =, v)$. An object $x \in U$ satisfies an atomic formula $(a, =, v)$ if the value of x on attribute a is v , that is, $I_a(x) = v$. We write:

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

The granule corresponding to the atomic formula $(a, =, v)$ is:

$$m(a, =, v) = \{x \in U \mid I_a(x) = v\}.$$

The granule $m(a, =, v)$ is also referred to as the block defined by the attribute-value pair (a, v) [5]. Blocks correspond to the atomic formulas and are used to construct

Table 2. An Information Table

Object	Height	Hair	Eyes	Class
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	red	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

rules. Low order rules can be expressed based on the equality relation =. An example of a low order rule is:

$$\bigwedge_{i=1}^n (I_{a_i}(x) = v_{a_i}) \Rightarrow \bigwedge_{j=1}^m (I_{d_j}(x) = v_{d_j}),$$

where $x \in U$, $v_{a_i} \in V_{a_i}$, $v_{d_j} \in V_{d_j}$, $a_i \in At$, and $d_j \in At$.

For high order rules, the language is given by $\mathcal{L}(\{(a, =)\}, U \times U)$ with atomic formulas of the form of $(a, =)$. A pair of objects $(x, y) \in U \times U$ satisfies an atomic formula $(a, =)$ if the value of x equals to the value of y on attribute a , that is, $I_a(x) = I_a(y)$. We write:

$$(x, y) \models (a, =) \text{ iff } I_a(x) = I_a(y).$$

The granule corresponding to the atomic formula $(a, =)$ is:

$$m(a, =) = \{(x, y) \in U \times U \mid I_a(x) = I_a(y)\}.$$

High order rules can be expressed by using the equality relation =. An example of a high order rule is:

$$\bigwedge_{i=1}^n (I_{a_i}(x) = I_{a_i}(y)) \Rightarrow \bigwedge_{j=1}^m (I_{d_j}(x) = I_{d_j}(y)),$$

where $(x, y) \in U \times U$, $a_i \in At$, $d_j \in At$.

Example 1. Table 2 is an information table taken from [9]. Each object is described by four attributes. The column labeled by ‘‘Class’’ denotes an expert’s classification of the objects.

An example of a low order rule in this information table is:

$$LR_1 : (I_{\text{Hair}}(x) = \text{blond}) \wedge (I_{\text{Eyes}}(x) = \text{blue}) \Rightarrow (I_{\text{Class}}(x) = +).$$

That is, if an object has blond hair and blue eyes, then it belongs to class +. An example of a high order rule is:

$$HR_1 : (I_{\text{Height}}(x) = I_{\text{Height}}(y)) \wedge (I_{\text{Eyes}}(x) = I_{\text{Eyes}}(y)) \Rightarrow (I_{\text{Class}}(x) = I_{\text{Class}}(y)).$$

That is, if two objects have the same height and the same eye color, then they belong to the same class. By using the probabilistic quantitative measures, we have:

$$accuracy(LR_1) = 1, \quad coverage(LR_1) = 2/3.$$

The association between $(Hair, =, blond) \wedge (Eyes, =, blue)$ and $(Class, =, +)$ reaches the maximum value 1, and the applicability of the rule is also high. For rule HR_1 , we have:

$$accuracy(HR_1) = 7/11, \quad coverage(HR_1) = 7/17.$$

In this case, $(Height, =) \wedge (Eyes, =)$ does not tell us too much information about the overall objects classification in terms of both accuracy and coverage.

4.2 Dominance-based Rough Set Analysis

The dominance-based rough set analysis proposed by Greco, Matarazzo and Slowinski [2–4] is based on preference relations on attribute values. It is used for the extraction of rules for ranking and attribute dependency analysis. Several different types of rules are introduced in dominance-based rough set analysis. In what follows, we interpret two types of such rules by using the language \mathcal{L} , as demonstrated in [11, 15, 16].

For low order rules, the language is given by $\mathcal{L}(\{(a, \succeq_a, v)\}, U)$. The granule corresponding to the atomic formula (a, \succeq_a, v) is defined as:

$$m(a, \succeq_a, v) = \{x \in U \mid I_a(x) \succeq_a v\}.$$

Low order rules can be expressed by using preference relations. An example of a low order rule is:

$$\bigwedge_{i=1}^n (I_{a_i}(x) \succeq_{a_i} v_{a_i}) \Rightarrow \bigwedge_{j=1}^m (I_{d_j}(x) \succeq_{d_j} v_{d_j}),$$

where $x \in U$, $v_{a_i} \in V_{a_i}$, $v_{d_j} \in V_{d_j}$, $a_i \in At$, and $d_j \in At$.

For high order rules, the language is given by $\mathcal{L}(\{(a, \succeq_a)\}, U \times U)$. The granule corresponding to the atomic formula (a, \succeq_a) is defined as:

$$m(a, \succeq_a) = \{(x, y) \in U \times U \mid I_a(x) \succeq_a I_a(y)\}.$$

High order rules can also be expressed by using the preference relations. An example of a high order rule is:

$$\bigwedge_{i=1}^n (I_{a_i}(x) \succeq_{a_i} I_{a_i}(y)) \Rightarrow \bigwedge_{j=1}^m (I_{d_j}(x) \succeq_{d_j} I_{d_j}(y)).$$

where $(x, y) \in U \times U$, $a_i \in At$, $d_j \in At$.

Table 3. An Information Table with Preference Relations

Objects	Size	Warranty	Price	Weight	Overall
p_1	middle	3 years	\$200	heavy	1
p_2	large	3 years	\$300	very heavy	3
p_3	small	3 years	\$300	light	3
p_4	small	3 years	\$250	very light	2
p_5	small	2 years	\$200	very light	3

Example 2. Table 3, taken from [11], is an information table with preference relations on attribute values. It is a group of five products by five manufactures, each product is described by four attributes. The final ranking labeled by Overall may be determined by their market share of the products. The preference relations induce the following orderings:

$$\begin{aligned}
 \succ_{\text{Size}}: & \quad \text{small} \succ_{\text{Size}} \text{middle} \succ_{\text{Size}} \text{large}, \\
 \succ_{\text{Warranty}}: & \quad 3 \text{ years} \succ_{\text{Warranty}} 2 \text{ years}, \\
 \succ_{\text{Price}}: & \quad \$200 \succ_{\text{Price}} \$250 \succ_{\text{Price}} \$300, \\
 \succ_{\text{Weight}}: & \quad \text{very light} \succ_{\text{Weight}} \text{light} \succ_{\text{Weight}} \text{heavy} \succ_{\text{Weight}} \text{very heavy}, \\
 \succ_{\text{Overall}}: & \quad 1 \succ_{\text{Overall}} 2 \succ_{\text{Overall}} 3.
 \end{aligned}$$

An example of a low order rule in this information table is:

$$\text{LR}_2 : (I_{\text{Size}}(x) \succeq \text{middle}) \wedge (I_{\text{Warranty}}(x) \succeq 3 \text{ years}) \Rightarrow I_{\text{Overall}}(x) \succeq 2.$$

That is, if a product's size is greater than or equal to middle and warranty is greater than or equal to 3 years, then its overall ranking will be greater than or equal to 2. An example of a high order rule is:

$$\text{HR}_2 : (I_{\text{Size}}(x) \succeq I_{\text{Size}}(y)) \wedge (I_{\text{Price}}(x) \succeq I_{\text{Price}}(y)) \Rightarrow I_{\text{Overall}}(x) \succeq I_{\text{Overall}}(y).$$

That is, if one product's size is smaller than or the same as another product and the price is not higher, then this product's overall ranking will be greater than or equal to the other product. By using the quantitative measures, we have:

$$\text{accuracy}(\text{LR}_2) = 2/3, \quad \text{coverage}(\text{LR}_2) = 1.$$

There exists a strong association between the two concepts, and applicability of the rule reaches the highest level. Similarly, for rule HR_2 , we have:

$$\text{accuracy}(\text{HR}_2) = 11/13, \quad \text{coverage}(\text{HR}_2) = 11/18.$$

The concept $(\text{Size}, \succeq) \wedge (\text{Price}, \succeq)$ reflects the overall objects ranking positively in terms of both accuracy and coverage.

5 Conclusion

A granular computing based interpretation is presented in this paper. By extracting the high-level similarity from existing decision logic languages [7, 16], we introduce a more general language \mathcal{L} for granular computing. Two basic features of the language \mathcal{L} are the set of atomic formulas \mathcal{A} and a model M consisting of individuals. For each formula, the collection of all individuals satisfying the formula form a granule, called the meaning of the formula. A rule is therefore expressed as connection between two formulas and interpreted based on the corresponding granules of the two formulas.

Depending on particular applications, we can construct concrete languages by using different types of atomic formulas and the associated models. This flexibility of the language \mathcal{L} is demonstrated by considering two rough set approaches, namely, the standard rough set analysis and dominance-based rough set analysis. The main differences of the two approaches are their respective treatments of atomic formulas and models. An information table is used to construct a concrete granular computing model. For standard rough set analysis, two types of granules are constructed based on two families of atomic formulas. One consists of a set of objects that share the same attribute value. The other consists of object pairs that cannot be distinguished based on the values of an attribute. Low and high order rules are defined to describe relationships between these two types of granules. For dominance-based rough set analysis, similar interpretations can be obtained by using two different families of atomic formulas.

The results of the paper suggest that one may study rule mining at a more abstract level. Algorithms and evaluation measures can indeed be designed uniformly for both low and high order rules.

References

1. Demri, S., Orłowska, E.: Logical Analysis of Indiscernibility. In: Orłowska, E. (Ed.): Incomplete Information: Rough Set Analysis. Physica Verlag, Heidelberg (1997) 347-380
2. Greco, S., Matarazzo, B., Slowinski, R.: Rough Approximation of a Preference Relation by Dominance Relations. *European Journal of Operational Research* **117** (1999) 63-83
3. Greco, S., Matarazzo, B., Slowinski, R.: Rough Approximation by Dominance Relations. *International Journal of Intelligent Systems* **17** (2002) 153-171
4. Greco, S., Slowinski, R., Stefanowski, J.: Mining Association Rules in Preference-ordered Data. *Proceedings of the 13th International Symposium on Foundations of Intelligent Systems (ISMIS'02)* (2002) 442-450
5. Grzymala-Busse, J.W.: Incomplete Data and Generalization of Indiscernibility Relation, Definability, and Approximations. *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Proceedings of 10th International Conference, LNAI* **3641** (2005) 244-253
6. Nguyen, H.S., Skowron, A., Stepaniuk, J.: Granular Computing: a Rough Set Approach. *Computational Intelligence* **17** (2001) 514-544
7. Pawlak, Z.: *Rough Sets - Theoretical Aspects of Reasoning about Data*. Kluwer Publishers, Boston. (1991)
8. Pawlak, Z., Skowron, A.: Rough Sets: Some Extensions. *Information Science* **177** (2007) 28-40

9. Quinlan, J.R.: Learning Efficient Classification Procedures and Their Application to Chess End-games. In: Michalski, J.S., Carbonell, J. G., Mirchell, T.M. (Eds.): Machine Learning: An Artificial Intelligence Approach. Vol. 1. Morgan Kaufmann. (1983) 463-482
10. Quinlan, J.R.: Learning Logical Definitions from Relations. *Machine Learning* **5** (1990) 239-266
11. Sai, Y., Yao, Y.Y., Zhong, N.: Data Analysis and Mining in Ordered Information Tables. Proceedings of the 2001 IEEE International Conference on Data Mining. (2001) 497-504
12. Tsumoto, S. Automated Discovery of Plausible Rules Based on Rough Sets and Rough Inclusion. Proceedings of PAKDD'99, LNAI **1574** (1999) 210-219
13. Wille, R.: Concept Lattices and Conceptual Knowledge Systems. *Computers Mathematics with Applications* **23** (1992) 493-515
14. Yao, Y.Y., Zhong, N.: An Analysis of Quantitative Measures Associated with Rules. Proceedings of PAKDD'99, LNAI **1574** (1999) 479-488
15. Yao, Y.Y., Sai, Y.: On Mining Ordering Rules. *New Frontiers in Artificial Intelligence, Joint JSAI 2001 Workshop Post-Proceedings, LNCS* **2253** (2001) 316-321
16. Yao, Y.Y.: Mining High Order Decision Rules. In: Inuiguchi, M., Hirano, S., and Tsumoto, S. (Eds.): *Rough Set Theory and Granular Computing*. Springer, Berlin. (2003) 125-135