ELSEVIER

Information Sciences 113 (1999) 271–292

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

Rules in incomplete information systems

Marzena Kryszkiewicz [1]

Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

Received 1 February 1998; received in revised form 26 May 1998; accepted 10 August 1998

Abstract

A new method of computing all optimal certain rules from an incomplete information system is presented and proved. The method does not require changing the size of the original incomplete system. Additionally, several existing rough set methods of computing decision rules from incomplete information systems are analyzed and compared. We show which of these methods are capable of generating all optimal certain rules or a class of optimal certain rules and which methods may lead to generation of false rules.

Keywords: Incomplete information systems; Decision rules; Rough sets

1. Introduction

The problem of knowledge discovering from incomplete information systems is considered. By an incomplete system we mean a system with missing data (null values). We do not consider the case of null value meaning *inapplicable* value. This problem may be solved by adding a special symbol denoting inapplicable value to the attribute domains. In the paper we deal with the problem of *unknown* values.

Several solutions to the problem of generating decision tree from the training set of examples with unknown values have been proposed in the area of Artificial Intelligence (AI). The simplest among them consist in removing examples with unknown values or replacing unknown values with the most

[1] E-mail: mkr@ii.pw.edu.pl

0020-0255/99/$19.00
PII: S0020-0255(98)10065-8

common values. More complex approaches were presented in [1,2]. A Bayesian formalism is used in [1] to determine the probability distribution of the unknown value over the possible values from the domain. This method could either choose the most likely value or divide the object into fractional objects, each with one possible value weighted according to the probabilities determined. It is suggested in [2] to predict the value of an attribute based on the value of other attributes of the object, and the class information.

The problem of rules generation from incomplete systems was also investigated in the context of Rough Sets [3–5]. Modelling uncertainty caused by the appearance of unknown values by means of fuzzy sets was discussed in [4]. Two methods of treating unknown values are available in the LERS system [3]. The first one consists in transforming an incomplete system into a complete system, where each object incompletely described in the source system is replaced by a set of possible subobjects in the target system. This method is hardly applied in practice because of the large size of a new table. The second method is the AI simple method mentioned earlier which reduces the size of the original table by removing objects with unknown values. The methodology proposed in [5] allows to generate generalized rules directly from the original incomplete decision table.

In the paper, a new method of computing all certain rules from an incomplete information system is presented and proved. The method does not require changing the size of the original system. Additionally, selected rough set methods of computing decision rules from incomplete information systems are analyzed and compared. We show which of these methods are capable of generating all certain rules or a class of certain rules and which methods may lead to generation of false rules.

The paper is organized as follows. In Section 2 basic notions related to information systems are presented. Section 3 provides definitions and properties of an indiscernibility relation, a similarity relation and set approximations. Basic notions related to decision tables and decision rules are presented in Section 4. Section 5 describes generation of certain decision rules from complete decision tables by means of Boolean reasoning. The notion of a certain rule in an incomplete system is defined and investigated in Section 6. As a result of the obtained properties, a new method of generating all optimal certain rules from an incomplete system is proposed. In Sections 7–9, the capabilities of the selected rough set methods [3,5] of certain rules generation are investigated.

2. Information systems

Information system (*IS*) is a triplet $\mathcal{S} = (\mathcal{O}, AT, f)$, where $\mathcal{O}$ is a non-empty finite set of *objects* and AT is a non-empty finite set of *attributes*, such that

$f_a: \mathcal{O} \to V_a$ for any $a \in AT$, where V_a is called *domain* of an attribute a. $Inf(x) = \{(a, f_a(x)) \mid a \in AT\}$ is called an *information vector* of x. Any attribute domain V_a may contain special symbol "*" to indicate that the value of an attribute is *unknown*. Here, we assume that an object $x \in \mathcal{O}$ possesses only one value for an attribute a, $a \in AT$, in reality. Thus, if the value of an attribute a is missing then the real value must be one from the set $V_a \setminus \{*\}$. Any domain value different from "*" will be called *regular*. A system in which values of all attributes for all objects from $\mathcal{O}$ are regular (known) is called *complete*, otherwise it is called *incomplete*.

Let $\mathcal{S} = (\mathcal{O}, AT, f)$. $\mathcal{S}' = (\mathcal{O}', AT', f')$ is called an *extension* of $\mathcal{S}$ iff $\mathcal{O}' = \mathcal{O}$, $AT' = AT$ and $f_a(x) \neq *$ implies $f'_a(x) = f_a(x)$ for any $a \in AT$ and $x \in \mathcal{O}$. We say that $\mathcal{S}''$ is a completion of $\mathcal{S}$ iff $\mathcal{S}''$ is a complete information system which is an extension of $\mathcal{S}$. The set of all extensions of the system $\mathcal{S}$ will be denoted by $EXTN(\mathcal{S})$, whereas the set of all completions of $\mathcal{S}$ will be denoted by $COMP(\mathcal{S})$. We will indicate that a notion is considered in some extension of the system $\mathcal{S}$ by adding the respective upper index denoting that extension. In the paper, we will also refer to extensions of $\mathcal{S}$ of a particular kind. $\mathcal{S}'$ is called *x-extension* of $\mathcal{S}$ iff $\mathcal{S}' \in EXTN(\mathcal{S})$ and for any $y \in \mathcal{O} \setminus \{x\}$: $Inf(y) = Inf'(y)$ and for any $a \in AT$: $f'_a(x)$ is regular. The set of all x-extensions of $\mathcal{S}$ will be denoted by $CIEXTN(\mathcal{S}, x)$.

Example 2.1. Table 1 illustrates an exemplary incomplete information system $\mathcal{S}$. Fig. 1 presents extensions $\mathcal{S}'$ and $\mathcal{S}''$ of the system $\mathcal{S}$. $\mathcal{S}''$ is a 2-extension of $\mathcal{S}$, so it differs from $\mathcal{S}$ only for object 2 and all attribute values of object 2 are regular in $\mathcal{S}''$. All completions of $\mathcal{S}$ are presented in Fig. 2.

In the sequel, any attribute-value pair (a,v), $a \in AT$, $v \in V_a$ will be called an *atomic* property. Any *atomic property* or its conjunction will be called *descriptor*. Conjunction of atomic properties (a,v), where $a \in A \subseteq AT$, will be called *A-descriptor*. Descriptor that does not possess null values will be called *complete*. The set of objects having the atomic property (a,v) will be denoted by

Table 1
Exemplary incomplete information system $\mathcal{S}$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	*	1	1
3	2	1	1	1
4	1	2	*	1
5	1	*	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
4	1	2	*	1
5	1	1	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}'$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
4	1	2	*	1
5	1	*	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}''$

Fig. 1. $\mathcal{S}' \in EXTN(\mathcal{S})$, $\mathcal{S}'' \in CIEXTN(\mathcal{S}, 2)$, where $\mathcal{S}$ is the system from Table 1.

$||(a,v)||$ (i.e. $||(a,v)|| = \{x \in \mathcal{O} \mid f_a(x) = v\}$). Let us note that $||(a,*)|| \cap ||(a,v)|| = \emptyset$, if $v \neq *$. The set of objects satisfying any descriptor t will be denoted by $||t||$ and will be computed in the usual way, e.g. $||t \wedge s|| = ||t|| \cap ||s||$.

3. Indiscernibility of objects and set approximations

3.1. Indiscernibility of objects

Let $\mathcal{S} = (\mathcal{O}, AT, f)$. Each subset of attributes $A \subseteq AT$ determines a binary *indiscernibility relation IND(A)*:

$$IND(A) = \{(x,y) \in \mathcal{O} \times \mathcal{O} \mid \forall a \in A, f_a(x) = f_a(y)\}.$$

The relation $IND(A)$, $A \subseteq AT$, is an equivalence relation and constitutes a partition of $\mathcal{O}$.

Let $I_A(x)$ denote the set of objects $\{y \in \mathcal{O} \mid (x,y) \in IND(A)\}$. Objects from $I_A(x)$ are indiscernible with regard to their description in the system, but they may have different properties in reality, unless the system is complete. In a complete system, objects perceived as indiscernible in the system are indiscernible also in reality.

A *similarity relation SIM(A)*:

$$SIM(A) = \{(x,y) \in \mathcal{O} \times \mathcal{O} \mid \forall a \in A, f_a(x) = f_a(y) \text{ or } f_a(x) = * \text{ or } f_a(y) = *\}$$

treats two objects as similar if they may have the same properties in reality. Similarity relation is reflexive and symmetric, but may not be transitive, so it is a tolerance relation.

By $S_A(x)$ we will denote the set of possibly indiscernible objects: $\{y \in \mathcal{O} \mid (x,y) \in SIM(A)\}$. Of course, $SIM(A)$ and $IND(A)$, $A \subseteq AT$, are equivalent relations in a complete system.

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
4	1	2	1	1
5	1	1	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^1$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
4	1	2	1	1
5	1	2	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^2$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
4	1	2	2	1
5	1	1	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^3$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
4	1	2	2	1
5	1	2	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^4$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	2	1	1
3	2	1	1	1
4	1	2	1	1
5	1	1	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^5$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	2	1	1
3	2	1	1	1
4	1	2	1	1
5	1	2	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^6$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	2	1	1
3	2	1	1	1
4	1	2	2	1
5	1	1	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^7$

$\mathcal{O}$	a	b	c	d
1	1	1	1	1
2	1	2	1	1
3	2	1	1	1
4	1	2	2	1
5	1	2	1	2
6	2	2	2	2
7	1	1	1	2

$\mathcal{S}^8$

Fig. 2. Completions $COMP(\mathcal{S})$ of the system $\mathcal{S}$ from Table 1.

Property 3.1. *Let $B \subseteq A \subseteq AT$. The following properties hold for any information system $\mathcal{S} = (O, AT, f)$ and its extension $\mathcal{S}'$:*

$I_A(x) \subseteq I_B(x)$;

$S_A(x) \subseteq S_B(x)$;

$I_A(x) \subseteq S_A(x)$;

$S'_A(x) \subseteq S_A(x)$.

Property 3.2. *Let $\mathcal{S} = (O, AT, f)$ be a complete information system and $A \subseteq AT$. Then*

$$I_A(x) = S_A(x).$$

Property 3.3. *Let $\mathscr{S} = (\mathbb{O}, AT, f)$ be an information system and $A \subseteq AT$. Then*

$$S_A(x) = \{y \in \mathbb{O} \mid \exists \mathscr{S}' \in COMP(\mathscr{S}), y \in I'_A(x)\}.$$

Proof. $y \in I'_A(x)$ for some $\mathscr{S}' \in COMP(S)$ iff for any $a \in AT: f'_a(y) = f'_a(x)$ for some $\mathscr{S}' \in COMP(\mathscr{S})$ iff for any $a \in AT: f_a(y) = f_a(x)$ or $f_a(y) = *$ or $f_a(x) = *$ iff $y \in S_A(x)$. □

Property 3.4. *Let $\mathscr{S} = (\mathbb{O}, AT, f)$ be an information system and $A \subseteq AT$. Then*

$$\bigcup_{\mathscr{S}' \in COMP(\mathscr{S})} I'_A(x) = S_A(x).$$

Proof. It is stated in Property 3.3 that $S_A(x)$ is the set of all objects each of which is indiscernible with x by attributes A in some completion of $\mathscr{S}$. Hence,

$$S_A(x) = \bigcup_{\mathscr{S}' \in COMP(\mathscr{S})} I'_A(x). \quad \square$$

Example 3.1. Let us consider the properties of the system presented in Table 1 and the properties of its completions presented in Fig. 2. Let $A = \{a,b,c\}$.

Let us consider object 2: $S_A(2) = \{1,2,4,5,7\}, I_A(2) = \{2,5\}$, so $I_A(2) \subseteq S_A(2)$ (see also Property 3.1).

$S^1_A(2) = I^1_A(2) = \{1,2,5,7\}$ (see also Property 3.2).
$S^2_A(2) = I^2_A(2) = \{1,2,7\}$ (see also Property 3.2).
$S^3_A(2) = I^3_A(2) = \{1,2,5,7\}$ (see also Property 3.2).
$S^4_A(2) = I^4_A(2) = \{1,2,7\}$ (see also Property 3.2).
$S^5_A(2) = I^5_A(2) = \{2,4\}$ (see also Property 3.2).
$S^6_A(2) = I^6_A(2) = \{2,4,5\}$ (see also Property 3.2).
$S^7_A(2) = I^7_A(2) = \{2\}$ (see also Property 3.2).
$S^8_A(2) = I^8_A(2) = \{2,5\}$ (see also Property 3.2).

$\bigcup_{\mathscr{S}' \in COMP(\mathscr{S})} I'_A(2) = \{1,2,4,5,7\} = S_A(2)$ (see also Property 3.4). □

3.2. Set approximations

Let $X \subseteq \mathbb{O}$ and $A \subseteq AT$. First we recall classical rough set definitions of *lower approximation* $\underline{A}_{IND}X$ and *upper approximation* $\overline{A}_{IND}X$ of X in a complete *IS*:

$$\underline{A}_{IND}X = \{x \in \mathbb{O} \mid I_A(x) \subseteq X\}.$$

$$\overline{A}_{IND}X = \{x \in \mathcal{C} \mid I_A(x) \cap X \neq \emptyset\}.$$

$\underline{A}_{IND}X$ is the set of objects that belong to X with certainty, while $\overline{A}_{IND}X$ is the set of objects that possibly belong to X.

We generalize these notions for the case of incomplete *IS*:

$\underline{A}_{SIM}X$ is *lower approximation* of X, iff $\underline{A}_{SIM}X = \{x \in \mathcal{C} \mid S_A(x) \subseteq X\}$.

$\overline{A}_{SIM}X$ is *upper approximation* of X, iff $\overline{A}_{SIM}X = \{x \in \mathcal{C} \mid S_A(x) \cap X \neq \emptyset\}$.

$\underline{A}_{SIM}X$ is the set of objects that belong to X with certainty in each completion of $\mathcal{S}$. $\overline{A}_{SIM}X$ is the set of objects that possibly belong to X in completions of $\mathcal{S}$. Obviously, $\underline{A}_{SIM}X = \underline{A}_{IND}X$ and $\overline{A}_{SIM}X = \overline{A}_{IND}X$ in a complete *IS*.

4. Decision tables and decision rules

Decision table (*DT*) is an information system $\mathcal{S} = (\mathcal{C}, AT \cup \{d\}, f)$, where d such that $d \notin AT$ and $* \notin V_d$ is a distinguished attribute called the *decision*, and the elements of AT are called *conditions*. If *DT* is a complete *IS* then it is called a *complete decision table*, otherwise it is called an *incomplete decision table*.

Let us define the function $\partial_A : \mathcal{C} \to \mathcal{P}(V_d)$, $A \subseteq AT$, as follows:

$$\partial_A(x) = \{f_d(y) \mid y \in S_A(x)\}.$$

∂_A will be called the *generalized decision* in *DT*. $\partial_{AT}(x)$, $x \in \mathcal{C}$, determines to which decision classes the object x may be classified to based on the available information on x. If $card(\partial_{AT}(x)) = 1$ then x can be classified without ambiguity.

Property 4.1. *Let* $x \in \mathcal{C}$, $A \subseteq AT$.

$$card(\partial_A(x)) = 1 \quad iff \quad S_A(x) \subseteq I_{\{d\}}(x).$$

Example 4.1. Table 2 describes an incomplete decision table containing information about cars. *Price*, *Mileage*, *Size* and *Max-Speed* are the conditional attributes of the system, whereas d is the decision attribute. (In the sequel, *P*, *M*, *S*, *X* will stand for *Price*, *Mileage*, *Size* and *MaX-Speed*, respectively.) The attribute domains are as follows: $V_{Price} = \{high, low\}$, $V_{Mileage} = \{high, low\}$, $V_{Size} = \{full, compact\}$, $V_{Max\text{-}Speed} = \{high, low\}$, $V_d = \{poor, good, excellent\}$. Additionally, the system in Table 2 is extended by the column containing the values of the generalized decision ∂_{AT} for all objects in the system.

Table 3 presents the complete decision table, which is a completion of the system from Table 2.

Table 2
Incomplete car table

Car	*Price*	*Mileage*	*Size*	*MaX-Speed*	*d*	∂_{AT}
1	high	low	full	low	good	{good}
2	low	*	full	low	good	{good}
3	*	*	compact	low	poor	{poor}
4	high	*	full	high	good	{good, excellent}
5	*	*	full	high	excellent	{good, excellent}
6	low	high	full	*	good	{good, excellent}

Table 3
Complete car table

Car	*Price*	*Mileage*	*Size*	*MaX-Speed*	*d*	∂_{AT}
1	high	low	full	low	good	{good}
2	low	low	full	low	good	{good}
3	high	high	compact	low	poor	{poor}
4	high	low	full	high	good	{good, excellent}
5	high	low	full	high	excellent	{good, excellent}
6	low	high	full	high	good	{good}

It is trivial to observe that the value of the generalized decision ∂_{AT} for an object in an incomplete decision table $\mathscr{S}$ is a superset of its generalized decision's value in the completion of $\mathscr{S}$ (see $\partial_{AT}(6)$ in Tables 2 and 3).

The knowledge hidden in decision tables data may be discovered and expressed in the form of *decision rules*: $t \rightarrow s$, where $t = \bigwedge(c, v)$, $c \in A \subseteq AT$, $v \in V_c \setminus \{*\}$, and $s = \bigvee(d, w)$, $w \in V_d$. In the sequel, we will call t and s *condition* and *decision part* of a rule, respectively. A rule with a single decision value in the decision part will be called *definite*, otherwise it will be called *non-definite*. We will say that an object x, $x \in U$, *supports* a rule $t \rightarrow s$ in $\mathscr{S}$ iff x has both property t and s in $\mathscr{S}$.

Example 4.2. The following rules may be induced from rows in Table 2:

$(P, high) \wedge (M, low) \wedge (S, full) \wedge (X, low) \rightarrow (d, good)$ // object 1
$(P, low) \wedge (S, full) \wedge (X, low) \rightarrow (d, good)$ // object 2
$(S, compact) \wedge (X, low) \rightarrow (d, poor)$ // object 3
$(P, high) \wedge (S, full) \wedge (X, high) \rightarrow (d, good) \vee (d, excellent)$ // object 4
$(S, full) \wedge (X, high) \rightarrow (d, good) \vee (d, excellent)$ // object 5
$(P, low) \wedge (M, high) \wedge (S, full) \rightarrow (d, good) \vee (d, excellent)$ // object 6

The first three rules are definite, while the last three rules are non-definite.

We can easily observe that the second rule in Example 4.2 is more general than the first rule and is supported both by object 2 and 1. Hence, rule 1, which is supported only by object 1, may be removed. We may also notice that the third rule may be replaced by the shorter rule: $(S, compact) \rightarrow (d, poor)$.

5. Complete decision tables

An information system $\mathscr{S} = (\mathcal{O}, AT \cup \{d\}, f)$ considered in Section 5 is assumed to be a complete decision table.

5.1. Certain decision rules

Any decision rule $t \rightarrow s$ is called *certain* in $\mathscr{S}$ iff $t \rightarrow s$ is definite and $||t|| \subseteq ||s||$ in $\mathscr{S}$. Any decision rule $t \rightarrow s$ is *optimal certain* in $\mathscr{S}$ iff it is certain in $\mathscr{S}$ and no other rule constructed from a proper subset of atomic properties occurring in t is certain in $\mathscr{S}$.

Property 5.1. *Let $x \in \mathcal{O}$. x supports a certain rule in $\mathscr{S}$ iff $I_{AT}(x) \subseteq I_{\{d\}}(x)$ in $\mathscr{S}$.*

Proof. ($\Rightarrow$) Let $t \rightarrow s$ be a certain rule supported by x in $\mathscr{S}$. Let T be the set of attributes occurring in t. Since x supports $t \rightarrow s$ in the complete information system $\mathscr{S}$ then $I_T(x) = ||t||$ and $I_{\{d\}}(x) = ||s||$. Additionally, since $t \rightarrow s$ is certain then $||t|| \subseteq ||s||$. By means of Property 3.1 we also have: $I_{AT}(x) \subseteq I_T(x)$. Thus, $I_{AT}(x) \subseteq I_T(x) = ||t|| \subseteq ||s|| = I_{\{d\}}(x)$ and finally $I_{AT}(x) \subseteq I_{\{d\}}(x)$.

($\Leftarrow$) It is necessary to prove that: if $I_{AT}(x) \subseteq I_{\{d\}}(x)$ then there is some certain rule $t \rightarrow s$ supported by x in $\mathscr{S}$. Let $t = \bigwedge(a, f_a(x))$, $a \in AT$, and $s = (d, f_d(x))$. Then $||t|| = I_{AT}(x)$ and $||s|| = I_{\{d\}}(x)$. Hence, $||t|| \subseteq ||s||$ and $t \rightarrow s$ is definite. Thus, $t \rightarrow s$ is a certain rule supported by x. □

Property 5.2. *Let $x \in \mathcal{O}$ and $A \subseteq AT$.*

$$card(\partial_A(x)) = 1 \quad iff \quad I_A(x) \subseteq I_{\{d\}}(x).$$

Proof. Follows immediately from Properties 4.1 and Property 3.2. □

5.2. Computation of optimal certain rules

Here we will present a method of computing all optimal certain rules supported by an arbitrary object $x \in \mathcal{O}$. Any certain rule supported by x will have a conjunction of some atomic properties of x in the condition part and the

property $(d, f_d(x))$ in the decision part. So, x may be used as a generator of rules. Let us also note that certain rules are supported only by objects x such that $I_{AT}(x) \subseteq I_{\{d\}}(x)$ (i.e. $card(\partial_{AT}(x)) = 1$). In order to determine the condition part of an optimal decision rules supported by an object x we may employ the notion of an x-reduct:

Let $A \subseteq AT$ and $x \in \mathscr{C}$ and $I_{AT}(x) \subseteq I_{\{d\}}(x)$ (i.e. $card(\partial_{AT}(x)) = 1$). The set A is a *certain x-reduct* in $\mathscr{S}$ iff A is a minimal set such that:

$$I_A(x) \subseteq I_{\{d\}}(x).$$

If A is an x-reduct then the condition part of an optimal certain rule will be a conjunction of atomic properties $(a, f_A(x))$, where $a \in A$.

In order to compute reducts of DT we will exploit the idea of *discernibility functions* [6]. Their main properties are that they are monotonic Boolean functions and their prime implicants determine reducts uniquely. Constructing suitable discernibility functions for generation of decision rules from complete systems has been subject of many papers e.g. [7–13].

Let $\alpha(x, y) = \{a \in AT \mid (x, y) \notin SIM(\{a\})\}$. Let $\sum \alpha(x, y)$ be a Boolean expression which is equal to 1, if $\alpha(x, y) = \emptyset$. Otherwise, let $\sum \alpha(x, y)$ be a disjunction of variables corresponding to attributes contained in $\alpha(x, y)$.

Let $x \in \mathscr{C}$ and $I_{AT}(x) \subseteq I_{\{d\}}(x)$ (i.e. $card(\partial_{AT}(x)) = 1$). $\Delta_c(x)$ is a *certain x-discernibility function* iff

$$\Delta_c(x) = \prod_{y \in Y_c} \sum \alpha(x, y), \quad \text{where } Y_c = \mathscr{C} \setminus I_{\{d\}}(x).$$

Example 5.1. We will illustrate the method of optimal certain rules generation for the case of the complete system from Table 3. Certain rules may be supported only by objects with single value generalized decisions. Hence, we will use objects 1–3 and 6 as rules' generators.

First, let us consider object 1: $I_{\{d\}}(1) = ||(d, good)|| = \{1, 2, 4, 6\}$. Hence, $Y_c = \mathscr{C} \setminus I_{\{d\}}(1) = \{3, 5\}$. $\Delta_c(1) = \sum \alpha(1, 3) \wedge \sum \alpha(1, 5) = (M \vee S) \wedge (X) = MX \vee SX$.

Thus, there are two certain 1-reducts $\{M, X\}$ and $\{S, X\}$. Therefore object 1 supports two optimal certain rules:

$$(M, low) \wedge (X, low) \rightarrow (d, good),$$
$$(S, full) \wedge (X, low) \rightarrow (d, good).$$

Similarly, one can find optimal certain rules supported by objects 2–3 and 6: $I_{\{d\}}(2) = ||(d, good)|| = \{1, 2, 4, 6\}$. Hence, $Y_c = \mathscr{C} \setminus I_{\{d\}}(2) = \{3, 5\}$. $\Delta_c(2) = \sum \alpha(2, 3) \wedge \sum \alpha(2, 5) = (P \vee M \vee S) \wedge (P \vee X) = P \vee MX \vee SX$. Hence, there are three optimal certain rules supported by object 2:

$(P, low) \rightarrow (d, good)$.

$(M, low) \wedge (X, low) \rightarrow (d, good)$.

$(S, full) \wedge (X, low) \rightarrow (d, good)$.

$I_{\{d\}}(3) = ||(d, poor)|| = \{3\}$. Hence, $Y_c = \mathcal{C} \setminus I_{\{d\}}(3) = \{1,2,4,5,6\}$. $\Delta_c(x) = \sum \alpha(3,1) \wedge \sum \alpha(3,2) \wedge \sum \alpha(3,4) \wedge \sum \alpha(3,5) \wedge \sum \alpha(3,6) = (M \vee S) \wedge (P \vee M \vee S) \wedge (M \vee S \vee X) \wedge (M \vee S \vee X) \wedge (P \vee S \vee X) = (M \vee S) \wedge (P \vee S \vee X) = S \vee MP \vee MX$.

Hence, there are three optimal certain rules supported by object 3:
$(S, compact) \rightarrow (d, poor)$.
$(P, high) \wedge (M, high) \rightarrow (d, poor)$.
$(M, high) \wedge (X, low) \rightarrow (d, poor)$.

$I_{\{d\}}(6) = ||(d, good)|| = \{1,2,4,6\}$. Hence, $Y_c = \mathcal{C} \setminus I_{\{d\}}(6) = \{3,5\}$. $\Delta_c(6) = \sum \alpha(6,3) \wedge \sum \alpha(6,5) = (P \vee S \vee X) \wedge (P \vee M) = P \vee MS \vee MX$. Hence, there are three optimal certain rules supported by object 6:

$(P, low) \rightarrow (d, good)$.

$(M, high) \wedge (S, full) \rightarrow (d, good)$.

$(M, high) \wedge (X, high) \rightarrow (d, good)$.

6. Incomplete decision tables

6.1. Certain decision rules

Following the approach to incomplete information systems presented by Lipski in [14], we propose the following definition of a certain rule: Any decision rule $t \rightarrow s$ is called *certain* in $\mathcal{S}$ iff it is certain in every completion of $\mathcal{S}$. Any decision rule $t \rightarrow s$ is *optimal certain* in $\mathcal{S}$ iff it is certain in $\mathcal{S}$ and no other rule constructed from a proper subset of atomic properties occurring in t is certain in $\mathcal{S}$.

Proposition 6.1. *$t \rightarrow s$ is certain in $\mathcal{S}$ if it is certain in every completion of $\mathcal{S}$ in which $||t|| \neq \emptyset$.*

Proof. By definition of a certain rule in a complete IS, a rule $t \rightarrow s$ is certain in any completion of $\mathcal{S}$ in which $||t|| \subseteq ||s||$. Hence, it is certain in every completion of $\mathcal{S}$ in which $||t|| = \emptyset$. Proposition is an immediate consequence of this fact and the definition of a certain rule. □

Proposition 6.2. *Let $Desc_T(Y)$ denote the set of T-descriptors of all objects from Y.* Let $t \rightarrow s$ be a definite decision rule supported in some completion of $\mathscr{S}$ and T be the set of all attributes occurring in t. $t \rightarrow s$ *is certain in* $\mathscr{S}$ *iff* $t \notin \bigcup_{\mathscr{S}' \in COMP(\mathscr{S})} Desc'_T(\mathscr{C} \setminus ||s||)$.

Proof. In order to prove the proposition we will prove the following equivalent statement: $t \rightarrow (d,v)$ *is not certain in* $\mathscr{S}$ *iff* $t \in \bigcup_{\mathscr{S}' \in COMP(\mathscr{S})} Desc'_T(\mathscr{C} \setminus ||(d,v)||)$. The rule $t \rightarrow (d,v)$ is not certain in $\mathscr{S}$ iff there is some completion $\mathscr{S}'$ of $\mathscr{S}$ in which $||t||' \not\subseteq ||(d,v)||'$ iff there is a completion $\mathscr{S}'$of $\mathscr{S}$ in which some object possesses the properties t and (d,w), $w \neq v$, iff $t \in \bigcup_{\mathscr{S}' \in COMP(\mathscr{S})} Desc'_T(\mathscr{C} \setminus ||(d,v)||)$. □

Proposition 6.2 justifies the correctness of a rules generation approach, in which an initial incomplete system $\mathscr{S}$ is transformed into a complete $\mathscr{S}'$ which contains all possible descriptors of the objects incompletely described in $\mathscr{S}$ (see [3]). Then rules are generated from the complete system $\mathscr{S}'$. Next propositions will justify another approach to rules generation, which allows to compute rules directly from x-extensions of the initial incomplete system.

Proposition 6.3. *Let $x \in \mathscr{C}$, $\mathscr{S}^e$ be an x-extension of $\mathscr{S}$ and $x \in ||t||^e$. If $t \rightarrow s$ is certain in $\mathscr{S}^e$ then $t \rightarrow s$ is certain in $\mathscr{S}$.*

Proof. It is necessary to prove that if $t \rightarrow s$ is certain in an x-extension $\mathscr{S}^e$ of $\mathscr{S}$ where $x \in ||t||^e$ then $t \rightarrow s$ is certain in $\mathscr{S}$. Let us consider an arbitrary $\mathscr{S}'$ from $COMP(\mathscr{S})$. From the definition there exists a completion $\mathscr{S}''$ of $\mathscr{S}^e$ which can differ from $\mathscr{S}'$ at most on x. So it is enough to prove that for any $y \in \mathscr{C}$: *if* $y \in ||t||'$ *then* $y \in ||s||'$ follows from: *if* $y \in ||t||''$ *then* $y \in ||s||''$. Let us assume that $y \in ||t||'$. Suppose $y \notin ||s||'$. It means that for some $y \neq x$ such that $y \in ||t||'$ we have $d(x) \neq d(y)$. However, $Inf''(y)$ is also an information vector in $\mathscr{S}''$, which contradicts $||t||'' \subseteq ||s||''$. □

Proposition 6.4. *Let $x \in \mathscr{C}$ and $\mathscr{S}^e$ be an x-extension of $\mathscr{S}$. An optimal certain rule supported by x in $\mathscr{S}^e$ is optimal certain in $\mathscr{S}$.*

Proof. Let $t \rightarrow s$ be an optimal certain rule supported by x in $\mathscr{S}^e$. Immediate conclusion from Proposition 6.3 is that $t \rightarrow s$ is certain in $\mathscr{S}$. Additionally, since $t \rightarrow s$ is optimal certain in $\mathscr{S}^e$ then any rule $r \rightarrow s$ constructed from a proper subset of atomic properties occurring in t is not certain in any completion of $\mathscr{S}^e$ and hence $r \rightarrow s$ is not certain in $\mathscr{S}$. The above observations allow us to conclude that $t \rightarrow s$ is an optimal certain rule in $\mathscr{S}$. □

Property 6.1. *Let $x \in \mathscr{C}$ and $\mathscr{S}^e$ be an x-extension of $\mathscr{S}$. x supports a certain rule in $\mathscr{S}^e$ iff $S^e_{AT}(x) \subseteq I_{\{d\}}(x)$ in $\mathscr{S}^e$.*

Proof. Let $t \rightarrow s$ be a certain rule supported by x in $\mathscr{S}^e$. The object x supports $t \rightarrow s$ in $\mathscr{S}^e$ iff x supports $t \rightarrow s$ in every completion $\mathscr{S}'$ of $\mathscr{S}^e$ iff $I'_{AT}(x) \subseteq I_{\{d\}}(x)$ (by Property 5.1) in each completion $\mathscr{S}'$ of $\mathscr{S}^e$ iff $\bigcup_{\mathscr{S}' \in COMP(\mathscr{S}^e)} I'_{AT}(x) \subseteq I_{\{d\}}(x)$ *iff* $S^e_{AT}(x) \subseteq I_{\{d\}}(x)$ (by Property 3.4). □

6.2. Computation of optimal certain rules

It is stated in Proposition 6.4 that optimal certain rules supported by an object x in any x-extension $\mathscr{S}^e$ of $\mathscr{S}$ are optimal certain in $\mathscr{S}$. Additionally, it follows from Property 6.1 that the condition $S^e_{AT}(x) \subseteq I_{\{d\}}(x)$ in $\mathscr{S}^e$ (i.e. $card(\partial^e_{AT}(x)) = 1$) is necessary for an object x to support a certain rule. Clearly, the set of all optimal certain rules each of which is supported by x in some completion of $\mathscr{S}$ is equal to the set of all optimal certain rules each of which is supported by x in some x-extension of $\mathscr{S}$. In this subsection we show how to compute optimal certain rules (certain x-reducts) in an x-extension $\mathscr{S}^e$. First we show how to obtain this goal by examining all completions of $\mathscr{S}^e$. Next we prove that the rules may be computed directly from $\mathscr{S}^e$.

Let $A \subseteq AT$, $x \in U$, $\mathscr{S}^e$ be an x-extension of $\mathscr{S}$ and $S^e_{AT}(x) \subseteq I_{\{d\}}(x)$ (i.e. $card(\partial^e_{AT}(x)) = 1$). The set A is a *certain x-reduct* in $\mathscr{S}^e$ iff A is a minimal set such that:

$$I'_A(x) \subseteq I_{\{d\}}(x)$$

in each completion $\mathscr{S}'$ of $\mathscr{S}^e$.

Let $x \in U$, $\mathscr{S}^e$ be an x-extension of $\mathscr{S}$ and $S^e_{AT}(x) \subseteq I_{\{d\}}(x)$ (i.e. $card(\partial^e_{AT}(x)) = 1$). $\Delta^e_x(x)$ is a *certain x-discernibility function* iff

$$\Delta^e_x(x) = \prod_{\mathscr{S}' \in COMP(\mathscr{S}^e)} \prod_{y \in Y_x} \sum \alpha'(x,y), \quad \text{where } Y_x = U \setminus I_{\{d\}}(x).$$

The prime implicants of $\Delta^e_x(x)$ determine x-reducts in $\mathscr{S}^e$ uniquely.

Proposition 6.5. *Let $x \in U$, $\mathscr{S}^e$ be an x-extension of $\mathscr{S}$ and $S^e_{AT}(x) \subseteq I_{\{d\}}(x)$ (i.e. $card(\partial^e_{AT}(x)) = 1$). Then*

$$\Delta^e_x(x) = \prod_{y \in Y_x} \sum \alpha^e(x,y), \quad \text{where } Y_x = U \setminus I_{\{d\}}(x).$$

Proof. Let $\mathscr{S}'$ be a completion of $\mathscr{S}^e$ such that for any $y \in U \setminus \{x\}$: $f^e_a(y) = *$ implies $f'_a(y) = f^e_a(x)$. Let $\mathscr{S}'' \neq \mathscr{S}'$ be an arbitrary completion of $\mathscr{S}^e$. One may easily notice that for each $y \in Y_x$, $\sum \alpha'(x,y) \wedge \sum \alpha''(x,y) = \sum \alpha'(x,y) = \sum \alpha^e(x,y)$. Therefore, $\Delta^e_x(x) = \prod_{\mathscr{S}' \in COMP(\mathscr{S}^e)} \prod_{y \in Y_x} \sum \alpha''(x,y) = \prod_{y \in Y_x} \sum \alpha'(x,y) = \prod_{y \in Y_x} \sum \alpha^e(x,y)$. □

Example 6.1. We will illustrate the method of optimal certain rules generation from an incomplete system by means of Proposition 6.5. Table 2 presents the system $\mathscr{S}$ to be considered. Object 5 will be used as a rule generator. There are four possible complete AT-descriptors of this object:

1. $(P, low) \wedge (M, low) \wedge (S, full) \wedge (X, high)$.
2. $(P, low) \wedge (M, high) \wedge (S, full) \wedge (X, high)$.
3. $(P, high) \wedge (M, low) \wedge (S, full) \wedge (X, high)$.
4. $(P, high) \wedge (M, high) \wedge (S, full) \wedge (X, high)$.

where P, M, S, X stand for *Price*, *Mileage*, *Size* and *Max-Speed*, respectively.

(a) Let $\mathscr{S}^c \in C1EXTN(S, 5)$ and object 5 has the descriptor in $\mathscr{S}^c$ as in the case 1. Table 4 illustrates the extension $\mathscr{S}^c$.

$S^c_{AT}(5) = \{5\}$ and $I_{\{d\}}(5) = ||(d, excellent)|| = \{5\}$, so $S^c_{AT}(5) \subseteq I_{\{d\}}(5)$ (i.e. $card(\partial^c_{AT}(5)) = 1$) and according to Property 6.1 object 5 supports some certain rule in $\mathscr{S}^c$. $Y_c = U \setminus I_{\{d\}}(5) = \{1, 2, 3, 4, 6\}$. Hence, $\Delta^c_5(5) = \sum x^c(5, 1) \wedge \sum x^c(5, 2) \wedge \sum x^c(5, 3) \wedge \sum x^c(5, 4) \wedge \sum x^c(5, 6) = (P \vee X) \wedge (X) \wedge (S \vee X) \wedge (P) \wedge (M) = PMX$.

Thus, there is only one certain 5-reduct $\{P, M, X\}$ in $\mathscr{S}^c$, which means that only one optimal certain rule is supported by object 5 in $\mathscr{S}^c$, namely:

$$(P, low) \wedge (M, low) \wedge (X, high) \rightarrow (d, excellent).$$

(b) Let $\mathscr{S}^c \in C1EXTN(S, 5)$ and object 5 has the descriptor in $\mathscr{S}^c$ as in the case 2. Table 5 illustrates the extension $\mathscr{S}^c$.

$S^c_{AT}(5) = \{5, 6\}$ and $I_{\{d\}}(5) = ||(d, excellent)|| = \{5\}$, so $S^c_{AT}(5) \not\subseteq I_{\{d\}}(5)$ (i.e. $card(\partial^c_{AT}(5)) \neq 1$). Therefore, by Property 6.1, object 5 does not support any certain rule in $\mathscr{S}^c$.

Similarly, there are no optimal certain rules supported by object 5 in the 5-extensions of Table 2 in which object 5 has the descriptor as in the cases 3 and 4, respectively. In these 5-extensions $card(\partial^c_{AT}(5)) \neq 1$.

Performing analogous computations for each object we would receive the following set of all optimal certain rules in Table 2:

Table 4
A 5-extension of Table 2

Car	*Price*	*Mileage*	*Size*	*MaX-Speed*	d	∂_{AT}
1	high	low	full	low	good	{good}
2	low	*	full	low	good	{good}
3	*	*	comp.	low	poor	{poor}
4	high	*	full	high	good	{good}
5	low	low	full	high	Excel.	{Excel.}
6	low	high	full	*	good	{good}

Table 5
Another 5-extension of Table 2

Car	*Price*	*Mileage*	*Size*	*MaX-Speed*	*d*	∂_{IT}
1	high	low	full	low	good	{good}
2	low	*	full	low	good	{good}
3	*	*	comp.	low	poor	{poor}
4	high	*	full	high	good	{good}
5	low	high	full	high	excel.	{good, excel.}
6	low	high	full	*	good	{good, excel.}

$(S, full) \wedge (X, low) \rightarrow (d, good)$. // found in some x-extensions, where $x = 1,2,6$

$(S, compact) \rightarrow (d, poor)$. // found in all 3-extensions

$(P, high) \wedge (M, high) \wedge (X, low) \rightarrow (d, poor)$. // found in some 3-extension

$(P, low) \wedge (M, low) \wedge (X, high) \rightarrow (d, excellent)$. // found in some 5-extensions

7. Computing certain rules from original incomplete decision table

In this section we will show that an important class of optimal certain rules which are supported in all completions of $\mathcal{S}$ can be computed directly from the original incomplete system $\mathcal{S}$. We will prove that definite generalized rules, which are a special case of generalized rules presented in [5], constitute such a class of rules.

7.1. Generalized rules

In this subsection we will provide the formal definition of generalized rules (after [5]) and present a method of computing them. Next we will examine specific properties of definite generalized rules.

Any decision rule $t \rightarrow s$ is called *generalized* in $\mathcal{S}$ iff $\overline{T}_{SM}(\|t\|) \subseteq \|s\|$, where T is the set of all attributes occurring in t. Any decision rule $t \rightarrow s$ is *optimal generalized* in $\mathcal{S}$ iff it is generalized in $\mathcal{S}$ and no other rule constructed from a proper subset of atomic properties occurring in t or s is generalized in $\mathcal{S}$. The decision part of an optimal generalized decision rule generated from the information vector of x, $x \in C$, is equal to $(d, w_1) \vee (d, w_2) \vee \ldots \vee (d, w_n)$, where $\{w_1, w_2, \ldots, w_n\} = \partial_{IT}(x)$. The reduced set of condition attributes can be computed as generalized x-reducts:

Let $A \subseteq AT$ and $x \in \mathcal{C}$. The set A is a *generalized x-reduct* in $\mathcal{S}$ iff A is a minimal set such that:

$$\partial_A(x) = \partial_{AT}(x).$$

Let $x \in \mathcal{C}$. $\Delta_g(x)$ is a *generalized x-discernibility function* in $\mathcal{S}$ iff

$$\Delta_g(x) = \prod_{y \in Y_g} \sum \alpha(x,y), \quad \text{where } Y_g = \mathcal{C} \setminus \{y \in \mathcal{C} \mid d(y) \in \partial_{AT}(x)\}.$$

The prime implicants of $\Delta_g(x)$ determine generalized x-reducts uniquely.

Example 7.1. Let us illustrate the method of generation of optimal generalized rules directly from an incomplete system. Table 2 presents the system $\mathcal{S}$ under consideration. Objects 1 and 5 will be used as exemplary rules generators.

Object 1: $Y_g = \mathcal{C} \setminus \{y \in \mathcal{C} \mid d(y) \in \partial_{AT}(1)\} = \mathcal{C} \setminus \{1,2,4,6\} = \{3,5\}$. Hence, $\Delta_g(1) = \sum \alpha(1,3) \wedge \sum \alpha(1,5) = (S) \wedge (X) = SX$.

Thus, there is only one optimal generalized rule supported by object 1 in $\mathcal{S}$, namely:

$$(S, full) \wedge (X, low) \rightarrow (d, good).$$

Object 5: $Y_g = \mathcal{C} \setminus \{y \in \mathcal{C} \mid d(y) \in \partial_{AT}(5)\} = \mathcal{C} \setminus \{1,2,4,5,6\} = \{3\}$. Hence, $\Delta_g(5) = \sum \alpha(5,3) = (S \vee X) = S \vee X$.

Thus, there are two optimal generalized rules supported by object 5 in $\mathcal{S}$, namely:

$$(S, full) \rightarrow (d, good) \vee (d, excellent),$$

$$(X, high) \rightarrow (d, good) \vee (d, excellent).$$

Altogether, the following set of all optimal generalized decision rules can be induced from Table 2:

$(S, full) \wedge (X, low) \rightarrow (d, good)$.	// generated by objects: 1,2
$(S, compact) \rightarrow (d, poor)$.	// generated by object: 3
$(S, full) \rightarrow (d, good) \vee (d, excellent)$.	// generated by objects: 4,5,6
$(X, high) \rightarrow (d, good) \vee (d, excellent)$.	// generated by objects: 4,5

The first two generalized rules are definite. Let us note that the third rule, which is non-definite, is supported by the objects: 1, 2, 4, 5, 6, but it is generated only by the objects: 4, 5, 6. The objects 1 and 2 generate more specific, but definite rule 1.

The next two properties refer to definite generalized rules.

Property 7.1. *Let $x \in \mathcal{O}$. x supports a definite generalized rule in $\mathcal{S}$ iff $S_{AT}(x) \subseteq I_{\{d\}}(x)$ in $\mathcal{S}$.*

Proof. ($\Rightarrow$) Let $t \rightarrow s$ be a definite generalized rule supported by x in $\mathcal{S}$. Let T be the set of attributes occurring in t. Since x supports $t \rightarrow s$ in $\mathcal{S}$ then $I_T(x) = ||t||$ and $S_T(x) = \overline{T}_{SIM}||t||$ and $I_{\{d\}}(x) = ||s||$. Additionally, since $t \rightarrow s$ is generalized then $\overline{T}_{SIM}||t|| \subseteq ||s||$. By means of Property 3.1 we also have: $S_{AT}(x) \subseteq S_T(x)$. Hence, $S_{AT}(x) \subseteq S_T(x) = \overline{T}_{SIM}||t|| \subseteq ||s|| = I_{\{d\}}(x)$ and finally $S_{AT}(x) \subseteq I_{\{d\}}(x)$.

($\Leftarrow$) It is necessary to prove that: if $S_{AT}(x) \subseteq I_{\{d\}}(x)$ then there is some definite generalized rule $t \rightarrow s$ supported by x in $\mathcal{S}$. Let $t = \bigwedge(a, f_a(x))$, $a \in T$, where $T = \{a \in AT \mid f_a(x) \neq *\}$, and $s = (d, f_d(x))$. Then $S_{AT}(x) = \overline{AT}_{SIM}||t|| = \overline{T}_{SIM}||t||$ and $||s|| = I_{\{d\}}(x)$. Hence, $\overline{T}_{SIM}||t|| \subseteq ||s||$ and $t \rightarrow s$ is definite. Thus, $t \rightarrow s$ is a definite generalized rule supported by x. □

Property 7.2. *Let $A \subseteq AT$ and $x \in \mathcal{O}$ and $S_{AT}(x) \subseteq I_d(x)$. Then*

$$\Delta_g(x) = \prod_{y \in Y_g} \sum \alpha(x, y), \quad \text{where } Y_g = \mathcal{O} \setminus I_{\{d\}}(x).$$

Proof. By Property 4.1, $S_{AT}(x) \subseteq I_d(x)$ implies $card(\partial_{AT}(x)) = 1$. Hence, $\partial_{AT}(x) = \{d(x)\}$. Thus, $Y_g = \mathcal{O} \setminus \{y \in \mathcal{O} \mid d(y) \in \{d(x)\}\} = \mathcal{O} \setminus \{y \in \mathcal{O} \mid d(y) = d(x)\} = \mathcal{O} \setminus I_{\{d\}}(x)$. □

7.2. Certain rules and definite generalized rules

In this subsection, we discuss the relationship between certain and definite generalized rules.

Proposition 7.1. *The set of objects each of which supports some optimal definite generalized rule in $\mathcal{S}$ is a subset of objects each of which supports at least one optimal certain rule in some x-extension of $\mathcal{S}$.*

Proof. Let $\mathcal{S}^c$ be an x-extension of $\mathcal{S}$. By Property 6.1, certain rules in $\mathcal{S}^c$ are supported only by objects x, $x \in \mathcal{O}$, such that $S^c_{AT}(x) \subseteq I_{\{d\}}(x)$. On the other hand, by Property 7.1, definite generalized rules are supported only by objects x such that $S_{AT}(x) \subseteq I_{\{d\}}(x)$. By Property 3.1, $S^c_{AT}(x) \subseteq S_{AT}(x)$. Thus, the set of objects $\{x \in \mathcal{O} \mid S_{AT}(x) \subseteq I_{\{d\}}(x)\}$ supporting optimal definite generalized rules in $\mathcal{S}$ is a subset of objects $\{x \in \mathcal{O} \mid S^c_{AT}(x) \subseteq I_{\{d\}}(x)\}$ supporting optimal certain rules in $\mathcal{S}^c$. □

Proposition 7.2. *The set of all optimal definite generalized rules supported by an object x in $\mathcal{S}$ is a subset of all optimal certain rules supported by x in any x-extension of $\mathcal{S}$.*

Proof. Let $\mathscr{S}^c$ be an x-extension of $\mathscr{S}$. By Property 7.1, an object that supports some optimal definite generalized rule in $\mathscr{S}$ supports also some optimal certain rule in $\mathscr{S}^c$. Thus, it is enough to show that the set of generalized x-reducts in $\mathscr{S}$ is a subset of certain x-reducts in $\mathscr{S}^c$ if $S_{AT}(x) \subseteq I_{\{d\}}(x)$. To this end, we will consider the respective x-discernibility functions.

Let us consider an object $x \in \mathscr{U}$ such that $S_{AT}(x) \subseteq I_d(x)$. Then $S^c_{AT}(x) \subseteq I_d(x)$ since $S^c_{AT}(x) \subseteq S_{AT}(x)$ (by Property 3.1). Let $\mathscr{U} \setminus I_{\{d\}}(x) = \{y_1, \ldots, y_n\}$. The generalized x-discernibility function in $\mathscr{S}$ has the following form:

$$\Delta_g(x) = \prod_{y \in \mathscr{U} \setminus I_{\{d\}}(x)} \sum \alpha(x, y) = \Sigma\alpha(x, y_1) \wedge \ldots \wedge \Sigma\alpha(x, y_n),$$

whereas the certain x-discernibility function in $\mathscr{S}^c$ looks as follows:

$$\Delta^c_c(x) = \prod_{y \in \mathscr{U} \setminus I_{\{d\}}(x)} \sum \alpha^c(x, y) = \Sigma\alpha^c(x, y_1) \wedge \ldots \wedge \Sigma\alpha^c(x, y_n).$$

We may easily notice that for any $y_i, i = 1 \ldots n$: $\alpha^c(x, y_i) \cap \alpha(x, y_i) = \alpha(x, y_i)$. Hence,

$$\Delta^c_c(x) = \Sigma(\alpha(x, y_i) \cup \varphi(x, y_i)) \wedge \ldots \wedge \Sigma(\alpha(x, y_n) \cup \varphi(x, y_n)),$$
$$\text{where } \varphi(x, y_i) = \alpha^c(x, y_i) \setminus \alpha(x, y_i),\ i = 1..n.$$

The obtained form of $\Delta^c_c(x)$ allow us to infer that the set of all prime implicants of $\Delta_g(x)$ is a subset of all prime implicants of $\Delta^c_c(x)$. This means that the set of all optimal definite generalized rules supported by an object x in $\mathscr{S}$ is a subset of all optimal certain rules supported by x in $\mathscr{S}^c$. □

8. Computing certain rules by replacing examples

In this approach, an initial incomplete system $\mathscr{S}$ is transformed into a complete system $\mathscr{S}'$ as follows: any information vector in an incomplete system $\mathscr{S}$ with unknown values is replaced with a set of all possible information vectors consistent with the incomplete original information vector. Next, any method computing optimal certain rules from the complete system may be applied, e.g. the method described in Section 5. One can find in [3] an algorithmic description of how to generate certain rules in accordance with the presented approach without the use of discernibility functions.

We proved the validity of this approach in Proposition 6.2. $\mathscr{S}'$ possesses all information vectors that could occur in any completion of $\mathscr{S}$, so all objects-generators of certain rules may be used in a discovery process if required. As a result, all optimal certain rules supported in $\mathscr{S}$ can be induced from $\mathscr{S}'$. The

drawback of the replacing examples' method is a possible exponential growth of the information system.

Example 8.1. According to the presented approach, the incomplete system $\mathcal{S}$ from Table 2, would be transformed into the complete system $\mathcal{S}'$ presented in Table 6.

The following set of all optimal certain rules can be inferred from Table 6:

$(S, full) \wedge (X, low) \rightarrow (d, good)$. //supported by objects: 1,2a,2b,6a

$(S, compact) \rightarrow (d, poor)$. //supported by objects: 3a,3b,3c,3d

$(P, high) \wedge (M, high) \wedge (X, low) \rightarrow (d, poor)$. //supported by object: 3d

$(P, low) \wedge (M, low) \wedge (X, high) \rightarrow (d, excellent)$. //supported by object: 5a

9. Computing certain rules by removing examples

The method consists in removing all the information vectors with unknown values from the initial system. The rules obtained by this method may not be certain at all or some certain rules will not be generated from such a reduced system, which illustrates Example 9.1.

Table 6

Car	*Price*	*Mileage*	*Size*	*MaX-Speed*	*d*	d_{AT}
1	high	low	full	low	good	{good}
2a	low	low	full	low	good	{good}
2b	low	high	full	low	good	{good}
3a	low	low	comp	low	poor	{poor}
3b	low	high	comp	low	poor	{poor}
3c	high	low	comp	low	poor	{poor}
3d	high	high	comp	low	poor	{poor}
4a	high	low	full	high	good	{good, excel.}
4b	high	high	full	high	good	{good, excel.}
5a	low	low	full	high	excel.	{excel.}
5b	low	high	full	high	excel.	{good, excel.}
5c	high	low	full	high	excel.	{good, excel.}
5d	high	high	full	high	excel.	{good, excel.}
6a	low	high	full	low	good	{good}
6b	low	high	full	high	good	{good, excel.}

Table 7

Car	Price	Mileage	Size	MaX-Speed	d
1	high	low	full	low	good

Example 9.1. Table 7 shows the result of removing objects with unknown attribute values from the incomplete system $\mathcal{S}$ presented in Table 2.

The following certain optimal rule can be induced from Table 7:

$$\rightarrow (d, good).$$

The obtained rule is not certain in the original system $\mathcal{S}$.

10. Conclusion

The indiscernibility relation and similarity relation are useful notions for expressing the relationship between an incomplete system and its completions. Among the completions there is one which reflects the real world described by the incomplete system. Unlike the majority of the existing methods of knowledge discovery, we did not try to find out which completion is the most likely so that to perform the discovery process in it. A new method, we presented, allows to discover the knowledge in the form of optimal certain decision rules that are valid in all completions. The method requires only completing the information vector of an object-generator in the process of computing rules supported by that object. The information vectors of other objects remain unchanged. If all optimal rules supported by the object-generator are of interest then the computations of rules should be performed for all possible information vectors of the object. The computations for different descriptors of the object-generator may be performed in parallel. The method guarantees the discovery of all optimal certain rules supported in any completion of the system.

The new method is an extension of the method presented in [5], where the problem related to generalized rules was considered. Optimal generalized rules can be computed directly from the initial system. No completion of information vectors is performed. In the paper we proved that all optimal definite generalized rules constitute a subset of all optimal certain rules.

Additionally, we showed that all certain rules may be generated when applying the method of replacing examples. Unfortunately, space complexity of the method may prevent its usage. We also showed that removing objects incompletely described may lead to generation of rules that are not certain in the original system. This method does not assure either that the rules generated

from a reduced system will cover the whole set of certain rules that could be induced from the initial system.

The work [15] is a continuation of this paper. In [15], we investigate several interesting properties of incomplete systems and propose new methods of computing other kinds of decision rules.

Acknowledgements

I would like to thank the reviewers of this paper for their valuable comments and suggestions.

References

[1] I. Kononenko, I. Bratko, E. Roskar, Experiments in automatic learning of medical diagnostic rules, Technical Report, Jozef Stefan Institute, Ljubljana, Yugoslavia, 1984.

[2] J.R. Quinlan, Induction of decision trees, in: J.W. Shavlik, T.G. Dietterich (Eds.), Readings in Machine Learning, Morgan Kaufmann, Los Altos, CA, 1990, pp. 57–69.

[3] M.R. Chmielewski, J.W. Grzymala-Busse, N.W. Peterson, S. Than, The rule induction system LERS – A version for personal computers, Found. Comput. Decision Sci. 18 (3-4) (1993) 181–212.

[4] R. Slowinski, J. Stefanowski, Rough-set reasoning about uncertain data, Fund. Inform. 27 (2-3) (1996) 229–244.

[5] M. Kryszkiewicz, Rough set approach to incomplete information systems, in: Proceedings of Second Annual Joint Conference on Information Sciences: Fuzzy Logic, Neural Computing, Pattern Recognition, Computer Vision, Evolutionary Computing, Information Theory, Computational Intelligence, Wrightsville Beach, North Carolina, USA, 28 September 1 October 1995, pp. 194–197; the extended version of the paper will appear in J. Inform. Sci.

[6] A. Skowron, C. Rauszer, The Discernibility Matrices and Functions in Information Systems, in: R. Slowinski (Ed.), Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory, Kluwer Academic Publisher, Dordrecht, 1992, pp. 331–362.

[7] Z. Pawlak, A. Skowron, A rough set approach to decision rules generation, ICS Research Report 23/93, Warsaw University of Technology, 1993.

[8] A. Skowron, A synthesis of decision rules: Applications of discernibility matrix, in: Proceedings of the International Conference on Intelligent Information Systems, Augustow, Poland, 1993, pp. 30–46.

[9] A. Skowron, Boolean reasoning for decision rules generation, in: J. Komorowski, Z. Ras (Eds.), Proceedings of the Seventh International Symposium ISMIS'93, Trondheim, Norway, 1993, Lecture Notes in Artificial Intelligence, vol. 689, Springer, Berlin, 1993, pp. 295–305.

[10] M. Kryszkiewicz, The Algorithms on Knowledge Reduction in Information Systems, Ph.D. Thesis, Warsaw University of Technology, Institute of Computer Science, 1994.

[11] A. Skowron, Extracting laws from decision tables, Comput. Intelligence 11 (2) (1995) 371–388.

[12] A. Skowron, J. Stepaniuk, Generalized approximation spaces, in: T.Y. Lin, A.M. Wildberger (Eds.), Soft Computing, Simulation Councils, San Diego, 1995, pp. 18–21.

[13] A. Skowron, L. Polkowski, Synthesis of decision systems from data tables, in: T.Y. Lin, N. Cercone (Eds.), Rough Sets and Data Mining, Analysis of of Imprecise Data, Kluwer Academic Publishers, Dordrecht, 1997, pp. 259–299.

[14] W.J. Lipski, On semantic issues connected with incomplete information databases, ACM Trans. Databases Systems 4 (1979) 262–296.
[15] M. Kryszkiewicz, Properties of incomplete information systems in the framework of rough sets, in: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery 1, Physica-Verlag, 1998, pp. 422–450.