Data versus Logic A Rough Set View

by

Zdzisław Pawlak

ICS Research Report 28/96
Data versus Logic A Rough Set View

Zdzislaw Pawlak
Institute of Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Baltycka 5, 44 000 Gliwice, Poland

zpw@ii.pw.edu.pl

Abstract: In this paper we consider some aspects of decision rule generation methods based on rough set theory. Particularly, we propose to associate with every set of decision rules a connection graph, similar to that used in switching theory, so that sets of decision rules can be represented as a kind of a switching circuit, which depicts relations between decision rules. The connection graph can be understood as a visual representation (explanation) of relations in data, or as a specification of a discrete dynamic system (e.g., controller, program, etc.). The connection graph can be also viewed as an alternative to the well known Petri Nets concept for concurrent systems analysis.

Keywords: Rough sets, decision rules, connection graph, Petri Nets.

1 Introduction

The paper contains a discussion on the relationship between data and its logical representation in a form of a decision rules – from the rough set perspective.

The rough set concept has been introduced by the author in [14]. Rough set theory has developed during the past fourteen years to a mature discipline with many real-life applications to its credit, e.g., medicine, pharmacology, market analysis, engineering, pattern recognition, data bases and many others [2,3,5,6,7,8,9,10,11,13,17,20,22,23,27,30,31,33]. Basics of rough set theory can be found in [15,16,21,25].

Data can represent results of observations, measurements, knowledge of an agent or group of agents, etc. Decision rules derived from data reveal hidden patterns in the data, and can be understood as description of laws governing the observed phenomena or processes. In other words decision rules are meant to represent a "cause-effect" relations in the observed process. It is well known that a relationship between data not necessarily implies corresponding relationship in reality. We will however refrain from philosophical discussion of this issue here.

The topic of discovering rules from data has been pursued for a long time in statistics, but many new methodologies based on some AI techniques, e.g., machine learning, data mining, emerged recently. Rough sets proved to be a valuable tool in this area as well, and many rough set based methods of rule induction from data have been developed [1,5,18,24,26,28,29,34].

In this paper we are going to analyze decision rule generation methods based on rough set theory. Besides, with every set of decision rules a connection graph will be associated. To this end notation similar to that used in the switching theory will be employed, so that the set of decision rules can be represented as a switching circuits, which depicts relations between decision rules. The connection graph can be understood as a visual representation (explanation) of the relations in data or as a specification of a discrete dynamic system (e.g., controllers).

Decision tables are used in software engineering [12] for program specification. It seems that the rough set approach can be as well deployed to this aim.

The presented approach can be also seen as an alternative to the well known Petri Nets model of concurrent system behaviour [19].

2 Information Tables, Decision Tables and Decision Rules

Data are often represented as a table, columns of which are labeled by attributes, rows by objects of interest and entries of the table are attribute values. Such tables are known as information systems, attribute-value tables or information tables. We will use here the term information table. Below an example of information table is shown.

Suppose we are given data about 6 patients, as shown in Table 1.

<table>
<thead>
<tr>
<th>Patient</th>
<th>headache</th>
<th>Muscle pain</th>
<th>Temperature</th>
<th>Flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>no</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p2</td>
<td>yes</td>
<td>no</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p3</td>
<td>yes</td>
<td>yes</td>
<td>very high</td>
<td>yes</td>
</tr>
<tr>
<td>p4</td>
<td>no</td>
<td>yes</td>
<td>normal</td>
<td>no</td>
</tr>
<tr>
<td>p5</td>
<td>yes</td>
<td>no</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>p6</td>
<td>no</td>
<td>yes</td>
<td>very high</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 1
Columns of the table are labelled by attributes (symptoms) and rows – by objects (patients), whereas entries of the table are attribute values. Thus each row of the table can be seen as information about specific patient. For example, patient p2 is characterized in the table by the following attribute-value set:

(Headache, yes), (Muscle-pain, no), (Temperature, high), (Flu, yes),

which form information about the patient.

Sometimes we distinguish in an information table two classes of attributes, called condition and decision (action) attributes. For example in Table 1 attributes Headache, Muscle-pain and Temperature can be considered as condition attributes, whereas the attribute Flu – as a decision attribute.

Each row of a decision table determines a decision rule, which specifies decisions (actions) that should be taken when conditions pointed out by condition attributes are satisfied. For example in Table 1 the condition (Headache, no), (Muscle-pain, yes), (Temperature, high) determines uniquely the decision (Flu, yes). Decision rules are often presented as implications and are called "if... then..." rules. For example, rule 1) in Table 1 can be presented as implication

if (Headache, no) and (Muscle-pain, yes) and (Temperature, high), then (Flu, yes).

With each decision table we can associate the set of all decision rules determined by the table. Hence with Table 1 we can associate the following decision rules:

if (Headache, no) and (Muscle-pain, yes) and (Temperature, high), then (Flu, yes);
if (Headache, yes) and (Muscle-pain, no) and (Temperature, high), then (Flu, yes);
if (Headache, yes) and (Muscle-pain, yes) and (Temperature, very high), then (Flu, yes);
if (Headache, no) and (Muscle-pain, yes) and (Temperature, normal), then (Flu, no);
if (Headache, yes) and (Muscle-pain, no) and (Temperature, high), then (Flu, no);
if (Headache, no) and (Muscle-pain, yes) and (Temperature, very high), then (Flu, yes).

Decision rules 2) and 5) in Table 1 have the same conditions but different decisions. Such rules are called inconsistent (nondeterministic, conflicting); otherwise the rules are referred to as consistent (deterministic, nonconflicting). Sometimes consistent decision rules are called sure (certain) rules, and inconsistent rules are called possible rules. Decision tables containing inconsistent decision rules are called inconsistent (nondeterministic, conflicting); if the table do not contain inconsistent rules it is called consistent (deterministic, nonconflicting).

The number of consistent rules to all rules in a decision table can be used as a consistency measure of the decision table, and will be denoted by $\gamma(C,D)$, where $C$ and $D$ are sets of condition and decision attributes respectively. Thus if $\gamma(C,D) = 1$, the decision table is consistent, and if $\gamma(C,D) \neq 1$ the decision table is inconsistent. For example for Table 1, $\gamma(C,D) = 4/6$.

Notice the distinction between decision tables and the corresponding set of decision rules. A decision table is a collection of data, whereas decision rules are implications, e.g., logical expressions. To deal with data we use various mathematical methods, e.g., statistics, but to analyze implications we must employ logical tools.

3 Rough Sets and Approximations

The starting point of rough set theory is an indiscernibility relation, generated by information about objects of interest. The indiscernibility relation is intended to express the fact that due to the lack of knowledge we are unable to discern some objects employing the available information. Let us illustrate this idea by an example.

In Table 1 patients p2, p3 and p5 are indiscernible with respect to the attribute Headache, patients p3 and p6 are indiscernible with respect to attributes Muscle-pain and Flu, and patients p2 and p5 are indiscernible with respect to attributes Headache, Muscle-pain and Temperature.

Patient p2 has flu, whereas patient p5 does not. Because they are indiscernible with respect to the attributes Headache, Muscle-pain and Temperature, hence flu cannot be characterized in terms of attributes Headache, Muscle-pain and Temperature. The remaining patients p1, p3 and p6 display symptoms which enable us to classify them with certainty as having flu, patients p2 and p5 cannot be excluded as having flu and patient p4 for sure does not have flu, in view of the displayed symptoms.

Now we present above considerations more formally.

Suppose we are given two finite, non-empty sets $U$ and $A$, where $U$ is the universe, and $A$ – a set attributes. With every attribute $a \in A$ we associate a set $V_a$, of its values, called the domain of $a$. Any subset $B$ of $A$ determines a binary relation $I(B)$ on $U$, which will be called an indiscernibility relation, and is defined as follows:

$xI(B)y$ if and only if $a(x) = a(y)$ for every $a \in A$, where $a(x)$ denotes the value of attribute $a$ for element $x$.

Obviously $I(B)$ is an equivalence relation. The family of all equivalence classes of $I(B)$, i.e., partition de-
terminated by \( B \), will be denoted by \( U/I(B) \), or simple \( U/B \). any equivalence class of \( I(B) \), i.e., the block of the partition \( U/B \), containing \( z \) will be denoted by \( U(z) \).

If \((x, y) \) belongs to \( I(B) \) we will say that \( x \) and \( y \) are \( B \)-indiscernible. Equivalence classes of the relation \( I(B) \) (or blocks of the partition \( U/B \)) referred to as \( B \)-elementary sets. In the rough set approach the elementary sets are the basic building blocks (concepts) of our knowledge about reality.

The indiscernibility relation will be used to define the following two operations on sets

\[
B_*(X) = \{ x \in U : B(x) \subseteq X \},
\]

\[
B^*(X) = \{ x \in U : B(x) \cap X \neq \emptyset \},
\]

assigning to every subset \( X \) of the universe \( U \) two sets \( B_*(X) \) and \( B^*(X) \), called the \( B \)-lower and the \( B \)-upper approximation of \( X \), respectively. The set

\[
BN_B(X) = B^*(X) - B_*(X)
\]

will be referred to as the \( B \)-boundary region of \( X \).

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

For example, the lower approximation of the set "flu" \( \{p_1, p_2, p_3, p_6\} \), is \( \{p_1, p_3, p_6\} \), the upper approximation of this set is \( \{p_1, p_2, p_3, p_5, p_6\} \), whereas the boundary region is \( \{p_2, p_5\} \). Similarly \( p_4 \) does not have flu and \( p_2, p_5 \) cannot be excludes as having flu, thus the lower approximation of "not flu" \( \{p_4, p_5\} \) is \( \{p_4\} \), the upper approximation is \( \{p_2, p_4, p_5\} \), and the boundary region is \( \{p_2, p_5\} \).

4 Dependence of Attributes

Another important issue in data analysis is discovering dependencies between attributes. Intuitively, a set of attributes \( D \) depends totally on a set of attributes \( C \), denoted \( C \rightarrow D \), if all values of attributes from \( D \) are uniquely determined by values of attributes from \( C \). In other words, \( D \) depends totally on \( C \), if there exists a functional dependency between values of \( D \) and \( C \). In Table 1 there are not total dependencies whatsoever. If in Table 1, the value of the attribute Temperature for patient \( p_6 \) were "no" instead of "high", there would be a total dependency \{Temperature\} \( \rightarrow \{\text{Flu}\} \), because to each value of the attribute Temperature there would correspond unique value of the attribute Flu.

Formally, dependency can be defined in the following way. Let \( D \) and \( C \) be subsets of \( A \). We say that \( D \) depends totally on \( C \), if and only if \( I(C) \subseteq I(D) \). That means that the partition generated by \( C \) is finer than the partition generated by \( D \). Notice, that the concept of dependency discussed above corresponds to that considered in relational databases.

We would need also a more general concept of dependency of attributes, called a partial dependency of attributes. Let us first depict the idea by example, referring to Table 1. In this table, for example, the attribute Temperature determines uniquely only some values of the attribute Flu. That is, (Temperature, very high) implies (Flu, yes), similarly (Temperature, normal) implies (Flu, no), but (Temperature, high) does not imply always (Flu, yes). Thus the partial dependency means that only some values of \( D \) are determined by values of \( C \).

Formally, the above idea can be formulated as follows. Let \( D \) and \( C \) be subsets of \( A \). We say that \( D \) depends in degree \( k, 0 < k < 1 \), on \( C \), denoted \( C \Rightarrow_k D \), if

\[
k = \frac{|POS_C(D)|}{|U|},
\]

where

\[
POS_C(D) = \bigcup_{X \in U/D} C_*(X).\]

The expression \( POS_C(D) \), called a positive region of the partition \( U/D \) with respect to \( C \), is the set of all elements of \( U \) that can be uniquely classified to blocks of the partition \( U/D \), by means of \( C \). It can be easily seen that \( k \) can be interpreted as the consistency measure \( \gamma(C, D) \).

Thus the coefficient \( k \) expresses the ratio of all elements of the universe, which can be properly classified to blocks of the partition \( U/D \), employing set of attributes \( C \). Notice that for \( k = 1 \) we get the previous definition of total dependency.

Obviously, a decision table is consistent if and only if \( k = 1 \), otherwise, i.e., if \( k \neq 1 \), the decision table is inconsistent; if \( k = 0 \) we will say that the decision table is totally inconsistent.

For example, for dependency \{Headache, Muscle-pain, Temperature\} \( \Rightarrow \{\text{Flu}\} \) we get \( k = 4/6 = 2/3 \), because four out of six patients can be uniquely classified as having flu or not, employing attributes Headache, Muscle-pain and Temperature.

If we were interested how exactly patients can be diagnosed using only the attribute Temperature, that is - in the degree of the dependence \{Temperature\} \( \Rightarrow \{\text{Flu}\} \), we would get \( k = 3/5 = 1/2 \), since in this case only three patients \( p_3, p_4 \) and \( p_5 \) out of six can be uniquely classified as having flu. In contrast to the previous case patient \( p_4 \) cannot be classified now as having flu or not. Hence the single attribute Temperature offers worse classification than the whole set of attributes Headache, Muscle-pain and Temperature. It is interesting to observe that neither Headache nor Muscle-pain can be used to recognize flu, because for both dependencies \{Headache\} \( \Rightarrow \{\text{Flu}\} \) and \{Muscle-pain\} \( \Rightarrow \{\text{Flu}\} \) we have \( k = 0 \), i.e., the table is totally inconsistent.
5 Reduction of Attributes and Values of Attributes

We often face a question whether we can remove some data from a data-table preserving its basic properties. Suppose we are given a dependency \( C \Rightarrow D \). It may happen that \( D \) depends not on the whole set \( C \) but on its proper subset \( C' \). In order to investigate this problem we need the notion of a reduct, which will be defined and discussed next.

Let \( C, D \subseteq A \) be a condition and decision attributes respectively.

- We will say that attribute \( a \in C \) is \( D \)-dispensable in \( C \), if \( POS_C(D) = POS_{C\setminus\{a\}}(D) \); otherwise the attribute \( a \) is \( D \)-indispensable in \( C \).
- If all attributes \( a \in C \) are \( C \)-indispensable in \( C \), then \( C \) will be called \( D \)-independent.
- Subset \( C' \subseteq C \) is a \( D \)-reduct (a reduct with respect to \( D \)) of \( C \), iff \( C' \) is \( D \)-independent and \( POS_C(D) = POS_{C'}(D) \).

For example in Table 1 there are two reducts with respect to \( Flu \): \{Headache, Temperature\} and \{Muscle-pain, Temperature\}. That means that either the attribute Headache or Muscle-pain can eliminated from the table and consequently instead of Table 1 we can use either

<table>
<thead>
<tr>
<th>Patient</th>
<th>Headache</th>
<th>Temperature</th>
<th>Flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>no</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p2</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p3</td>
<td>yes</td>
<td>very high</td>
<td>yes</td>
</tr>
<tr>
<td>p4</td>
<td>no</td>
<td>normal</td>
<td>no</td>
</tr>
<tr>
<td>p5</td>
<td>yes</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>p6</td>
<td>no</td>
<td>very high</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 2

or

<table>
<thead>
<tr>
<th>Patient</th>
<th>Muscle-pain</th>
<th>Temperature</th>
<th>Flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p2</td>
<td>no</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p3</td>
<td>yes</td>
<td>very high</td>
<td>yes</td>
</tr>
<tr>
<td>p4</td>
<td>yes</td>
<td>normal</td>
<td>no</td>
</tr>
<tr>
<td>p5</td>
<td>no</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>p6</td>
<td>yes</td>
<td>very high</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 3

where \( C_a = C \setminus \{a\} \);
otherwise the value of attribute \( a \) is \( D \)-indispensable for \( x \).
- If for every attribute \( a \in C \) value of \( a \) is \( D \)-indispensable for \( x \), then \( C \) will be called \( D \)-independent for \( x \).
- Subset \( C' \subseteq C \) is a \( D \)-reduct of \( C \) for \( x \) (a value reduct), iff \( C' \) is \( D \)-independent for \( x \) and

\[ C(x) \subseteq D(x) \text{ implies } C'(x) \subseteq D(x). \]

Using the concept of a value reduct, Table 2 and Table 3 can be simplified as follow

<table>
<thead>
<tr>
<th>Patient</th>
<th>Headache</th>
<th>Temperature</th>
<th>Flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>yes</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p2</td>
<td>no</td>
<td>high</td>
<td>yes</td>
</tr>
<tr>
<td>p3</td>
<td>yes</td>
<td>very high</td>
<td>yes</td>
</tr>
<tr>
<td>p4</td>
<td>no</td>
<td>normal</td>
<td>no</td>
</tr>
<tr>
<td>p5</td>
<td>yes</td>
<td>high</td>
<td>no</td>
</tr>
<tr>
<td>p6</td>
<td>no</td>
<td>very high</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 4

where "-" denotes "don't care values".

We can also present the obtained results as a set of decision rules.

For Table 4 we get

if (Headache, no) and (Temperature, high), then (Flu, yes);
if (Headache, yes) and (Temperature, high), then (Flu, yes);
if (Temperature, very high), then (Flu, yes);
if (Temperature, normal), then (Flu, no); and for Table 5 we have

if (Muscle-pain, yes) and (Temperature, high), then (Flu, yes);
if (Muscle-pain, no) and (Temperature, high), then (Flu, yes);
if (Temperature, very high), then (Flu, yes);
if (Temperature, normal), then (Flu, no);
6 Decision Rules and Connection Graph

Suppose we are given data obtained from observation of certain processes, phenomena etc. Our task is to find out, on the basis of observed data, rules governing the perceived process. That means that we have to find all (or some) decision rules which describe relationship between data. This will be called explanation of data. For example, let us consider Table 1. It follows from this example that the decision rules (Muscle-pain, no) implies (Temperature-high), (Headache, no) implies (Muscle-pain, yes), (Temperature-high) implies (Flu, yes) are valid in the table.

For simplicity of notation let us replace Table 1, by Table 6

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6

where attributes Headache, Muscle-pain, Temperature and Flu are replace by letters a, b, c and d respectively, and values of attributes are replaced by numbers 0,1 and 2 in an obvious way.

More precisely, by explanation we will understand here finding all simple minimal decision rules from data. By a simple decision rule we mean the rule having only one decision attribute, and a decision rule is minimal if all its condition attribute-value pairs are reduced with respect to the decision attribute.

In order to solve this problem we have to compute all partial dependencies of the form $(A - \{a\}) \Rightarrow \{a\}$ for all $a \in A$ and compute all minimal decision rules corresponding to these dependencies. To this end we have to find out all value reducts of each set of attributes $A - \{a\}$ for each object belonging to $POS(A - \{a\})\{a\}$.

For the attribute a we get:

$RED_{a}^{1}(b, c, d) = \{b, c\},$  
$RED_{a}^{2}(b, c, d) = \{b\},$  
$RED_{a}^{3}(b, c, d) = \{c\},$  
$RED_{a}^{4}(b, c, d) = \{b, c\},$

$RED_{a}^{5}(b, c, d) = \{b\}$ means that the attribute-value pair $(b, a)$ determines uniquely attribute-value pair $a_{0}$, i.e., that the decision rule $A \Rightarrow a_{0}$ is valid. Proceeding similarly we get following decision rules:

$b_{0} \rightarrow a_{1},$  
$c_{0} \rightarrow a_{0},$  
$b_{1}d_{0} \rightarrow a_{0},$  
$c_{1}d_{0} \rightarrow a_{1}.$

In the same way for the attribute b we get

$RED_{b}^{1}(a, c, d) = \{a\},$  
$RED_{b}^{2}(a, c, d) = \{a, c\},$  
$RED_{b}^{3}(a, c, d) = \{c\},$  
$RED_{b}^{4}(a, c, d) = \{a, c\},$  
$RED_{b}^{5}(a, c, d) = \{a, c\},$

hence the following decision rules are valid

$a_{0} \rightarrow b_{1},$  
$a_{1}c_{1} \rightarrow b_{0},$  
$c_{2} \rightarrow b_{1},$  
$c_{0} \rightarrow b_{1},$  
$a_{1}d_{0} \rightarrow b_{0},$  
$c_{1}d_{0} \rightarrow b_{0}.$

For the attribute c we have

$RED_{c}^{1}(a, b, d) = \{b\},$  
$RED_{c}^{2}(a, b, d) = \{a, b\},$  
$RED_{c}^{3}(a, b, d) = \{a, b\},$  
$RED_{c}^{4}(a, b, d) = \{b\},$  
$RED_{c}^{5}(a, b, d) = \{b\},$

which yields decision rules

$b_{0} \rightarrow c_{1},$  
$a_{1}b_{1} \rightarrow c_{2},$  
$a_{0}d_{0} \rightarrow c_{0},$  
$b_{1}d_{0} \rightarrow c_{0},$  
$a_{1}d_{0} \rightarrow c_{1}.$

For the attribute d we get

$RED_{d}^{1}(a, b, c) = \{a, c\},$  
$RED_{d}^{2}(a, b, c) = \{c\},$  
$RED_{d}^{3}(a, b, c) = \{c\},$  
$RED_{d}^{4}(a, b, c) = \{c\},$

which gives

$a_{0}c_{1} \rightarrow d_{1},$  
$b_{1}c_{1} \rightarrow d_{1},$  
$c_{2} \rightarrow d_{1},$  
$c_{0} \rightarrow d_{0}.$
The above decision rules can be presented in a more compact form

\[
\begin{align*}
    b_1 c_1 \lor b_1 d_0 \lor c_0 & \rightarrow a_0, \\
    b_0 \lor c_1 d_0 & \rightarrow a_1, \\
    a_1 c_1 \lor a_1 d_0 \lor c_1 d_0 & \rightarrow b_0, \\
    a_0 \lor c_0 \lor c_2 & \rightarrow b_1, \\
    a_2 d_0 \lor b_1 d_0 & \rightarrow c_0, \\
    b_0 \lor a_1 d_0 & \rightarrow c_1, \\
    a_1 b_1 & \rightarrow c_2, \\
    c_0 & \rightarrow d_0, \\
    a_0 c_1 \lor b_1 c_1 \lor c_2 & \rightarrow d_1.
\end{align*}
\]

Relations between all decision rules can be depicted as a connection graph. Vertices of the graph represent attribute-value pairs (drawn in a similar way as multistate devices, e.g., flip-flops, in the case of binary circuits), whereas branches of the graph show connections (implications) between corresponding attribute-value pairs. The connection graph for the above set of decision rules is shown Fig. 1. Thus Table 6 can be seen as a concise denotation of all minimal decision rules by the table.

Table 6 considered in this section can be also understood not as a result of observations but, as a specification of requirements for a special discrete systems behavior. In this case Fig. 1 is to be understood not as a visual aid of "cause-effect" relations occurring in the observed data, but as a schemata of an algorithm specified be the table.

Acknowledgments

This work was supported by grant No. 8 S503 021 06 from State Committee for Scientific Research.

References


Figure 31: Connection Graph
Recently published Research Reports
of the Institute of Computer Science, W.U.T.

13/96 Tomasz Martyn, Frakタルne techniki modelowania zjawisk naturalnych, maj 1996.

14/96 Jerzy Stefanowski, Krzysztof Slowiński, Rough sets as a tool for studying attribute dependencies in the urinary stones treatment data set, June 1996.

15/96 Janusz Sosnowski, Andrzej Kuśmierczyk, Pseudorandom testing of microprocessor, June 1996.

16/96 Salvatore Greco, Benedetto Matarazzo, Roman Slowiński, Rough approximation of a preference relation by dominance relations, June 1996.

17/96 Jerzy Pułaczewski, Jacek Sienkiewicz, A rough set approach to process control - simulations and results, June 1996.

18/96 Ireneusz R. Moraczewski, Druga młodość kanonów Milla. Rzecz o zbiorach przybliżonych, czerwiec 1996.

19/96 Ireneusz R. Moraczewski, Barbara Sudnik-Wójcikowska, Wojciech Borkowski, Rough sets in floristic discription of the inner-city of Warsaw, June 1996.

20/96 Ireneusz R. Moraczewski, Fuzzy rough sets as a tool for ecologioal data analysis, June 1996.

21/96 Anna Derezińska, Algorytmy zbierania sygnatury w systemach transputerowych, czerwiec, 1996.

22/96 Anna Derezińska, Weryfikacja modelu algorytmu rozsiewającego w systemach transputerowych, czerwiec 1996.


24/96 Zdzisław Piasta, Andrzej Lenarcik, Rule induction with probabilistic rough classifiers, August 1996.

25/96 Mariusz Muraszkiewicz, Online interface for WWW servers and CDS/ISIS databases, August 1996.

26/96 Zbigniew Stepaniuk, Properties and applications of rough relations, August 1996.