BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **13**, No 4

Sofia • 2013 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2013-0058

An Efficient Algorithm for Determining the Set of All Reductive Attributes in Incomplete Decision Tables

Janos Demetrovics[1], *Vu Duc Thi*[2], *Nguyen Long Giang*[2]

[1] *Institute of Computer and Automation, Hungarian Academy of Sciences*
[2] *Institute of Information Technology, Vietnam Academy of Science and Technology (VAST). 18 Hoang Quoc Viet Road, Ha Noi, Viet Nam*
Emails: Demetrovics.Janos@sztaki.mta.hu vdthi@ioit.ac.vn nlgiang@ioit.ac.vn

Abstract: *In rough set theory, the number of all reducts for a given decision table can be exponential with respect to the number of attributes. This paper investigates the problem of determining the set of all reductive attributes which are present in at least one reduct of an incomplete decision table. We theoretically prove that this problem can be solved in polynomial time. This result shows that the problem of determining the union of all reducts can be solved in polynomial time, and the problem of determining the set of all redundant attributes which are not present in any reducts can also be solved in polynomial time.*

Keywords: *Rough sets, reducts, incomplete decision tables, tolerance matrix, tolerance function.*

1. Introduction

Feature selection is one of the core problems in machine learning and data mining. The accuracy of many classification algorithms depends on the quality of selected attributes. The rough set approach to feature selection problem is based on *reducts*, which are in fact the minimal sets of attributes that preserve some necessary amount of information. However, the number of all reducts for a given decision table can be exponential with respect to the number of attributes. Therefore, we are forced to search either for minimal length reducts or for core attributes, i.e., the attributes that occur in all reducts. The minimal reduct problem is NP-hard while the searching for a core attribute problem can be solved in polynomial time.

In decision tables, a conditional attribute is called a *reductive attribute* if it is present in at least one reduct. A conditional attribute is called a *redundant attribute* if it is not a reductive attribute, i.e., it is not present in any reduct. Reductive attributes are used for object classification, while redundant attributes do not play any role in object classification. Redundant attributes must be eliminated before rule extraction. It is easy to see that the problem of determining the set of all redundant attributes becomes a problem of determining the set of all reductive attributes. To solve this problem, a common approach is to calculate the union of all reducts. However, the number of all reducts can be exponential with respect to the number of attributes. Thus the problem of calculating all reducts is not efficient for large datasets and we are forced to find another approach. In consistent complete decision tables, by considering a decision table as a relation, N g u y e n L o n g G i a n g and V u D u c T h i [3] have proposed an algorithm for determining the set of all reductive attributes in polynomial time based on some results of the relation database.

In fact, there are many cases where the decision tables contain missing values for at least one conditional attribute in the value set of this attribute and these decision tables are called *incomplete decision tables*. To obtain decision rules directly from incomplete decision tables, M. K r y s z k i e w i c z [2] has defined a tolerance relation based on the equivalent relation in a classical rough set and proposed a tolerance rough set. In the tolerance rough set, M. Kryszkiewicz has proposed an attribute reduction method in incomplete decision tables based on Boolean reasoning approach [2].

In this paper, we investigate the problem of determining the set of all reductive attributes in incomplete decision tables based on Boolean reasoning approach [1, 2, 6]. As a result, we prove that this problem can be solved completely in polynomial time. This result shows that the problem of determining the set of all redundant attributes can also be solved in polynomial time.

The structure of this paper is as follows. Section 2 presents some basic concepts in the tolerance rough set in incomplete information systems. Section 3 presents the Boolean reasoning approach to reduct calculation problems in incomplete decision tables, as well as the computational complexity of these problems. The conclusions and future remarks are presented in the last section.

2. Basic concepts

An Information System is a pair $\mathrm{IS} = (U, A)$, where the set *U* denotes the *universe of objects* and *A* is the set of *attributes*, i.e., mappings of the form: $a : U \to V_a$, and V_a is called the *value set* of attribute *a*. If V_a contains a missing value for at least one attribute $a \in A$, then IS is called an *incomplete information system*, otherwise it is *complete*. Further on, we will denote the missing value by *.

An incomplete decision table is an Incomplete Information System $\mathrm{IDS} = (U, A \cup \{d\})$ where $d, d \notin A$, and $* \notin V_d$, is a distinguished attribute called decision attribute, and the elements of A are called conditional attributes.

Let us consider the incomplete decision table below (Table 1). Attributes Price, Mileage, Size and Max-Speed are conditions, whereas Decision is the decision attribute.

Table 1. An example incomplete decision table

Car	Price	Mileage	Size	Max-speed	Decision
u_1	High	High	*	*	Poor
u_2	Low	*	Full	Low	Good
u_3	Low	Low	Compact	High	Poor
u_4	Medium	High	Compact	High	Good
u_5	Medium	High	Compact	High	Good
u_6	Medium	High	Compact	*	Good

We will refer to decision attribute *Decision* as *d*, and to conditional attributes *Price, Mileage, Size* and *Max-Speed* as to $a_1,...,a_4$ in this order.

The tolerance rough set has been introduced by Kryszkiewicz [2] as a tool for extracting decision rules directly from incomplete decision tables. The idea is to define a tolerance relation in incomplete information systems based on the equivalent relation in complete information systems. For a subset of attributes $B \subseteq A$ we define *B-tolerance relation* $\mathrm{SIM}(B)$ (defined on $U \times U$) as follows:

$$\mathrm{SIM}(B) = \left\{(u,v) \in U \times U \middle| \forall a \in B,\ a(u) = a(v) \text{ or } a(u) = * \text{ or } a(v) = *\right\}.$$

The relation $\mathrm{SIM}(B)$ is a tolerance relation and it defines a covering of U into tolerance classes which we denote by $S_B(u)\ (u \in U)$, where $S_B(u) = \{v \in U \,|\, (u,v) \in \mathrm{SIM}(B)\}$. Clearly, $\mathrm{SIM}(B) = \bigcap_{a \in B} \mathrm{SIM}(\{a\})$ [2].

Let $\mathrm{IDS} = (U, A \cup \{d\})$ be an incomplete decision table. For $B \subseteq A$, $u \in U$, $\partial_B(u) = \{d(v) | v \in S_B(u)\}$ is called a *generalized decision* in IDS. If $\mathrm{card}(\partial_A(u)) = 1$ for any $u \in U$ where $\mathrm{card}(\partial_B(u))$ is the number of elements in $\partial_B(u)$, then IDS is *consistent*, otherwise it is *inconsistent* [2].

It has been shown that one of the crucial concepts in rough set theory is a *reduct* or *decision reduct* (see [5]). In general, reducts are minimal subsets (with respect to the set inclusion relation) of attributes which contain a necessary portion of information about the set of all attributes [1, 6].

According to Kryszkiewicz [2], a reduct of an incomplete decision table is a minimal attribute set which preserves the generalized decision for all objects.

Definition 1. [2] Let $IDS = (U, A \cup \{d\})$ be an incomplete decision table. If $R \subseteq A$ satisfies

(1) $\partial_R(u) = \partial_A(u)$ for any $u \in U$,

(2) $\forall R' \subset R$, $\partial_{R'}(u) \neq \partial_A(u)$,

then *R* is called a *reduct* of IDS based on a generalized decision.

The set of all reducts of a given incomplete decision table $\mathrm{IDS} = (U, A \cup \{d\})$ is denoted by

$$\mathrm{RED}(\mathrm{IDS}) = \{R \subseteq A : R \text{ is a reduct of } \mathrm{IDS}\}.$$

The attribute $a \in A$ called a *core attribute* if and only if *a* presents in all reducts of *A*. The set of all core attributes is denoted by

$$\mathrm{CORE}(\mathrm{IDS}) = \bigcap_{R \in \mathrm{RED}(\mathrm{IDS})} R.$$

The attribute $a \in A$ is called *reductive attribute* if and only if *a* presents in at least one reduct of *A*. The set of all reductive attributes is denoted by

$$\mathrm{REAT}(\mathrm{IDS}) = \bigcup_{R \in \mathrm{RED}(\mathrm{IDS})} R.$$

It is obvious that $\mathrm{CORE}(\mathrm{IDS}) \subseteq R \subseteq \mathrm{REAT}(\mathrm{IDS})$. for any reduct $R \in \mathrm{RED}(\mathrm{IDS})$.

An attribute is called a *redundant attribute* if it is not a reductive attribute. In other words, a redundant attribute is not presented in any reduct of *A*. The set of all redundant attributes is denoted by

$$\mathrm{REDU}(\mathrm{IDS}) = A - \mathrm{REAT}(\mathrm{IDS}).$$

For example, with the incomplete decision table given in Table 1, $S_A(u_1) = \{u_1\}$, $S_A(u_2) = \{u_2\}$, $S_A(u_3) = \{u_3\}$, $S_A(u_5) = S_A(u_5) = S_A(u_6) = \{u_4, u_5, u_6\}$, $\mathrm{card}(\partial_A(u_1)) = 1$, $\mathrm{card}(\partial_A(u_2)) = 1$, $\mathrm{card}(\partial_A(u_3)) = 1$, $\mathrm{card}(\partial_A(u_4)) = 1$, $\mathrm{card}(\partial_A(u_5)) = 1$, $\mathrm{card}(\partial_A(u_6)) = 1$. The set of all reducts of the incomplete decision table is $\mathrm{RED}(\mathrm{IDS}) = \{\{a_1, a_3\}, \{a_1, a_4\}\}$. Thus, $\mathrm{CORE}(\mathrm{IDS}) = \{a_1\}$, $\mathrm{REAT}(\mathrm{IDS}) = \{a_1, a_3, a_4\}$, $\mathrm{REDU}(\mathrm{IDS}) = \{a_2\}$.

3. Boolean reasoning approach in incomplete decision tables and complexity results

Boolean reasoning approach to a reduct calculation problem in complete decision tables has been explained in [1, 6]. Boolean reasoning approach to the reduct calculation problem in incomplete information systems without decision attribute has been explained in [2]. The nice idea of the approach is a tool showing that the reduct calculation problem is equivalent to the prime implicit problem for discernibility functions.

To calculate reducts in incomplete decision tables based on this approach, firstly we define a *tolerance relationship discernibility matrix* to calculate discernibility functions, called *tolerance matrix* for short.

Let $\mathrm{IDS} = (U, A \cup \{d\})$ be an incomplete decision table. We denote

$$D_d(u,v) = \begin{cases} \{a \in A \mid a(u) \neq a(v) \wedge a(u) \neq * \wedge a(v) \neq *\} & \text{if } d(v) \notin \partial_A(u), \\ \varnothing & \text{if } d(v) \in \partial_A(u). \end{cases}$$

Then $D_d(u,v)$ is called a *discernibility attribute set* with respect to the tolerance relation of *IDS*, and $M_d(A)=(D_d(u,v):\ u,v\in U)$ is called a *tolerance matrix* of *IDS*.

It is easy to see that a tolerance matrix $M_d(A)=[a_{ij}]$ is a $n\times n$ table, where n is the number of objects, and the entry a_{ij} is referring to the pair of objects (u_i,u_j) that belong to tolerance classes which are with discernibility by the generalized decision. The entry a_{ij} is the set of all conditional attributes which discern these two objects with respect to the tolerance relation, i.e., it is a list of attributes *a*, such that $a(u_i)\neq a(u_j)$ and $a(u_i)\neq *$ and $a(u_j)\neq *$.

In Table 2 we present a compact form of a tolerance matrix corresponding to the decision table from Table 1, where the objects corresponding to class *Poor* are listed as columns and the objects corresponding to class *Good* are listed as rows.

Table 2. The compact form of the tolerance matrix corresponding to the decision table in Table 1

u_i	u_1	u_3
u_2	a_1	a_3, a_4
u_4	a_1	a_1, a_2
u_5	a_1	a_1, a_2
u_6	a_1	a_1, a_2

The *Boolean tolerance relationship discernibility function* corresponding to the tolerance matrix, called the *tolerance function* for short, is defined as follows:

$$\Delta_d(a_1,\dots,a_k)=\prod_{i,j:\,d(u_i)\neq d(u_j)}\ \sum_{a\in a_{ij}} a$$

where $a_1,\dots,a_k$ are the Boolean variables related to attributes from *A*, and $\prod$, $\sum$ denote the Boolean conjunction and Boolean disjunction operators. Thus, for the tolerance matrix in Table 2, the tolerance function is as follows:

(3) $$\Delta_d(a_1,\dots,a_4)=(a_1)(a_1)(a_1)(a_1)(a_3+a_4)(a_1+a_2)(a_1+a_2)(a_1+a_2).$$

It is shown in [1, 6], that the discernibility matrix and discernibility function are very important tools for calculation and analysis of reducts in complete decision tables. As a consequence of this fact, we also use the tolerance matrix and the tolerance function to solve reduct calculation problems in incomplete decision tables.

In the sequel, we define the reduct based on a tolerance matrix and prove that the reduct based on a tolerance matrix is equivalent to the reduct based on a generalized decision [2].

Definition 2. Let $\mathrm{IDS}=(U,A\cup\{d\})$ be an incomplete decision table and $M_d(A)=(D_d(x,y):\ x,y\in U)$ is the tolerance matrix of IDS. If $R\subseteq A$ satisfies

$(1)\ R\cap D_d(u,v)\neq\varnothing$ for any $D_d(u,v)\neq\varnothing,u,v\in U$,

$(2)\ \forall r\in R,\ R'=R-\{r\}$ is not satisfied (1)

then R is called a reduct of IDS based on tolerance matrix.

Theorem 1. Let $\mathrm{IDS}=\left(U,A\cup\{d\}\right)$ be an incomplete decision table and $M_d\left(A\right)=\left(D_d\left(u,v\right):\ u,v\in U\right)$ is the tolerance matrix of IDS. For any attribute set $R\subseteq A$, *if* $\partial_R(u)=\partial_A(u)\ \left(\forall u\in U\right)$ then

$$R\cap D_d\left(u,v\right)\neq\varnothing\ \left(\forall D_d\left(u,v\right)\neq\varnothing,u,v\in U\right).$$

P r o o f. Suppose that there exists $D_d\left(u_0,v_0\right)\neq\varnothing$ such that $R\cap D_d\left(u_0,v_0\right)=\varnothing$. Since $D_d\left(u_0,v_0\right)\neq\varnothing$, then $d\left(v_0\right)\notin\partial_A\left(u_0\right)$. Because $R\cap D_d\left(u_0,v_0\right)=\varnothing$, according to the definition of a tolerance matrix we have $\left(u_0,v_0\right)\in\mathrm{SIM}\left(R\right)$, that is $v_0\in S_R\left(u_0\right)$ or $d\left(v_0\right)\in\partial_R\left(u_0\right)$. From $d\left(v_0\right)\notin\partial_A\left(u_0\right)$, $d\left(v_0\right)\in\partial_R\left(u_0\right)$ we can conclude that $\partial_R(u_0)\neq\partial_A(u_0)$. This is in contradiction with the precondition, so the assumption is not true, thus $\partial_R(u)=\partial_A(u)$ holds for any $u\in U$.

Theorem 2. Let $\mathrm{IDS}=\left(U,A\cup\{d\}\right)$ be an incomplete decision table and $M_d\left(A\right)=\left(D_d\left(u,v\right):\ u,v\in U\right)$ is the tolerance matrix of IDS. For any attribute set $R\subseteq A$, *if* $R\cap D_d\left(u,v\right)\neq\varnothing\ \left(\forall D_d\left(u,v\right)\neq\varnothing,u,v\in U\right)$ then $\partial_R(u)=\partial_A(u)\ \left(\forall u\in U\right)$.

P r o o f. Suppose that there exists $u_0\in U$ such that $\partial_R(u_0)\neq\partial_A(u_0)$. Because $\partial_A(u_0)\subseteq\partial_R(u_0)$, then there exists d_1 such that $d_1\in\partial_R(u_0)\wedge d_1\notin\partial_A(u_0)$. Suppose that $d_1=d\left(u_1\right)$, it is easy to see that $u_1\in S_R\left(u_0\right)$ and $u_1\notin S_A\left(u_0\right)$. From $d\left(u_1\right)\notin\partial_A(u_0)$ and $u_1\notin S_A\left(u_0\right)$, we have $D_d\left(u_0,u_1\right)\neq\varnothing$. For any $a\in D_d\left(u_0,u_1\right)$ we have $a\left(u_0\right)\neq a\left(u_1\right)$ and $a\left(u_0\right)\neq *$ and $a\left(u_1\right)\neq *$. From $u_1\in S_R\left(u_0\right)$, $u_1\notin S_A\left(u_0\right)$ we can conclude that $a\in A-R$. This implies that $D_d\left(u_0,u_1\right)\cap R=\varnothing$. This is in contradiction with the precondition, so the assumption is not true, thus $\partial_R(u)=\partial_A(u)$ holds for any $u\in U$.

Theorem 3. Let $\mathrm{IDS}=\left(U,A\cup\{d\}\right)$ be an incomplete decision table. Then, the reduct of IDS based on generalized decision is the same as the reduct based on a tolerance matrix.

P r o o f. Suppose that $\mathrm{RED}\left(\mathrm{IDS}\right)$ is the set of all reducts of IDS based on a generalized decision and $\mathrm{IRED}\left(\mathrm{IDS}\right)$ is the set of all reducts of IDS based on a tolerance matrix.

(1) For any $R\in\mathrm{RED}\left(\mathrm{DS}\right)$ we have $\partial_R(u)=\partial_A(u)\ \left(\forall u\in U\right)$, according to Theorem 1, we have $R\cap D_d\left(u,v\right)\neq\varnothing\ \left(\forall D_d\left(u,v\right)\neq\varnothing,u,v\in U\right)$. Moreover, if there exists $u_0\in U,R^{'}\subset R$ such that $\partial_{R^{'}}(u_0)\neq\partial_A(u_0)$, according to the method proving Theorem 2, we can conclude that there exists $u_0,u_1\in U$ such that

$D_d\left(u_0,u_1\right)\cap R^{'}=\varnothing$. According to Definition 2, R is a reduct of IDS based on a tolerance matrix. This implies that $\mathrm{RED}\left(\mathrm{IDS}\right)\subseteq\mathrm{IRED}\left(\mathrm{IDS}\right)$

(2) For any $R\in\mathrm{IRED}\left(\mathrm{DS}\right)$ we have $R\cap D_d\left(u,v\right)\neq\varnothing\ \left(\forall D_d\left(u,v\right)\neq\varnothing,u,v\in U\right)$, according to Theorem 2, we have $\partial_R(u)=\partial_A(u)\ \left(\forall u\in U\right)$. Moreover, if there exists $R^{'}\subset R$, $D_d\left(u_0,v_0\right)\neq\varnothing$, such that $R^{'}\cap D_d\left(u_0,v_0\right)=\varnothing$, according to the method proving Theorem 1, we can conclude that there exists $u_0\in U$ such that $\partial_{R^{'}}(u_0)\neq\partial_A(u_0)$. According to Definition 1, R is a reduct of *IDS* based on the generalized decision. This implies that $\mathrm{IRED}\left(\mathrm{IDS}\right)\subseteq\mathrm{RED}\left(\mathrm{IDS}\right)$.

From (1) and (2) we have $\mathrm{RED}\left(\mathrm{IDS}\right)=\mathrm{IRED}\left(\mathrm{IDS}\right)$. In other words, the reduct based on a generalized decision is the same as the reduct based on a tolerance matrix.

It is shown from Theorem 3 and the Boolean reasoning approach that the reduct calculation problem in incomplete decision tables is the same as that in complete decision tables [1, 6]. This means that the set of attributes $R=\left\{a_{i_1},...,a_{i_j}\right\}$ is a reduct in an incomplete decision table if and only if the monomial $m_R=a_{i_1}\cdot...\cdot a_{i_j}$ is a prime implicant of $\Delta_d\left(a_1,...,a_k\right)$. As a consequence of this fact, both the problem of searching for minimal length reducts, as well as the problem of searching for all reducts of a given incomplete decision table, are NP-hard.

The question is related to the computational complexity of the problems of reductive attributes. We will use the tolerance matrix and tolerance function to prove that this problem can be solved in *polynomial time*. Therefore, the proof is also based on Boolean reasoning approach.

The main idea is based on the absorption law in Boolean algebra, which states that $x+\left(x\cdot y\right)=x$ and $x\cdot\left(x+y\right)=x$ where *x*, *y* are some Boolean functions. In other words, in Boolean algebra, the longer expressions are absorbed by the shorter ones. For the Boolean function in (1), $\left(a_1\right)$ absorbs $\left(a_1+a_2\right)$.

The Boolean expression is called the irreducible CNF if it is in CNF (Conjunctive Normal Form) and it is not possible to apply the absorption law on its clauses.

As an example, the irreducible CNF of the discernibility function in (1) is as follows: $\Delta_d\left(a_1,...,a_4\right)=a_1\cdot\left(a_3+a_4\right)$. We have the following

Theorem 4. For any incomplete decision table $\mathrm{IDS}=\left(U,A\cup\left\{d\right\}\right)$. If

$$\Delta_d\left(a_1,...,a_k\right)=\left(\sum_{a\in C_1}a\right)\cdot\left(\sum_{a\in C_2}a\right)\cdots\left(\sum_{a\in C_m}a\right)$$

is the irreducible CNF of discernibility function $\Delta_d\left(a_1,...,a_k\right)$ then

$$\mathrm{REAT}\left(\mathrm{IDS}\right)=\bigcup_{i=1}^{m}C_i.$$

P r o o f: We will prove the inclusion in both directions that Equation 2 holds:

(1) Let $a\in\mathrm{REAT}\left(\mathrm{IDS}\right)$. From the definition, there exists a reduct $R\in\mathrm{RED}\left(\mathrm{IDS}\right)$, such that $a\in R$. From Definition 2 we have $R\cap C_i\neq\varnothing$ for $i=1,...,m$. If $a\notin C_i$ then $R-a\cap C_i=R\cap C_i\neq\varnothing$. This implies that if $a\notin\bigcup_{i=1}^{m}C_i$

then there exists a subset of $R-\{a\}$ which is also a reduct of IDS, and this is a contradiction. Hence we have $a \in \bigcup_{i=1}^{m} C_i$ i.e.,

$$\mathrm{REAT}(\mathrm{IDS}) \subseteq \bigcup_{i=1}^{m} C_i .$$

(2) We can use the fact that the irreducible CNF of a monotone Boolean function is unique to prove the inverse inclusion:

$$\bigcup_{i=1}^{m} C_i \subseteq \mathrm{REAT}(\mathrm{IDS}).$$

If $a \in \bigcup_{i=1}^{m} C_i$ and a is a redundant attribute, then

$$\Delta_d^{(1)}(a_1,...,a_k) = \prod_{i=1}^{m}\left(\sum_{a_j \in C_i} a_j\right) \text{ and } \Delta_d^{(2)}(a_1,...,a_k) = \prod_{i=1}^{m}\left(\sum_{a_j \in C_i - a} a_j\right)$$

are the two different irreducible CNF forms of the discernibility function $\Delta_d(a_1,...,a_k)$, which is the contradiction. Consequently, if $a \in \bigcup_{i=1}^{m} C_i$ then $a \in \mathrm{REAT}(\mathrm{IDS})$ i.e. $\bigcup_{i=1}^{m} C_i \subseteq \mathrm{REAT}(\mathrm{IDS})$.

The following algorithm is the straightforward application of Theorem 4.

Algorithm 1: Determining all reductive attributes of an incomplete decision table.

D a t a: an incomplete decision table $\mathrm{IDS} = (U, A \cup \{d\})$.

R e s u l t: $\mathrm{REAT}(\mathrm{IDS})$ – the set of all reductive attributes of IDS.

Step 1. Calculate the tolerance matrix $M_d(A)$.

Step 2. Reduce $M_d(A)$ using the absorption law; assume that $C_1,...,C_m$ are the nonempty entries of $M_d(A)$ after reduction.

Step 3. Return $\mathrm{REAT}(\mathrm{IDS}) = \bigcup_{i=1}^{m} \subseteq C_i$ as the set of all reductive attributes of IDS.

If $|A| = k$ and $|U| = n$, then the construction of the tolerance matrix requires $O(n^2k)$ steps and the reducing phase using an absorption law requires at most $O(n^4k)$ steps. Therefore, the problem of calculation of all reductive attributes can be solved in $O(n^4k)$ steps.

For example, with the incomplete decision table in Table 1, the compact form of the tolerance matrix is as Table 2, the irreducible CNF of the discernibility function is $\Delta_d(a_1,...,a_4) = a_1 \cdot (a_3 + a_4)$, that is $C_1 = (a_1)$, $C_2 = (a_3 + a_4)$. Consequently, $\mathrm{REAT}(\mathrm{IDS}) = \{a_1, a_3, a_4\}$ and $\mathrm{REDU}(\mathrm{IDS}) = \{a_2\}$.

4. Conclusions

In this paper we have presented an approach to the problem of determining the set of all reductive attributes for an incomplete decision table. The approach is based on tolerance matrix and Boolean reasoning methodology. As a result, an algorithm for determining the set of all reductive attributes of an incomplete decision table was proposed. We also proved that the time complexity of the proposed algorithm is polynomial in the number of rows and columns of the incomplete decision table. This result shows that we can determine all redundant attributes in incomplete decision tables in polynomial time. This plays an important role in eliminating redundant attributes in incomplete decision tables before rule extraction. However, the method proposed seems to have quite high complexity. In the worse case, the proposed solutions may need $O\left(n^4k\right)$ steps, where n is the number of objects and k is the number of attributes in the incomplete decision table.

We are planning to work on more efficient methods to reduce the time complexity of the proposed solutions. The idea may be based on the attempt to realize the same algorithms without implementation of a tolerance matrix.

Acknowledgments. This work was funded by Vietnam's National Foundation for Science and Technology Development (NAFOSTED) via a research grant for fundamental sciences, Grant number: 102.01-2010.09.

References

1. H u n g S o n N g u y e n. Approximate Boolean Reasoning. Foundations and Applications in Data Mining. – Transactions on Rough Sets, Vol. **5**, 2006, 334-506.
2. K r y s z k i e w i c z, M. Rough Set Approach to Incomplete Information Systems. – Information Science, Vol. **112**, 1998, 39-49.
3. N g u y e n L o n g G i a n g, V u D u c T h i. Some Problems Concerning Condition Attributes and Reducts in Decision Tables. – In: Proceeding of the Fifth National Symposium "Fundamental and Applied Information Technology Research" (FAIR), Dong Nai, Viet Nam, 2012, 142-152.
4. P a w l a k, Z. Rough Sets. – In: International Journal of Information and Computer Sciences, Vol. **11**, 1982, 5341-5356.
5. P a w l a k, Z. Rough Sets – Theoretical Aspects of Reasoning about Data. Dordrecht, Kluwer Academic Publishers, 1991.
6. S k o w r o n, A., C. R a u s z e r. The Discernibility Matrices and Functions in Information Systems. – In: Słowi´nski [8]. Chapter 3. 331-362.
7. S k o w r o n, A., Z. P a w l a k, J. K o m o r o w s k i, L. P o l k o w s k i. A Rough Set Perspective on Data and Knowledge. – In: W.Kloesgen and ˙ J. Zytkow, Eds. Handbook of KDD. Oxford, Oxford University Press, 2002, 134-149.
8. R. Słowi´nski, Ed. Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory. Vol. 11 of D: System Theory, Knowledge Engineering and Problem Solving. Dordrecht, Netherlands, Kluwer Academic Publishers, 1992.