Forschungsinstitut für anwendungsorientierte Wissensverarbeitung

an der Universität Ulm

Forschungsinstitut für anwendungsorientierte Wissensverarbeitung

Workshop "Mathematics and AI"
Vol. II

Schloß Reisensburg (Günzburg), 19th-22nd December 1988

Program / Abstracts

FAW - Bericht B -89008
Mai 1989

Organizing Committee:

Prof. Dr. P. Hammer
RUTCOR
New Brunswick
New Jersey 08903
USA

Prof. Dr.Dr. F. J. Radermacher

FAW Ulm
Postfach 2060
D-7900 Ulm

Prof. Dr. M. M. Richter
FB Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

Helmholtzstraße 16
Postfach 2060
D-7900 Ulm/Donau

Prof. Dr. Z. Pawlak
Rough sets and their applications

On data reduction and analysis - A rough set approach
Rough sets - basic concepts and definitions

ON DATA REDUCTION AND ANALYSIS - A ROUGH SET APPROACH

ZDZISŁAW PAWLAK

WARSAW

1. Introduction

The paper describes basis theoretical ideas underlying the application of rough set concept to imprecise data reduction and analysis, although no reference to any specific application is mentioned. Our claim is that data reduction and analysis is of primary importance in many areas of artificial intelligence. For example machine learning, expert systems, decision support systems, decision tables and others may be considered in this scope, and have proved to be very suitable domains of rough set based analysis. (cf. Arciszewski and Ziarko (1986)), Fibak at al (1986), Mrózek (1987), Słowiński at al (1988), Ziarko and Katzber (1989).

The departure point of our considerations is the notion of an information system (cf. Pawlak (1982)), called also sometimes knowledge representation system or attribute-value system.

We are going to show in this paper basic properties of information systems having in mind the above said applications.

2. Information Systems

Information system can be perceived as a data table, columns of which are labelled by attributes rows are labelled by objects (states, processes etc.) and each row represents an information about the corresponding object.

The data table can be obtained as a result of measurements, observations or represent knowledge of an agent or a group of agents. The origin of the data table is not important from our point of view and we shall be interested only in some the formal properties of such tables.

Informally speaking by an information system we mean a finite collection of data about some objects (states, processes etc.). We assume that objects are characterized by some features expressed as pairs (attribute, value). For example the following pairs (color, red), (height, tall), (sex, male), (age,young) are possible features of some objects.

Main problems we are going to deal with consist in discovering dependencies among data, reducing data redundancies in the tables and generating decision rules. In other words we are interested in detecting attribute dependencies and reducing the set of attributes .

It turns out that many problems of fundamentally different nature can be reduced to the above mentioned data table analysis. Machine learning, pattern recognition, decision table or expert systems are exemplary applications problems where the proposed approach seems to give novel insight and algorithms.

3. Formal definition of an Information System

An information system is a quadruple $S = (U, A, V, f)$ where

U – is a nonempty, finite set called the *universe*.

A – is a finite set of *attributes*.

$V = \bigcup_{a \in A} V_a$; V_a – is called the *domain* of attribute a.

$f : U \times A \longrightarrow V$ – is an *information function* (total), such that $f(x, a) \in V_a$ for every $a \in A$ and $x \in U$.

In what follows we shall need the notion of information about an object – in the information system – which is defined below:

Let $x \in U$. The function $f_x : A \longrightarrow V$, such that $f_x(a) = f(x, a)$ for every $a \in A$ will be called information on x in S.

Thus information on x is simply the set of values of an attributes assigned to an object x, or in other words – description of object x in terms of attributes available in the information system.

The example which follows will illustrate the definition.

Example 1. Let us consider the following information system.

U	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

Table 1

The universe U consist of 8 elements numbered 1,2,3,4,5,6,7 and 8, the set of attributes is A = {a,b,c,d,e}, whereas

$$V = V_a = V_b = V_c = V_d = V_e = \{0,1,2\}.$$

■

4. Indiscernibility Relation

It should be quite clear that some objects may have identical values of some attributes, i.e.they cannot be distinguished by attributes. This observation is fundamental one in our approach, and it is used to define the indiscernibility relation, which is the basis of rough set philosophy.

Let us express this more formally.

Let $S = (U, A, V, f)$ be an information system and let $P \subseteq A$. By $IND(P)$ we shall denote a binary relation over U defined as: $(x, y) \in IND(P)$ if and only if $f_x(a) = f_y(a)$ for every $a \in P$.

It is easily seen that $IND(P)$ is an equivalence relation for every P. Thus every subset of attributes generates the indiscernibility relation in the information system, i.e. elements of U having the same values of attributes of P are indiscernible by the values of attributes P.

The family of all equivalence classes of the relation $IND(P)$ will be denoted by P^*, and elements of P^* are referred to as *blocks* or *indiscernibility classes* of P^*. An equivalence class of the relation $IND(P)$ containing the element x is denoted by $[x]_P$. An example of such indiscernibility relation, generated by the information system shown in Table 1 is given next.

Example 2. In the table 1 elements 1,4 and 5 of U are indiscernible by attribute a, elements 2,7 and 8 are indiscernible by attributes b and c, and elements 2 and 7 are indiscernible by attributes d and e.

Exemplary partitions generated by attributes in this system are given below.

$a^* = \{\{2,8\},\{1,4,5\},\{3,6,7\}\}$

$b^* = \{\{1,3,5\},\{2,4,7,8\},\{6\}\}$

$\{c,d\}^* = \{\{1\},\{2\},\{3,6\},\{2,7\},\{4\},\{5\},\{8\}$

$\{a,b,c\}^* = \{\{1,5\},\{2,8\},\{3\},\{4\},\{6\},\{7\}\}$

■

The following are easy properties of indiscernibility relations:

Proposition 1

(a) $IND(P) = \bigcap_{a \in P} IND(a)$, for every $P \subseteq A$

(b) $IND(P \cup Q) = IND(P) \cap IND(Q)$

(c) If $P \subseteq Q$ then $IND(Q) \subseteq IND(P)$

(d) $IND(\emptyset) = U \times U$

(e) $[x]_P = \bigcap_{a \in P} [x]_a$.

■

5. Approximations of Sets

Having defined the indiscernibility relation, we are able to define now the concepts of lower and upper approximations.

Let $P \subseteq A$ and $X \subseteq U$. The *P-lower approximation of* X, denoted $\underline{P}X$, and the *P-upper approximation* of X, denoted $\bar{P}X$ are, defined as below.

$$\underline{P}X = \bigcup\{Y \in P^*: Y \subseteq X\}$$

$$\bar{P}X = \bigcup\{Y \in P^*: Y \cap X \neq \emptyset\}$$

The *boundary* of X is defined as $BN_P(X) = \bar{P}X - \underline{P}X$.

Set $\underline{P}X$ is the set of all elements of U which can be with certainty classified as elements of X, employing the set of attributes P; $\bar{P}X$ is the set of elements of U which can be *possibly* classified as elements of X, using the set of attributes P. The set $BN_P(X)$ is the set of elements which cannot be classified either to X or to $-X$ using the set of attributes P.

If $\underline{P} \neq \overline{P}$ we say that X is *P-rough* (or *rough* if P is understood); otherwise X is *P-exact*. We shall also, use terminology – *rough (exact) with respect to P.*

Example 3. Consider the data table 1, the set of attributes $C = \{a,b,c\}$ and the subset of the universe $X = \{1,2,3,4,5\}$. Then $\underline{C}X = \{1,2,3,4,5\}$, $\overline{C}X = \{1,2,3,4,5,8\}$ and $BN_C(X) = \{2,8\}$.

Thus the set X is rough with respect to the attributes C, which is to say that we are unable to decide whether elements 2 and 8 are members of the set X or not. For the rest of the universe classification of elements, using the set C of attributes, is possible.

■

One can easly prove the following properties of approximations.

Proposition 2.

1) $\underline{P}X \subseteq X \subseteq \overline{P}X$
2) $\underline{P}\emptyset = \overline{P}\emptyset = \emptyset$; $\underline{P}U = \overline{P}U = U$
3) $X \subseteq Y$ implies $\underline{P}X \subseteq \underline{P}Y$ and $\overline{P}X \subseteq \overline{P}Y$
4) $\underline{P}\ (X \cup Y) = \overline{P}X \cup \overline{P}Y$
5) $\underline{P}(X \cap Y) = \underline{P}X \cap \underline{P}Y$
6) $\underline{P}(X \cup Y) \supseteq \underline{P}X \cup \underline{P}Y$
7) $\overline{P}(X \cap Y) \subseteq \overline{P}X \cap \overline{P}Y$
8) $\underline{P}(-X) = -\overline{P}X$

9) $\bar{P}(-X) = -\underline{P}X$

10) $\underline{P}\underline{P}X = \bar{P}\underline{P}X = \underline{P}X$

11) $\bar{P}\bar{P}X = \underline{P}\bar{P}X = \bar{P}X$

■

6. Reduction of Attributes

Application of rough set to data reduction and analysis consists in comparison of classifications (partitions) induced by various sets of attributes.

The question we are going to discuss in this section is whether some attributes can be abandoned in the information system without loose of information about objects. Let us discuss this problem in some detail.

In many practical applications we are interested in reducing those attributes which are redundant with respect to the whole set of attributes. The classification of objects obtaining in the absence of such attributes is as good, in the sense of preserving the original classification, as the classification based on all attributes. To this end we introduce the concept of a reduct, that is, of a subset of attributes which is characterized by the following conditions:

1. it preservers the original classification,
2. none of the attributes can be removed from the reduct without destroying the property 1.

In what follows we introduce the necessary definitions in more systematic and precise terms.

Let $S = (U, A, V, f)$ be an information system and let $P, Q \subseteq A$, $a \in P$.

By a *P-positive region* of the partition Q^*, denoted $POS_P(Q^*)$, we understand the set

$$POS_P(Q^*) = \bigcup_{X \in Q^*} \underline{P}X$$

We say that a is *Q-indispensable in P*, if

$$POS_P(Q^*) \neq POS_{P-\{a\}}(Q^*);$$

otherwise a is *Q-dispensable in P*.

If all attributes in P are *Q-indispensable* we shall say that set of attributes P is *Q-independent*.

The set of all Q-indispensable attributes in P will be called a *Q-core* of P, and will be denoted as $\mathrm{CORE}_Q(P)$.

Set $R \subseteq P$ will be called a *Q-reduct* of P, if R is Q-independent subset of P, and $POS_R(Q^*) = POS_P(Q^*)$.

The following property explains the relationship between the core and reduces of attributes.

Proposition 3.

$$\mathrm{CORE}_Q(P) = \bigcap_{R \in \mathrm{RED}_Q(P)} R$$

where $\mathrm{RED}_Q(P)$ the family of all Q-reducts of P.

■

Let us briefly comment the above defined notions.

Set $\mathrm{POS}_P(Q^*)$ is the set of all objects which can be classified to blocks of the partition (Q^*) employing the set of attributes P.

The set of attributes is independent if non its attributes can be removed without changing the classification provided by the whole set of attributes.

The core is the set of most important attributes and reduct of the set of attributes is its subset which provides the same ability to classify objects as the whole set of attributes.Let us observe the the set of attributes can have more then one reduct.

In the example we below provide simple illustration of the introduced notions.

Example 4. Let us consider the table 2, shown below where B = {a,b,c,d} and C = {e,f}.

U	a	b	c	d	e	f
1	1	0	0	1	1	2
2	1	0	0	0	1	2
3	1	1	0	1	1	2
4	0	0	0	0	0	2
5	0	0	0	1	0	2
6	1	0	0	2	1	1
7	1	1	0	2	1	1
8	2	1	0	2	1	1
9	2	1	1	2	1	0
10	2	2	1	2	1	0
11	2	2	1	1	1	0
12	2	1	1	1	1	0
13	2	0	1	1	1	0

Table 2

It is easy to verify that the set B of attributes is B-independent but it is C- dependent. The only C-reduct of attributes B is the set {a,c,d}, which is at the same time the C- core of B ■

We end this section with the following important remark.

The idea of the reduct can be modified to provide for further reduction of attributes. Let $Q^* = \{X_1, ..., X_n\}$ be the partition (classification) generated by the set of attributes Q. We can apply the concept of a reduct of attributes to distinguish the class X_i from the remaining classes $X_1, ..., X_{i-1}, X_{i+1}, ..., X_n$, thus obtaining the set of attributes characteristic to each class X_i. This procedure leads to n reducts (possibly different) R_i each of which is associated with the class X_i. Of course in general case each class can have more then one reduct. This kind of reduct will be called a binary reduct. Binary reduct are particularly useful in decision rules generation and deep simplification of decision tables (cf. Ziarko (1987)). A detail discussion of this problem is left to the reader.

7. Dependency of Attributes

The dependency of attributes is the fundamental concept in the presented approach.

Intuitively speaking subset of attributes Q depends on subset of attributes A, if values of attributes in A are uniquely determined by values of attributes in P, i.e. if there exists a function which assigns to each set of values of P set values of Q. This means that dependency can be also explained in terms of classification, i.e.if Q depends on P that is we are able to classify objects to blocks of Q^* using the set of attributes P.Hence to define the dependency of attributes we can use the concept of the positive region of a set as shown below.

Let $S = (U, A, V, f)$ be an information system and $P, Q \subseteq A$. We say that the set of attributes Q *depends* in a degree k $(0 \le k \le 1)$ on the set of attributes P (in S), symbolically $P \longrightarrow^{k} Q$, if

$$k = \gamma_P(Q^*) = \frac{card\ POS_P(Q^*)}{card\ (U)}$$

where card denotes cardinality of the set.

If $k = 1$ we will say that Q *depends totally* on P (or in short depends); if $0 < k < 1$, we say that Q *depends roughly (partially)* on P, and if $k = 0$ we say that Q is *totally independent* of P. If $P \longrightarrow^{1} Q$ we shall also write $P \dashrightarrow Q$. That is, if $k = 1$ then we have total functional dependency among corresponding attributes. If $0 < k < 1$ then the functional dependency is confined to some , but not all objects in the table. That kind of dependency can be also referred to as partial functional dependency. Finally, if $k = 0$ none of the values of the attributes in P are sufficient to determine corresponding values of attributes in Q. In this case there is entirely no functional dependency between P and Q. The idea of functional dependency can be also interpreted in terms of our ability to classify objects. More precisely, from the definition of dependency follows that if $P \dashrightarrow^{k} Q$ then the positive region of the partition Q^* induced by Q covers k*100 percent of all objects represented in the table. On the other hand only those objects belonging to positive region of the partition can be uniquely classified. This means that k*100 percent of objects can be classified into blocks of partition Q^* based on values of attributes belonging to P. Thus the coefficient $\gamma_P(Q^*)$ can be understood as a degree of dependency between Q and P. In other words if we restrict the set of objects in the information system $S = (U, A, V, f)$ to the set $POS_P Q^*$, we would obtain the system $S' = (POS_P Q^*, A, V, f)$ in which $P \longrightarrow Q$ is a total functional dependency. Of course one could use another measure of rough dependency but the one assumed here seems to be very well suited to various applications and it is also easy to compute and interpret. The measure k of dependency $P \dashrightarrow^{k} Q$ does not capture how actually this partial dependency is distributed among decision classes. For example some decision classes can be fully characterized by attributes in P whereas others may be characterized only partially. To this end we will need also a coefficient $\gamma_P(X) = card\ \underline{P}X / card\ X$, where $X \in Q^*$, which says how elements of each class of Q^* can be classified by employing only the set of attributes P. Thus the two numbers $\gamma_P(Q^*)$ and $\gamma_P(X)$, $X \in Q^*$ give us full information about the "classification power" of the set at attributes P with respect to the classification Q^*.

Example 5. Let us compute the degree of dependency of attributes $D = \{d,e\}$ from the attributes $C = \{a,b,c\}$ in the table 1. The partition D^* consists of the following blocks, $X_1 = \{1\}$, $X_2 = \{2,7\}$, $X_3 = \{3,6\}$, $X_4 = \{4\}$, $X_5 = \{5,8\}$ and the partition C^* consists of blocks $Y_1 = \{1,5\}$, $Y_2 = \{2,8\}$, $Y_3 = \{3\}$, $Y_4 = \{4\}$, $Y_5 = \{6\}$ and $Y_6 = \{7\}$. Because $\underline{C}X_1 = \emptyset$, $\underline{C}X_2 = Y_6$, $\underline{C}X_3 = Y_3 \cup Y_5$, $\underline{C}X_4 = Y_4$ and $\underline{C}X_5 = \emptyset$, thus $POS_C(D^*) = Y_3 \cup Y_4 \cup Y_5 \cup Y_6 = \{3,4,6,7\}$. That is to say that only these elements can be classified into blocks of the partition D^* employing the set $C = \{a,b,c\}$ attributes. Hence the degree of dependency between Q and P is $\gamma_C(D^*) = 4/8 = 0{,}5$.

■

8. Decision Tables

In this section we will consider special, important class of information systems, called *decision tables*. Basics of the decision tables can be found in Hurley 1983.

Decision table is a finite set of decision rules, which specify what decisions (actions) should be undertaken when some conditions are satisfied.

It turns out that the information system provides a very good framework as a basis of decision tables theory.

Decision tables can be defined in terms of information system as follows.

Let $S = (U,A,V,f)$ be an information system and $C,D \subset A$ two subsets of attributes such that $C \cap D = \emptyset$ and $C \cup D = A$, called *condition* and *decision attributes* respectively. Information system S with distinguished condition and decision attributes will be called a *decision table*, and will be denoted $S = (U,C,D,V,f)$.

Equivalence classes of the relations $IND(C)$ and $IND(D)$

will be called *condition* and *decision classes*, respectively.

The function $f_x : A \longrightarrow V$, such that $f_x(a) = f(x,a)$, for every $a \in A$, $x \in U$ will be called a *decision rule* (in S). If g is a decision rule, then the restriction of g to C, denoted $g|C$, and the restriction of g to D, denoted $g|D$ will be called *conditions* and *decisions (actions)* of g respectively.

The decision rule is *deterministic* (in S) if for every $y \neq x$, $f_x|C = f_y|C$ implies $f_x|D = f_y|D$; otherwise f_x is *nondeterministic*.

A decision table is *deterministic (consistent)* if all its decision rules are deterministic; otherwise a decision table is *nondeterministic (inconsistent)*.

The following is the important property that establishes relationship between determinism (consistency) and dependency of attributes in a decision table.

Proposition 4. A decision table $S = (U,C,D,V,f)$ is deterministic (consistent) if and only if $C \longrightarrow D$.

■

9 Simplification of Decision Tables

Simplification of decision tables is of primary importance in many applications. Example of simplification is the reduction of condition attributes in a decision table. In the reduced decision table the same decisions can be based on smaller number of conditions. This kind of simplification eliminates the need for checking unnecessary conditions or, in some applications, performing expensive test to arrive at a conclusion which eventually could be achieved by simpler means. Simplification of decision tables has been investigated by many authors(cf.Hurley(1983)), and there is a variety of informal approaches, to this problem.

Let us also mention that the simplification of boolean functions in the context of digital circuits design (cf.Muroga 1973)) may be also viewed as a simplification of decision tables.

We should note that in contrast to the general notion of an information system rows do not represent here any real objects. Consequently duplicate rows can be eliminated as they correspond to the same decision.

Thus the proposed method consists in removing superfluous condition attributes (columns), duplicate rows and in addition to that irrelevant values of condition attributes.

In this way we obtain "incomplete" decision table, containing only those values of condition attributes which are necessary to make decisions.

From mathematical point of view, removing attributes and removing values of attributes are alike and will be explained in what follows.

For the sake of simplicity we assume that the set of condition attributes is already reduced, i.e. there are not superfluous condition attributes in the decision table.

As we mentioned in before with every subset of attributes P we can associate the partition P^*, and consequently set of condition and decision attributes define partitions of objects into condition and decision classes.

Because we want to discern every decision class using minimal number of conditions - our problem can be reduce now to searching for reducts of condition classes with respect to decision classes. To this end we can use similar methods to that of finding reducts of attributes.

Now we define all necessary notions needed in this section.

Suppose we are given a family of set $F = \{X_1, \ldots, X_n\}$, $X_i \subseteq U$ and a subset $Y \subseteq U$, such that $\bigcap F \subseteq Y$.

We say that X_i is *Y-dispensable* F, if $\bigcap(F-\{X_i\}) \subseteq Y$; otherwise the set X_i is *Y-indispensable* in F.

A family F is Y-independent if each its set is *Y*-indispensable.

A family $H \subseteq F$ is *y-core* of F if H is the family of all *Y*-indispensable sets in F.

A family $H \subseteq F$ is a *Y-reduct* of F, if H is *Y*-ind*ependent* $\bigcap H \subseteq Y$.

As we can see the introduced definitions again differ from the one discussed previously only in this regard that instead of relations we deal now with sets.

Let us also notice that the counterpart of Propositions 2. is also valid in the present framework as shown below.

Proposition 5.

$$\bigcap_{H \in RED_Y(F)} H = CORE_Y(F)$$

where $RED_Y(F)$ is the family of all reducts Y-reducts of F.

■

Now we are in a position to explain how to reduce superfluous values of condition attributes form a decision table.

From Proposition 1 it follows that with every subset of attributes $P \subseteq A$ and object x we may associate set $[x]_P$. Thus with each row labelled by object x in the decision table, and set of condition attributes C, we may associate set $[x]_C = \bigcap_{a \in C} [x]_a$. But each set $[x]_a$ is uniquely determined by attribute value $f(x,a)$, hence in order to remove superfluous values of condition attributes we have to eliminate all superfluous equivalence classes from the equivalence class $[x]_C$. Thus problems of elimination of superfluous values of attributes and elimination of corresponding equivalence classes are similar.We shall illustrate this idea by means of the following example:

Example 6. Let us consider the decision Table 3, where a,b,c and d are condition attributes and e and f are decision attributes, denoted respectively by C and D.

U	a	b	c	d	e	f
1	1	0	0	1	1	2
2	1	0	0	0	1	2
3	1	1	0	1	1	2
4	0	0	0	0	0	2
5	0	0	0	1	0	2
6	1	0	0	2	1	1
7	1	1	0	2	1	1
8	2	1	0	2	1	1
9	2	1	1	2	1	0
0	2	2	1	2	1	0
1	2	2	1	1	1	0
2	2	1	1	1	1	0
3	2	0	1	1	1	0

Table 3

It is easy to check that the set of condition attributes is C-independent, but it is D-dependent. The only D-reduct of condition attributes is the set $\{a,c,d\}$.

Thus after removing duplicate decision rules the decision Table 3 can be simplified as shown below.

U	a	c	d	e	f
1	1	0	1	1	2
2	1	0	0	1	2
4	0	0	0	0	2
5	0	0	1	0	2
7	1	0	2	1	1
8	2	0	2	1	1
9	2	1	2	1	0
11	2	1	1	1	0

Table 4

Let us also remark that both decision tables 3 and 4 are deterministic.

In the decision table there are four kinds of possible decisions. The possible decision are specified by the following pairs of values of decision attributes e and f : (1,2), (0,2), (1,1) and (1,0), denoted in what follows by I, II, III and IV, respectively.

Thus we can represent decision Table 4 as shown in the table 5.

U	a	c	d	e f
1	1	0	1	I
2	1	0	0	
4	0	0	0	II
5	0	0	1	
7	1	0	2	III
8	2	0	2	
9	2	1	2	IV
11	2	1	1	

Table 5

It is easy to check that in the first row we have two indispensable attribute-values a_1 and d_1 (a_i denotes value i of attribute a) and $\{a_1, d_1\}$ is the core and also the reduct of the set corresponding to this row.

Proceeding in the same way with the remaining rows and removing all duplicate rows we obtain the following table of value cores:

U	a	c	d	e	f
1	1		1		I
2	1		0		
4	0				II
7			2		III
8		0			
9		1			IV

Table 6

It can be easily seen that in the decision classes I, II and IV core values in each row are also reducts of values. For the decision class III however values a_2, c_0 do not form value reducts of corresponding rows. For each row in this class we have two reducts , thus we have four possible combinations of value reducts in this class,i.e. there are for possible simplification of the decision table.

The simplest solution is shown in the table below.

U	a	c	d	e	f
1	1	×	1		I
2	1	×	0		
4	0	×	×		II
8	×	0	2		III
9	×	1	×		IV

Table 7

Crosses in the table denote "don't care" values of attributes.Table 7 could be also obtained using the binary reduct, the remaining three solutions however can not be obtained in this way.

■

In summary, to simplify a decision table we should first find reducts of conditions attributes, remove duplicate rows and then find value-reducts of condition attributes and again , if necessary, remove duplicate rows .This method leads to a simple algorithm for decision table simplification or generation of decision rules (algorithms) from examples, which, according to our experiments, out performs other methods, in terms of achievable degree in the number of conditions and what more , gives all possible solutions to the problem.

We conclude this section with the following remark. Because, in general, a subset of attributes may have more that one reduct the simplification of decision tables does not yield unique results. The table possibly can be also optimized according to pre assumed criteria.

10. Conclusion.

As mentioned at the very begining of this article basic concept of the presented approach is that of the partition of objects of universe of discourse induced by data (attribute-value characterisation of objects). Next these partitions are used to define approximations of sets, and further reduction and dependences of attributes - which form fundamental set of mathematical concepts needed to reason from data. Reasoning in the proposed framework is rather semantical then syntactical in its nature, for we do not employ syntactical rules of inference, like in logic, but we refer to meaning of data being consider. Let us also observe that the we deal rather with qualitative then quantitative data here, since in order to discribe partitions, numbers are not essential and it is enough to use in this case qualitative concepts like "small"," big" , etc, which is very important in many applications.

References

Arciszewski, T. & Ziarko,W, (1986). Adaptive Expert System for Preliminary Engineering Design. *Proc. 6-th International Workshop on Expert Systems and their Applications*, Avignon,France,695-712.

Boryczka, M. & Słowiński, (1989). Properties of Optimal Decision Algorithms from Decision Tables Using Rough Sets. *Bull. Polish Acad. Sci. Tech.* (to appear).

Fibak, J.Słowiński, K. & Słowiński R. (1986).The Application of Rough Set Theory to the Verification of Indications for Treatment of Duodenal Ulcer By HSV. *Proc. 6-th International Workshop on Expert Systems and their Applications*.Avignon, France, 587-599.

Hunt, E.D., Martin, J. & Stone, P.J. (1986). Experiments in

Induction. *Academic Press, New York.*

Hurley, R.B. (1983). Decision Tables in Software Engineering. *Van Nostrand Reinhold Company, New York.*

Montalbano, M. (1974). Decision Tables. *Science Research Associates.* Chicago 1974.

Mrózek, A. (1987).Rough Sets and Some Aspects of Expert Systems Realizations.*Proc.7 -th International Workshop on Expert Systems and their Applications.*Avignon,France,597-611.

Muroga, S. (1979). Logic Design and Switching Theory. *Wiley,* New York.

Novotny, M. & Pawlak, Z. (1983). On a Representation of Rough Sets by Means of Information Systems. *Fundamenta Informaticae* 6 (3-4), pp.286-296.

Novotny, M. & Pawlak, Z. (1989). Partial Dependency of Attributes. *Bull. Polish Acad. Sci. Math.* (to appear).

Novotny, M. & Pawlak, Z. (1989). Independence of Attributes. *Bull.Polish Acad.Sci.Math.* (to appear).

Orłowska, E. (1983). Dependencies of attributes in Pawlak's Information Systems. *Fundamenta Informaticae* 6 (3-4), pp.247-256.

Pawlak, Z. (1981). Information Systems-Theoretical Foundations. *Information Systems* 6, 205-218.

Pawlak, Z. (1982). Rough Sets. *International Journal of Information and Computer Sciences* 11, 341-356.

Pawlak, Z. (1985a). Rough Sets and Decision Tables. *Lecture Notes in Computer Science* 208. Springer Verlag pp.186-196.

Pawlak, Z. (1985b). On Rough Dependency of Attributes in Information Systems. *Bull. Polish Acad. Sci. Tech.* 33, pp.551-559.

Pawlak, Z. Słowiński, K. & Słowiński, R (1986). Rough Classification of Patients after Highly Selective Vagotomy for Duodenal Ulcer. *Int. J. Man-Machine Studies*. 24, 413-433.

Pawlak, Z. (1987). Decision Tables - A Rough Sets Approach. *Bull. of EATCS* 33, 85-96.

Rauszer, C.M. (1989). Algebraic Properties of Functional Dependencies. *Bull. Polish Acad. Sci. Math.* (to appear).

Słowiński, R. & Stefanowski, J. (1988). Rough Classification in Incomplete Information Systems. *Mathematical Modeling* (to appear).

Wille, R. (1982). Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. *In: I.Rival (Ed.). Ordered Sets, Reidel*, Dordrecht - Boston, pp.445-470.

Wong, S.K.M. & Ziarko, W. (1986). On Optimal Decision Rules in Decision Tables. *Bull. Polish Acad. Sci. Math.* 33 (11-12), pp.693- 696.

Ziarko, W. (1987).On Reduction of Knowledge Representation.*Proc. 2-nd International Symp.on Methodologies of for Intelligent Systems*. Charlotte,NC.North Holland.

Ziarko, W.(1988).& Katzberg J.D. (1989) Control Algorithm Acquisition, Analysis and Reduction: a Machine Learning Approach.*Knowledge Based System Diagnosis, Supervision and Control,Plenum Press*, (to appear).