PODSTAWY STEROWANIA
18, 3-4 (1988), 175-200

Rough sets and information systems

Zdzisław PAWLAK

Department of Complex Control Systems, Polish Academy of Sciences
44-100 Gliwice ul. Bałtycka 5,

Received on 88.03.15

1. Introduction

In this paper we are going to give some basic ideas underlying the concept of a rough set, introduced by the author in [13] in order to deal with the vague and imprecise data.

The most interesting case is when data is arranged in the form of an information system (see [12]). The application of rough sets to the analysis of information systems is shown and discussed here.

The proposed approach has ben applied successfully in many areas (see e.g. [1, 11] and [15]).

The rough set concept can be vieved as an alternative to the fuzzy sets (see [18]). Comparison of these two concepts can be found in [2, 14] and [17].

More properties concerning rough sets and information systems are published in [2–10] and [16].

2. Rough Sets

In this section we give basic definitions and properties of concepts necessary to explain the idea of a rough set.

2.1. Approximation space

An approximation space is an ordered pair Apr = (Univ, Ind), where Univ $\neq \emptyset$ is a finite set called an universe, and Ind $= \{R_1, R_2, \ldots, R_n\}$, $R_i \subseteq$ Univ $\times$ Univ is a family of primitive indiscernibility relations. We assume throughout this paper that every primitive indiscernibility relation R_i is an equivalence relation. If Ind consist of one indiscernibility relation, then the approximation space Apr = (Univ, Ind) will be called simple. Finite intersection of indiscernibility relations in Apr is also an indiscernibility relation in Apr. If $F \subseteq$ Ind, then $\tilde{F} = \bigcap_{R \in F} R$. Equivalence classes of an indiscernibility relation R in Apr are called R-elementary sets in Apr. The family of all R-elementary sets in Apr will be denoted R^*. If R is an indiscernibility relation in Apr = (Univ, Ind), $x, y \in$ Univ and $(x, y) \in R$, then we say that x and y are R-indiscernible in Apr (indiscernible with respect to R in Apr).

Every union of R – elementary sets in Apr will be called R – discernible set in Apr, or discernible set, if R is understood; Otherwise the set is called R – indiscernible in Apr, rough with respect to R in Apr. We assume that the empty set is R – discernible for every R and Apr. The family of all R – discernible sets in Apr will be denoted by $\text{Ind}_R(\text{Apr})$ or short Ind_R. Certainly Ind_R is a boolen algebra, i.e. the family of all R – discernible sets is closed under intersection, union and complement of sets.

If Apr = (Univ, Ind) and Apr′ = (Univ′, Ind′) are two approximation spaces and Univ′ $\subseteq$ Univ, Ind′ $\subseteq$ Ind we say that Apr is a subapproximation space of Apr. If Univ′ $\subset$ Univ and each indiscernibility relation in Ind′ is restriction of certain indiscernibility relation in Ind to Univ′ we say that the approximation space Apr′ is a subapproximation space of Apr restricted to Univ′, Apr′ = Apr/Univ′. If Univ = Univ′ and Ind′ $\subset$ Ind we say that Apr′ is subapproximation space of Apr restricted to Ind′, (Apr′ = Apr/Ind′).

If R and Q are two indiscernibility relations in Apr = (Univ, Ind) and $R \subset Q$ we say that R is finer then Q, or Q is coarser then R.

2.2. Approximation of sets

Let Apr = (Univ, Ind) be an approximation space, $X \subseteq$ Univ and R an indiscerminility relation in Apr. For any X and R we define two sets

$$\underline{R}X = \{x \in \text{Univ}: x_R \subseteq X\}$$

$$\bar{R}X = \{x \in \text{Univ}: x_R \cap X \neq \emptyset\}$$

called the R-lower and the R-upper approximation of X in Apr, respectively.

The set $Bn_R(X) = \bar{R}X - \underline{R}X$ will be called a R-boundary of X in Apr.

We shall employ also the following denotations:

$\text{Int}_R(X) = \underline{R}X$, R – interior of X,

$\text{Ext}_R(X) = \text{Univ} - \bar{R}X$, R – exterior of X.

The notions introduced above can be depicted as shown in Fig 1.

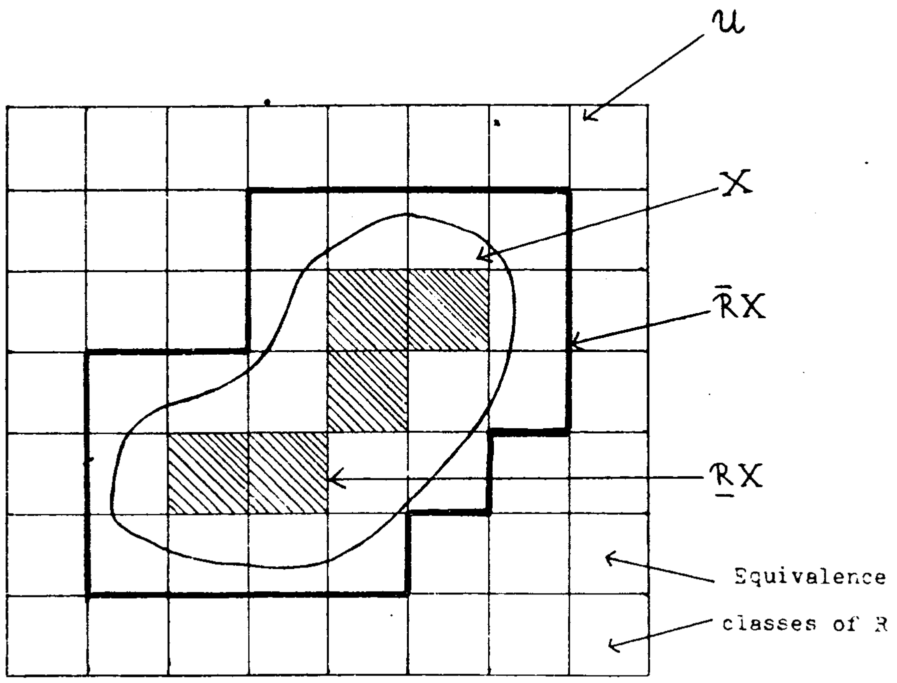

Fig. 1

The number

$$\alpha_R(X) = \frac{\operatorname{card} \underline{R}X}{\operatorname{card} \bar{R}X}$$

will be called R-accuracy (or accuracy with respect to R) of X in Apr, and the number

$$\varrho_R(X) = 1 - \alpha_R(X)$$

is called R-roughness (or roughness with respect to R) of X in Apr.

Of course $0 \leqslant \alpha_R(X),\ \varrho_R(X) \leqslant 1$.

Obviously a set X is R-discernible iff $\underline{R}X = \bar{R}X$; otherwise, i.e. if $\underline{R}X \neq \bar{R}X$, a set X is X is R-indiscernible or rough with respect to R.

2.3. Properties of approximations

Each indiscernibility relation R in Apr = (Univ, Ind) defines the topological space T_R = (Univ, Ind_R(Apr)), where Ind_R(Apr) is the topology for Univ and it is the family of all open and closed lets in T_R. The family of all R-elementary sets in Apr is a base of T_R. The R-lower and R-upper approximation of X in A are interior and closure operations in the topological space T_R respectively, and the following properties are true:

A1) $\underline{R}X \subseteq X \subseteq \bar{R}X,$

A2) $\underline{R}\,\text{Univ} = \bar{R}\,\text{Univ} = \text{Univ},$

A3) $\underline{R}\emptyset = \bar{R}\emptyset = \emptyset,$

A4) $\underline{R}(X \cup Y) = \bar{R}X \cup \bar{R}Y,$

A5) $\underline{R}(X \cup Y) \supseteq \underline{R}X \cup \underline{R}Y,$

A6) $R(X \cap Y) \subseteq RX \cap \bar{R}Y,$

A7) $\underline{R}(X \cap X) = \underline{R}X \cap \underline{R}Y,$

A8) $\bar{R}(-X) = -\underline{R}(X),$

A9) $\underline{R}(-X) = -\bar{R}(X),$

A10) $\underline{R}\underline{R}X = \bar{R}\underline{R}X = \underline{R}X,$

A11) $\bar{R}\bar{R}X = \underline{R}\bar{R}X = \bar{R}X.$

Using the notions of R-lower and R-upper approximations we can define two membership functions $\underline{\in}_R$ and $\bar{\in}_R$ (called a strong and a weak membership, respectively) as follows

$$x \underline{\in}_R X \quad \text{iff } x \in \underline{R}X,$$

$$x \bar{\in}_R X \quad \text{iff } x \in \bar{R}X,$$

If $x \underline{\in}_R X$ we say that "x surely belongs to X with respect to R", and if $x \bar{\in}_R X$ we say that "x possibly belongs to X with respect to R".

Let us also note that $\underline{R}X$ is the maximal R-discernible set contained in X and $\bar{R}X$ is the minimal R-discernible set containing X.

In other words $\underline{R}X$ is the intersection of all sets having the same R-lower approximation and $\bar{R}X$ is the union of all sets having the same R-upper approximation.

2.4. Classification of rough sets

Let Apr = (Univ, Ind), R be an indiscernibility relation in Apr and $X \subseteq$ Univ be an R-indiscernible (rough set in Apr). We can classify rough sets as follows:

Set X is roughly R-discernible in Apr, if $\underline{R}X \neq \emptyset$ and $\bar{R}X \neq$ Univ.

Set X is externally R-indiscernible in Apr, if $\underline{R}X \neq \emptyset$ and $\bar{R}X =$ Univ.

Set X is internally R-indiscernible in Apr, if $\underline{R}X = \emptyset$ and $\bar{R}X \neq$ Univ.

Set X is totally R-indiscernible in Apr, if $RX = \emptyset$ and $\bar{R}X =$ Univ.

Let us notice that if X is R-discernible (roughly R-discernible, totally R-indiscernible) so is $-X$. If X is externally (internally) R-indiscernible, then $-X$ is internally (externally) R-indiscernible.

The above classification says how a set of objects X can be discern by means of indiscernibility relation R.

Example 2.4.1

Let Univ $= \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$ and let R, be an indiscernibility relation with the following equivalence classes:

$$E_1 = \{x_0, x_1\}, \quad E_2 = \{x_2, x_6, x_9\}, \quad E_3 = \{x_3, x_5\},$$
$$E_4 = \{x_4, x_8\}, \quad E_5 = \{x_7, x_{10}\}$$

The sets:

$$X_1 = \{x_0, x_1, x_4, x_8\},$$
$$Y_1 = \{x_3, x_5, x_7, x_8\}$$
$$Z_1 = \{x_2, x_3, x_5, x_6, x_9\}$$

are examples of R-discernible sets.

The sets

$$X_2 = \{x_0, x_3, x_4, x_5, x_8, x_{10}\},$$
$$Y_2 = \{x_1, x_7, x_8, x_{10}\},$$
$$Z_2 = \{x_2, x_3, x_4, x_8\}$$

are examples of roughly R-discernible sets. The corresponding approximations boundaries and accurately:

$$\underline{R}X_2 = E_3 \cup E_4 = \{x_3, x_4, x_5, x_8\},$$
$$\bar{R}X_2 = E_1 \cup E_3 \cup E_4 = \{x_0, x_1, x_3, x_4, x_5, x_7, x_8, x_{10}\},$$
$$Bn_R(X_2) = E_1 = \{x_0, x_1\},$$
$$\alpha_R(X_2) = 4/8 = 1/2,$$
$$\underline{R}Y_2 = E_5 = \{x_7, x_{10}\},$$
$$\bar{R}Y_2 = E_1 \cup E_4 \cup E_5 = \{x_0, x_1, x_4, x_7, x_8, x_{10}\},$$
$$Bn_R(Y_2) = E_1 \cup E_4 = \{x_4, x_7, x_8, x_{10}\},$$
$$\alpha_R(Y_2) = 2/6 = 1/3,$$
$$\underline{R}Z_2 = E_4 = \{x_4, x_8\},$$
$$\bar{R}Z_2 = R_2 \cup E_3 \cup E_4 = \{x_2, x_3, x_4, x_5, x_6, x_8, x_9\},$$
$$Bn_R(Z_2) = E_2 \cup E_3 = \{x_2, x_3, x_5, x_6, x_9\},$$
$$\alpha_R(Z_2) = 2/7.$$

The sets:

$$X_3 = \{x_0, x_1, x_2, x_3, x_4, x_7\},$$
$$X_3 = \{x_1, x_2, x_3, x_6, x_8, x_9, x_{10}\},$$
$$Z_3 = \{x_0, x_2, x_3, x_4, x_8, x_{10}\}$$

are examples of externally R-indiscernible sets.

The corresponding approximations, boundaries and accurately are as follows:

$$\underline{R}X_3 = E_1 = \{x_0, x_1\},$$
$$\underline{R}X_3 = \text{Univ},$$
$$Bn_R(X_3) = E_2 \cup E_3 \cup E_4 \cup E_5 = \{x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\},$$
$$\alpha_R(X_3) = 2/10 = 1/5,$$
$$\underline{R}Y_3 = E_2 = \{x_2, x_6, x_4\},$$
$$\cdot \underline{R}Y_3 = \text{Univ},$$
$$Bn_R(Y_3) = E_1 \cup E_3 \cup E_4 \cup E_5 = \{x_0, x_1, x_3, x_4, x_5, x_7, x_8, x_{10}\},$$
$$\alpha_R(\cdot Y_3) = 3/10,$$
$$\underline{R}Z_3 = E_4 = \{x_4, x_8\},$$
$$\bar{R}Z_3 = \text{Univ},$$
$$Bn_R(Z_3) = E_1 \cup E_2 \cup E_3 \cup E_{\underset{\sim}{5}} = \{x_0, x_1, x_2, x_3, x_5, x_6, x_7, x_9, x_{10}\},$$
$$\alpha_R(Z_3) = 2/10 = 1/5.$$

The sets:

$$X_4 = \{x_0, x_2, x_3\},$$
$$X_4 = \{x_1, x_2, x_4, x_7\},$$
$$Z_4 = \{x_2, x_3, x_4\}$$

are examples of internally R-indiscernible sets.

Below are given upper approximation of these sets:

$$\bar{R}X_4 = E_1 \cup E_2 \cup E_3 = \{x_0, x_1, x_2, x_3, x_5, x_6, x_9\},$$
$$\bar{R}Y_4 = E_1 \cup E_2 \cup E_4 \cup E_7 = \{x_0, x_1, x_2, x_4, x_6, x_7, x_8, x_9, x_{10}\},$$
$$\bar{R}Z_4 = E_2 \cup E_3 \cup E_4 = \{x_3, x_3, x_4, x_5, x_6, x_8, x_9\}.$$

Certainly lower approximations of these sets are empty sets, and accuracy is equal to zero.

Below are given examples of sets which are totally R-indiscernible:

$$X_5 = \{x_0, x_2, x_3, x_4, x_7\},$$
$$X_5 = \{x_1, x_5, x_6, x_8, x_{10}\},$$
$$Z_5 = \{x_6, x_2, x_5, x_8\}.$$

2.5. Dependency of indiscernibility relations

Let Apr = (Univ, *Ind*) be an approximation space and P, Q indiscernibility relations in Apr.

The number

$$\gamma_P(Q^*) = \frac{\operatorname{card} \operatorname{Int}_P(Q^*)}{\operatorname{card} \operatorname{Univ}},$$

where

$$\operatorname{Int}_P(Q^*) = \bigcup_{X \in Q^*} \underline{P}X$$

will be called the accuracy of the classification Q^* with respect to P (P-accuracy of Q^*).

Obviously $0 \leqslant \gamma_P(Q^*) \leqslant 1$ for any P and Q. We say that Q *depends in a degree* k on P (in Apr), $P \xrightarrow{k} Q$, if $k = \gamma_P(Q^*)$.

If $P \xrightarrow{1} Q$ we say that depends totally on P (or in short depends); if $0 < k < 1$ and $P \xrightarrow{k} Q$ we say that Q partially depends on P; if $P \xrightarrow{0} Q$ we say that Q is totally independent on P.

If $P \xrightarrow{1} Q$ we shall also writte $P \to Q$.

The intuitive meaning of the dependency relation is as follows: if $P \to Q$ then $(x, y) \in P$ implies $(x, y) \in Q$ for every $x, y \in \operatorname{Univ}$; if Q partially depends on P, then $(x, y) \in P$ implies $(x, y) \in Q$ for some $x, y \in \operatorname{Univ}$; if Q is totally independent on P, then $(x, y) \in P$ does not imply $(x, y) \in Q$ for any $x, y \in \operatorname{Univ}$.

Property 2.5.1

$P \xrightarrow{k} Q$ in Apr iff $P \xrightarrow{1} Q$ in Apr $\operatorname{Int}_P(Q^*)$ and $P \xrightarrow{0} Q$ in Apr $Bn_P(Q^*)$, where $Bn_P(Q^*) = \bigcup_{X \in Q^*} Bn_P X$.

Property 2.5.2

The following conditions are equivalent:

1) $P \to Q$,
2) $P \subseteq Q$,
3) $P \cap Q = P$,
4) $\operatorname{Int}_P(Q^*) = \operatorname{Univ}$.

2.6. Reduction of indiscernibility relations

Let Apr = (Univ, Ind), F, G and $H \subseteq \operatorname{Ind}$.

We say that F is a reduct of G with respect to H (H-reduct) in Apr if F is a minimal subset of G such that

$$\mathrm{Int}_F(H^*) = \mathrm{Int}_G(H^*).$$

If $H = G$, then F will be called a reduct of G.

Property 2.6.1

If F is a reduct of G, then $\tilde{F} = \tilde{G}$.

Subset $F \subseteq \mathrm{Ind}$ of indiscernibility relations is independent in Apr if the only reduct of F is F itself, otherwise F is dependent in Apr.

Property 2.6.2

1) $F \subseteq \mathrm{Ind}$ is independent in Apr iff for every $G \subset F$, $G \neq F$.

2) $F \subseteq \mathrm{Ind}$ is dependent in Apr iff there exists $G \subseteq F$ such that $\tilde{G} = \tilde{F}$.

3) G is a reduct of F in Apr iff G is a maximal independent subset of indiscernibility relations in F.

Let $F, G \subseteq \mathrm{Ind}$ and $R \subset F$. We say that R is superfluous with respect to G (G-superfluous) in F if

$$\mathrm{Int}_F(G^*) = \mathrm{Int}_{F-R}(G^*);$$

otherwise R is indispensable with respect to G (G-indispensable) in F.

The set of all G-indispensable relations in F will be called the core of F with respect to G (G-core).

Property 2.6.3

Let F_G denote the G-cors of F and let $G(F)$ be the family of all G-reducts of F. The following is true:

$$F_G = \bigcap_{H \in G/F} H.$$

The core is the set of "most" important attributes, in this sense that they can not be removed without accetting the "descriptive power" of the set attributes.

3. Information Systems

In this section we show the interpretation of previously introduced concepts in information systems. Intuitively – information system is a collection of data in which some objects (processes, states, etc) are described in terms of some features. For example a collection of data concerning patients suffering from a certain disease – in terms of some symptoms (like temperature, blood presure etc.) is an information system.

Information system seems to be a very good tool to discuss some problems concerning decision tables theory, machine learning, data analysis, pattern recognition, and others.

We show in this section how the rough set concept can be used to analyse these kind of data.

3.1. Information system and approximation space

An information system is a 4-tuple

$$S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f),$$

where

Univ – is a finite set of objects – called the universe,

Att – is a finite set of attributes, $\mathrm{Val} = \bigcup_{a \in \mathrm{Att}} \mathrm{Val}_a$, Val_a – is a finite set of values of $a \in \mathrm{Att}$,

$f\colon \mathrm{Univ} \times \mathrm{Att} \to \mathrm{Val}$ – is a total function, such that

$$f(x, a) \in \mathrm{Val}_a \quad \text{for every } a \in \mathrm{Att} \text{ and } x \in \mathrm{Univ}.$$

In other words an information system is a finite table column of which are labeled by attributes and rows are labeled by objects. Fach entry of column a and row x is the value $f(x, a)$.

An example of an information system is given in Tab. 1.

Table 1

Univ	a	b	c
x_1	1	0	2
x_2	0	1	1
x_3	2	0	0
x_4	1	0	2
x_5	1	0	0
x_6	0	1	1
x_7	2	0	0
x_8	1	0	0
x_9	1	0	2
x_{10}	0	1	1

With every subset of attributs $A \subseteq \mathrm{Att}$, we associate an indiscernibility relation $\tilde{A}$ defined thus

$$\tilde{A} = \{(x, y) \colon f(x, a) = f(y, a) \quad \text{for every } a \in A\}.$$

Of course $\tilde{A}$ is an equivalence relation. Hence with every information system S we can associate an approximation space $A_S = (\text{Univ}, \text{Ind})$, where

$$\text{Ind} = \{\tilde{a}: a \in \text{Att}\}.$$

Thus every a is a primitive indiscernibility relation in Apr for any a Att. In what follows we shall write A instead $\tilde{A}$ if there will be no confusion of the set attributes A and the relation $\tilde{A}$.

Conversely, with every approximation space Apr = (Iniv, Ind) we can associate an information system S_{Apr} defined as follows:

with every primitive indiscernibility relation R in Apr we associate uniquelly a name a_R of R and we define the function f in such that $f(x, a_R) = f(y, a_R)$ iff x and belong to the same equivalence class of the relation R.

Thus the concepts of an approximation space and information space are isomorphic and can be mutually replaced.

3.2. Dependency of attributes in information systems

Let $S = (\text{Univ}, \text{Att}, \text{Val}, f)$ be an information system and A, $B \subseteq \text{Att}$ subset of attributes.

In order to define the dependency of attributes we can employ the definition given in Section 2.5.

Set of attributes B depends on set of attributes B in a degree k $(0 \leqslant k \leqslant 1)$ in S, in symbols $A \xrightarrow{k} B$, if $k = \gamma_A(B) = \dfrac{\text{card}\,\text{Int}_A(B^*)}{\text{card}\,\text{Univ}}$.

The intuitive meaning of this definition is following: if $A \xrightarrow{1} B$ that is to mean that the values of attributes $b \in B$ can be determined when values of attributes $a \in A$ are known. If $k \neq 1$, that is to mean that the dependency between A and B is partial, i.e. the dependency holds for objects belonging to $\text{Int}_A(B^*)$ only (see Property 2.5.1).

For example in the information system shown in Tabl. 1 the following dependency is valid: $B \xrightarrow{0.5} c$, where $B = \{a, b\}$, and c is the abbreviation for $\tilde{c}$.

Because

$$c^* = \{X_1, X_2, X_3\},$$

where

$$X_1 = \{x_1, x_5, x_9\},$$

$$X_2 = \{x_2, x_7, x_{10}\},$$

$$X_3 = \{x_3, x_4, x_6, x_8\}$$

and

$$B^* = \{Y_1, Y_2, Y_3\},$$

where

$$Y_1 = \{x_1, x_4, x_5, x_8, x_9\},$$
$$Y_2 = \{x_2, x_7, x_{10}\},$$
$$Y_3 = \{x_3, x_6\}.$$

Hence

$$\underline{B}X_1 = \emptyset, \quad \underline{B}X_2 = Y_2, \quad \underline{B}X_3 = Y_3$$

and

$$\mathrm{Int}_B(c^*) = \underline{B}X_1 \cup \underline{B}X_2 \cup \underline{B}X_3 = Y_1 \cup Y_3 = \{x_2, x_3, x_6, x_7, x_{10}\}.$$

Thus $k = 5/10 = 0{,}5$.

This is to mean that the objects $\{x_2, x_3, x_6, x_7, x_{10}\}$ only can be property classified, to the classes of c^*, employing the set of attributes B.

Let us also notice that

$$\bar{B}X_1 = Y_1, \quad \bar{B}X_2 = Y_2, \quad \bar{B}X_3 = Y_1 \cup Y_3,$$

i.e. set X_1 is internally B-indiscernible, X_2 is B-discernible and X_3 is roughly B-discernible, the corresponding accuracy coefficients are:

$$\alpha_B(X_1) = 0/5 = 0,$$
$$\alpha_B(X_2) = 3/3 = 1,$$
$$\alpha_B(X_3) = 2/7.$$

3.3 Reduction of attributes

Let S = (Univ, Att, Val, f) be an information system, $A, B, C \subseteq$ Att subsets of attributes.

If A is a reduct of B will respect to C (see definition section 2.6), that is to mean that set of attributes $A \subseteq B$ provides the same discernibility of classes of the classification C^* as the set of attributes B.

Hence instead using set of attributes B we can use smaler set of attributes A, to classify objects to classes of the classification C^*.

It is easy to see, for example, that in the information system shown in Tab. 1 set $\{a, c\}$ is the only reduct of set of attributes $\{a, b, c\}$, for $\{a, c\} = \{a, b, c\}$.

There are two reducts a and c of the set attributes $\{a, c\}$ with respect to b. Note that the b-core of $\{a, c\}$ is the empty set. Moreover we have $a \to a$ and $c \to b$ in the information system considered.

4. Information Language

In this chapter we introduce the concept of an information language, which will be associated with every information system. The information language will be used to describe decision rules and decision algorithms in a syntactical way, which allows employing standard logical methods to analyse and investigate these concepts.

4.1. Syntax of the information language

With every information system $S = (\text{Univ}, \text{Att}, \text{Val}, f)$ we associate an information language L_S (L – when S is understood), which consists of terms, formulas and decision algorithms.

Terms are built up from some constants by means of boolean operations $+, \cdot, -$; we assume that 0,1 are constants and Att, Val are finite sets of constants called attributes and attribute values, respectively.

The set of terms is the least set satisfying the conditions:

1) Constants 0 and 1 are terms in L.

2) Any expression of the form $(a := v)$, where $a \in \text{Att}$ and $v \in \text{Val}_a$ – is a term in L.

3) If s and t are terms in L, so are $-t$, $(t+s)$ and $(t \cdot s)$ (or simple ts).

In what follows we shall drop unnecessary parenthesis in a usual way in terms.

For example the following

$$-((a := 0)+(b := 2))(c := 1),$$
$$(a := 2)(-(b := 1))$$

are terms in a certain information language.

The set of formulas in an information language L in the least set satisfying the conditions:

1) Constants T (for truth) and F (for falsity) are formulas in L.

2) If t and s are terms in L, then $t = s$ and t, s are formulas in L.

3) If φ and ψ are formulas in L, then $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$ and $(\sim \varphi)$ are formulas in L.

For example the following

$(a := 0) = (b := 2)$,
$(a := 1)(b := 0) = (c := 2)$,
$(a := 2) \Rightarrow (b := 0)+(c := 1)$,
$((a := 0) \Rightarrow (b := 0)) \cup ((a := 1) \Rightarrow (c := 0))$

are formulas in a certain information language.

4.2. The meaning (the semantics) of terms and formulas

In this section we shall define formally the meaning of terms and formulas in an information system $S = (\text{Univ}, \text{Att}, \text{Val}, f)$. Terms are intended to mean subsets of the universe and the meaning of formulas is truth of falsity.

In order to define the meaning of terms and formulas we shall employ the meaning function g_s: Ter $\times$ For $\to P(\text{Univ}) \cup \{T, F\}$, where Ter and For denote the set of all terms and formulas, respectively.

The meaning function for terms is defined as follows (we omit the subscript S if S is understood):

1) $g(0) = \varnothing$, $g(1) = \text{Univ}$.
2) $g(a := v) = \{x \in \text{Univ}: f(x, a) = v\}$.
3) $g(-t) = \text{Univ} - g(t)$,
   $g(t+s) = g(t) \cup g(s)$,
   $g(ts) = g(t) \cap g(s)$.

For example the meaning of certain terms in the information system shown in Tab. 1 is shown below:

$$g((a:=1)(b:=0)) = \{x_1, x_4, x_8, x_8\},$$
$$g((a:=0)+(c:=2)) = \{x_1, x_2, x_5, x_7, x_9, x_{10}\},$$
$$g(-a:=2)) = \{x_1, x_2, x_4, x_5, x_7, x_8, x_9, x_{10}\}.$$

The meaning of formulas is defined in the following way:

1) $g(T) = T$; $g(F) = F$.
2) $g(t = s) = \begin{cases} T, & \text{if } g(t) = g(s), \\ F, & \text{if } g(t) \neq g(s). \end{cases}$
3) $g(t \Rightarrow s) = \begin{cases} T, & \text{if } g(t) \subseteq g(s), \\ F, & \text{otherwise}. \end{cases}$
4) $g(\sim \varphi) = \begin{cases} T, & \text{if } g(\varphi) = F, \\ F, & \text{if } g(\varphi) = T. \end{cases}$
5) $g(\varphi \vee \psi) = g(\varphi) \vee g(\psi)$.
6) $g(\varphi \wedge \psi) = g(\varphi) \wedge g(\psi)$.

If $g_S(\varphi) = T$ we say that φ is true in S; if $g_S(\varphi) = F$ then φ is said to be false in S. If φ is true in S we shall write $\models_S \varphi$ or simple $\models \varphi$ when S is known.

If $\models_S (t = s)$ we say that t and s are equivalent in S; if $\models_S (t \Rightarrow s)$ we say that the term t implies the term s in S.

Examples of false and true formulas in the information system given in Tab. 1 are shown below:

$(a:=0) = (b:=1)$ – true,
$(a:=2) = (c:=0)$ – false,
$(c:=0) = (b:=0)$ – true,
$(c:=1) = (b:=0)$ – false.

For the transformation of terms we shall use the axioms of boolean algebra and the following specific axiom

$$(a := v) = -\sum_{\substack{u \neq v \\ u \in \mathrm{Val}_a}} (a := u).$$

For the transformation of formulas we shall employ the axioms of propositional calculus.

4.3. Normal form of terms

A term t in L is A-elementary ($A \subseteq \mathrm{Att}$) if $t = \prod_{a \in A} (a := v_a)$.

A term t in L is in A-normal from if $t = \sum s$, where all s are A-elementary. For example the terms

$$(a := 1)(b := 0), \quad (a := 2)(b := 1), \quad (a := 0)(b := 1)$$

are $\{a, b\}$-elementary and the terms

$$(a := 2)(b := 1)(c := 2),$$
$$(a := 0)(b := 0)(c := 1)$$

are $\{a, b, c\}$-elementary.

The term

$$(a := 1)(b := 0) + (a := 2)(b := 1)$$

is in $\{a, b\}$-normal form and the term

$$(a := 2)(b := 1)(c := 2) + (a := 0)(b := 0)(c := 1)$$

is in $\{a, b, c\}$-normal form.

Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be an information system, $A \subseteq \mathrm{Att}$ subset of attributes, and L_A — an information language with the set of attributes A.

Property 4.3.1

For every term t in L_A there exists the terms s in L_A, in A-normal form such that $\models_S t = s$; s is referred to as the A-normal form of t in L_A.

For example the $\{a, b\}$-normal form of the term $(a := 1)$ is the term $(a := 1)(b := 0) + (a := 1)(b := 1)$ (in the information system shown in Tab. 1).

Subset $X \subseteq \mathrm{Univ}$ is said to be A-describable in $L (A \subseteq \mathrm{Att})$ if there exists a term t in L such that $g_S(t) = X$; the term t is called the A-description of X in L.

If set $X \subseteq \mathrm{Univ}$ is non A-describable in L, then there exist terms t and s such that $g_S(t) = \underline{A}X$ and $g_X(s) = \bar{A}X$, called the $\underline{A}$ — lower and $\bar{A}$ — upper description of X in L, respectively. This is to mean that some subset of objects be described by a given subset of attributes not exactly but with some approximation only.

For example the subset $X = \{x_3, x_4, x_7\}$ of objects in the information system shown in Tab. 1 has the following A-lower and A-upper description ($A = \{a, b, c\}$).

A-lower description of X is

$$(a:=2)(b:=0)(c:=0);$$

A-upper description of X is

$$(a:=2)(b:=0)(c:=0)+(a:=1)(b:=0)(c:=2).$$

4.4 Decision rules

Any formula of the form $t \Rightarrow s$ will be called a decision rule in L; t is referred to as a condition and s – as a decision of the decision rule, respectively. If the decision rule $t \Rightarrow s$ in true we shall also say that it is consistent; otherwise the decision rule is inconsistent.

Let $t \Rightarrow s$ be a decision rule in L and let A, $B \subseteq$ Att be sets of all attributes which occur in t and s, respectively. We shall call $t \Rightarrow s$ (A, B) – decision rule. If A and B are one element sets, for the sake of simplicity, we shall use the expression (a, b)-decision rule.

Let $S = (\text{Univ}, \text{Att}, \text{Val}, f)$ be an information system and $t \Rightarrow s$ (A, B)-decision rule in L.

We say that the (A, B)-decision rule is deterministic in S, if $g_S(s) \in B^*$, i.e. $g_S(s)$ in a description of a certain equivalence class of the equivalence relation B; otherwise the decision rule is nondeterministic.

We say that $a(A, B)$-decision rule $t \Rightarrow s$ is in (A, B)-normal form if t and s are in (A, B)-normal form.

Property 4.4.1

A (A, B)-decision rule $t \Rightarrow s$ is consistent in S iff all non-empty elementary terms occurring in (A, B)-normal form of t occur also in the (A, B)-normal form of s.

This property unable us to prove the validity of any decision rule in a simple syntactical way.

Example 4.4.1

Consider the information system shown in Tab. 2.

Table 2

Univ	a	b
x_1	1	0
x_2	1	1
x_2	0	2

Let us check whether the decision rule $(a:=1) \Rightarrow (b:=0)$ is consistent or not.

The (a, b)-normal form of the rule is

$$(a:=1)(b:=0)+(a:=1)(b:=1) \Rightarrow (a:=1)(b:=0).$$

Because the elementary term $(a:=1)(b:=1)$ occurs only in the condition of th (a, b)-normal form of the rule, for the rule $(a:=1)\Rightarrow(b:=0)$ is inconsistent.

We can also check whether the decision rule is consistent or not, using th meaning function (semantics), namely:

$$g_S(a:=1)=\{x_1, x_2\}$$

and

$$g_S(b:=0)=\{x_1\},$$

hence $g_S(a:=1)\not\subseteq g_S(b:=0)$ and the decision rule is inconsistent.

On the other hand the decision rule $(b:=0)\Rightarrow(a:=1)$ is consistent because the (a, b)-normal form of the rule is of the form:

$$(a:=1)(b:=0)\Rightarrow(a:=1)(b:=0)+(a:=1)(b:=1)$$

and the only (a, b)-elementary term $(a:=1)(b:=0)$ in the condition of the decision rule occurs also in the decision of the rule.

Employing the definition of semantics we have

$$g_S(a\text{: }1)=\{x_1, x_2\} \quad \text{and} \quad g_S(b:=0)=\{x_1\},$$

hence $g_S(b:=0)\subset g_S(a:=1)$ and the decision rule is consistent.

Let us also notice that both decision rules are deterministic in the information system.

4.5. Decision algorithms

Any finite set of decision rules in L is called a decision algorithm in L. An example of a decision algorithm is shown below:
$(a:=1)\Rightarrow(b:=2)(c:=1)$,
$(b:=2)+(a:=1)\Rightarrow(c:=2)$,
$(a:=0)+(b:=2)\Rightarrow(c:=1)$.

With every decision algorithm $\mathfrak{a}=\{t_1\Rightarrow s_i\}_n$, $1\leqslant i\leqslant m$, in L we associate the formula

$$\Psi\mathfrak{a}=\bigwedge_{i=1}^{n}(t_i\Rightarrow s_i)$$

called the decision formula of $\mathfrak{a}$ in L.

A decision algorithm is said to be consistent if all its decision rules are consistent; otherwise the decision algorithm is inconsistent.

Property 4.5.1

A decision algorithm $\mathfrak{a}$ in L is consistent (in S) iff $\models_S \Psi\mathfrak{a}$.

Example 4.5.1

It is easy to see that the decision algorithm
$(a := 1)(b := 0) \Rightarrow (c := 2) + (c := 0)$,
$(a := 0) \Rightarrow (c := 1)$,
$(a := 2) + (b := 1) \Rightarrow (c := 0) + (c := 1)$.
is consistent in the information system shown in Tab. 1, and the decision algorithm
$(\mathrm{a} := 1) \Rightarrow (c := 2)$
$(a := 0) \Rightarrow (c := 0)$
is incosistent in the system.

A decision algorithms is deterministic in S if all its decision rules are deterministic in S; otherwise the algorithm is nondeterministic.

For example the two above decision algorithms are nondeterministic.

If A and B are the sets of all attributes occurring in the conditions and decisions of the decision rules of the algorithm α, then α will be called the (A, B)-decision algorithm and denoted $\alpha(A, B)$.

Property 4.5.2

(A, B)-decision algorithm is deterministic in S iff $A \rightarrow B$ in S.

A (A, B)-decision algorithm is total in S if for every equivalence class X of the equivalence relation B there exists a decision rule $t \Rightarrow s$ in α such that $g_S(s) = X$; otherwise the decision algorithm is partial in S.

Example 4.5.2.

The algorithm
$(\mathrm{a} := 1)(b := 0) \Rightarrow (c := 2)$,
$(a := 0) \Rightarrow (c := 1)$,
$(a := 2) = (b := 1) \Rightarrow (c := 0)$
is total in the information system shown in Tab. 1, whereas the decision algorithm
$(a := 0)(b := 0) \Rightarrow (c := 2)$,
$(a := 0)(b := 1) \Rightarrow (c := 1)$
is partial in the information system.

In what follows we shall consider total algorithms only (if not otherwise stated).

The following property can be used as transformation rule for decision algorithms:

Property 4.5.2

$$\models_S \bigwedge_{i=1}^{n} (t_i \Rightarrow s) \equiv \left(\sum_{i=1}^{n} t_i \Rightarrow s\right),$$

where $\equiv$ is defined in usual way.

The meaning of this property is obvious.

The following property establishes the relationship between the consistency and determinism of a decision algorithm.

Property 4.5.3

It $A \rightarrow B$ in S, then $\models_S \Psi a(A, B)$.

5. Examples

In this section we will depict previously introduced notions by means of three examples: discrimination analysis, learning from examples and decision tables.

We will show that all these cases can be reduced to the schemes discussed in the previous sections of the paper.

5.1. Discrimination analysis

Suppose we are given a data file, for example concerning patients suffering from a certain disease. With every patient several items of information (symptoms) are associated. Beside this patients are classifical according to a certain preassumed rules, for example, age, disease advance, etc. The classification can be based on existing symptoms or it can be given be an expert. The main problem of discrimination analysis consists in describing each class of the classification in terms of available symptoms.

Thus the problem can be reduced to that discussed in previous sections, namely we can treat the data file as an information system and ask whether $A \rightarrow c$, or not, where c is an attribute representing the classification and A is the set of atrributes representing symptoms. The second problem is to find reducts of A in order to find the minimal number of symptoms necessary to classify objects property.

An example given below will explains the idea of discrimination more exactly.

Example 5.1.1

Let us consider an information system shown in Tab. 3.

Table 3

Univ	a	b	c	d
x_1	1	0	2	1
x_2	0	1	1	2
x_3	2	0	0	3
x_4	1	1	0	1
x_5	1	0	2	2
x_6	2	0	0	3
x_7	0	1	1	2
x_8	0	1	1	2

Let attributes a, b, c represent some “symptoms” and the attribute d classification of objects.

Thus our first problem consist in checking wheather $\{a, b, c\} \to d$, and the second one is the reduction of “symptoms” $\{a, b, c\}$.

Let us denote $\{a, b, c\} = A$, and let us compute A^* and d^*. The A-elementary sets are classes of the classification A^* and are as follows:

$$X_1 = \{x_1, x_5\}, \quad X_2 = \{x_2, x_7, x_8\},$$
$$X_3 = \{x_3, x_6\}, \quad X_4 = \{x_4\}.$$

Classes of the classification d^* are:

$$Y_1 = \{x_1, x_4\},$$
$$Y_2 = \{x_2, x_5, x_7, x_8\},$$
$$Y_3 = \{x_3, x_6\}.$$

The corresponding approximations are as follows:

$$\underline{A}Y_1 = X_4 = \{x_4\}, \quad \underline{A}Y_1 = X_7 \cup X_4 = \{x_1, x_4, x_5\},$$
$$Bn_A(Y_1^*) = X_1 = \{x_1, x_5\}, \quad \gamma_A(Y_1^*) = 1/3,$$
$$\underline{A}Y_2 = X_2 = \{x_2, x_7, x_8\}, \quad \underline{A}Y_2 = X_2 \cup X_2 = \{x_1, x_2, x_4, x_5, x_7, x_8\},$$
$$Bn_A(Y_2^*) = X_1 = \{x_1, x_4\}, \quad \gamma_A(Y_2) = 1/2,$$
$$\underline{A}Y_3 = X_3 = \{x_3, x_6\}, \quad \bar{A}Y_3 = X_3 = \{x_3, x_6\},$$
$$Bn_A(Y_3^*) = \emptyset, \quad \gamma_A(Y_3^*) = 1,$$
$$\mathrm{Int}_A(d^*) = \underline{A}Y_1 \cup \underline{A}Y_2 \cup \underline{A}Y_3 = X_2 \cup X_3 \cup X_4 = \{x_2, x_3, x_4, x_6, x_7, x_8\},$$
$$\gamma_A(d^*) = \frac{\text{card}\,\mathrm{Int}_A(d^*)}{\text{card Univ}} = \frac{6}{8} = 3/4.$$

That is to mean that $A \xrightarrow{0.75} d$.

Thus we are unable to classify objects according to the classification d^* using set of attributes A. Only 6 out at 8 objects can be classify correctly using the set attributes A (namely objects $\{x_2, x_3, x_4, x_6, x_7, x_8\}$).

The accuraces of particular classes are shown below

Class	α
1	0,5
2	0,5
3	1,0

That means that classes 1 and 2 are roughly discernible by the set of attributes A and class 3 is discernible be the set of attributes.

It is easy to compute that the core of A with respect to d is the empty set and the set A has tree reducts $\{a, b\}$, $\{b, c\}$ and $\{a, c\}$ with respect to d. Thus any pair of attributes from A will suffice to classify objects with the same accuracy as that provided by the whole set of attributes A.

We can also give a decision algorithm, which enable us to classify objects to proper classes using properties expresses by attributes from A.

An example of a decision algorithms is shown below:

$(a := 1)(b := 1) \Rightarrow (d := 1)$,

$(a = 1)(b := 0) \Rightarrow (d := 1) + (d := 2)$,

$(a := 0) \Rightarrow (d := 2)$,

$(a := 2) \Rightarrow (d := 3)$.

Let us notice that the algorithm is nondeterministic, total and consistent.

This scheme of data analysis can be employed in psychology, sociology, agriculture, engineering and other areas.

5.2. Learning from examples

Suppose we are given a finite set of Univ of objects. Elements of Univ are called training examples and Univ is called a training set. Assume further that Univ is classified into disjoint classes $X_1, X_2, \ldots, X_n$ $(n > 2)$ by a teacher (expert, environment) etc. The classification represents the teacher's knowledge of objects from Univ. Furthermore assume that a "student" is able to characterize each objects from Univ in terms of attributes from preassumed set A. Descriptions of objects in terms of attributes from A represents the student's knowledge of objects from Univ.

We can say that the student has a syntactical knowledge and the teacher – the semantical knowledge – about objects from Univ.

The problem of learning from examples is, wheather the student's knowledge can be mached with the teacher's knowledge, or more precisely, whether the teacher's classification can be described in terms of attributes available to the student.

Thus learning from examples consists in describing classes $X_1, X_2, \ldots, X_n$ in terms of attributes from A, or more exactly, in finding a decision algorithm which provides the teacher's classification on the basis of properties of objects expressed in terms of attributes from A.

It is easily seen that the problem of learning from examples can be easily formulated in terms of notations introduced before. Training examples from the universe Univ, "students" attributes A and teacher attribute e – is the set of attributes Att.

Thus the problem of learning from examples reduces to the question whether $A \rightarrow e$ (or wherther e^* – is A-discernible), i.e. – whether there exists an algorithm to "learn" classification e^* by attributes of training examples.

This can be easily done by using methods discussed in the previous sections.

For example let us consider the following information system:

Example 5.2.1

Table 4

Univ	a	b	c
x_1	1	0	2
x_2	0	1	1
x_3	2	0	0
x_4	1	0	2
x_5	1	0	0
x_6	0	1	1
x_7	2	0	0
x_8	1	0	0
x_9	0	1	1
x_{10}	2	0	0
x_{11}	1	0	0
x_{12}	1	0	2

where Univ is the training set, $A = \{a, b\}$ are students attributes and $c =$ is teacher attribute.

In order to check wheter the concepts represented by examples $x_1, \ldots, x_{12}$ can be learned using attributes from A-according to the teacher knowledge expressed by the classification c^*, we have to compute degree k of dependency $A \xrightarrow{k} c$.

In order to do so let us compute classes of the classification c^*, which are as follows:

$$Y_1 = \{x_1, x_4, x_{12}\},$$
$$Y_2 = \{x_2, x_6, x_9\},$$
$$Y_3 = \{x_3, x_5, x_7, x_9, x_{10}, x_{11}\}$$

and classes of the classification A^*, which are given below:

$$X_1 = \{x_1, x_4, x_5, x_8, x_{11}, x_{12}\},$$
$$X_2 = \{x_2, x_6, x_9\},$$
$$X_3 = \{x_3, x_7, x_{10}\}.$$

The corresponding approximation are:

$$\underline{A}Y_1 = \emptyset, \quad \bar{A}Y_1 = X_1,$$
$$Bn_A(Y_1^*) = X_1, \quad \gamma_A(Y_1^*) = 0,$$
$$\underline{A}Y_2 = X_2, \quad \bar{A}Y_2 = X_2,$$
$$Bn_A(Y_2^*) = \emptyset, \quad \gamma_A(Y_2) = 1,$$
$$\underline{A}Y_3 = X_3, \quad \bar{A}Y_3 = X_1 \cup X_3,$$
$$Bn_A(Y_3^*) = X_1, \quad \gamma_A(Y_3) = 0,5,$$

$$\mathrm{Int}_A(c^*) = \underline{A}Y_1 \cup AY_2 \cup AY_3 = X_2 \cup X_3 = \{x_2, x_3, x_6, x_7, x_9, x_{10}\},$$
$$\gamma_A(c^*) = 6/12 = 0{,}5.$$

Thus the class Y_1 — is internally A-indiscernible, Y_2 — is A-discernible and Y_3 is roughly A-discernible.

Thus it is impossible to learn positive instances of Y_1, but it is possible to learn negative instances of Y_1 (if $x \cup X_2 \cup X_3$ we known that x is not in Y_1). In other words it is impossible to classify correctly $\{x_1, x_4, x_{12}\}$ by checking their features expressed by attributes a and b.

Set Y_2 can be learn, i.e. all elements of Y_2 can be classifical property examining their features a and b.

Set Y_3 can be learned roughly, i.e. only examples $\{x_3, x_7, x_{10}\}$ can be recognized on the basis of a and b as elements of Y_3; objects $\{x_2, x_6, x_9\}$ can be excluded being members of Y_1 and $X_1 = \{x_1, x_4, x_5, x_8, x_{11}, x_{12}\}$ is the boundary of Y_3, i.e. if can not decided whether elements of X_1 belong to Y_3 or not — employing the attributes a and b.

The corresponding decision algorithm is shown below:
$(a := 1)(b := 0)(c := 2) + (c := 0)$,
$(a := 0)(c := 1)$,
$(a := 2)(c := 0)$.
The algorithm is nondeterministic, total and consistent.

5.3. Decision Tables

Decision tables are important tools in computer applications. We are going to show in this section, the application of rough sets to decision table analysis. The proposed approach seems to be very suitable to formulate and slove many basic problems concerning decision table analysis and implementation.

Decision table can be considered as an information system in which attributes are divided intwo two classes called condition and decision attributes, denoted Con and Dec respectively.

Each row to the decision table is called a decision rule. In other word the decision rule in a function f_x: Att $\rightarrow$ Val such that $f_x(a) = f(x, a)$ for every $x \in$ Univ and $a \in$ Att. The restriction of f_x to condition attributes, denoted f_x/Con, will be called condition and the restriction of f_x to Dec, denoted f_x/Dec — decision of the rule f_x. Decision rules will be written in the form $p \Rightarrow g$, where p in the condition and g the decision of the rule (see section 4.4).

An example of decision table is shown below:

Example 5.3.1

Let us notice that the decision rule $(a := 0)(b := 1)(c := 1) \Rightarrow (d := 1)(c := 2)$ is non-deterministic whereas the rule $(a := 2)(b := 0)(c := 0) \Rightarrow (d := 1)(e := 1)$ in deterministic in Tab. 5.

Table 5

Univ	a	b	c	d	e
1	1	0	2	2	0
2	0	1	1	1	2
3	2	0	0	1	1
4	1	1	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	1	1	1	2
8	0	1	1	0	1

A decision table in deterministic if all its decision rules are deterministic, otherwise the decision table in non-deterministic.

For example the decision table shown in example 5.3.1 (Tab. 1) is non--deterministic.

Property 5.3.1

A decision table S is deterministic iff Con $\rightarrow$ Dec in S.

A decision table S in said to be roughly deterministic if Con $\xrightarrow{k}$ Dec and $0 < k < 1$; a decision table is totally non-deterministic if Con $\xrightarrow{0}$ Dec in S.

The property 2.5.1 can be also presented in the following form:

Property 5.3.2

Con $\rightarrow$ Dec in S iff Con $\xrightarrow{1}$ Dec in S $(\mathrm{Int}_{\mathrm{Con}}(\mathrm{Dec}^*))$ and Con $\xrightarrow{0}$ Dec in $Bn_{\mathrm{Con}}(\mathrm{Dec}^*)$.

This property can be used to decompose the decision table in two parts (possibly empty) such that first one is deterministic and the second totally non-deterministic.

Example 5.3.2

The decision table shown in Tab. 5 is non-deterministic and we have Con $\xrightarrow{0.5}$ Dec, and the table can be decomposed into two decision tables as shown below:

Table 6

Univ	a	b	c	d	e
3	2	0	0	1	1
4	1	1	0	2	2
6	2	2	0	1	1
7	2	1	1	1	2

Table 7

Univ	a	b	c	d	e
1	1	0	2	1	0
2	0	1	1	1	2
5	1	0	2	0	1
8	0	1	1	0	1

The decision table shown in Tab. 6 is deterministic and the table shown in Tab. 7 is totally non-deterministic.

Let us also notice that $\gamma_{\mathrm{Con}}(\mathrm{Dec}^*) = 1/2$ in the table.

If can be easily shown that the set of condition attributes in decision table shown in Tab. 5 has only one reduct $\{a, b\}$ and that the set of decision attributes is independent.

Thus the decision table can be presented as shown in Tab. 8.

Table 8

Univ	a	b	d	e
1	1	0	2	0
2	0	1	1	2
3	2	0	1	1
4	1	1	2	2
5	1	0	0	1
6	2	2	1	0
7	2	1	1	2
8	0	1	0	1

The next example shows more complicated simplification of a decision table.

Example 5.3.3.

The decision table shown in Tab. 8 below is deterministic

Table 9

Univ	a	b	c	d	e	f
1	3	2	2	2	2	4
2	3	2	2	1	2	4
3	2	2	2	1	1	4
4	2	2	2	2	1	4
5	3	2	2	3	2	3
6	3	3	2	3	2	3
7	4	3	2	3	2	3
8	4	3	3	3	2	2
9	4	4	3	3	2	2
10	4	4	3	2	2	2
11	4	3	3	2	2	2
12	4	2	3	2	2	2
13	3	3	2	2	2	4

Attributes a, b, c, d are conditions attributes and e, f are decision attributes in the table.

The set $\{a, b, c, d\}$ of conditions attributes is independent.

Now we can compute reducts of Con with respect to the classification $\{e, f\}^*$, and we get, the following reducts:

$\{a, b\}$ for $e := 2$ and $f := 4$,

$\{a\}$ for $e := 1$ and $f := 4$,

$\{e, a\}$ for $e := 2$ and $f := 3$,

$\{e\}$ for $e := 2$ and $f := 2$.

Hence we can simplify the table as shown below:

Table 10

Univ	a	b	c	d	e	f
1′	3	–	–	1	2	3
2′	3	–	–	2	2	4
3′	2	–	–	–	1	4
4′	–	–	2	3	2	3
5′	–	–	3	–	2	2

Consequently we get the following decision algorithm:

1′: $(a:=3)((d:=1)+(d:=2))\Rightarrow(e:=2)(f:=4)$,

2′: $(a:=2)\Rightarrow(e:=1)(f:=4)$,

3′: $(c:=2)(d:=3)\Rightarrow(e:=2)(f:=3)$,

4′: $(c:=3)\Rightarrow(e:=2)(f:=2)$.

6. Conclusion

When data are gathered about objects, states, processes etc., in terms of attributes (features), it may happend that some objects have the some data, and consequently they are indiscernible by the available features.

In order to deal with this kind of situations the concept of the rough set has been introduced, which turned out to be very suitable mathematical tool in many ares of artificial intelligence, where vague or imprecise data are to be analyzed.

References

[1] T. Arciszewski, W. Ziarko, *Adaptive Expert System for Preliminary Engineering Design*, Proc. of the International Workshop on Expert Systems and their Applications, Avignon, France, pp. 695–712, 1986.

[2] D. Dubois, H. Prade, *Twofold Fuzzy Sets and Rough Sets*, Fuzzy Sets and Systems, Vol. **23**, 1, pp. 3–28, 1987.

[3] J. Grzymała-Busse, *Algebraic Properties of Knowledge Representation Systems*, Proc. ACM Sigart, The International Symposium on Methodologic S for Intelligent Systems, Knoxville, USA, pp. 432–440, 1986.

[4] J. Grzymała-Busse, *On the Reduction of Knowledge Representation Systems*, Proc. of the 6th Interantional Workshop on Expert System s and their Applications, Avignon, France, vol. **1**, pp. 463–478, 1986.

[5] T. Iwiński, *Algebraic Approach to Rought Sets Bull*, Polish Acad. Sci. Math., Vol. **35**, No. 9–10, pp. 673–683, 1987.

[6] T. Iwiński, *Contraction of Attributes*, Bull. Polish Acad. Sci. Math. (to appear).

[7] M. Novotny, Z. Pawlak, *On Representation of Rough Sets by Information Systems*, Fundamenta Informaticae, Vol. **VI**, No. 3–4, pp. 289–296, 1985.

[8] M. Novotny, Z. Pawlak, *Black Box Analysis and Rough Top Equality*, Buh. Polish Acad. Sci. Math., Vol. **33**, No. 1-2, pp. 105–113, 1985.
[9] M. Novotny, Z. Pawlak, *Characterization of Rough Top Equalities and Rough Bottom Equalities*, Ball. Polish Acad. Sci Math., Vol. **33**, No. 1-2, pp. 91-97, 1985.
[10] M. Novotny, Z. Pawlak, *On Rough Equalities*, Buh. Polish Acad. Sci Math., Vol. **33**, No. 1-2, pp. 91 97.
[11] A. Mrózek, *Rough Sets and Some Aspects of Expert Systems Realization*, Proc. of the International Workshop on Expert Systems and their Applications, Avignon, France, pp. 597-611, 1987.
[12] Z. Pawlak, *Information Systems-Theoretical Foundations*, Information Systems, Vol. **6**, pp. 205–218, 1981.
[13] Z. Pawlak, *Rough Sets*, International Journal of Information and Computer Sciences, Vol. **11**, No. 5, pp. 341–356, 1981.
[14 Z. Pawlak, *Rough Sets and Fuzzy Sets*, J. of Fuzzy Sets and Systems, **17**, pp. 99–102, 1985.
[15] Z. Pawlak, R. Słowiński, K. Słowiński, *Rough Sets Based Decision Algorithm for Treatment of Duodenal Ulcer by HSV*, Bull. Polish Accad. Sci. Biology, Vol. **34**, No. 10–12, pp. 227–246, 1987.
[16] S. K. M. Wong, Ye Le, W. Ziarko, *Comparison of Rough Sets and Statistical Methods in Inductive Learning*, Int. J. Man-Matime Studies, Vol. **24**, pp. 53–72, 1986.
[17] S. K. M. Wong, W. Ziarko, *Comparison of the Probabilistic Approximate Classification and the Fuzzy Set Model*, Fuzzy Sets and Systems, Vol. **21**, pp. 357–362, 1987.
[18] L. Zadeh, *Fuzzy Sets*, Information and Control, **8**, pp. 338–353, 1965.

Zbiory przybliżone i systemy informacyjne

W pracy przedstawiono podstawowe idee dotyczące pojęcia zbiorów przybliżonych i ich wykorzystania do analizy danych.

Szczegółowo przedstawiono wykorzystanie zbiorów przybliżonych do analizy danych reprezentowanych w postaci systemu informacyjnego.

Zamieszczono przykłady ilustrujące wprowadzane pojęcia i metody analizy danych.