ISBN 978-83-60810-51-4

On elimination of redundant attributes in decision tables

Long Giang Nguyen
Institute of Information Technology,
VAST, Viet nam
Email: nlgiang@ioit.ac.vn

Hung Son Nguyen
Institute of Mathematics, Warsaw University
Banach 2, 02-097, Warsaw, Poland
Email: son@mimuw.edu.pl

***Abstract*—Most decision support systems based on rough set theory are related to the minimal reduct calculation problem, which is NP-hard. This paper investigates the problem of searching for the set of useful attributes that occur in at least one reduct. By compliment, this problem is equivalent to searching for the set of redundant attributes, i.e. the attributes that do not occur in any reducts of the given decision table. We show that the considered problem is equivalent to a Sperner system for relational data base system and prove that it can be solved in polynomial time. On the base of these theoretical results, we also propose some algorithms for elimination of redundant attributes in decision tables.**

***Index Terms*—rough sets, reducts, relational database, minimal keys, Sperner system**

I. Introduction

FEATURE selection is one of the crucial problems in machine learning and data mining. The accuracy of many classification algorithms depends on the quality of selected attributes. Rough set approach to feature selection problem is based on reducts, which are in fact the minimal (with respect to inclusion) sets of attributes that preserve some necessary amount of information. Unfortunately, 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 whilst the searching for core attribute problem can be solved in polynomial time.

This paper investigates the problem of identifying the set of attributes, that are present in at least one reduct. Such attributes are called the *reductive attributes*. The not reductive attributes are called *redundant attributes* because they do not play any role in object classification. For a given decision table, the problem of searching for all reductive attributes becomes the problem of determining the union of all reducts of the given decision table, or determining the set of all redundant attributes of a decision table.

In this paper we present two approaches to the investigated problem. Firstly, we present the fundamental analysis of the problem of searching for reductive attributes. Using Boolean reasoning approach we prove that the problem can be solved completely in polynomial time. Moreover, we can consider the decision table as the relation over the set of attributes and apply some results in relational database theory to solve the mentioned problems. We propose an algorithm to determine the set of all reductive attributes of consistent decision tables based on the methods of searching for keys, antikeys and prime attributes in decision table (see [1], [2]).

The structure of this paper is as follows. Section II and Section III presents some basic concepts in rough set theory as well as the computational complexity of the reduct calculation problems. Section IV presents the concept of reducts in decision table from the view point of relational database theory. We also propose an algorithm to determine the set of all reductive attributes of a consistent decision table. In Section V, we perform some experiments of the proposed algorithm. The conclusions and future remarks are presented in the last section.

II. Basic concepts

An *information system* is a pair $\mathbb{A} = (U, A)$, where the set U denotes the *universe of objects* and A is the set of *attributes*, i.e. the mappings of the form: $a : U \rightarrow V_a$. The set V_a is called the *domain* or *the value set* of attribute a.

A decision system is an information system $\mathbb{D} = (U, A \cup \{dec\})$ where dec is a distinguished attribute called the *decision attribute* or briefly *decision*. The remaining attributes are called *conditional attributes* or briefly *conditions*. For convenience, we assume that the domain of decision attribute consists of two or very few values. For any $k \in V_{dec}$ the set

$$CLASS_k = \{u \in U : dec(u) = k\}$$

is called the *decision class* of $\mathbb{D}$

As an example, let us consider the decision system below (Table I). Attributes *Diploma, Experience, French* and *Reference* are *condition attributes*, whereas *Decision* is the decision attribute. We will refer to decision attribute *Decision* as dec, and to conditional attributes *Diploma, Experience, French* and *Reference* as to $a_1, \ldots, a_4$ in this order. In this example there are two decision classes related to the values *Accept* and *Reject* of the decision attribute domain. These decision classes are as follow:

$$CLASS_{Accept} = \{x_1, x_4, x_6, x_8\}$$

$$CLASS_{Reject} = \{x_2, x_3, x_5, x_7\}$$

Rough set theory has been introduced by Professor Z.Pawlak [6] as a tool for concept approximation under

TABLE I
AN EXAMPLE DECISION SYSTEM REPRESENTED AS A TABLE.

	Diploma	Experience	French	Reference	Decision
x_1	MBA	Medium	Yes	Excellent	Accept
x_2	MBA	Low	Yes	Neutral	Reject
x_3	MCE	Low	Yes	Good	Reject
x_4	MSc	High	Yes	Neutral	Accept
x_5	MSc	Medium	Yes	Neutral	Reject
x_6	MSc	High	Yes	Excellent	Accept
x_7	MBA	High	No	Good	Accept
x_8	MCE	Low	No	Excellent	Reject

uncertainty. The idea is to approximate the concept by two descriptive sets called *lower and upper approximations*. One of the assumptions in rough set theory that differs it from other methods in soft computing and concept approximation is that the lower and upper approximations must be extracted from the information that is available in training data.

One of the simplest ways to define the lower and upper approximations has been proposed by Prof. Z.Pawlak in [7]. This approach to concept approximation is based on the indiscernibility relation.

For a subset of attributes $B \subseteq A$ we define *B-indiscernibility relation* $IND(B)$ and *decision-relative indiscernibility relation* $IND_{dec}(B)$ (both defined on $U \times U$) as follows:

$$\begin{aligned}(x, y) \in IND(B) &\iff \forall_{a \in A} a(x) = a(y)\\ (x, y) \in IND_{dec}(B) &\iff dec(x) = dec(y) \vee\\ &\qquad \forall_{a \in A} a(x) = a(y)\end{aligned}$$

The relation $IND(B)$ is an equivalence relation and it defines a partitioning of U into equivalence classes which we denote by $[x]_B$ ($x \in U$). The complement of $IND(B)$ in $U \times U$ is called *discernibility relation*, denoted $DISC(B)$. The lower and upper approximations of a concept X (using attributes from B) are defined by

$$\mathbf{L}_B(X) = \left\{x \in U : [x]_{IND(B)} \subseteq X\right\} \quad \text{and}$$
$$\mathbf{U}_B(X) = \left\{x \in U : [x]_{IND(B)} \cap X \neq \varnothing\right\}.$$

The main philosophy of rough set approach to concept approximation problem is based on minimizing the difference between upper and lower approximations (also called the *boundary region*). This simple, but brilliant idea, leads to many efficient applications of rough sets in machine learning and data mining like feature selection, rule induction, discretization or classifier construction [9].

It has been shown that all of the problems mentioned above are related to one of the crucial concepts in rough set theory, called *reducts* or *decision reducts* (see [7]). 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 [8][5].

There are several ways to define reducts in Rough set theory. We will further focus on the following one.

A *decision-relative reduct* is a minimal set of attributes $R \subseteq A$ such that

$$IND_{dec}(R) = IND_{dec}(A).$$

This condition guarantees that R contains all information necessary to discern objects belonging to different classes. The set of all decision reducts of a given decision table $\mathbb{D} = (U, A \cup \{dec\})$ is denoted by

$$\mathcal{RED}(\mathbb{D}) = \{R \subseteq A : R \text{ is a reduct of } \mathbb{D}\}$$

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

$$CORE(\mathbb{D}) = \bigcap_{R \in \mathcal{RED}(\mathbb{D})} R$$

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

$$REAT(\mathbb{D}) = \bigcup_{R \in \mathcal{RED}(\mathbb{D})} R$$

It is obvious that

$$CORE(\mathbb{D}) \subseteq R \subseteq REAT(\mathbb{D})$$

for any reduct $R \in \mathcal{RED}(\mathbb{D})$.

The attribute is called *redundant attribute* if it is not a reductive attribute. In other words, redundant attribute is not presented in any reduct of A.

For example, the set of all reducts of the decision table in Table I is $\mathcal{RED}(\mathbb{D}) = \{\{a_1, a_2\}, \{a_2, a_4\}\}$. Thus

$$CORE(\mathbb{D}) = \{a_2\} \quad REAT(\mathbb{D}) = \{a_1, a_2, a_4\}$$

In this example, a_3 is the redundant attribute.

III. COMPLEXITY RESULTS

The concept of decision reducts using discernibility matrix has been explained in [8]. This simple and nice idea is also a tool for showing that the reduct calculation problem is equivalent to the prime implicant problem for boolean functions.

In fact, discernibility matrix for a given decision table $\mathbb{D} = (U, A \cup \{dec\})$, denoted by $\mathbb{M}_{\mathbb{D}}(A) = [C_{ij}]$, is a $n \times n$ table, where n is the number of object, and the entry c_{ij} is referring to the pair of objects (x_i, x_j) that belong to different decision classes. The entry c_{ij} is the set of all conditional attributes which discern the objects x_i and x_j, i.e.

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\}$$

In Table II we present a compact form of *decision-relative discernibility matrix* corresponding to the decision system from Table I, where the objects corresponding to class *Accept* are listed as columns and the objects corresponding to class *Reject* are listed as rows.

TABLE II
THE COMPACT FORM OF DECISION-RELATIVE DISCERNIBILITY MATRIX CORRESPONDING TO DECISION SYSTEM IN TABLE I.

	x_1	x_4	x_6	x_7
x_2	a_2, a_4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4
x_3	a_1, a_2, a_4	a_1, a_2, a_4	a_1, a_2, a_4	a_1, a_2, a_3
x_5	a_1, a_4	a_2	a_2, a_4	a_1, a_2, a_3, a_4
x_8	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1, a_2, a_4

The Boolean reasoning approach to reduct calculation problem is based on encoding it by the boolean *discernibility function* defined as follows:

$$\Delta_{\mathbb{D}}(a_1, \ldots, a_k) = \prod_{i,j:d(x_i)\neq d(x_j)} \sum_{a\in C_{ij}} a$$

where $a_1, \ldots, a_k$ are the boolean variables related to attributes from A, and $\prod, \sum$ denote the Boolean conjunction and Boolean disjunction operators. Thus, for the discernibility matrix in Table II, the discernibility function can be written as follows:

$$\begin{aligned}\Delta_{\mathbb{D}}(a_1, \ldots, a_4) = {} & (a_2 + a_4)(a_1 + a_2)(a_1 + a_2 + a_4)\\ & (a_2 + a_3 + a_4)(a_1 + a_2 + a_4)(a_1 + a_2 + a_4)\\ & (a_1 + a_2 + a_4)(a_1 + a_2 + a_3)(a_1 + a_4)(a_2)(a_2 + a_4)\\ & (a_1 + a_2 + a_3 + a_4)(a_1 + a_2 + a_3)(a_1 + a_2 + a_3 + a_4)\\ & (a_1 + a_2 + a_3)(a_1 + a_2 + a_4) \qquad (1)\end{aligned}$$

It has been shown in [8],[5] that the set of attributes $R = \{a_{i_1}, \ldots, a_{i_j}\}$ is a reduct in $\mathbb{D}$ if and only if the monomial

$$m_R = a_{i_1} \cdot \ldots \cdot a_{i_j}$$

is a prime implicant of $\Delta_{\mathbb{D}}(a_1, \ldots, a_k)$. 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 decision table are NP-hard.

In terms of decision-relative discernibility matrix, a decision reduct R is a minimal subset of attributes so that for each non-empty entry C_{ij} of $\mathbb{M}(\mathbb{A})$, $C_{ij} \cap R \neq \emptyset$.

Discernibility matrix and discernibility function are very important tools for calculation and analysis of reducts. Let us recall the following well known fact (see [8],[5]).

Theorem 1: *For any attribute $a \in A$, a is a core attribute if and only if a occurs in discernibility matrix as a singleton. As a consequence, the problem of searching for core attributes can be solved in polynomial time.*

For the example from Table I, according to the Theorem 1, attribute a_2 (Experience) is the core attribute, because this is the only attribute that discerns x_4 and x_5 (see also Table II). And we can determine it without calculation of all reducts of this table.

The question is related to computational complexity of the problems of reductive attributes. We will use the discernibility matrix and discernibility function to prove that this problem can be solved in polynomial time. Therefore, once again, the Boolean reasoning approach shows to be a useful tool for reduct calculation problem.

The main idea is based on the *absorption law* in Boolean algebra, which states that

$$x + (x \cdot y) = x \qquad x \cdot (x + y) = x$$

where x, y are the arbitrary Boolean functions. In other words, in Boolean algebra, the longer expressions are absorbed by the shorter ones. For the Boolean function in Equation 1, $(a_1 + a_2)$ absorbs $(a_1 + a_2 + a_4)$ but, at the same time, it is absorbed by (a_2).

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 Equation 1 is as follows:

$$\Delta_{\mathbb{D}}(a_1, \ldots, a_4) = a_2 \cdot (a_1 + a_4)$$

We have the following theorem

Theorem 2: *For any decision table $\mathbb{D} = (U, A \cup \{dec\})$, if*

$$\Delta_{\mathbb{D}}(a_1, \ldots, a_k) = \left(\sum_{a\in C_1} a\right) \cdot \left(\sum_{a\in C_2} a\right) \ldots \left(\sum_{a\in C_m} a\right) \qquad (2)$$

is the irreducible CNF of discernibility function $\Delta_{\mathbb{D}}(a_1, \ldots, a_k)$, then

$$REAT(\mathbb{D}) = \bigcup_{i=1}^{m} C_i \qquad (3)$$

Proof of Theorem 2: As (2) is the irreducible CNF of discernibility function, the family $\{C_1, \ldots, C_m\}$ should satisfy the following properties:

- It is an antichain, i.e. $C_i \nsubseteq C_j$ and $C_j \nsubseteq C_i$ for any $i, j \in \{1, \ldots, m\}$
- If R is a reduct, i.e. $R \in \mathcal{RED}(\mathbb{D})$, then $R \cap C_i \neq \emptyset$ for any $i \in \{1, \ldots, m\}$.

We will prove that the inclusions in both directions of the Equation (3) hold:

1) $REAT(\mathbb{D}) \subseteq \bigcup\limits_{i=1}^{m} C_i$:
   Let $a \in REAT(\mathbb{D})$. From the definition, there exists a reduct $R \in \mathcal{RED}(\mathbb{D})$ such that $a \in R$. This means that $R \cap C_i \neq \emptyset$ for $i = 1, \ldots, m$.
   If $a \notin \bigcup_{i=1}^{m} C_i$ then for any $i \in \{1, \ldots, m\}$ we have $a \notin C_i$, which implies that
   $$(R - \{a\}) \cap C_i = R \cap C_i \neq \emptyset.$$
   Thus there exists a subset of $R - \{a\}$ which is also a reduct of $\mathbb{D}$, and this is a contradiction.
   Hence we have $a \in \bigcup_{i=1}^{m} C_i$.
2) $\bigcup\limits_{i=1}^{m} C_i \subseteq REAT(\mathbb{D})$:
   We can use the fact that the irreducible CNF of monotone Boolean function is unique to prove this inverse inclusion.

Indeed, if $a \in \bigcup_{i=1}^{m} C_i$ and a is a redundant attribute, then $R \cap C_i - \{a\} \neq \emptyset$ for each reduct $R \in \mathcal{RED}(\mathbb{D})$. Thus

$$\Delta_{\mathbb{D}}^{(1)}(a_1, \ldots, a_k) = \prod_{i=1}^{m} \left(\sum_{a_j \in C_i} a_j \right)$$

and

$$\Delta_{\mathbb{D}}^{(2)}(a_1, \ldots, a_k) = \prod_{i=1}^{m} \left(\sum_{a_j \in C_i - \{a\}} a_j \right)$$

are the two different irreducible CNF form of the discernibility function $\Delta_{\mathbb{D}}(a_1, \ldots, a_k)$, what is the contradiction.

■

The following algorithm is the straightforward application of the above theorem:

Algorithm 1: Determining all reductive attributes of a decision table.

```
Data: a consistent decision table $\mathbb{D} = (U, A \cup \{dec\})$;
Result: $REAT(A)$ – the set of all reductive attributes of $\mathbb{D}$;
Step 1: Calculate the discernibility matrix $\mathbb{M}_{\mathbb{D}}(A)$;
Step 2: Reduce $\mathbb{M}_{\mathbb{D}}(A)$ using absorption law; Assume
    that $C_1, \ldots, C_m$ be the nonempty entries of $\mathbb{M}_{\mathbb{D}}(A)$ after
    reduction;
Step 3: Return $REAT(A) = \bigcup_{i=1}^{m} C_i$ as the set of all
    reductive attributes of $\mathbb{D}$.
```

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

IV. Decision Tables in terms of Relational Databases

Let us give some necessary definitions and results of the theory of relation database that can be found in [1], [2], [3], [4], [11], [12].

A. Relational Database Theory

Let $A = \{a_1, \ldots, a_k\}$ be a finite set of attributes and let $D(a_i)$ be the set of all possible values of attribute a_i, for $i = 1, \ldots, k$. Any subset of the Cartesian product

$$\mathcal{R} \subseteq D(a_1) \times D(a_2) \times \ldots \times D(a_k)$$

is called *the relation over* A. In other words, relation over A is the set of tuples $\{h_1, \ldots, h_n\}$ where

$$h_j : A \to \bigcup_{a_i \in A} D(a_i),$$

is a function that $h_j(a_i) \in D(a_i))$ for $1 \leq j \leq n$.

Let $\mathcal{R} = \{h_1, \ldots, h_n\}$ be a given relation over $A = \{a_1, \ldots, a_k\}$. Any pair of attribute sets $B, C \subseteq A$ is called the functional dependency (FD for short) over A, and denoted by $B \to C$, if and only if for any pair of tuples $h_i, h_j \in \mathcal{R}$:

$$\forall_{a \in B}(h_i(a) = h_j(a)) \implies \forall_{a \in C}(h_i(a) = h_j(a))$$

The set $\mathcal{F}_{\mathcal{R}} = \{(B, C) : B, C \subset A; B \to C\}$ is called the *full family of functional dependencies in* $\mathcal{R}$.

Let $\mathbb{P}(A)$ be the power set of attribute set A. A family $\mathcal{R} \subset \mathbb{P}(A) \times \mathbb{P}(A)$ is called *an f-family over* A if and only if for all subsets of attributes $P, Q, S, T \subseteq A$ the following properties hold:

$$R1.(P, P) \in \mathcal{R} \tag{4}$$

$$R2.(P, Q) \in \mathcal{R}, (Q, S) \in \mathcal{R} \implies (P, S) \in \mathcal{R} \tag{5}$$

$$R3.(P, Q) \in \mathcal{R}, P \subseteq S, T \subseteq Q \implies (S, T) \in \mathcal{R} \tag{6}$$

$$R4.(P, Q) \in \mathcal{R}, (R, S) \in \mathcal{R} \implies (P \cup R, Q \cup T) \in \mathcal{R} \tag{7}$$

Clearly $\mathcal{F}_{\mathcal{R}}$ is an f-family over A. It is also known that if $\mathcal{F}$ is an f-family over A then there is a relation $\mathcal{R}$ such that $\mathcal{F}_{\mathcal{R}} = \mathcal{F}$.

A pair $\mathbb{S} = (A, \mathcal{F})$, where A is a set of attributes and $\mathcal{F}$ is a set of FDs on A, is called the *relation scheme*. Let us denote by $\mathcal{F}^+$ the set of all FDs, which can be derived from $\mathcal{F}$ by using the rules $R1 - R4$.

For any $B \subseteq A$, the set

$$B^+ = \{a \in A : B \to a \in \mathcal{F}^+\}$$

is called *the closure* of B on $\mathbb{S}$. It is clear that $B \to C \in \mathcal{F}^+$ if and only if $C \subseteq B^+$.

A set of attributes $B \subset A$ is called the *key* of $\mathbb{S} = (A, \mathcal{F})$ iff $B \to A \in \mathcal{F}^+$. The set B is the *minimal key* of $\mathbb{S} = (A, \mathcal{F})$ if B is a key of $\mathbb{S}$ and any proper subset of B is not a key of $\mathbb{S}$. Let us denote by $\mathcal{K}(\mathbb{S})$ the set of all minimal keys of the given relation scheme $\mathbb{S}$.

Recall that a family $\mathcal{K} \subseteq \mathbb{P}(A)$ is a Sperner system if for any $K_1, K_2 \in \mathcal{K}$ implies $K_1 \nsubseteq K_2$. Clearly $\mathcal{K}(\mathbb{S})$ is Sperner system.

Let $\mathcal{K} = \mathcal{K}_{\mathbb{S}}$ be a Sperner system over A containing all minimal keys of $\mathbb{S}$. We defined the set of antikeys of $\mathcal{K}$, denoted by $\mathcal{K}^{-1}$, as follows:

$$\begin{aligned}\mathcal{K}^{-1} = \{B \subseteq A : &(\forall_{C \in \mathcal{K}} C \nsubseteq B) \wedge \\ &(B \subseteq D \to \exists_{C \in \mathcal{K}} C \subseteq D)\}\end{aligned}$$

It is easy to see that $\mathcal{K}^{-1}$ is the set of subsets of A, which does not contain the elements of $\mathcal{K}$ and which is maximal for this property. They are the maximal non-keys. Clearly, $\mathcal{K}^{-1}$ is also a Sperner system.

For the given relation $\mathcal{R} = \{h_1, \ldots, h_n\}$ over $A = \{a_1, \ldots, a_k\}$, let

$$\mathbb{E}(\mathcal{R}) = \{E_{ij} : 1 \leq i < j \leq n\},$$

where $E_{ij} = \{a \in A : h_i(a) = h_j(a)\}$. The family $\mathbb{E}(\mathcal{R})$ is called *the equality set of* $\mathcal{R}$. It is easy to notice that in the worse case, $\mathbb{E}(\mathcal{R})$ contains $O(n^2)$ subsets of attributes, where n is the number of records in relation $\mathcal{R}$.

It is known (see [1]) that for each subset of attributes $B \subseteq A$, the following property holds:

$$B^+ = \begin{cases} \bigcap_{ij=1}^{n} E_{ij} & \text{if } B \subseteq E_{ij} \text{ for some } E_{ij} \in \mathbb{E}(\mathcal{R}) \\ A & otherwise \end{cases}$$

Let $\mathbb{S} = (A, \mathcal{F})$ be a relation scheme over attribute set A. For any attribute $a \in A$, the set

$$\mathcal{K}_{\mathbb{S}}(a) = \{B \subseteq A : B \to a \wedge \nexists_{C \subset B} C \to a\}$$

is called the *family of minimal sets of the attribute a over $\mathbb{S}$*. It is known that $\{a\} \in \mathcal{K}_{\mathbb{S}}(a)$, $A \notin \mathcal{K}_{\mathbb{S}}(a)$ and $\mathcal{K}_{\mathbb{S}}(a)$ is a Sperner system over A.

B. Relational Database Theory and Reducts

In rough set theory the minimal sets from $\mathcal{K}_{\mathbb{S}}(a)$ are strongly related to the concept of decision reducts. Any decision table $\mathbb{D} = (U, A \cup \{dec\})$ can be treated as a relation $U = \{u_1, \ldots, u_n\}$ over the set of all attributes $A \cup \{dec\}$. It is clear that

$$\mathcal{K}_{\mathbb{S}}(dec) = \mathcal{RED}(\mathbb{D}) \cup \{dec\}$$

where $\mathbb{S}$ is the relation scheme induced from the decision table $\mathbb{D}$.

In relational database theory, the following important facts has been proven (see e.g. in [2]).

Lemma 1: *The following equality holds for any Sperner system $\mathcal{K}$ over the set of attribute A:*

$$\bigcup_{K \in \mathcal{K}} K = A - \bigcap_{K \in \mathcal{K}^{-1}} K$$

As the consequence we have the following theorem

Theorem 3: *Let $\mathbb{D} = (U, A \cup \{dec\})$ be a consistent decision table, the set of reductive attributes can be determined by:*

$$\begin{aligned} REAT(A) &= \bigcup_{K \in \mathcal{K}_{\mathbb{S}}(dec)} K - \{dec\} \\ &= A - \bigcap_{K \in (\mathcal{K}_{\mathbb{S}}(dec))^{-1}} K \end{aligned} \tag{8}$$

Therefore, the main problem is to calculate the family $(\mathcal{K}_{\mathbb{S}}(dec))^{-1}$. According to the theory of relational database in previous Section we have

$$\begin{aligned} (\mathcal{K}_{\mathbb{S}}(dec))^{-1} = \{B \subseteq A :& (B \to dec \notin \mathcal{F}^+) \wedge \\ & (B \subsetneq C \Rightarrow C \to dec \in \mathcal{F}^+)\} \end{aligned}$$

It is clear that for any set of attributes $B \subseteq A$ we have $B \in (\mathcal{K}_{\mathbb{S}}(dec))^{-1}$ if and only if

$$B \in \mathbb{E}(\mathcal{R}) \wedge \nexists_{C \in \mathbb{E}(\mathcal{R})} (dec \in C \text{ and } (B \subsetneq C - \{dec\})$$

The method of determining the set of reductive attributes using the equality set $\mathbb{E}(\mathcal{R})$ is presented in Algorithm 2.

Similar to Algorithm 1, the size of E_{ij} is $O(n^2k)$, where k and n are the number of attributes and number of objects. Thus, in the worse case, the construction of $\mathcal{M}(dec)$ requires at most $O(n^4k)$ steps. Therefore the problem of calculation of all reductive attributes can be solved in $O(n^4k)$ steps.

Algorithm 2: Determining the set of all reductive attributes of a decision table.

```
Data: a consistent decision table $\mathbb{D} = (U, A \cup \{dec\})$;
Result: $REAT(A)$ – the set of all reductive attributes of $\mathbb{D}$;
Step 1: Calculate the equality system
        $\mathbb{E}(\mathcal{R}) = \{E_{ij} : 1 \le i < j \le n\}$
    where $E_{ij}$ is the set of attributes that have the same values for $u_i$ and $u_j$;
Step 2: Let
        $\mathbb{E}_d = \{B \in \mathbb{E}(\mathcal{R}) : dec \in B\}$
        $\mathbb{E}_0 = \{B \in \mathbb{E}(\mathcal{R}) : dec \notin B\}$
Step 3: Construct the family of subsets of $A$:
        $\mathcal{M}(dec) = \{B \in \mathbb{E}_0 : \forall_{C \in \mathbb{E}_0} (B \cap C \neq B)\}$
Step 4: Construct the set $V = \bigcap_{K \in \mathcal{M}(dec)} K$
Step 5: Return $REAT(A) = A - V$ as the set of all reductive attributes of $\mathbb{D}$.
```

C. Example

Let us consider the exemplary decision table in Table I. The equality set $\mathbb{E}(\mathcal{R})$ of this table is presented in Table III.

TABLE III
THE EQUALITY SET OF THE DECISION TABLE FROM TABLE I

$\mathbb{E}(\mathcal{R})$: without dec	with dec
$\{a_1, a_3\}$	$\{a_3, dec\}$
$\{a_3, a_4\}$	$\{a_3, a_4, dec\}$
$\{a_3\}$	$\{a_1, dec\}$
$\{a_1\}$	$\{a_1, a_2, a_3, dec\}$
$\{a_4\}$	$\{a_2, dec\}$
$\{a_2, a_3\}$	$\{a_2, a_3, dec\}$
$\{a_1, a_3, a_4\}$	$\{a_1, a_2, dec\}$
	$\{dec\}$

In fact, $\mathbb{E}(\mathcal{R})$ consists of different subsets of the attribute set $A \cup \{dec\} = \{a_1, a_2, a_3, a_4, dec\}$. However, for the clear representation, we divided $\mathbb{E}(\mathcal{R})$ into two parts: the subsets that do not contain the decision dec are placed in left column and the subsets that contain the decision dec are placed in the right column.

One can see that the left column can be calculated by taking the complements of all entries of the discernibility matrix in Table II.

The next step is to calculate $\mathcal{M}(dec)$, which is the family maximal subsets among the subsets of $\mathbb{E}(\mathcal{R})$ that do not contain dec. In this example

$$\mathcal{M}(dec) = \{\{a_2, a_3\}, \{a_1, a_3, a_4\}\}$$

Thus $V = \{a_3\}$ and $REAT(A) = \{a_1, a_2, a_4\}$

TABLE IV
THE RESULTS OF EXPERIMENT ON SOME BENCHMARK DATA SETS USING THE PROPOSED ALGORITHM

Sequence number	**Data sets**	$\|U\|$	$\|A\|$	t	**The reductive attributes**	**The redundant attributes**
1	Adult stretch	20	4	0.93	$3, 4$	$1, 2$
2	Soybean small data	47	35	2.74	$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 20, 21, 22, 23, 24, 25, 26, 27, 28, 35$	$11, 13, 14, 15, 16, 17, 18, 19, 29, 30, 31, 32, 33, 34$
3	Sponge.data	76	45	2.1	$1, \ldots, 11, 13, \ldots, 34, 36, \ldots, 45$	$12, 35$
4	Zoo.data	101	17	3.19	$1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17$	$3, 6, 16$

V. EXPERIMENTS

The experiments on PC (Pentium Dual Core 2.13 GHz, 1GB RAM, WINXP) are performed on 4 data sets obtained from UCI Machine Learning Repository[1]. We present the results of calculation the set of all reductive attributes and the set of all redundant attributes in Table IV. In this Table $|U|$, $|A|$ are the numbers of objects and condition attributes, and t is the time of operation (calculated by second). Conditional attributes will be denoted by $1, 2, \ldots |A|$.

VI. CONCLUSIONS

In this paper, we presented two alternative approaches to the problem of determining the set of all reductive attributes for a decision table. The first approach is based on discernibility matrix and Boolean reasoning methodology.

The second approach is based on Sperner system using the equality set. We defined the family of all minimal sets of an attribute over a relation based on the definition of the family of minimal sets of an attribute over a relation scheme [2], so the concept of reduct in decision tables is equivalent to that of minimal set of an attribute in a relation. As a result, an algorithm for determining all reduced attributes of a consistent decision table was proposed based on some results proposed in [1]. We also proved that the time complexity of proposed algorithm is polynomial in the number of rows and columns of the decision table. This results play an important role in rejecting redundant attributes in decision tables before attribute reduction and rule extraction.

The positive result is related to the fact that the set of reductive attributes can be calculated in polynomial time. However both proposed methods seem to have quite a high complexity. In the worst case, the proposed solutions may need $O(n^4k)$ steps, where n in the number of objects and k is the number of attributes in the decision table.

We are planing to work on the 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 discernibility matrix.

[1] The UCI machine learning repository, http://archive.ics.uci.edu/ml/

ACKNOWLEDGMENT

This work is partially supported by the Vietnam's National Foundation for Science and Technology Development (NAFOSTED) via a research grant for fundamental sciences, grant number: 102.01-2010.09; the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: "Inter-disciplinary System for Interactive Scientific and Scientific-Technical Information"; Polish Ministry of Science and Higher Education grant N N516 077837 and Polish National Science Centre grant 2011/01/B/ST6/03867.

REFERENCES

[1] Demetrovics J., Thi V.D. Keys, antikeys and prime attributes. Ann. Univ. Scien. Budapest Sect. Comput. 8 (1987), 37-54.

[2] Demetrovics J., Thi V.D. Some remarks on generating Armstrong and inferring functional dependencies relation, Acta Cybernetica, 12 (1995), 167-180.

[3] Demetrovics J. and Thi V.D., Describing candidate keys by hypergraphs, Computers and Articial Intelligence, 18 (2) (1999), 191-207.

[4] Demetrovics J. and Thi V.D., Some computational problems related to Boyce-Codd normal form, Annales Univ. Sci. Budapest. Sect. Comp., 19 (2000), 119-132.

[5] Nguyen H.S.: Approximate Boolean Reasoning: Foundations and Applications in Data Mining. Transactions on Rough Sets: Volume 5, 2006, pages. 334-506 (2006)

[6] Pawlak Z. Rough sets. *International Journal of Information and Computer Sciences*, 11(5):341–356, 1982.

[7] Pawlak Z. *Rough Sets. Theoretical Aspects of Reasoning about Data*. Springer, Formerly Kluwer Academic Publishers, Boston, Dordrecht, London, 1991.

[8] Skowron, A. and Rauszer, C. The discernibility matrices and functions in information systems. In Słowiński [10], chapter 3, pages 331–362.

[9] Skowron, A., Pawlak, Z., Komorowski, J., and Polkowski, L. (2002). A rough set perspective on data and knowledge. In Kloesgen, W. and Żytkow, J., editors, *Handbook of KDD*, pages 134–149. Oxford University Press, Oxford.

[10] Słowiński R., editor. *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, volume 11 of *D: System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1992.

[11] Thi V.D. and Son N.H., Some Problems Related to Keys and the Boyce-Codd Normal Form, Acta Cybernetica, 16 (2004), 473-483.

[12] Thi V.D. and Son N.H., On Armstrong Relations for Strong Dependencies, Acta Cybernetica, 17 (3) (2006), 521-531.