BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 2
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0015

Some Properties Related to Reduct of Consistent Decision Systems

Nguyen Long Giang[1], *Demetrovics Janos*[2], *Vu Duc Thi*[3], *Phan Dang Khoa*[3]

[1]*Institute of Information Technology, VAST, Viet Nam*
[2]*Institute for Computer Science and Control* (*SZTAKI*), *Hungarian Academy of Sciences, Hungary*
[3]*Institute of Information Technology, VNU, Viet Nam*
E-mails: nlgiang@ioit.ac.vn demetrovics@sztaki.mta.hu vdthi@vnu.edu.vn khoapd@vnu.edu.vn

***Abstract*:** *Reduct of decision systems is the topic that has been attracting the interest of many researchers in data mining and machine learning for more than two decades. So far, many algorithms for finding reduct of decision systems by rough set theory have been proposed. However, most of the proposed algorithms are heuristic algorithms that find one reduct with the best classification quality. The complete study of properties of reduct of decision systems is limited. In this paper, we discover equivalence properties of reduct of consistent decision systems related to a Sperner-system. As the result, the study of the family of reducts in a consistent decision system is the study of Sperner-systems.*

***Keywords*:** *Relational database, rough set theory, Sperner-system, decision system, reduct.*

1. Introduction

Reduction of attribute set is the most important problem in the pre-processing step of data mining and machine learning. The target of reduction of attribute set is to remove redundant and unnecessary attributes in order to find reduct attribute set (known as reduct) to increase the efficiency of knowledge extracting models. Rough Set Theory (RST) introduced by P a w l a k [1] is considered as an effective method to find reduct of decision systems. So far, researchers have proposed many algorithms to find reduct of decision systems based on RST and extended RST. However, all proposed algorithms are heuristic algorithms that find one reduct with the best classification quality. The complete study of properties of reducts of decision systems is limited.

On consistent decision systems, in recent years there has been a number of publications related to reducts of consistent decision systems according to the relational database theory approach [2-8]. In papers [2, 4], authors have constructed the algorithm to find all reductive attributes of consistent decision systems in polynomial time. On that basis, they propose a polynomial algorithm to build a new reduction decision system from a given decision system and build an algorithm to extract all the functional dependencies of a consistent decision system. Authors in [3]

propose an algorithm to build a consistent decision system from a given functional dependency set. Based on the results in [2], the authors in [5] develope an algorithm to find all reductive attributes of consistent incomplete decision systems in polynomial time. In paper [6], authors prove that the problem to find all reducts has the time complexity in exponentials in the cardinality of conditional attribute set. In paper [7], authors have calculated the complexity of some algorithms related to reduct of consistent decision systems. In addition, based on some results related to reduct, the authors in paper [8] propose an algorithm to reduce the object set in a decision system in order to obtain a new decision system with smaller size.

Sperner-system plays an important role in studying properties of relational database theory. In paper [6], authors have discovered some relationships between Sperner-systems and minimal sets of an attribute in a decision system. Based on relational database theory approach, in this paper we study the relationship between reduct of a consistent decision system and a Sperner-system. As the result, the study of the family of reducts in a consistent decision system is equivalent to the study of Sperner-systems.

The rest of the paper is structured as follows. Section 2 introduces some original concepts related to rough set theory and relational database. Section 3 studies the equivalence properties of reduct with a Sperner-system. Section 4 is conclusions and further research directions.

2. Some basic concepts and results

In this section, we introduce some original concepts and results related to rough set theory and relational database, which can be found in [1, 9].

Definition 1. Suppose that $R=\{a_1,...,a_n\}$ is an attribute set and $D(a_i)$ is the value domain of a_i. A relation *r* defined on *R* is the tuple set $\{t_1,...,t_m\}$ such that for any $1\le j\le m$, $t_j : R\to \bigcup_{a_i\in R} D(a_i)$ is a map that $t_j(a_i)\in D(a_i)$.

Assume that $r=\{t_1,...,t_m\}$ is a relation defined on $R=\{a_1,...,a_n\}$. For any $X,Y\subseteq R$, $X\to Y$ is a Functional Dependency defined on *R* (FD for short) if for any $t_i,t_j\in r$ $(\forall x\in X)(t_i(x)=t_j(x))\Rightarrow(\forall y\in Y)(t_i(y)=t_j(y))$. Assume that F_r is the set of all functional dependencies in r, then:

(1) $X\to X,$

(2) $X\to Y,\ Y\to Z\Rightarrow X\to Z,$

(3) $X\to Y,\ X\subseteq Z, Y\subseteq T\Rightarrow Z\to T,$

(4) $X\to Y,\ Z\to T\Rightarrow X\cup Z\to Y\cup T.$

Suppose that F is a FD on *R*. We denote F^+ as the FD set, F^+ can be obtained from F by using the rules (1)-(4).

A *relation-scheme* is defined as $s=(R,F)$, where R is an attribute set and F is a FD set defined on *R*. For any $X\subseteq R$, *the closure* of *X* on s is

$X^+ = \{a : X \to \{a\} \in F^+\}$. It can be shown that $X \to Y \in F^+$ iff $Y \subseteq X^+$. Similarly, *the closure* of *X* on r is $X_r^+ = \{a : X \to \{a\} \in F^+\}$.

$\mathcal{E} \subseteq P(R)$ is a Sperner-System defined on *R*, and if $\forall X, Y \in \mathcal{E}$, $X \not\subset Y$. We denote a Sperner-System as SS. For a SS, $\mathcal{E}$, the set $\mathcal{E}^{-1}$ is defined as follows:

$\mathcal{E}^{-1} = \{X \subset R : (Y \in \mathcal{E}) \Rightarrow (Y \not\subset X)\}$ and if $(X \subset Z) \Rightarrow (\exists Y \in \mathcal{E})(Y \subseteq Z)$

We can see that $\mathcal{E}^{-1}$ is a SS defined on *R* too, and $\mathcal{E}^{-1}$ is called anti-keys.

Suppose that $s = (R, F)$ is a relation-scheme defined on *R* and $a \in R$. Then $\mathcal{E}_a^r = \{X \subseteq R : X \to \{a\}, \nexists Y \subseteq R, Y \subset X : Y \to \{a\}\}$. $\mathcal{E}_a^r$ is minimal sets of *a* defined on *r*.

Definition 2. An Information System is $\mathrm{IS} = (U, A)$ in which *U* is the object set and *A* is the attribute set. For each attribute $a \in A$ by V_a we denote the Value domain of *a*, $V = \bigcup_{a \in A} V_a$; f is a map: $(U, A) \to V$ such that $f(u, a) \in V_a$.

Definition 3. A Decision System is $\mathrm{DS} = (U, C \cup \{d\})$ in which *C* is the Condition attribute set, d is the decision attribute and $\{d\} \notin C$.

A decision system DS is consistent when the FD, $C \to \{d\}$, is true. Conversely, DS is inconsistent**.** We denote a Consistent Decision System as CDS.

Definition 4. Given $\mathrm{DS} = (U, C \cup \{d\})$ be a CDS and an attribute $B \subseteq C$. *B* is called a reduct of DS if:

1) for $\forall x, y \in U$ if $B(x) = B(y)$ then $d(x) = d(y)$.

2) for $\forall E \subset B$ there exists $(x, y) \in U$ such that $E(x) = E(y)$ and $d(x) \neq d(y)$.

The above reduct is called Pawlak REDuct (PRED). Let $\mathrm{PRED}(C)$ be the set of all reducts of *C*.

In [6], authors have proved the following result.

Theorem 1 [6]. Assume that $\mathrm{DS} = (U, C \cup \{d\})$ is a CDS and $C = \{a_1, a_2, ..., a_n\}$, $U = \{x_1, x_2, ..., x_m\}$. Consider the relation $r = \{x_1, x_2, ..., x_m\}$ defined on $R = C \cup \{d\}$. We set $\mathcal{E}_r = \{E_{ij}\}$ where $E_{ij} = \{c \in R : c(x_i) = c(x_j)\}$, $1 \le i < j \le m$ and $\mathcal{M}_d = \{X \in \mathcal{E}_r : d \notin X, \nexists Y \in \mathcal{E}_r, X \subset Y : d \notin Y\}$. $\mathcal{E}_a^r = \{X \subseteq R : X \to \{a\}, \nexists Y \subseteq R, Y \subset X : Y \to \{a\}\}$ Then we have $\mathcal{M}_d = (\mathcal{K}_d^r)^{-1}$ where $\mathcal{K}_d^r$ is the minimal sets of *d* on relation *r*.

3. Some results related to the equivalent properties of the family of reducts with respect to Sperner-systems

In this part, we propose some results related to the equivalent properties of the family of reducts with respect to SS. First, we give some related results.

Theorem 2 [6]. Assume that $\mathrm{DS} = \left(U, C \cup \{d\}\right)$ is a CDS. Then $\left(\mathcal{K}_d^r\right)^{-1}$ is a SS defined on *C*. On the contrary, if $\mathcal{K}$ is a SS defined on *C* then there exists a CDS $\mathrm{DS} = \left(U, C \cup \{d\}\right)$ such that $\mathcal{K} = \left(\mathcal{K}_d^r\right)^{-1}$.

In the following, we give an algorithm to find anti-key set $\mathcal{K}^{-1}$ from $\mathcal{K}$ where $\mathcal{K}$ is a SS.

Algorithm 1 [11]. Finding $\mathcal{K}^{-1}$ from a given SS, $\mathcal{K}$.

Input: Let $\mathcal{K} = \{B_1, ..., B_m\}$ be a SS defined on *A*.

Output: $\mathcal{K}^{-1}$.

Step 1. We set $\mathcal{K}_1 = \{R - \{a\} : a \in B_1\}$. It is obviously that $\mathcal{K}_1 = \{B_1\}^{-1}$

Step *q*+1. (*q*<*m*). Assume that $\mathcal{K}_q = F_q \cup \{X_1, ..., X_{tq}\}$, where $X_1, ..., X_{tq}$ are elements of $\mathcal{K}_q$ containing B_{q+1}, $F_q = \{Z \in \mathcal{K}_q : B_{q+1} \not\subseteq Z\}$. For i=1,…, t_q, we compute $\{B_{q+1}\}^{-1}$ on X_i in the same way as $\mathcal{K}_1$. The results are denoted as $A_1^i, ..., A_{ri}^i$. Let $\mathcal{K}_{q+1} = F_q \cup \{A_p^i : A \in F_q \Rightarrow A_p^i \not\subset A_i\}$ where $1 \le i \le t_q, 1 \le p \le r_i$.

Finally, let $\mathcal{K}^{-1} = \mathcal{K}_m$.

Theorem 3 [11]. For $\forall q \ (1 \le q \le m)$ and $\mathcal{K}_q = \{B_1, ..., B_q\}^{-1}$, we have $\mathcal{K}_m = \mathcal{K}^{-1}$.

It is clear that $\mathcal{K}, \mathcal{K}^{-1}$ are unique and the definition may be drawn of $\mathcal{K}^{-1}$ that Algorithm 1 does not depend on the order of the sequence $B_1, ..., B_m$. Set $\mathcal{K}_q = \mathcal{F}_q \cup \{X_1, ..., X_{tq}\}$ and l_q ($1 \le q \le m-1$) is the cardinality of K_q.

Proposition 1. Algorithm 1 has the time complexity as $O\left(|R|^2 \sum_{q=1}^{m-1} t_q u_q\right)$ where $u_q = I_q - t_q$ if $I_q > t_q$ and $u_q = 1$ if $I_q = t_q$.

It is clear that at each step of the algorithm we have, $\mathcal{K}_q$ is a SS defined on *R*. We know that the size of any SS on *R* is not more than $C_n^{[n/2]}$, where $n = |R|$. It can be seen that $C_n^{[n/2]}$ is approximate equal to $2^{n+1/2}/(\Pi . n^{1/2})$. Therefore, Algorithm 1 has the time complexity in exponential in *n*. In the cases

$I_q \le I_m\left(\forall q, 1 \le q \le m-1\right)$, Algorithm 1 has the time complexity as $O\left(|R|^2 |\mathcal{K}| \left|\mathcal{K}^{-1}\right|^2\right)$. Consequently, the time complexity of algorithm for finding $\mathcal{K}^{-1}$ is polynomial in $|R|$, $|\mathcal{K}|$ and $|\mathcal{K}|^{-1}$. Algorithm 1 is effective when $|\mathcal{K}|$, $|\mathcal{K}|^{-1}$ are small.

The equivalence between the family of reducts of consistent decision systems and SS can be seen from the above results.

Theorem 4. Assume that $\mathrm{DS} = \left(U, C \cup \{d\}\right)$ is a CDS, then $\mathrm{PRED}\left(C\right)$ is a SS defined on *C*. Otherwise, if $\mathcal{K}$ is a SS defined on *C* then there exists a CDS, $\mathrm{DS} = \left(U, C \cup \{d\}\right)$, such that $\mathcal{K}$ = PRED(*C*).

Proof: Given $\mathrm{DS} = \left(U, C \cup \{d\}\right)$ be a CDS. According to the definition of reduct, PRED(*C*) is a SS defined *C*.

Suppose that $\mathcal{K} = \{A_1, A_2, ..., A_m\}$ is a SS defined on *C*. Based on Algorithm 1, from $\mathcal{K}$ we construct the anti-key set $\mathcal{K}^{-1}$. Suppose that $\mathcal{K}^{-1} = \{B_1, B_2, ..., B_m\}$. We construct a CDS, $\mathrm{DS} = \left(U, C \cup \{d\}\right)$, as follow: $U = \{x_0, x_1, ..., x_m\}$ for any $a \in C$: $a\left(x_0\right) = 0$ and $d\left(x_0\right) = 0$. For any i, $i = 1, ..., m$, and $a \in C$ we set $a\left(x_i\right) = 0$ if $a \in A_i$, otherwise $a\left(x_i\right) = i$. Set $d\left(x_i\right) = i$ where $R = C \cup \{d\}$. From Theorem 1 and Theorem 2 we have $\mathcal{K}^{-1} = \left(\mathcal{K}_d^r\right)^{-1}$. From the definition of SS, anti-key set and the definition of reduct of a CDS, we have $\mathcal{K}$ = PRED(*C*). The result is proved.

From this result, we can see that the study of the family of reducts of a CDS, $\mathrm{DS} = \left(U, C \cup \{d\}\right)$, is equivalent to the study of SS defined on *C*.

From Theorem 4 and Proposition 1, we have the following lemma:

Corollary 1. Given $\mathrm{DS} = \left(U, C \cup \{d\}\right)$ be a CDS, then the cardinality of $\mathrm{PRED}\left(C\right)$ is not more than $C_n^{[n/2]}$, where $n = |C|$.

Algorithm 2. Finding a CDS from a given SS $\mathcal{K}$ defined on *C*.

Input: Given $\mathcal{K} = \{B_1, ..., B_m\}$ be a SS defined on *C*.

Output: A CDS, $\mathrm{DS} = \left(U, C \cup \{d\}\right)$, such that $\mathcal{K} = \mathrm{PRED}\left(C\right)$.

Step 1. From $\mathcal{K}$ we construct $\mathcal{K}^{-1}$

Step 2. Suppose that $\mathcal{K}^{-1} = \{A_1, ..., A_t\}$, we set $U = \{x_0, x_1, ..., x_t\}$

- For any $a \in C$, we set $a\left(x_0\right) = 0$ and $d\left(x_0\right) = 0$
- For any $i, i = 1, ..., t$, we set $d\left(x_i\right) = i$ and $a\left(x_i\right) = 0$ if $a \in A_i$, $a\left(x_i\right) = i$ if $a \notin A_i$

Clearly, based on Theorem 4 and Algorithm 1 we have $\mathrm{DS} = \left(U, C \cup \{d\}\right)$ in which $U = \{x_0, x_1, ..., x_t\}$ such that $\mathcal{K} = \mathrm{PRED}(C)$.

Example 1. Let $C = \{a, b, c, e, f, g\}$ and $\mathcal{K} = \{(a, b), (b, c, e), (b, e, f), (e, g)\}$. From Algorithm 1 and Algorithm 2 we obtain:

$$\mathcal{K}_1 = \{(a, c, e, f, g), (b, c, e, f, g)\}$$

$$\mathcal{K}_1 = \mathcal{F}_1 \cup \{(b, c, e, f, g)\} \text{ where } \mathcal{F}_1 = \{(a, c, e, f, g)\}.$$

It is easy to see that the anti-keys of (b, c, e) on set (b, c, e, f, g) are $(c, e, f, g), (b, e, f, g), (b, c, f, g)$. So we have:

$\mathcal{K}_2 = \{(a, c, e, f, g), (b, e, f, g), (b, c, f, g)\}$. From this we obtain $\mathcal{K}_2 = \mathcal{F}_2 \cup \{(b, e, f, g)\}$ where $\mathcal{F}_2 = \{(a, c, e, f, g), (b, c, f, g)\}$.

It is shown that the antikeys of (*b*, *e*, *f*) on set (b, e, f, g) are $(e, f, g), (b, f, g), (b, e, g)$. Based on this we have:

$$\mathcal{K}_3 = \mathcal{F}_2 \cup \{(b, e, g)\} = \{(a, c, e, f, g), (b, c, f, g), (b, e, g)\}$$

Clearly, $\mathcal{K}_3 = \mathcal{F}_3 \cup \{(a, c, e, f, g), (b, e, g)\}$, where $\mathcal{F}_3 = \{(b, c, f, g)\}$

It is easy to see that the anti-keys of (*e*, *g*) on set (b, e, g) are $(b, g), (b, e)$ and the anti-keys of (*e*, *g*) on set (a, c, e, f, g) are $(a, c, f, g), (a, c, e, g)$. From this, we have

$$\mathcal{K}_4 = \mathcal{F}_3 \cup \{(a, c, f, g), (a, c, e, f), (b, e)\} = \{(b, c, f, g), (a, c, f, g), (a, c, e, g), (b, e)\}.$$

From Algorithm 2, the CDS, $\mathrm{DS} = \left(U, C \cup \{d\}\right)$, is constructed as follows: $U = (u_0, u_1, u_2, u_3, u_4)$ $\forall a \in C : a(x_0) = 0, d(x_0) = 0$. Denote $A_1 = (b, c, f, g)$, $A_2 = (a, c, f, g)$, $A_3 = (a, c, e, f)$, $A_4 = (b, e)$. We set $d(x_i) = i$ and $a(x_i) = 0$ if $a \in A_i$, $a(x_i) = i$ if $a \notin A_i$. From this results, we obtain the consistent decision system $\mathrm{DS} = \left(U, C \cup \{d\}\right)$ (Table 1).

Table 1. The obtained consistent decision system in Example 1

a	*b*	*c*	*e*	*f*	*g*	*d*
0	0	0	0	0	0	0
1	0	0	1	0	0	1
0	2	0	2	0	0	2
0	3	0	0	0	3	3
4	0	4	0	4	4	4

4. Conclusion

Attribute reduction problem is the most important problem in the data pre-processing in order to improve the efficiency of data mining and machine learning models. Based on relational database theory approach, in this paper we study the equivalence property of reducts in consistent decision systems with the Sperner-system. The results show that the study of the family of reducts in a consistent decision system is equivalent to the study of Sperner-systems. Our further research is to study a method of reducing rules on the obtained reducts to decrease the complexity of the classification model.

References

1. P a w l a k, Z. Rough sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, 1991.
2. G i a n g, N. L., V. D. T h i. Some Problems Concerning Condition Attributes and Reducts in Decision Tables. – In: Proc. of FAIR, Dong Nai, Vietnam, 2011, pp. 142-152.
3. T h i, V. D., N. L. G i a n g. A Method to Construct Decision Table from Relation Scheme. – Cybernetics and Information Technologies, Vol. **11**, 2011, No 3, pp. 32-41.
4. T h i, V. D., N. L. G i a n g. A Method for Extracting Knowledge from Decision Tables in Terms of Functional Dependencies. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 1, pp. 73-82.
5. D e m e t r o v i c s, J., V. D. T h i, N. L. G i a n g. An Efficient Algorithm for Determining the Set of All Reductive Attributes in Incomplete Decision Tables. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 4, pp. 118-126.
6. D e m e t r o v i c s, J., V. D. T h i, N. L. G i a n g. On Finding All Reducts of Consistent Decision Tables. – Cybernetics and Information Technologies, Vol. **14**, 2014, No 4, pp. 3-10.
7. D e m e t r o v i c s, J., V. D. T h i, T. H. D u o n g, N. L. G i a n g. On the Time Complexity of the Problem Related to Reduct of Consistent Decision Tables. – Serdica Journal of Computing, Vol. **9**, 2015, No 2, pp. 101-110.
8. D e m e t r o v i c s, J., V. D. T h i, H. M. Q u a n g, N. V. A n h. An Method to Reduc the Size of Consistent Decision Tables. – Acta Cybernetica, Vol. **23**, 2018, pp. 1039-1054.
9. D e m e t r o v i c s, J., V. D. T h i. Some Remarks on Generating Armstrong and Inferring Functional Dependencies Relation. – Acta Cybernetica, Vol. **12**, 1995, pp. 167-180.
10. A h o, A. V., J. E. H o f c r o f t, J. D. U l l m a n. The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, Mass. 1974.
11. T h i, V. D. Minimal Keys and Antikeys. – Acta Cybernetica, Vol. **7**, 1986, No 4, pp. 361-371.

Received: 26.07.2020; Second Version: 16.12.2020; Accepted: 03.02.2021