Mathematical Sciences Letters
An International Journal

http://dx.doi.org/10.18576/msl/060211

Trees in Class-Oriented Concept Lattices Constrained by Equivalence Relations

*Hua Mao**

Department of Mathematics, Hebei University, Baoding 071002, China

Received: 19 Apr. 2015, Revised: 28 Nov. 2016, Accepted: 29 Dec. 2016
Published online: 1 May 2017

Abstract: In this paper, we present necessary and sufficient conditions on input data for the output class-oriented concept lattice to form a tree after one removes its greatest element. In addition, we present algorithms for computing the tree of a class-oriented concept lattice constrained by equivalence relations.

Keywords: class-oriented concept lattices, tree, class-oriented concept, equivalence relation

1 Introduction

Formal concept analysis (FCA) is a method of exploratory data analysis that aims at the extraction of natural clusters from object-attribute data tables. There are many types of binary relations between objects have been studied [5] and [11]. One of the clusters, called class-oriented concepts, is naturally interpreted as human-perceived concepts in a traditional sense [5] and [7].

This paper presents conditions for input data which are necessary and sufficient for the output class-oriented concept lattice constrained by equivalence relations to form a tree after removing its greatest element. We provide two algorithms for tree in class-oriented concept lattices constrained by equivalence relations and give illustrative examples. Though some characterizing trees in concept lattices have been studied in [1], the differences between concept lattices and class-oriented concept lattices make this paper be valuable.

2 Preliminaries

We assume that a data set is given in terms of a formal context as [8]. For simplicity, we only consider finite cases. In this paper, all the notations and properties, FCA are referred to [1] and [8]; class-oriented concept lattices come from [1], [7] and [11]; lattice theory is seen [7] and [9]; graph theory is cf. [4] and [10].

An object-attribute data table describing which objects have which attributes can be identified with a triplet (X,Y,I) (called a *formal context*) where X is the set of objects, Y is the set of attributes, and $I \subseteq X \times Y$ is a relation. An object $x \in X$ has the set of attributes: $xI = \{y \in Y \mid xIy\} \subseteq Y$. Similarly, an attribute y is possessed by the set of objects: $Iy = \{x \in Y \mid xIy\} \subseteq X$. For each $A \subseteq X$ and $B \subseteq Y$ denote by

$A^* = \{y \in Y \mid \text{for each } x \in A : (x,y) \in I\}$ and $B^* = \{x \in X \mid \text{for each } y \in B : (x,y) \in I\}$.

In reality, we find $A^* = \{y \in Y \mid A \subseteq Iy\} = \bigcap_{x\in A} xI$ and $B^* = \{x \in X \mid B \subseteq xI\} = \bigcap_{y\in B} Iy$.

Two objects may be viewed as being equivalent if they have the same description. An *equivalence relation* can be defined by for $x, x' \in X, x \equiv_X x' \Leftrightarrow xI = x'I$.

With equivalence relation $\equiv_X$, an object has the same predecessor and successor neighborhood. For an object $x \in X$, the set of objects that are equivalent to x is called an *equivalence class of* x and defined by $\equiv_X x = \{x' \in X \mid x' \equiv_X x\} = \{x' \in X \mid x \equiv_X x'\} = x\equiv_X = [x]$.

The family of all equivalence classes is denoted by $X/\equiv_X = \{[x] | x \in X\}$. A new family of subsets, denoted by $\sigma(X/\equiv_X)$, can be obtained from $X/\equiv_X$ by adding the empty set $\emptyset$ and making it closed under set union, which is a subsystem of 2^X and the basis is $X/\equiv_X$. The following properties hold:

(E1) $A_1, A_2 \in \sigma(X/\equiv_X) \Rightarrow A_1 \cap A_2 \in \sigma(X/\equiv_X)$;
(E2) $A_1, A_2 \in \sigma(X/\equiv_X) \Rightarrow A_1 \cup A_2 \in \sigma(X/\equiv_X)$.

* Corresponding author e-mail: yushengmao@263.net

A pair $(A,B), A\subseteq X, B\subseteq Y$, is called a *class-oriented concept* if $A\in\sigma(X/\equiv_X)$ and $B=A^*$. The set of objects A is called the *extension* of the concept (A,B), and the set of attributes B is called the *intension*. For two class-oriented concepts (A_1,B_1) and (A_2,B_2), we say that

$(A_1,B_1)\leq(A_2,B_2)$ if and only if $A_1\subseteq A_2$ $\cdots\cdots\cdots\cdots\cdots\cdots\cdots$ (F1)

The family of all class-oriented concepts forms a complete lattice called *class-oriented concept lattice* which is denoted by $\mathscr{B}(X/\equiv_X)$ in this paper. It gives a hierarchical structure of the elements in $\sigma(X/\equiv_X)$ and their corresponding attributes. The meet $\wedge$ and the join $\vee$ are defined by

$(A_1,B_1)\wedge(A_2,B_2)=((A_1\cap A_2),(A_1\cap A_2)^*), (A_1,B_1)\vee(A_2,B_2)=((A_1\cup A_2),(B_1\cap B_2))$ $\cdots\cdots\cdots\cdots$ (F2).

3 Trees in class-oriented concept lattices

This paper will adopt the definition of trees similarly to that in [1] as follows.

A finite partially ordered set $(\sigma(X/\equiv_X),\leq_T)$ will be called a *tree* if for each $a,b\in X$:
(Ti) there is an infimum of a and b in $(\sigma(X/\equiv_X),\leq_T)$, and
(Tii) there is a supremum of a and b in $(\sigma(X/\equiv_X),\leq_T)\Leftrightarrow a$ and b are comparable.

Evidently, the notation tree above is actually a real rooted tree as that defined in graph theory (see [4]), but not vice versa. However, the authors in [1] use the similar tree to the above dealing with many properties for concept lattices. Analogously to [1], this section will work on the family of class-oriented concept lattices relative to trees.

According to lattice theory, we find that the condition (Ti) is satisfied for each $\mathscr{B}(X/\equiv_X)$; the condition (Tii) is satisfied if and only if $\mathscr{B}(X/\equiv_X)$ is linearly ordered.

Based on these, we can express that

the whole class-oriented concept lattice is a tree if and only if it is linearly ordered.

In fact, this case is not valuable to observe.

We mainly focus on trees which appear in $\mathscr{B}(X/\equiv_X)$ if we remove its greatest element (X,Y_X), where $Y_X=\bigcap_{x\in X}xI$. Let $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ denote $\mathscr{B}(X/\equiv_X)\setminus\{(X,Y_X)\}$. In what follows, we investigate the conditions which $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ becomes a tree.

3.1 Formal contexts generating trees

The following assertion characterizes $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ to be a tree.

Theorem 3.1. Let (X,Y,I) be a formal context. Then $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is a tree if and only if, for any concepts $(A,B),(C,D)\in\mathscr{B}^{\curlywedge}(X/\equiv_X)$ at least one of the following is true:

(1) $A\subseteq C$ or $C\subseteq A$,
(2) $A\cup C=X$.

Proof. Similarly to the proof for Theorem 2 in [1], the need consequences are obtained. □.

We will take advantage of the following notion.

Definition 3.1. Let (X,Y,I) be a formal context. We say that (X,Y,I) generates a tree if $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is a tree.

Recall that due to (F1) and the definition of class-oriented concepts, $([x],xI)$ is a class-oriented concept for any $x\in X$ and covers the least element $(\emptyset,Y)$ in $\mathscr{B}(X/\equiv_X)$. In addition, for any $(A,B)\in\mathscr{B}(X/\equiv_X)$, we receive $A=\bigcup_{x\in A}[x]$ and $B=\bigcap_{x\in A}xI$ according to definitions and the finite of $|X|$ and $|Y|$. The extensions of all the atoms $\{([x],xI): x\in X\}$ are important in the following theorem which characterizes contexts generating trees.

Theorem 3.2. Let (X,Y,I) be a formal context. Then (X,Y,I) generates a tree if and only if, for any $(A,B)\in\mathscr{B}^{\curlywedge}(X/\equiv_X)$, there exists $a\in A$ such that there is one and only one chain between $\emptyset$ and A in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ satisfying $\emptyset\subset[a]\subset A_1\subset\ldots\subset A_n=A$.

Proof. $(\Rightarrow)$ It is evident by Definition 3.1.

$(\Leftarrow)$ If (A,B) covers the least element in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$, i.e $A=[x]$ for $x\in A$, then the needed result is accepted.

Next, suppose $A\neq[x]$ for any $x\in A$. We prove the needed result.

Otherwise, in light of definition of a tree, one of the following cases will be happen.

Case 1. There are two incomparable elements $(A,B),(C,D)\in\mathscr{B}^{\curlywedge}(X/\equiv_X)$ such that

$(A,B)\vee(C,D)\in\mathscr{B}^{\curlywedge}(X/\equiv_X)$.

Case 2. $(A,B)\wedge(C,D)\notin\mathscr{B}^{\curlywedge}(X/\equiv_X)$ holds for some two elements $(A,B),(C,D)\in\mathscr{B}^{\curlywedge}(X/\equiv_X)$.

We analyze the two cases respectively.

Under the supposition of Case 1:

Both $(A,B)<(A,B)\vee(C,D)$ and

$(C,D)<(A,B)\vee(C,D)$

hold owing to the incomparable property of (A,B) and (C,D).

Since $\mathscr{B}(X/\equiv_X)$ is a complete lattice with the set of atoms $\{([x],xI)\mid x\in X\}$, we obtain $U=\bigcup_{x\in U}[x]$ for each element $(U,V)\in\mathscr{B}(X/\equiv_X)$. The given condition implies that there exists at least a chain between $\emptyset$ and A, between $\emptyset$ and C satisfying $\emptyset\subset[x_1]\subset a_1\subset\ldots\subset a_n=A$ and $\emptyset\subset[x_2]\subset c_1\subset\ldots\subset c_m=C$, where $x_1\in A$ and $x_2\in C$. Thus, for $(A,B)\vee(C,D)=(A\cup C,B\cap D)$, there are two different chains $\emptyset\subset[x_1]\subset a_1\subset\ldots\subset a_n=A\subset A_1\subset\ldots\subset A_t=A\cup C$ and $\emptyset\subset[x_2]\subset c_1\subset\ldots\subset c_m=C\subset C_1\subset\ldots\subset C_s=A\cup C$, a contradiction to the known condition.

Under the supposition of Case 2:

If (A,B) and (C,D) are comparable. We obtain the existence of infimum of (A,B) and (C,D) in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$. This follows a contradiction to the supposition. Hence, (A,B) and (C,D) are incomparable.

Because $\mathscr{B}(X/\equiv_X)$ is a complete lattice, we assert that the infimum $\alpha\wedge\beta\in\mathscr{B}(X/\equiv_X)$ holds for each $\alpha,\beta\in$

$\mathscr{B}(X/\equiv_X)$. In addition, it is easily seen $\alpha \wedge \beta < \alpha, \beta$ if α and β are incomparable.

Therefore, combining

$\mathscr{B}^{\curlywedge}(X/\equiv_X) = \mathscr{B}(X/\equiv_X) \setminus \{(X, Y_X)\}$

with the two incomparable $(A,B),(C,D) \in \mathscr{B}^{\curlywedge}(X/\equiv_X)$, we receive $(A,B) \wedge (C,D) \in \mathscr{B}^{\curlywedge}(X/\equiv_X)$, a contradiction to the supposition.

Summing up the above analysis, we may be assured that $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is a tree. $\square$.

3.2 Algorithms for trees in class-oriented concept lattices

Trees in class-oriented concept lattices, as they were introduced in the previous sections, can be computed by algorithms for computing class-oriented concepts.

We see that a concept is uniquely determined by its extension and all the extensions of elements in $\mathscr{B}(X/\equiv_X)$ with hierarchy order as $\mathscr{B}(X/\equiv_X)$ forms a complete lattice which is isomorphic to $\mathscr{B}(X/\equiv_X)$. This implies that we should focus on the construction of extension of a concept in $\mathscr{B}(X/\equiv_X)$. Here, by Theorem 3.1 and Theorem 3.2, we present algorithms to generate a tree in class-oriented concept lattice. Meanwhile, we search out all the class-oriented concepts.

The core of an algorithm for computing the tree is shown in the following.

Step 1.1 We easily obtain $[x_j]$ from the binary table, $(j = 1,2,\ldots,|X|)$. Let $\mathscr{L}_1 = \{[x_i] : i = 1,\ldots,k\}$, where $[x_i] \cap [x_j] = \emptyset, (i \neq j; i, j = 1,\ldots,k)$ and $\bigcup_{i=1}^{k}[x_i] = X$.

Step 1.2. Evidently, it has $[x_{i_1}] \cup [x_{j_1}] \neq [x_{i_2}] \cup [x_{j_2}]$ if $\{i_1, j_1\} \neq \{i_2, j_2\}, (i_1, j_1, i_2, j_2 = 1,\ldots,k)$, and besides, $[x_i] \subset [x_i] \cup [x_j], (i \neq j; i, j = 1,\ldots,k)$. Let $\mathscr{L}_2 = \{[x_i] \cup [x_j] : i \neq j \text{ and } i, j = 1,\ldots,k\}$.

$\ldots\ldots$

Step 1.t. Let $\mathscr{L}_t = \{[x_{i_1}] \cup [x_{i_2}] \cup \ldots \cup [x_{i_t}] : i_p \neq i_q; p \neq q; i_1, i_2, \ldots, i_t = 1,\ldots,k\}$. $(t = 3,\ldots,k)$.

Therefore, we obtain $\{\mathscr{L}_1, \mathscr{L}_2, \ldots, \mathscr{L}_k\}$ of the extensions of elements in $\mathscr{B}(X/\equiv_X) \setminus \{(\emptyset, Y), (X, Y_X)\}$ and the cover relations among them defined by (F1). From the binary table, we easily obtain the constitutions of x_iI, $(i = 1,\ldots,|X|)$ and receive the set of atoms in $\mathscr{B}(X/\equiv_X)$ as $\{([x_i], x_iI) : i = 1,\ldots,k)\}$. Moreover, it follows $\mathscr{H}_t = \{([x_{i_1}] \cup \ldots \cup [x_{i_t}], \bigcap_{j=1}^{t} x_{i_j}I) \mid [x_{i_1}] \cup \ldots \cup [x_{i_t}] \in \mathscr{L}_t\}$, $(t = 1,\ldots,k-1,k)$, and further, $\{(\emptyset, Y)\} \cup \mathscr{H}_1 \cup \ldots \cup \mathscr{H}_{k-1}$ constitutes $\mathscr{B}^{\curlywedge}(X/\equiv_X)$. The algorithm of Breath First Searching (BFS) for a spanning tree is used here. Actually, a spanning tree of the diagram of $\mathscr{B}(X/\equiv_X) \setminus \{(X, Y_X)\}$ is just $\mathscr{B}^{\curlywedge}(X/\equiv_X)$. This approach is a little like to the BFS algorithm searching out the tree of a graph. So, we call it BFS algorithm.

Next we provide a method to search out all the extension of $\mathscr{B}^{\curlywedge}(X/\equiv_X)$.

$[x_\alpha] \subset [x_\alpha] \cup [x_i] \subset [x_\alpha] \cup [x_{i_1}] \cup [x_{i_2}] \subset \ldots \subset [x_\alpha] \cup [x_{i_1}] \cup [x_{i_2}] \cup \ldots \cup [x_{i_t}]$ is a maximal chain from $[x_\alpha]$ to $[x_\alpha] \cup [x_{i_1}] \cup \ldots \cup [x_{i_t}]$, where $\alpha = 1,\ldots,k; \alpha \notin \{i_1,\ldots,i_t\}; i_p \neq i_q, p, q = 1,\ldots,t; i_1, i_2, \ldots, i_t = 1,2,\ldots,k$ and $|\{1,\ldots,t\} \cup \{\alpha\}| < k$. Hence, we obtain many chains. For every chain, we give it a positive integer number and all these numbers are not repeated. In other words, all the chains is sequenced in number natural order. Theorem 3.2 states that every elements in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is in a unique maximal chain.

Since $\mathscr{B}^{\curlywedge}(X/\equiv_X) = \mathscr{B}(X/\equiv_X) \setminus \{(X, Y_X)\}$ follows that for $(A,B) \in \mathscr{B}(X/\equiv_X)$, if (A,B) not as the greatest element (X, Y_X), then $(A,B) \in \mathscr{B}^{\curlywedge}(X/\equiv_X)$. Thus, every element in class-oriented concept lattice constrained by an equivalence relation can be found if we search out $\mathscr{B}^{\curlywedge}(X/\equiv_X)$.

From these ideas, we have a method as follows.

Step 2.1. It is easy to obtain $[x_j]$ from the binary table, $(j = 1,2,\ldots,|X|)$. Let $\mathscr{L}_1 = \{[x_i] : i = 1,\ldots,k\}$, where $[x_i] \cap [x_j] = \emptyset, (i \neq j, i, j = 1,\ldots,k)$ and $\bigcup_{i=1}^{k}[x_i] = X$.

Step 2.2. Let $\alpha = 1$. We obtain all the chains $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_{\gamma_1}$ between $[x_\alpha]$ and the extension of the maximal elements in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ which contain $[x_\alpha]$ respectively. Let $\mathscr{C}_1 = \mathbb{C}_1, \mathscr{C}_t = (\mathbb{C}_t \setminus \bigcup_{j=1}^{t-1}\mathbb{C}_j) \cup \max(\bigcap_{j=1}^{t}\mathbb{C}_j), (t = 2,\ldots,\gamma_1)$. Let $\mathscr{A}_1 = \{\mathscr{C}_j | (\mathbb{C}_j \setminus \bigcup_{i=1}^{j-1}\mathbb{C}_i) \neq \emptyset, j = 2,\ldots,\gamma_1\} \cup \{\mathscr{C}_1\}$ and $\mathfrak{S}_1$ stands for the family of all the constituent elements in all $C \in \mathscr{A}_1$.

Suppose we have obtain all the $\mathscr{A}_\alpha$ and $\mathfrak{S}_\alpha$ in $\alpha \leq m < k$.

Now let $\alpha = m+1$. In $\mathscr{B}^{\curlywedge}(X/\equiv_X)$, we gain all the chains between $[x_\alpha]$ and the maximal elements which contain $[x_\alpha]$ but not contain $[x_1],\ldots,[x_{\alpha-2}]$ and $[x_{\alpha-1}]$, i.e. we need only focus on the chains provided by $\{[x_j] : j = \alpha, \alpha+1,\ldots,k\}$. Let $\mathbb{C}_{\alpha_1},\ldots,\mathbb{C}_{\alpha_\delta}$ be all these different chains. Let $\mathscr{C}_{\alpha_1} = \mathbb{C}_{\alpha_1}, \mathscr{C}_{\alpha_t} = (\mathbb{C}_{\alpha_t} \setminus \bigcup_{j=1}^{t-1}\mathbb{C}_{\alpha_j}) \cup \max(\bigcap_{j=1}^{t}\mathbb{C}_{\alpha_j}), (t = 2,\ldots,\delta)\}$. Let $\mathscr{A}_\alpha = \{\mathscr{C}_{\alpha_t} | (\mathbb{C}_{\alpha_t} \setminus \bigcup_{j=1}^{t-1}\mathbb{C}_{\alpha_j}) \neq \emptyset, j = 2,\ldots,\delta\} \cup \{\mathscr{C}_{\alpha_1}\}$ and $\mathfrak{S}_\alpha$ stands for the family of all the constituent elements in all $C \in \mathscr{A}_\alpha$.

Step 2.3. The set of extensions of members in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is $\{\mathfrak{S}_\alpha : \alpha = 1,\ldots,k\} \cup \{\emptyset\}$ and the cover relationships defined by (F1) are shown by every $\mathscr{A}_\alpha$, $(\alpha = 1,\ldots,k)$, and besides, we receive $\emptyset \subset S$ for each $S \in \mathfrak{S}_\alpha, (\alpha = 1,\ldots,k)$.

Therefore, we obtain $\mathscr{B}^{\curlywedge}(X/\equiv_X)$. Evidently, $\mathscr{A}_\alpha \neq \emptyset$ is real owing to $[x_\alpha] \in \mathscr{C}_{\alpha_1}$ for each $\alpha, (\alpha = 1,\ldots,k)$. We call this method DFS (depth first searching) algorithm.

We present an illustrative example to show DFS algorithm above to obtain $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ for a given formal context.

Example 3.1. A context (X,Y,I) in [5] is Table 1. The correspondent class-oriented concept lattice shown [6].

We get the corresponding $\mathscr{B}^{\curlywedge}(X,Y,I)$, which is depicted in Figure 1, is a tree.

Table 1 A formal context

	po	lm	sm	vm	vc	at	cs	dm	ic	ad
x_1 : picture completion	×	×		×						×
x_2 : picture arrangement	×					×	×	×	×	×
x_3 : block design	×				×	×		×		×
x_4 : object assembly	×				×	×		×		×
x_5 : digit symbol			×	×	×					

po: perceptual organization, lm: ong-term memory, sm: short-term memory, vm: visual memory, vc: visual-motor coordination, at: abstract thought, cs: common sense, dm: decision making,
ic: intellectual curiosity, ad: attention to detail.

The process of getting the extensions of all the members in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ as follows.

We can indicate from Table 1 that in $\mathscr{B}(X/\equiv_X)$, the least element is $(\emptyset,Y)$ and the greatest element is $(X,\emptyset)$; in addition,
$[x_1]=\{x_1\},[x_2]=\{x_2\},[x_3]=\{x_3,x_4\},[x_5]=\{x_5\}$; $x_1I=\{po,lm,vm,ad\},x_2I=\{po,at,cs,dm,ic,ad\},x_3I=x_4I=\{po,vc,at,dm,ad\},x_5I=\{sm,vm,vc\}$.

Step 1. All the extension of the atoms are

$$\mathscr{L}_1=\{[x_1],[x_2],[x_3],[x_5]\}.$$

Step 2. First, we compute all the maximal chains generated with $\{[x_1]\}$ as follows.

$\mathbb{C}_1 : [x_1]\subset[x_1]\cup[x_2]\subset[x_1]\cup[x_2]\cup[x_3]$; $\mathbb{C}_2 : [x_1]\subset[x_1]\cup[x_2]\subset[x_1]\cup[x_2]\cup[x_5]$;

$\mathbb{C}_3 : [x_1]\subset[x_1]\cup[x_3]\subset[x_1]\cup[x_3]\cup[x_2]$; $\mathbb{C}_4 : [x_1]\subset[x_1]\cup[x_3]\subset[x_1]\cup[x_3]\cup[x_5]$;

$\mathbb{C}_5 : [x_1]\subset[x_1]\cup[x_5]\subset[x_1]\cup[x_5]\cup[x_2]$; $\mathbb{C}_6 : [x_1]\subset[x_1]\cup[x_5]\subset[x_1]\cup[x_5]\cup[x_3]$.

Second, we receive all the chains $\mathscr{C}_j$ $(j=1,\dots,6)$ produced by $\{\mathbb{C}_j : j=1,\dots,6\}$.

$\mathscr{C}_1=\mathbb{C}_1$; $\mathscr{C}_2=(\mathbb{C}_2\setminus\mathbb{C}_1)\cup\max(\mathbb{C}_1\cap\mathbb{C}_2) : [x_1]\cup[x_2]\subset[x_1]\cup[x_2]\cup[x_5]$;

$\mathscr{C}_3=(\mathbb{C}_3\setminus(\mathbb{C}_1\cup\mathbb{C}_2))\cup\max(\mathbb{C}_1\cap\mathbb{C}_2\cap\mathbb{C}_3) : [x_1]\subset[x_1]\cup[x_3]$;

$\mathscr{C}_4=\mathbb{C}_4\setminus(\mathbb{C}_1\cup\mathbb{C}_2\cup\mathbb{C}_3)\cup\max(\bigcap_{j=1}^{4}\mathbb{C}_j) : [x_1]\subset[x_1]\cup[x_3]\cup[x_5]$;

$\mathscr{C}_5=(\mathbb{C}_5\setminus(\mathbb{C}_1\cup\mathbb{C}_2\cup\mathbb{C}_3\cup\mathbb{C}_4))\cup\max\bigcap_{j=1}^{4}\mathbb{C}_j : [x_1]\subset[x_1]\cup[x_5]$;

$\mathscr{C}_6 : \mathbb{C}_6\setminus((\mathbb{C}_1\cup\mathbb{C}_2\cup\mathbb{C}_3\cup\mathbb{C}_4\cup\mathbb{C}_5)=\emptyset$.

Thus, we obtain $\mathscr{A}_1=\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4,\mathscr{C}_5\}$ and $\mathfrak{S}_1=\{[x_1],[x_1]\cup[x_2],[x_1]\cup[x_3],[x_1]\cup[x_5],[x_1]\cup[x_2]\cup[x_3],[x_1]\cup[x_2]\cup[x_5],,[x_1]\cup[x_3]\cup[x_5]\}$.

Third, we compute all the maximal chains generated with $\{[x_2]\}$ and not containing $[x_1]$ as follows. Certainly, $[x_2]$ belongs to $\mathscr{L}_1\setminus\{[x_1]\}$.

$\mathbb{C}_7 : [x_2]\subset[x_2]\cup[x_3]\subset[x_2]\cup[x_3]\cup[x_5]$; $\mathbb{C}_8 : [x_2]\subset[x_2]\cup[x_5]\subset[x_2]\cup[x_5]\cup[x_3]$;

Fourth, we get all the chains $\{\mathscr{C}_j : j=7,8\}$ generated with $\{\mathbb{C}_7,\mathbb{C}_8\}$.

$\mathscr{C}_7=\mathbb{C}_7$; $\mathscr{C}_8=(\mathbb{C}_8\setminus\mathbb{C}_7)\cup\max(\mathbb{C}_7\cap\mathbb{C}_8)=[x_2]\subset[x_2]\cup[x_5]$.

At the same time, we obtain $\mathscr{A}_2=\{\mathscr{C}_7,\mathscr{C}_8\}$ and $\mathfrak{S}_2=\{[x_2],[x_2]\cup[x_3],[x_2]\cup[x_5],[x_2]\cup[x_3]\cup[x_5]\}$.

Fifth, we compute all the maximal chains generated with $\{[x_3]\}$ and not containing any elements in $\{[x_1],[x_2]\}$ as follows. Certainly, $[x_3]$ belongs to $\mathscr{L}_1\setminus\{[x_1],[x_2]\}$. $\mathbb{C}_9 : [x_3]\subset[x_3]\cup[x_5]$.

Sixth, we get the chain $\mathscr{C}_9$ generated with $\{\mathbb{C}_9\}$. $\mathscr{C}_9 : [x_3]\subset[x_3]\cup[x_5]$.

Meanwhile, we obtain $\mathscr{A}_3=\{\mathscr{C}_9\}$ and $\mathfrak{S}_3=\{[x_3],[x_3]\cup[x_5]\}$.

Seventh, we compute all the maximal chains produced by $\{[x_5]\}$ and not containing any elements in $\{[x_1],[x_2],[x_3]\}$ as follows. Certainly, $[x_5]$ belongs to $\mathscr{L}_1\setminus\{[x_1],[x_2],[x_3]\}$. $\mathbb{C}_{10} : [x_5]$.

Eighth, we get $\mathscr{C}_{10}$ produced by $\{\mathbb{C}_{10}\}$. $\mathscr{C}_{10} : [x_5]$. Simultaneously, $\mathscr{A}_4=\{\mathscr{C}_{10}\}$ and $\mathfrak{S}_4=\{[x_5]\}$.

Step 3. The set of extensions of elements in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is $\{\mathfrak{S}_j : j=1,...,4\}\cup\{\emptyset\}$. The hierarchy relation among them are shown in $\{\mathscr{A}_j : j=1,\dots,4\}$. $\emptyset\subset S$ holds for each $S\in\mathfrak{S}_j,(j=1,...,4)$.

The diagram of all the extensions of elements in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is Figure 1.

In fact, the diagram of $\mathscr{B}^{\curlywedge}(X/\equiv_X)$ is Figure 1 if a node stands for the extension of a corresponding element in $\mathscr{B}^{\curlywedge}(X/\equiv_X)$, that is, $\emptyset : \{(\emptyset,Y)\}$; $[x_j] : \{([x_j],x_jI)\}$; $[x_i]\cup[x_j] : \{([x_i]\cup[x_j],x_iI\cap x_jI)\}$; $[x_i]\cup[x_j]\cup[x_t] : \{([x_i]\cup[x_j]\cup[x_t],x_iI\cap x_jI\cap x_tI)\}$, where $p\neq q,p,q\in\{i,j,t\};i,j,t=1,2,3,5$.

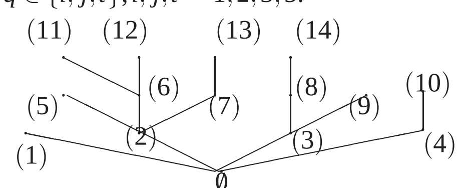

Figure 1 Formal context generating a tree

where (1)=$[x_5]$; (2)=$[x_1]$; (3)=$[x_2]$; (4)=$[x_3]$; (5)=$[x_1]\cup[x_5]$; (6)=$[x_1]\cup[x_2]$; (7)=$[x_1]\cup[x_3]$; (8)=$[x_2]\cup[x_3]$; (9)=$[x_2]\cup[x_5]$; (10)=$[x_3]\cup[x_5]$; (11)=$[x_1]\cup[x_2]\cup[x_5]$; (12)=$[x_1]\cup[x_2]\cup[x_3]$; (13)=$[x_1]\cup[x_3]\cup[x_5]$; (14)=$[x_2]\cup[x_3]\cup[x_5]$.

4 Conclusion

Using the idea of tree which is similar to that in graph theory, we find out the relationships between formal contexts and trees in class-oriented concept lattices. Simultaneously, we present an algorithm to compute the trees in a class-oriented concept lattice. Through an example, the algorithm is illustrated. In the future, we hope to find much more results with the assistance of graph theory to apply in the research of class-oriented concept lattices, and also of FCA.

Acknowledgement

The author acknowledges the financial support by the National Nature Science Foundation of China (61572011) and Nature Science Foundation of Hebei Province (A2017201007).

References

[1] R. Bělohlávek, B.D. Baets, J. Qutrata and V. Vychodil, International J. of Uncertainty, Fuzziness and Knowledge-Based Systems, **16**, 1-15 (2008).
[2] R. Bělohlávek, V. Sklenář and J. Zacpal, Concept lattices constrained by equivalence relations, In: Proceedings of the CLA 2004 International Workshop on Concept Lattices and Their Applications, Ostrava, 2004, 58-66.
[3] R. Bělohlávek, V. Sklenář and J. Zacpal, Concept lattices constrained by systems of partitions, In: Proceeding Znalosti 2005, 4th Annual Congerence, Stara Lesna, 5-8 (2005).
[4] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications,Elsevier Science Publishing, New York, 1976.
[5] Y. Chen and Y. Yao, Information Science, **178**, 1-20 (2008).
[6] B.A. Davey and H.A. Priestley, Introduction to Lattices and Order, 2nd. edn., Cambridge University Press, Cambridge, 2002.
[7] I. Düntsch and G. Gediga, Modal-style operators in qualitative data analysis, In: Proceedings of the 2002 IEEE International Conference on Data Mining, 155-162, (2002).
[8] B. Ganter and R. Wille, Formal Concept Analysis: Mathematical Foundations, Berlin: Springer-Verlag, 1999.
[9] G. Grätzer, General Lattice Theory, 2nd. edn. Birkhäuser Verlag, Basel, 1998.
[10] B. Korte and J. Vygen, Combinatorial Oprimization Theory and Algorithms, Springer Verlag Berlin Heidelberg, New York, 2000.
[11] Y.Y. Yao, Lecture Notes in Computer Science **2639**, 44-51 (2003).

Hua Mao is a mathematics professor at Hebei University,China. She received the PhD degree in Applied Mathematics at Xidian University. Her research interests are in the areas of pure and applied mathematics including matroid theory, lattice theory, concept lattice theory and information science. She has published research articles in many international journals of mathematical and information sciences.