BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 3
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0025

Investigation of Dense Family of Closure Operations

Nguyen Hoang Son[1], *J. Demetrovics*[2], *Vu Duc Thi*[3], *Nguyen Ngoc Thuy*[4]

[1]*Faculty of Mathematics, University of Sciences, Hue University, Vietnam*
[2]*Computer and Automation Institute Hungarian Academy of Sciences, Hungary*
[3]*Information Technology Institute, Vietnam National University, Hanoi, Vietnam*
[4]*Faculty of Information Technology, University of Sciences, Hue University, Vietnam*
E-mails: nhson@hueuni.edu.vn demetrovics@sztaki.mta.hu vdthi@vnu.edu.vn nnthuy.cs@hueuni.edu.vn

Abstract: *As a basic notion in algebra, closure operations have been successfully applied to many fields of computer science. In this paper we study dense family in the closure operations. In particular, we prove some families to be dense in any closure operation, in which the greatest and smallest dense families, including the collection of the whole closed sets and the minimal generator of the closed sets, are also pointed out. More important, a necessary and sufficient condition for an arbitrary family to be dense is provided in our paper. Then we use these dense families to characterize minimal keys of the closure operation under the viewpoint of transversal hypergraphs and construct an algorithm for determining the minimal keys of a closure operation.*

Keywords: *Closure operation, dense family, minimal key, hypergraph, transversal hypergraph.*

1. Introduction

Closure operations are popularly used in a variety of areas, such as databases, data mining, fuzzy sets, rough sets, matroids, logic programming, etc. [7, 8, 10-13, 15]. In recent years, many interesting research results have appeared concerning combinatorial problems in the closure operations, especially for minimal keys, antikeys and closed sets of the closure operations. It is well-known that the minimal keys of the closure operation have an essential role in the theory of databases. Under the impact of the minimal key, data will be determine the individual uniquely [7, 9, 16, 18]. Besides, we are often interested in the antikeys of the closure operation, which are understood as maximal non-keys. The antikeys are widely applied in finding minimal keys as well as in some extremal problems of the closure operations [9, 16, 18]. Moreover, the closed sets are also important notions related to the closure operations see, e.g., [3, 4, 7]. The set of the wholly closed sets constitutes a meet-semilattice.

Hypergraphs are an undoubtedly useful tool for addressing many combinatorial problems [2, 9, and 17]. It is well known that the hypergraph theory is one of the very important fields of discrete mathematics which is successfully applied in theoretical and applied computer science. In the hypergraph theory, the concept of transversals is vital. Under the viewpoint of the hypergraphs, the antikeys or the minimal keys of the closure operations can be organized as hypergraphs, even, as simple hypergraphs.

In the present paper, we shall give a notion of dense family with respect to the closure operation *h*. We then point out that calculating minimal keys of *h* is equivalent to producing minimal transversals of the dense family in *h*. Based on this result, we will construct an effective algorithm based on the dense family approach to discover the minimal keys of the closure operation.

The remainder of the present paper is organized as follows. In the next section, we recall several basic concepts as well as some results related to the closure operations and the theory of hypergraphs. Then, Section 3 introduces the concept of dense family of closure operations and then provides some dense families of any closure operation. This section also presents a condition for determining a family to be dense. Then, very interesting dense families based approach for finding the wholly minimal keys of the closure operation will be presented in Section 4. We also construct an effective algorithm for extracting the minimal keys of the closure operation.

2. Preliminary

We shall recall here basic definitions and results relative to meet-semilattice, closure operations and hypergraphs, which can be also found in [2, 4, 7, 9, 14, 16].

Let us consider a nonempty and finite set *U* (also called the universe). Then, we denote the power set of *U* as 2^U which also means the set of the whole subsets of *U*. We will begin with first important concepts.

A family $\mathcal{M} \subseteq 2^U$ is called a *meet-semilattice* on the universe *U* if two following conditions are held:

$$\text{(M1) } U \in \mathcal{M},$$

$$\text{(M2) } \forall \mathcal{N} \subseteq 2^U, \varnothing \neq \mathcal{N} \subseteq \mathcal{M} \Rightarrow \bigcap \mathcal{N} \in \mathcal{M}.$$

A map $h: 2^U \to 2^U$ is called a *closure operation* on the universe *U* if, for all $X, Y \subseteq U,$ the map *h* satisfies three conditions:

$$\text{(L1) } X \subseteq h(X),$$

$$\text{(L2) } X \subseteq Y \Rightarrow h(X) \subseteq h(Y),$$

$$\text{(L3) } h(h(X)) = h(X).$$

Then, the collection of the whole closure operations on the universe *U* will be denoted as $\mathrm{Cl}(U)$. Given $h \in \mathrm{Cl}(U)$, we define a set F_h on *h* as follows:

(1) $$F_h = \{(X,Y) : Y \subseteq h(X)\},$$

where, F_h is called a *f-family* on *U*.

We now consider $h \in \mathrm{Cl}(U)$. We are often interested in a very special collection of subsets *X* of *U* where the set *X* will not be changed under the influence of the closure operation *h*. Such subsets will be called *closed set*. Here, we shall denote the set of all closed sets by $\text{Closed}(h)$. Fortunately,

(2) $$\text{Closed}(h) = \{X \subseteq U : X = h(X)\}.$$

Obviously, $\text{Closed}(h)$ is a meet-semilattice on *U*. Therefore, we can also determine the closed set, as follows:

(3) $$\text{Closed}(h) = \{h(X) : X \subseteq U\}.$$

Proposition 2.1 [7].

1) If $h \in \text{Cl}(U)$, then $\text{Closed}(h)$ is a meet-semilattice on *U*;

2) If $\mathcal{M}$ is a meet-semilattice on *U*, then the map $h_{\mathcal{M}} : 2^U \to 2^U$, in which for each $X \in 2^U, h_{\mathcal{M}}(X) = \bigcap\{B \in \mathcal{M} : X \subseteq B\}$ is a closure operation on *U*.

In the following, we present some basic concepts in the hypergraph theory.

Firstly, we consider a family $\mathcal{H}$ consisting of subsets of *U*, $\mathcal{H} = \{E_i : E_i \in 2^U, i = 1, 2, \ldots, m\}$. Then, $\mathcal{H}$ will be called a *hypergraph on the universe U* if E_i is not empty for each *i* (in [2] the union of E_i, $i = 1, 2, \ldots, m$ must be equal to *U*, however, this condition is not required in our paper). Each element of *U* is called a *vertex*, while each E_i is called a *hyperedge* or *edge of* $\mathcal{H}$. We denote the set of all hypergraphs on *U* by $\text{HG}(U)$.

For every $E_i, E_j \in \mathcal{H}$ if $E_i \subseteq E_j$ implies $E_i = E_j$, then the hypergraph $\mathcal{H}$ is called *simple hypergraph*. The set of all simple hypergraph on *U* is denoted by $\text{SH}(U)$.

Given a $\mathcal{H} \in \text{HG}(U)$, we denote $\min(\mathcal{H})$ and $\max(\mathcal{H})$ are respectively the sets of minimal and maximal edges of $\mathcal{H}$ by using set inclusion. More specifically,

(4) $$\min(\mathcal{H}) = \{E_i \in \mathcal{H} : \nexists E_j \in \mathcal{H}, E_j \subset E_i\},$$

and

(5) $$\max(\mathcal{H}) = \{E_i \in \mathcal{H} : \nexists E_j \in \mathcal{H}, E_j \supset E_i\}.$$

It can be seen that both $\max(\mathcal{H})$ and $\min(\mathcal{H})$ are uniquely determined by $\mathcal{H}$, moreover, $\min(\mathcal{H}), \max(\mathcal{H}) \in \text{SH}(U)$. If a subset *T* of *U* meets all edges of $\mathcal{H}$, then *T* is called a *transversal of the hypergraph* $\mathcal{H}$. Formally, *T* is a transversal of $\mathcal{H}$ if $T \cap E \neq \varnothing$ for each $E \in \mathcal{H}$. We denote the set of all transversals of $\mathcal{H}$ by $\text{Trs}(\mathcal{H})$. Then, the transversal *T* is *minimal* if $\forall S \subset T$, $S \notin \text{Trs}(\mathcal{H})$. The set of all minimal transversals of the hypergraph $\mathcal{H}$ will be denoted as $\text{Tr}(\mathcal{H})$ and it is also regarded as a *transversal hypergraph* of $\mathcal{H}$.

Proposition 2.2. [17] Let $\mathcal{H} \in \text{HG}(U)$. Then $\text{Tr}(\mathcal{H}) \in \text{SH}(U)$ and $\text{Tr}(\mathcal{H}) = \text{Tr}(\min(\mathcal{H}))$.

From Proposition 2.2, it is clear that we can find the minimal transversals based on $\min(\mathcal{H})$ for decreasing the computational cost. Next, we shall consider several interesting properties related to the minimal transversal of the simple hypergraphs.

Proposition 2.3. [2] Let $\mathcal{H}_1, \mathcal{H}_2 \in \mathrm{SH}(U)$. Then

1) $\mathcal{H}_1 = \mathrm{Tr}(\mathcal{H}_2)$ iff $\mathcal{H}_2 = \mathrm{Tr}(\mathcal{H}_1),$.
2) $\mathrm{Tr}(\mathcal{H}_1) = \mathrm{Tr}(\mathcal{H}_2)$ iff $\mathcal{H}_1 = \mathcal{H}_2$,
3) $\mathrm{Tr}(\mathrm{Tr}(\mathcal{H}_1)) = \mathcal{H}_1$.

We now provide an algorithm to calculate the minimal transversals of a given hypergraph.

Algorithm 2.1 [9]. Extracting the Minimal Transversals

Input: a hypergraph $\mathcal{H} = \{E_1, E_2, \ldots, E_m\}$ on *U*.

Output: the transversal hypergraph $Tr(\mathcal{H})$.

Step 0. Set $\mathcal{L}_1 = \{\{a\} : a \in E_1\}$.

Step *k* + 1 ($k < m$ **).** Suppose that

$$\mathcal{L}_k = \mathcal{S}_k \cup \{B_1, B_2, \ldots, B_{t_k}\},$$

where in which $B_i \cap E_{k+1} = \emptyset, i = 1, 2, \ldots, t_k$, and $\mathcal{S}_k = \{A \in \mathcal{L}_k : A \cap E_{k+1} \neq \emptyset\}$.

For each $i = 1, 2, \ldots, t_k$ generate the set $\{B_i \cup \{a\} : a \in E_{k+1}\}$. Denote them by $A_1^i, A_2^i, \ldots, A_{r_i}^i$,

$$\mathcal{L}_{k+1} = \mathcal{S}_k \cup \{A_l^i : A \in \mathcal{S}_k \Rightarrow A \not\subset A_l^i, 1 \leq i \leq t_k, 1 \leq l \leq r_i\}.$$

Step *m* + 1. Return $\mathrm{Tr}(\mathcal{H}) = \mathcal{L}_m$.

It is easy to determine that the computational complexity of the above algorithm is exponential in *n*. In many cases, however, the computational time is not greater than $\mathcal{O}(n^2 m |\mathrm{Tr}(\mathcal{H})|^2)$ [9]. Then, if *m* is small, the algorithm is very effective.

Example 2.1. Consider the universe $U = \{a_1, a_2, a_3, a_4\}$ and the hypergraph $\mathcal{H} = \{\{a_1, a_3\}, \{a_2\}, \{a_3, a_4\}\}$, we have.

$\mathcal{L}_1 = \{\{a_1\}, \{a_3\}\}$,

$\mathcal{L}_2 = \{\{a_1, a_2\}, \{a_2, a_3\}\}$,

$\mathcal{L}_3 = \{\{a_1, a_2, a_4\}, \{a_2, a_3\}\}$.

Therefore, $\mathrm{Tr}(\mathcal{H}) = \{\{a_1, a_2, a_4\}, \{a_2, a_3\}\}$.

3. *h*-dense family of closure operations

We will introduce in this section a concept of a dense family with respect to the closure operation *h* and its related results.

First, let us consider a collection $\mathcal{E}$ consisting of subsets of *U* and a set $F_{\mathcal{E}}$ on $\mathcal{E}$, as below:

(6) $$F_{\mathcal{E}} = \{(X, Y) : \forall A \in \mathcal{E}, X \subseteq A \Rightarrow Y \subseteq A\}.$$

Then, we will say that the family $\mathcal{E}$ is *h-dense* (or *dense in h*) if $F_h = F_{\mathcal{E}}$. It is easy to show that for every $\mathcal{E}_1, \mathcal{E}_2 \subseteq 2^U$, if $\mathcal{E}_1 \subseteq \mathcal{E}_2$ then $F_{\mathcal{E}_2} \subseteq F_{\mathcal{E}_1}$.

We now point out that dense families with respect to an arbitrary closure operation *h* always exist.

Proposition 3.1. $\mathrm{Closed}(h)$ is the greatest *h*-dense family.

Proof: First, we show $\mathrm{Closed}(h)$ is a *h*-dense family.

We treat $(X, Y) \in F_h$. Let $h(Z) \in \mathrm{Closed}(h)$ satisfying $X \subseteq h(Z)$. From this, and according to properties of closure operation, we always have $h(X) \subseteq h(Z)$. By the definition of F_h, we obtain $Y \subseteq h(Z)$. Consequently, $(X, Y) \in F_{\mathrm{Closed}(h)}$. Conversely, let $(X, Y) \in F_{\mathrm{Closed}(h)}$. Because $X \subseteq h(X)$ and $h(X) \in \mathrm{Closed}(h)$, we have $Y \subseteq h(X)$. Hence, $(X, Y) \in F_h$.

Now, assume that $\mathcal{E}$ is an arbitrary *h*-dense family. Note that

$$F_h = \{(X, Y) : \forall A \in \mathcal{E},\, X \subseteq A \Rightarrow Y \subseteq A\}.$$

Let $A \in \mathcal{E}$. Because $(A, h(A)) \in F_h$ and $A \subseteq A$, we have $h(A) \subseteq A$. On the other hand, we have $A \subseteq h(A))$. Thus, $h(A) = A$. This also means $A \in \mathrm{Closed}(h)$.

□

Thus $\mathrm{Closed}(h)$ is a *h*-dense family and it is, moreover, the greatest one with respect to *h*. Interestingly, even when we remove several special elements, $\varnothing$ and *U*, from $\mathrm{Closed}(h)$, we still receive a *h*-dense family. Specifically, we define

(7) $$\mathrm{Closed}^*(h) = \mathrm{Closed}(h) - \{\varnothing, U\}.$$

Proposition 3.2. $\mathrm{Closed}^*(h)$ is a *h*-dense family.

Proof. According to Proposition 3.1, we have $F_h \subseteq F_{\mathrm{Closed}^*(h)}$.

Given $(X,Y) \in F_{\mathrm{Closed}^*(h)}$, we consider two cases. First, if *X* is a key of *h* then $(X, Y) \in F_h$. Second, if *X* is not a key of *h*, $h(X) \in \mathrm{Closed}^*(h)$. Furthermore, $X \subseteq h(X)$. Therefore, from the definition of $F_{\mathrm{Closed}^*(h)}$, that is,

$$F_{\mathrm{Closed}^*(h)} = \{(X, Y) : \forall A \in \mathrm{Closed}^*(h),\, X \subseteq A \Rightarrow Y \subseteq A\},$$

we imply $Y \subseteq h(X)$. Thus, $(X,\ Y) \in F_h$ which means that $F_{\mathrm{Closed}^*(h)} \subseteq F_h$.

In summary – $F_{\mathrm{Closed}^*(h)} = F_h$. □

With respect to a given closure operation *h*, we will further show a very special dense family, which is the smallest one in *h*.

Let us first consider a family $\mathcal{S} \subseteq 2^U$. Then, the family $\mathcal{S}$ is called a *generator* of meet-semilattice $\mathcal{M}$ if $\mathcal{S}^+ = \mathcal{M}$, in which $\mathcal{S}^+ = \{\cap\mathcal{T} : \mathcal{T} \subseteq \mathcal{S}\}$. It is obvious that *U* is included in $\mathcal{S}^+$, but it is not included in $\mathcal{S}$. This is because it is the intersection of the empty collection sets. We next set $\mathcal{P} = \{X \in \mathcal{M} : X \neq \cap\{Y \in \mathcal{M} : X \subset Y\}\}$. In [5] it is proved that $\mathcal{P}$ is the minimal generator of $\mathcal{M}$, and even it is unique. Therefore, for any generator Q of $\mathcal{M}$, we always have $\mathcal{P} \subseteq Q$. It can be seen that $\mathcal{P}$ consists of subsets which are not the intersection of two others in $\mathcal{M}$. Then, $\mathrm{Gen}(h) = \{X \in \mathrm{Closed}(h) : X \neq \cap\{Y \in \mathrm{Closed}(h) : X \subset Y\}\}$ will be the minimal generator of meet-semilattice $\mathrm{Closed}(h)$. The following proposition is clear.

Proposition 3.3. $\mathrm{Gen}(h)$ is the smallest *h*-dense family.

Example 3.1. We consider the mapping $h: 2^U \to 2^U$, with $U = \{a_1, a_2, a_3, a_4\}$, which is determined by:

X	$h(X)$	X	$h(X)$	X	$h(X)$	X	$h(X)$
$\varnothing$	$\varnothing$	$\{a_4\}$	$\{a_1, a_2, a_4\}$	$\{a_2, a_3\}$	$\{a_2, a_3\}$	$\{a_1, a_2, a_4\}$	$\{a_1, a_2, a_4\}$
$\{a_1\}$	$\{a_1\}$	$\{a_1, a_2\}$	$\{a_1, a_2, a_4\}$	$\{a_2, a_4\}$	$\{a_1, a_2, a_4\}$	$\{a_1, a_3, a_4\}$	U
$\{a_2\}$	$\{a_2\}$	$\{a_1, a_3\}$	$\{a_1, a_3\}$	$\{a_3, a_4\}$	U	$\{a_2, a_3, a_4\}$	U
$\{a_3\}$	$\{a_3\}$	$\{a_1, a_4\}$	$\{a_1, a_2, a_4\}$	$\{a_1, a_2, a_3\}$	U	U	U

Evidently, *h* is a closure operation, i.e., $h \in \mathrm{Cl}(U)$. We then easily obtain some *h*-dense families as follows:

$\mathrm{Closed}(h) = \{\varnothing, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_4\}, U\}$,

$\mathrm{Closed}^*(h) = \{\{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_4\}\}$,

$\mathrm{Gen}(h) = \{\{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_4\}\}$.

Thus, we have proved several families to be *h*-dense. However, suppose a family $\mathcal{E}$ is given. Then a more important issue is that how to determine whether the family $\mathcal{E}$ is dense in *h*. In order to deal with this problem, in what follows, we will establish a condition for determining a family to be *h*-dense.

It can be easily seen that if we define $F = \{(X, Y) : X, Y \subseteq U\}$ and $g_F(X) = \{a \in U : (X, \{a\}) \in F\}$ for all $X \subseteq U$, then $g_{F_h} = h$.

Theorem 3.1. Let $h \in \mathrm{Cl}(U)$ and $\mathcal{E} \subseteq 2^U$. Then $\mathcal{E}$ is *h*-dense if and only if for every $X \subseteq U$,

$$h(X) = \begin{cases} \bigcap\limits_{X \subseteq E} E & \text{if } \exists E \in \mathcal{E} : X \subseteq E, \\ U & \text{otherwise.} \end{cases} \tag{8}$$

Proof: It is easy to see that $F_h = F_{\mathcal{E}}$ if and only if $g_{F_h} = g_{F_{\mathcal{E}}}$. Hence, the proof of Theorem 3.1 is now transformed to proving that

$$g_{F_{\mathcal{E}}}(X) = \begin{cases} \bigcap\limits_{X \subseteq E} E & \text{if } \exists E \in \mathcal{E} : X \subseteq E, \\ U & \text{otherwise,} \end{cases}$$

for every $X \subseteq U$.

We treat a set *X* satisfying for every $E \in \mathcal{E}$, $X \not\subseteq E$. According to the definition of $F_{\mathcal{E}}$, we can easily imply that $(X, U) \in F_{\mathcal{E}}$, and thus $g_{F_{\mathcal{E}}}(X) = U$.

We always have $\varnothing \subseteq \bigcap\{E : E \in \mathcal{E}\} \subseteq E$. Besides, based on the definition of $F_{\mathcal{E}}$ we obtain

$$g_{F_{\mathcal{E}}}(\varnothing) = \bigcap\{E : E \in \mathcal{E}\}.$$

On the contrary, if *X* is non-empty and $E \in \mathcal{E}$ such that $X \subseteq E$ exists. We then set

$$\mathcal{P} = \{E : X \subseteq E, E \in \mathcal{E}\},$$

and

$$I = \bigcap \{E : E \in \mathcal{P}\}.$$

Evidently, $X \subseteq I, (X, I) \in F_{\mathcal{E}}$ and $I \subseteq g_{F_{\mathcal{E}}}(X)$.

Let us now assume that there is $a \in U$ such that $a \notin I$. It follows that $E \in \mathcal{P}$ such that $a \notin E$ exists.

Then by the definition of $g_{F_{\mathcal{E}}}$,

$$g_{F_{\mathcal{E}}}(X) = \{a \in U : (X, \{a\}) \in F_{\mathcal{E}}\}.$$

We imply $(X, I \cup a) \notin F_{\mathcal{E}}$. This means that $g_{F_{\mathcal{E}}}(X) = I$, or

$$g_{F_{\mathcal{E}}}(X) = \bigcap \{E : E \in \mathcal{P}\}.$$

■

4. Minimal keys of closure operations

We will begin this section with the definitions relative to keys and minimal keys of the closure operations.

Given $h \in \mathrm{Cl}(U)$ and a subset *K* of *U*, we will say that *K* is a *key of the operation h* if *h(K) = U*. The key *K* is *minimal* if every proper subset of *K* is not a key, i.e., $\forall k \in K$, $h(K - \{k\}) \neq U$. We shall denote $\mathrm{Key}(h)$ the set of all minimal keys of the operation *h*.

A subset $K^{-1} \subseteq U$ is called an *antikey* of *h* if $h(K^{-1}) \neq U$ and $\forall k \in U - K^{-1}$, $h(K^{-1} \cup \{k\}) = U$. We will also denote $\mathrm{Antikey}(h)$ the set of all antikeys of the operation *h*.

It can be easily seen that $\mathrm{Key}(h), \mathrm{Antikey}(h) \in \mathrm{SH}(U)$. In [16] we have proved the following relationship.

Proposition 4.1. If $h \in \mathrm{Cl}(U)$ then

$$\bigcup \mathrm{Key}(h) = U - \bigcap \mathrm{Antikey}(h).$$

In the previous section, we have introduced the dense families in the closure operations, as well as the way for determining a family to be dense. Next, we will observe the minimal keys of the closure operations from the viewpoint of the dense families. More specifically, we consider the dense family $\mathcal{E}$ in the closure operation *h*. We shall see that discovering all minimal keys of the operation *h* will be equivalent to producing the whole minimal transversals of the hypergraph $\overline{\mathcal{E}} - \{\emptyset\}$.

Theorem 4.1. Let $h \in \mathrm{Cl}(U)$ and $\mathcal{E} \subseteq 2^U$. If $\mathcal{E}$ is dense in *h*, then

$$\mathrm{Key}(h) = \mathrm{Tr}(\overline{\mathcal{E}} - \{\emptyset\}).$$

Proof: First, we consider the case when *M* is a minimal key of *h*. Then $(M, U) \in F_h$. Now given $A \in \mathcal{E} - \{U\}$, since

$$F_h = \{(X, Y) : (\forall E \in \mathcal{E}) X \subseteq E \Rightarrow Y \subseteq E\},$$

this implies that if $M \subseteq A$ then $A = U$. It follows that in case $A \neq U$ then $M \cap \overline{A} \neq \varnothing$. Therefore, $M \in \mathrm{Trs}(\overline{\mathcal{E}} - \{\varnothing\})$. Clearly, if there is a $N \subset M$ satisfying $N \in \mathrm{Trs}(\overline{\mathcal{E}} - \{\varnothing\})$, then $(N, U) \in F_{\mathcal{E}}$. Hence $(N, U) \in F_h$. This contradicts the hypothesis $M \in \mathrm{Key}(h)$. Thus, $M \in \mathrm{Tr}(\overline{\mathcal{E}} - \{\varnothing\})$ holds.

Conversely, assume that $M \in \mathrm{Tr}(\overline{\mathcal{E}} - \{\varnothing\})$. Then

$$\forall A \in \mathcal{E} - \{U\}, M \cap \overline{A} \neq \varnothing.$$

This implies that $M \not\subseteq A$. According to the definition of $F_{\mathcal{E}}$, we have $(M, U) \in F_{\mathcal{E}}$, or $(M, U) \in F_h$. Thus, M is a minimal key of h. Furthermore, if $N \in \mathrm{Key}(h)$ is a minimal key satisfying $N \subset M$, then $(N, U) \in F_h$. Since $\mathcal{E}$ is h-dense, we obtain

$$\forall A \in \mathcal{E} - \{U\}, N \cap \overline{A} \neq \varnothing.$$

This conflicts with the hypothesis that M is a minimal transversal of $\overline{\mathcal{E}} - \{\varnothing\}$. Therefore, $M \in \mathrm{Key}(h)$ holds. □

Clearly, the collection of all minimal keys of the operation h can be computed based on the dense families in h. Note that $\mathrm{Closed}(h)$, $\mathrm{Closed}^*(h)$ and $\mathrm{Gen}(h)$ have been proved to be h-dense families. As an immediate consequence of Theorem 4.1, hence, the corollary below is easily achieved.

Corollary 4.1. Let $h \in \mathrm{Cl}(U)$. Then:

1) $\mathrm{Key}(h) = \mathrm{Tr}(\min(\overline{\mathrm{Closed}(h)} - \{\varnothing\}))$,

2) $\mathrm{Key}(h) = \mathrm{Tr}(\min(\overline{\mathrm{Closed}^*(h)}))$,

3) $\mathrm{Key}(h) = \mathrm{Tr}(\overline{\mathrm{Gen}(h)})$.

Based on the obtained results, we will present an effective algorithm for extracting the minimal keys of the closure operation in terms of the dense families.

Algorithm 4.1. Algorithm for extracting the minimal keys (DENKEY)

Input: The operation $h \in \mathrm{Cl}(U)$ and the universe $U = \{a_1, a_2, \ldots, a_n\}$.

Output: $\mathrm{Key}(h)$.

Step 1. Compute the h-dense family $\mathrm{Closed}(h)$.

Step 2. Induce a directed graph $G = (V, E)$ where $V = \mathrm{Closed}(h)$ and $(X, Y) \in E$ if $X \supset Y$ and there does not exist $Z \in \mathrm{Closed}(h)$ satisfying $X \supset Z \supset Y$.

Set the h-dense family $\mathrm{Gen}(h) = \{X \in \mathrm{Closed}(h) : d(X) = 1\}$, in which $d(X)$ is the number of directed edges incident to X.

Step 3. Compute $\mathrm{Tr}(\overline{\mathrm{Gen}(h)})$.

Step 4. Return $\mathrm{Key}(h) = \mathrm{Tr}(\overline{\mathrm{Gen}(h)})$.

It is easy to see from Corollary 4.1 that the DENKEY algorithm extracts exactly $\mathrm{Key}(h)$. The computational time of DENKEY is exponential in n. However, it should be emphasized that $\mathrm{Gen}(h)$ computed in Step 2 is the smallest h-dense family.

Hence, $\text{Gen}(h)$ has usually the quite small number of elements. Moreover, we recall that Algorithm 2.1 has been pointed out in [9] to be very efficient for finding all minimal transversals. Although the computational complexity of Algorithm 2.1 is exponential, in many cases Algorithm 2.1 can execute only in the polynomial time. Therefore, based on Algorithm 2.1, our proposed DENKEY algorithm is also very efficient for finding all minimal keys of the closure operations.

Besides, an important point to remember is that in some cases when we need to find only one minimal key of the operation *h*, the computational process is very simple. Indeed, it can be easily seen that $U = \{a_1, a_2, \ldots, a_n\}$ is a key of the operation *h*. If we set $K_0 = U$, and for each $i = 1, 2, \ldots, n$,

$$K_i = \begin{cases} K_{i-1} - \{a_i\} & \text{if } h(K_{i-1} - \{a_i\}) = U, \\ K_{i-1} & \text{otherwise,} \end{cases} \tag{9}$$

then $K_n \in \text{Key}(h)$. Hence, one minimal key can be extracted in polynomial time.

Consider Example 3.1 again, we will compute the minimal keys of the operation *h* by using the DENKEY algorithm. First, we already know that

$$\text{Closed}(h) = \{\varnothing, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_4\}, U\}.$$

Then, we immediately get

$$\text{Gen}(h) = \{\{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_4\}\},$$

$$\overline{\text{Gen}(h)} = \{\{a_2, a_4\}, \{a_1, a_4\}, \{a_3\}\},$$

and

$$\text{Tr}(\overline{\text{Gen}(h)}) = \{\{a_1, a_2, a_3\}, \{a_3, a_4\}\}.$$

Thus, the set of the minimal keys of the operation *h* is

$$\text{Key}(h) = \{\{a_1, a_2, a_3\}, \{a_3, a_4\}\}.$$

As mentioned in the introduction section, the closure operations have been applied to many fields such as theories of rough sets and fuzzy sets, databases, etc. In the rest of the article, we shall provide an interesting application of our result to the attribute reduction problem in the rough set theory, which helps to enhance the efficient and effective of the classification problem by selecting important attributes and removing unnecessary attributes.

First, we recall briefly some basic concepts in the rough set theory, which can find in [10-12]. We will begin with the concept of information systems introduced by Pawlak. An *information system* is a pair of $\mathcal{I} = (U, A)$, where U and A are two finite, non-empty sets, also called the set of *objects*, and the set of *attributes*, respectively. Each attribute $a \in A$ determines a mappping $a: U \to V_a$, where each V_a is a set of values of the attribute a. We can observe an example for an information system with $U = \{u_1, u_2, \ldots, u_6\}$ and $A = \{\text{Temperature, Headache, Muscle\_pain}\}$, as follows:

U	Temperature (a_1)	Headache (a_2)	Muscle_pain (a_3)
u_1	high	no	yes
u_2	high	yes	no
u_3	very high	yes	yes
u_4	normal	no	yes
u_5	high	yes	no
u_6	very high	no	yes

Consider an information system $\mathcal{I} = (U, A)$ and an attribute subset $B \subseteq A$, and *B-indiscernibility relation on* U is defined by

$$\text{IND}(B) = \{(u, v) \in U \times U : \forall a \in B, a(u) = a(v)\}.$$

Then, if $(u, v) \in \text{IND}(B)$, then u and v are indiscernible on the subset of attributes *B*.

An attribute subset $B \subseteq A$ is called a *reduct of A* if IND(*B*) = IND(*A*) and $\forall a \in B, \text{IND}(B - \{a\}) \neq \text{IND}(B)$.

Next, we define a map $h: 2^A \to 2^A$, in which for each $X \in 2^A$, $h(X) = Y \supseteq X$ such that IND(*X*) = IND(*Y*) and for any $a \in Y$, $\text{IND}(Y - a\}) \neq \text{IND}(X)$. Obviously, h is a closure operation on A. Consider the above information system $\mathcal{I} = (U, A)$, the operation h is represented in details, as follows:

X	$h(X)$	X	$h(X)$
$\varnothing$	$\varnothing$	$\{a_1, a_2\}$	A
$\{a_1\}$	$\{a_1\}$	$\{a_1, a_3\}$	$\{a_1, a_3\}$
$\{a_2\}$	$\{a_2\}$	$\{a_2, a_3\}$	$\{a_2, a_3\}$
$\{a_3\}$	$\{a_3\}$	A	A

Then, we easily obtain

$\text{Closed}(h) = \{\varnothing, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_3\}, \{a_2, a_3\}, A\}$,

$\text{Gen}(h) = \{\{a_1\}, \{a_2\}, \{a_1, a_3\}, \{a_2, a_3\}\}$,

$\overline{\text{Gen}(h)} = \{\{a_2, a_3\}, \{a_1, a_3\}, \{a_2\}, \{a_1\}\}$,

and

$\text{Tr}(\overline{\text{Gen}(h)}) = \{\{a_1, a_2\}\}$.

Thus, the set of the minimal keys of the operation *h* is $\text{Key}(h) = \{\{a_1, a_2\}\}$. It is more interesting that $\text{Key}(h)$ is also exactly the set of all reducts of *A* in the information system $\mathcal{I} = (U, A)$.

5. Conclusion

We introduce the concept of dense family of closure operations and provide some dense families, such as the collection of all closed sets and the minimal generator of the closure operation. For an arbitrary family, we also give an effective way for determining whether it is dense or not. Based on the dense families, we present an interesting approach for discovering all minimal keys of the closure operation and propose an effective algorithm for calculating these minimal keys.

References

1. Armstrong, W. W. Dependency Structures of Data Base Relationships. – Information Processing, Vol. **74**, 1974, pp. 580-583.
2. Berge, C. Hypergraphs: Combinatorics of Finite Sets. Amsterdam, North-Holland, 1989.
3. Bertet, K., C. Demko, J. Viaud, C. Guéri. Lattices, Closures Systems and Implication Bases: A Survey of Structural Aspects and Algorithms. – In: Theoretical Computer Science. Vol. **743**. 2018, pp. 93-109.
4. Caspard, N., B. Monjardet. The Lattices of Closure Systems, Closure Operators, and Implicational Systems on a Finite Set: A Survey. – Discrete Applied Mathematics, Vol. **127**, 2003, No 2, pp. 241-269.
5. Demetrovics, J. Logical and Structural Study of a Relational Data Model (in Hungarian). – MTASZTAKI Tanulmanyok Budapest, Vol. **114**, 1980, pp. 1-97.
6. Demetrovics, J., N. X. Huy, Closed Sets and Translations of Relation Schemes, Hungarian Applied Mathematics. – Comput. Math. Appl., Vol. **21**, 1991, No 1, pp. 13-23.
7. Demetrovics, J., G. Hencsey, L. Libkin, I. Muchnik. On the Interaction between Closure Operations and Choice Functions with Applications to Relational Databases. – Acta Cybernetica, Vol. **10**, 1992, No 3, pp. 129-139.
8. Demetrovics, J., L. Libkin, I. B. Muchnik. Functional Dependencies in Relational Databases: A Lattice Point of View. – Discrete Applied Mathematics, Vol. **40**, 1992, No 2, pp. 155-185.
9. Demetrovics, J., V. D. Thi. Describing Candidate Keys by Hypergraphs. – Computers and Artificial Intelligence, Vol. **18**, 1999, No 2, pp. 191-207.
10. Demetrovics, J., V. D. Thi, N. L. Giang. On Finding All Reducts of Consistent Decision Tables. – Cybernetics and Information Technologies, Vol. **14**, 2014, No 4, pp. 3-10.
11. Demetrovics, J., V. D. Thi, N. L. Giang. 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.
12. Giang, N. L., J. Demetrovics, V. D. Thi, P. D. Khoa. Some Properties Related to Reduct of Consistent Decision Systems, – Cybernetics and Information Technologies, Vol. **21**, 2021, No 2, pp. 3-9.
13. Mao, H., S. Liu. Posets and Closure Operators Relative to Matroids. – Matematika, 2012, No 1, pp. 77-85.
14. Narayanamoorthy, S., A. Tamilselvi. Bipolar Fuzzy Line Graph of a Bipolar Fuzzy Hypergraph. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 1, pp. 13-17.
15. Nghia, V. D. Relationships between Closure Operations and Choice Functions Equivalent Descriptions of a Family of Functional Dependencies. – Acta Cybernetica, Vol. **16**, 2004, No 3, pp. 485-506.
16. Son, N. H., V. D. Thi. Some Combinatorial Characteristics of Closure Operations. – Algebra and Discrete Mathematics, Vol. **28**, 2019, No 1, pp. 144-156.
17. Son, N. H., V. D. Thi. Some Results Related to Dense Families of Database Relations. – Acta Cybernetica, Vol. **17**, 2005, No 1, pp. 173-182.
18. Thi, V. D. Minimal Keys and Antikeys. – Acta Cybernetica, Vol. **7**, 1986, No 4, pp. 361-371.

Received: 07.02.2022; Second Version: 20.05.2022; Third Version: 18.06.2022; Accepted: 28.06.2022