TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 8, August 2014, pp. 6332 ~ 6337
DOI: 10.11591/telkomnika.v12i8.5854

A Complete Lattice Lossless Compression Storage Model

Zhi Huilai
School of Computer Science and Technology, Henan Polytechnic University,
Jiaozuo Henan P.R.China, +86-0391-3987711
email: zhihuilai@126.com

Abstract

In this paper, a complete lattice lossless compression storage model is proposed to improve the storage efficiency. In order to build the proposed model, first all the upper and lower irreducible elements of the complete lattice are identified respectively, then an isomorphic mapping form the complete lattice to a concept lattice is founded, and finally a matrix is used to store the formal context of the concept lattice. Compared with using adjacent matrix, example and analysis show that the proposed method can improve the storage efficiency of complete lattice.

Keywords: *Lattice theory, complete Lattice, irreducible element, lossless compression storage*

1. Introduction

Lattice describes the partial order relations between objects, and is widely used in object clustering and hierarchical structure analysis. In recent years, lattice theory, especially the complete lattice theory, is used in many fields, such as graph querying [1], situation hierarchy manipulate [2], metabolic pathway analysis [3], set-valued variable representation [4], and so on. In theoretical research, adjacency matrix as the storage model is acceptable. However, in real applications the lattice usually contains numerous nodes and has a complicated structure. At this circumstance, adjacency matrix as the storage model will cost a lot of storage space and is not conducive to lattice retrieval and lattice isomorphism judgment. Lattice storage is no longer a insignificant problem, but a key theoretical issues of practical application value.

Formal Concept Analysis (FCA) after being produced by professor R. Wille [5], its core structure concept lattice has attracted broad attention and being used in various fields. As its unique advantages in data analysis and knowledge system development, it has become a means for external recognition [6]. For this reason, in this article we will propose a complete lattice storage model based on the theory of FCA.

2. Preliminaries

Before proceeding, we briefly recall the lattice terminology and properties[7], as well as fundamental definitions in FCA [8, 9].

Definition 2.1. Let P be a set. An order on P is a binary relation $\leq$ such that, for all $x, y, z \in P$,

(1) $x \leq x$,

(2) $x \leq y$ and $y \leq x$ imply $x = y$,

(3) $x \leq y$ and $y \leq z$ imply $x \leq z$.

These conditions are referred to, respectively, as reflexivity, anti-symmetry and transitivity.

Definition 2.2. Let P be an ordered set and let $x, y \in P$. We say x is covered by y (or y covers x), and write $x \prec y$ or $y \succ x$, if $x < y$ and $x \leq z < y$ implies $z = x$. Moreover, x is called the lower neighbor of y and y is called the upper neighbor of x.

Received February 24, 2014; Revised March 26, 2014; Accepted April 15, 2014

Definition 2.3. Let P be an ordered set and let $S \subseteq P$. An element $x \in P$ is an upper bound of S if $s \leq x$ for all $s \in S$. An lower bound is dually. The set of all upper bounds of S is denoted by S^u and the set of all lower bounds by S^l:

$S^u := \{x \in P \mid (\forall s \in S) s \leq x\}$ and $S^l := \{x \in P \mid (\forall s \in S) s \geq x\}$.

If S^u has a least element x, then x is called the least upper bound of S. Dually, if S^l has a greatest element x, then x is called the greatest lower bound of S.

The least upper bound of S is also called the supermum of S and is denoted by sup S; the greatest lower bound of S is also called the infimum of S and is denoted by inf S.

Notation 2.1. We write $x \vee y$ in place of sup$\{x, y\}$ when it exists and $x \wedge y$ in place of inf$\{x, y\}$ when it exists. Similarly, we write $\vee S$ and $\wedge S$ instead of sup S and inf S. It is sometimes necessary to indicate that the join or meet is being found in a particular ordered set P, in which case we write $\vee_P S$ or $\wedge_P S$.

Definition 2.4. Let P be an non-empty ordered set.

(1) If $x \vee y$ and $x \wedge y$ exists for all $x, y \in P$, then P is called a lattice;

(2) If $\vee S$ and $\wedge S$ exists for all $S \subseteq P$, then P is called a complete lattice.

Definition 2.5. Let L be a lattice. An elements $x \in L$ is join-irreducible if

(1) $x \neq 0$ (in case L has a zero)

(2) $x = a \vee b$ implies $x = a$ or $x = b$ for all $a, b \in L$.

A meet-irreducible element is defined dually.

Definition 2.6. Let P be an ordered set and $Q \subseteq P$. Then Q is called join-dense in P if for every element $a \in P$ there is a subset A of Q such that $a = \vee_P A$. The dual of join-dense is meet-dense.

Definition 2.7. A context is a triple $K = (G, M, I)$ where G and M are sets and $I \subseteq G \times M$. The elements of G and M are called objects and attributes respectively. As usual, instead of $(g, m) \in I$ we write gIm and say 'the object g has the attribute m'.

Definition 2.8. Given a formal context $K = (G, M, I)$, the derivation functions $f(.)$ and $g(.)$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$f(A) = \{m \in M \mid \forall g \in A : gIm\}; g(B) = \{g \in G \mid \forall m \in B : gIm\}.$$

Definition 2.9. A formal concept of formal context $K = (G, M, I)$ is a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $f(A) = B$, and $g(B) = A$.

A concept (A, B) is subconcept of (C, D) if $A \subseteq C$ (equivalently, $D \subseteq B$). In this case, (C, D) is called a superconcept of (A, B). We write $(A, B) \leq (C, D)$ and define the relations $\geq$, $<$, and $>$ as usual.

The set of all concepts ordered by the relation $\leq$ forms a lattice, which is denoted by $L(K)$ and called the concept lattice of the context K. The relation defines the covering graph of $L(K)$.

3. Complete Lattice Storage Model Based on FCA

Definition 3.1. Let $K=(G,M,I)$ be a formal context, object g is called a full attributes object, if and only if $f(g)=M$. Dually, attribute m is called a largest common attribute if and only if $g(m)=G$ [10].

Definition 3.2. Let $K=(G,M,I)$ be a formal context, object g is called a shaded object, if and only if there are a series of objects $\{g_i\}_{i\in T}$ and T is index set, that makes $\bigcap_{i\in T} f(g_i)=f(g)$. Dually, attribute m is called a shaded attribute, if and only if there are a series of attributes $\{m_i\}_{i\in T}$ and T is index set, that makes $\bigcap_{i\in T} g(m_i)=g(m)$.

Definition 3.3. Let $K=(G,M,I)$ be a formal context, K is called a purified formal context, if and only if there is no full attributes object, no largest common attribute, no shaded object and no shaded attribute.

Definition 3.4. Let $K=(G,M,I)$ be a formal context, a concept is called a object concept if it has the form $\left(g\left(f\left(g\right)\right), f\left(g\right)\right)$, $g\in G$, and g is called its object label; a concept is called a property concept if it has that has the form $\left(g\left(m\right), f\left(g\left(m\right)\right)\right), m\in M$, and m is called its attribute label.

Proposition 3.1. In a complete lattice, a join-irreducible element has only one lower neighbor, and a meet-irreducible element has only one upper neighbor [7].

Theorem 3.1. Let $K=(G,M,I)$ be a purified formal context, a object concept of K must be a join-irreducible element, and vice versa. Dually, an attribute concept of K must be a meet-irreducible element, and vice versa.

Proof: Proof by contraction and assume that a object concept (A,B) is not a join-irreducible element, then (A,B) has at least two lower neighbors, and we denote all these lower neighbors as $(A_t,B_t)_{t\in T}$, and T is the index set. Since (A,B) is a object concept, then it must exist an object g that makes $f(g)=B$. By basic theorem of concept lattice, we have $B=\bigcap_{t\in T} B_t=\bigcap_{t\in T} f(A_t)$, and also $f(A)=B$, then we get $f(A)=\bigcap_{t\in T} f(A_t)$, which means (A,B) is a shaded object, and this is contradict to the condition of the theorem that K is not a purified context. So the assumption fails, and the theorem holds. By Duality Principle, we directly get that an attribute concept of K must be a meet-irreducible element, and vice versa.

Theorem 3.2. [7] Let V be a complete lattice, let G and M be sets and assume that there exist mappings $\gamma: G\rightarrow V$ and $\mu: M\rightarrow V$ such that $\gamma(G)$ is join-dense in V and $\mu(M)$ is meet-dense in V. Define I by $gIm \Leftrightarrow \gamma(G)\le \mu(M)$, for all $g\in G$ $m\in M$. Then V is isomorphic to concept lattice $L(G,M,I)$.

We define $\gamma(g):=\left(g\left(f\left(g\right)\right), f\left(g\right)\right)$ and $\mu(m):=\left(g\left(m\right), f\left(g\left(m\right)\right)\right)$, then we have $gIm \Leftrightarrow g\in g(m) \Leftrightarrow g\left(f\left(g\right)\right)\subseteq g(f(g(m)))=g(m)\Leftrightarrow \gamma(g)\le\mu(m)$ and thus we have $gIm\Leftrightarrow \gamma(G)\le\mu(M)$, which means $\gamma(g)$ and $\mu(m)$ satisfy Theorem 3.2. Moreover, by Definition 3.4 we know that $\gamma(g)$ is a object concept, and $\mu(m)$ is an attribute concept.

According to the above discussion, given a complete lattice V, by using mapping $\gamma(g)$ and $\mu(m)$, we can get a concept lattice $L(G,M,I)$ which is isomorphic to the complete lattice V.

In the following Algorithm 1, by labeling irreducible element on V, we can get the objects and attributes contained in $L(G,M,I)$, which is corresponding the elements contained in G and M. Moreover, by using the connection between join-irreducible element and meet-irreducible element which is embodied in V, we can get the relation I. So we get the context $K=(G,M,I)$.

Algorithm 1: formal context acquisition form complete lattice

Input: complete lattice V;

Output: formal context K

Step 1: Traverse complete lattice V upward form its minimal element, if the currently visited element has only one upper neighbor, then labeling this element with a unique letter;

Step 2: Traverse complete lattice V downward form its maximal element, if the currently visited element has only one lower neighbor, then labeling this node with a unique digit;

Step 3: If m different digits and n different letters are used, then establish a context with m rows and n columns and store it by using a matrix $A=\{a_{ij}\}_{m\times n}$, and each digit corresponding to a row while each letter corresponding to a column, and initialize it to be a nil-matrix;

Step 4: Traverse complete lattice V, for each node labeled by a digit(assume this digit is i), visit its upper neighbors until meet the maximal element. And in this process if there exist an element labeled by a letter(assume this letter is j), then set the value of a_{ij} to 1;

Step 5: Return A.

Example 1. Let V be a complete and is shown in Figure 1. By using Algorithm 1, firstly we find its irreducible elements, which is shown in Figure 2. Secondary, we get its corresponding formal context K that is shown in Table 1. And according to K we get its storage matrix

$$A_1=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

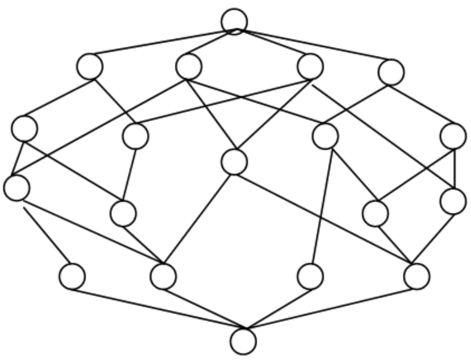

Figure 1. Complete Lattice V

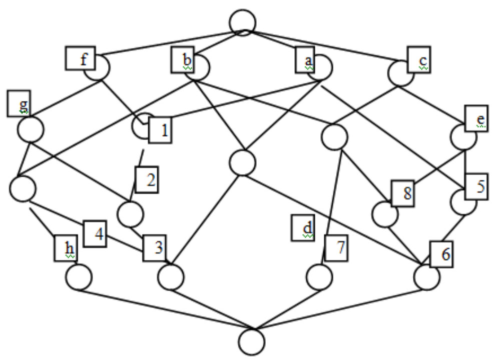

Figure 2. Irreducible Elements of V

Table 1. Formal Context K

	a	b	c	d	e	f	g	h
1	*					*		
2	*					*	*	
3	*	*				*	*	
4		*				*	*	*
5	*		*		*			
6	*	*	*		*			
7		*	*	*				
8		*	*		*			

If using adjcent matrix to store complete lattice V, we must establish a matrix A_2 with 19 rows and 19 columns, and the number of 1 elements is twice of the number of links between the nodes, i.e. 62.

The storage efficiency comparison between A_1 and A_2 is given is Table 2.

Table 2. Storage Efficiency Comparison

	size	non-zero elements	Percentage of the non-zero elements
A_1	8×8	26	40.6%
A_2	19×19	62	17.1%

Other than saving storage space, our method is also helpful to improve the efficiency for judging complete lattice isomorphism. Complete lattice isomorphism judgment can be converted into graph isomorphism judgment, and this is seen as a NP - complete problem by a majority of scholars [11]. If we judge graph isomorphism by performing row and column exchange of adjacent matrix, at the worst case, the total number of exchange will reach $r!\times c!$ times(r is the number of rows, as c is the of columns), and this is much greater than exponential time complexity. Our method reduces the scale of the storage matrix, thus it can improve the efficiency of complete lattice isomorphism judgment.

4. Conclusion

Based on Formal Concept Analysis, we propose a complete lattice storage method. The proposed method only stores irreducible elements, and the relationship between them. Compared with the adjacency matrix storage method, the proposed method can improve the storage efficiency.

Acknowledgements

The work presented in this paper is supported by Doctorial Foundation of Henan Polytechnic University (B2011-102).

References

[1] Yuan Dayu, Mitra Prasenjit. Lindex: A lattice-based index for graph databases. VLDB Journal. 2013; 22(2):229-252.
[2] Ye J, Coyle L, Dobson S, Nixon Paddy. Representing and manipulating situation hierarchies using situation lattices. Revue d'Intelligence Artificielle. 2008; 22(5):647-667.

[3] Yaron A, B. Goldstein, Alexander Bockmayr. A Lattice-Theoretic Framework for Metabolic Pathway Analysis. Computational Methods in Systems Biology. Lecture Notes in Computer Science. 2013; 130:178-191.

[4] Thierry Denoeux, Zoulficar Younes, Fahed Abdallah. Representing uncertainty on set-valued variables using belief functions. Artificial Intelligence. 2010; 174(7-8):479-499.

[5] Ganter, B., Wille, R.. Formal Concept Analysis: Mathematical Foundations. Berlin: Springer. 1999.

[6] Poelmans Jonas, Ignatov Dmitry I, Kuznetsov Sergei O. Formal concept analysis in knowledge processing: A survey on applications. Expert System with Applications. 2013; 40(16): 6538-6560.

[7] V. K. Garg, N. Mittal, A. Sen. Applications of lattice theory to distributed computing. ACM SIGACT Notes. 2003; 34(3):40-61.

[8] R. Wille. Concept lattices and conceptual knowledge systems. Computers & Mathematics with Applications. 1992; 23(6-9):493-515.

[9] R. Wille. Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts. I. Rival (Ed.), Ordered Sets, Reidel, Dordrecht, Boston (1982), pp. 445-470.

[10] Wenxue Hong, Jingying Mao, Jianping Yu, Linsong Jia. The complete definitions of attributes and abstract description of attribute features of the formal context. ICIC Express Letters. 2013; 7(3):997-1003.

[11] Karp R M. Reducibility among combinatorial problems. New York: Plenum Press, 1972.