Maintaining Class Membership Information

Anne Berry* Alain Sigayret*

7th June 2002

Abstract

Galois lattices (or concept lattices), which are lattices built on a binary relation, are now used in many fields, such as Data Mining and hierarchy organization, but may be of exponential size. In this paper, we propose a decomposition of a Galois sub-hierarchy which is of small size but contains useful inheritance information. We show how to efficiently maintain this information when an element is added to or removed from the relation, using a dynamic domination table which describes the underlying graph with which we encode the lattice.

1 Introduction

Galois lattices (also called concept lattices), are an emerging tool in research areas such as Data Mining, Database Managing and Object Hierarchy Organization (see [9], [15], [16], [19], [20], [21]).

A lattice has the advantage over a tree that it allows a much more complex structure, as every pair of elements not only has a greatest lower bound, but also has a lowest upper bound. In particular, this structure has been shown to be well adapted to representing multiple inheritance (a car can be both a wheeled vehicle and water-faring).

Concept lattices, moreover, are built from a binary relation, classically between a set $\mathcal{P}$ of properties and a set $\mathcal{O}$ of objects, which completely represents the information which is to be analysed. The elements of the lattice describe all possible maximal associations of the properties and objects. They are both a powerful investigation tool for Data Mining applications,

*LIMOS UMR CNRS 6158, Ensemble Scientifique des Cézeaux, Universit Blaise Pascal, 63170 Aubière, France. E-mail: berry@isima.fr, sigayret@isima.fr

and a complete underlying structuration for relationships, which makes them a good basis for extracting an organization into hierarchies.

The main drawback of such a lattice is that it may be exponential in size compared to the initial binary relation it is constructed from. Though it is known that, when one can give an upper bound on the number of properties for an object, the lattice is of polynomial size (see [8], [10], [11]), there is no known *general* characterization for relations which will define only a polynomial number of concepts. As a result, users of concept lattices are left with few options:

- use only a part of the lattice, by defining a sub-lattice or applying zooming techniques;
- use and maintain a relation which belongs to a class which is known to define only a polynomial number of concepts;
- use a polynomial representation of the lattice to extract the most pertinent information.

In this paper, we will investigate the third possibility, by using both an underlying polynomial-sized graph which we use to encode the lattice (see [4]) and by restricting the information to a variant of a Galois sub-hierarchy.

Moreover, we address the issue of maintaining such information when an element is added to or deleted from the relation, without re-computing the entire sub-hierarchy.

2 Concept Lattices and Galois Sub-hierarchies

Given a finite set $\mathcal{P}$ of "properties" (which may be attributes, methods, features, etc., and which we will denote by lowercase letters) and a finite set $\mathcal{O}$ of "objects" (which may be tuples, individuals, classes, etc., and which we will denote by numbers), we consider a binary relation R as a proper subset of the Cartesian product $\mathcal{P} \times \mathcal{O}$; we will refer to the triple $(\mathcal{P}, \mathcal{O}, R)$ as a **context**.

Given a context $C = (\mathcal{P}, \mathcal{O}, R)$, a **concept** or **closed set** of C, also called a **maximal rectangle** of R, is a sub-product $A \times B \subseteq R$ such that $\forall x \in \mathcal{O} - B, \exists y \in A \,|\, (y,x) \notin R$, and $\forall x \in \mathcal{P} - A, \exists y \in B \,|\, (x,y) \notin R$. A is called the **intent** of the concept, B is called the **extent**.

The set of concepts thus defined form a lattice when ordered by inclusion on the intents, or, dually, by inclusion on the extents, called a **concept lattice** or **Galois lattice**.

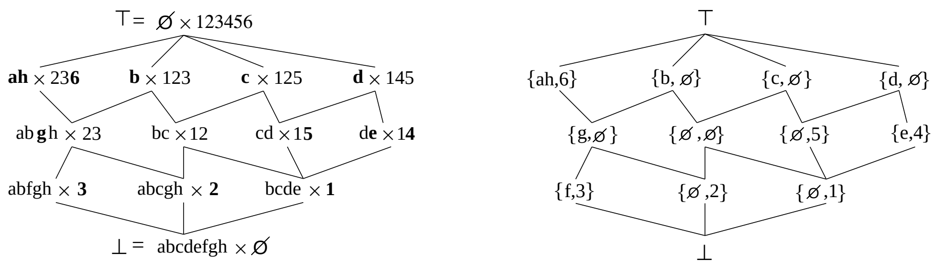

Figure 1: Concept lattice $\mathcal{L}(R)$ and corresponding simplified lattice of relation R of Example 2.1.

Example 2.1 $\mathcal{P} = \{a, b, c, d, e, f\}$, $\mathcal{O} = \{1, 2, 3, 4, 5, 6\}$. *Binary relation R:*

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

The associated concept lattice $\mathcal{L}(R)$ is shown in Figure 1.

Because of the inheritance rules associated with this lattice, the labels can be simplified into mentioning only properties or objects which occur for the first time, in a top-bottom fashion for properties, in a bottom-top fashion for objects.

Example 2.2 *The simplified concept lattice obtained from $\mathcal{L}(R)$ associated with relation R of Example 2.1 is shown in Figure 1.*

When the number of elements of the lattice is exponential, several authors ([9], [7], [15]) have found it useful to further simplify this lattice into a **Galois sub-hierarchy**, by defining a partially ordered set obtained by removing from this simplified lattice trivial nodes such as empty set pairs $\{\emptyset, \emptyset\}$, and usually the top and bottom elements.

Example 2.3 *Figure 2 shows the Galois sub-hierarchy obtained from the simplified lattice of Example 2.2. Note that this partial ordering does not*

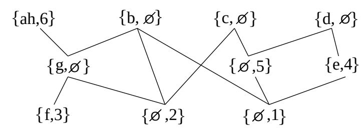

Figure 2: Galois sub-hierarchy obtained from the simplified lattice of Example 2.2.

define a lattice, as elements $\{\emptyset, 2\}$ *and* $\{\emptyset, 1\}$ *fail to have a unique nearest common descendent.*

3 Decomposing a Galois Sub-hierarchy Using Graph Domination

Our approach to encoding a Galois lattice (see [4]) is, surprisingly enough, to use an underlying graph G_R, constructed on the complement of the relation, defined, for a given context $(\mathcal{P}, \mathcal{O}, R)$ as $G_R = (V, E)$, with $V = \mathcal{P} \cup \mathcal{O}$, and with edges defined as:

1. internal edges which make $\mathcal{P}$ and $\mathcal{O}$ into cliques (if $x, y \in \mathcal{P}$ then $xy \in E$ and if $x, y \in \mathcal{O}$ then $xy \in E$).
2. external edges: if $x \in \mathcal{P}$ and $y \in \mathcal{O}$ then $xy \in E$ iff $(x, y) \notin R$.

Since $\mathcal{P}$ and $\mathcal{O}$ are trivially cliques, we will not represent their internal edges, nor will these have any influence on the complexity evaluations we discuss, as they need not be traversed. We will denote by $N^+(x)$ the external neighborhood of vertex x: if $x \in \mathcal{P}, N^+(x) = \{y \in \mathcal{O} | (x, y) \notin R\}$, and if $x \in \mathcal{O}, N^+(x) = \{y \in \mathcal{P} | (y, x) \notin R\}$. We will also use $\overline{N^+}(x)$ to denote $\mathcal{O} - N^+(x)$ for $x \in \mathcal{P}$, $\overline{N^+}(y)$ to denote $\mathcal{P} - N^+(y)$ for $x \in \mathcal{O}$. We use $n = |\mathcal{P}| + |\mathcal{O}|$, and $m = |\mathcal{P}| \times |\mathcal{O}| - |R|$.

Note that, though not much is known on the size of the concept lattice defined by a given relation, in general the lattice tends to be exponential in size when it is dense (i.e. when it has many crosses); in this case, for our graph, m will be of the order of n instead of n^2.

The reason we define this graph is that we have the remarkable property that a vertex set S of G_R is a minimal separator of G_R, separating connected component A from connected component B if, and only if $A \times B$ is a concept

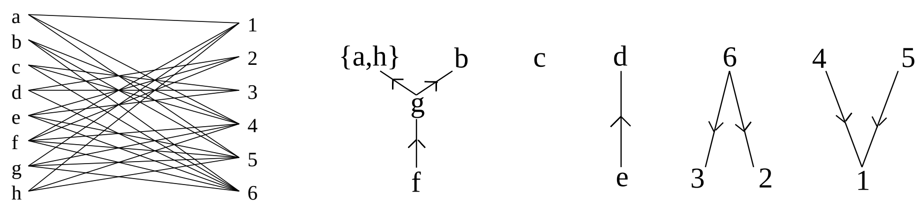

Figure 3: Graph G_R coding the relation from Example 2.1 and the corresponding domination relation.

defined by relation R. Now this may seem a little far-fetched, but a steady output of work done in the past decade has yielded many results on minimal separation, and in [4] we show this to be an efficient tool for concept lattice investigation and concept generation. It is interesting to note, however, that quite regularly work appears on the relationship between graphs and lattices (see [2], [13], [14], [17], [18]).

One of the related graph notions which turns out to be of primary importance for the study of concept lattices is that of **domination**: a vertex x is said to dominate another vertex y if $N^+(y) \subset N^+(x)$.

The domination relation defines a partial pre-ordering on V, which we decompose into the **property domination relation** and the **object domination relation**.

Example 3.1 *The relation of Example 2.1 yields the graph given in Figure 3.*
$N^+(a) = \{1,4,5\}$, $N^+(b) = \{4,5,6\}$, $N^+(c) = \{3,4,6\}$, $N^+(d) = \{2,3,6\}$, $N^+(e) = \{2,3,5,6\}$, $N^+(f) = \{1,2,4,5,6\}$, $N^+(g) = \{1,4,5,6\}$, $N^+(h) = N^+(a)$, $N^+(1) = \{a,f,g,h\}$, $N^+(2) = \{d,e,f\}$, $N^+(3) = \{c,d,e\}$, $N^+(4) = \{a,b,c,f,g,h\}$, $N^+(5) = \{a,b,e,f,g,h\}$, $N^+(6) = \{b,c,d,e,f,g\}$.

a and h share the same neighborhood and behave as a single vertex ah. f dominates g, g dominates a and b; by transitivity, this implies that f also dominates a and b. c is neither dominated nor dominating.

Figure 3 gives the property domination relation and the object domination relation.

It is clear from Example 3.1 that this domination relation is strongly related to the Galois sub-hierarchy shown in Figure 2.

In order to precisely describe this relationship, we introduce a decomposition of the Galois sub-hierarchy into the sub-hierarchy of intents and the sub-hierarchy of extents, by using only the left or right parts of the labels

Figure 4: Sub-hierarchies of intents and of extents derived from Figure 2.

of the sub-hierarchy. If, consistently with the original definition, we then remove the empty set nodes, we find the domination relation.

Example 3.2 *Figure 4 gives the sub-hierarchies of intents and of extents derived from Figure 2.*

Based on these considerations, we give the following property:

Property 3.3 *The property domination relation of G_R is equivalent to the Galois sub-hierarchy taken on the intents, from which all empty set nodes have been removed, and the object domination relation of G_R is equivalent to the Galois sub-hierarchy taken on the extents, from which all empty set nodes have been removed.*

4 Computing and Updating the Domination Information

Computing the domination relation of a graph costs roughly $O(nm)$ time. In this section, we will introduce a data structure which will enable us to update a relation by adding or deleting elements, with a cost of only $O(n)$ per update.

We will restrict our description of our process to computing the property domination relation; of course, the same process can be applied dually to computing the object domination relation.

4.1 Computing and Querying the Domination Table

In order to maintain property domination information, we construct a domination table, which, for each pair (x, y) of properties, lists the objects, the presence of which prevents x from dominating y, or from having a neighborhood which is the same as that of y. This just means that if for object i, $(x, i) \in R$ and $(y, i) \notin R$, i will appear in the list for (x, y).

Example 4.1 *Property domination table corresponding to relation R from Example 2.1:*

	a	b	c	d	e	f	g	h
a	$\emptyset$	$\{1\}$	$\{1,5\}$	$\{1,4,5\}$	$\{1,4\}$	$\emptyset$	$\emptyset$	$\emptyset$
b	$\{6\}$	$\emptyset$	$\{5\}$	$\{4,5\}$	$\{4\}$	$\emptyset$	$\emptyset$	$\{6\}$
c	$\{3,6\}$	$\{3\}$	$\emptyset$	$\{4\}$	$\{4\}$	$\{3\}$	$\{3\}$	$\{3,6\}$
d	$\{2,3,6\}$	$\{2,3\}$	$\{2\}$	$\emptyset$	$\emptyset$	$\{3\}$	$\{2,3\}$	$\{2,3,6\}$
e	$\{2,3,6\}$	$\{2,3\}$	$\{2,5\}$	$\{5\}$	$\emptyset$	$\{3\}$	$\{2,3\}$	$\{2,3,6\}$
f	$\{2,6\}$	$\{1,2\}$	$\{1,2,5\}$	$\{1,4,5\}$	$\{1,4\}$	$\emptyset$	$\{2\}$	$\{2,6\}$
g	$\{6\}$	$\{1\}$	$\{1,5\}$	$\{1,4,5\}$	$\{1,4\}$	$\emptyset$	$\emptyset$	$\{6\}$
h	$\emptyset$	$\{1\}$	$\{1,5\}$	$\{1,4,5\}$	$\{1,4\}$	$\emptyset$	$\emptyset$	$\emptyset$

This table is to be read as (x, y) pairs, each containing the information on x dominating y, where x labels a column and y a row.

The algorithmic process for constructing the property domination table is simple:

For each $x \in \mathcal{P}$
    For each $y \in \mathcal{P}$
        For each $z \in \mathcal{O}$
            If $(x, z) \in R$ and $(y, z) \notin R$ then add z to list (x, y);

The process for building the object domination table is symmetric, and obtained from the previous one by exchanging $\mathcal{P}$ and $\mathcal{O}$.

The global size of the tables is of $O(nm)$; as each entry can be computed in constant time using relation R, the total time cost for computing the property domination table and the object domination table will be in $O(nm)$.

From the table, we can deduce that:

- *a will dominate b if object 6 is deleted.*
- *e dominates d since $(e, d) = \emptyset$. The property domination relation shown in Figure 3 can easily be globally computed by examining this domination for each property.*
- *$N^+(a) = N^+(h)$, since a dominates h and h dominates a; note how columns a and h are identical.*

- *$N^+(g) = N^+(a) \cup N^+(b)$, as column g is the intersection of columns a and b in the domination table. This means that relation R fails to be a reduced relation: g appears in the intent of a concept of $\mathcal{L}(R)$ iff both a and b appear too.*

Object domination table corresponding to relation R from Example 2.1.

	1	*2*	*3*	*4*	*5*	*6*
1	$\emptyset$	$\{a,g,h\}$	$\{a,f,g,h\}$	$\emptyset$	$\emptyset$	$\{a,h\}$
2	$\{d,e\}$	$\emptyset$	$\{f\}$	$\{d,e\}$	$\{d\}$	$\emptyset$
3	$\{c,d,e\}$	$\{c\}$	$\emptyset$	$\{d,e\}$	$\{c,d\}$	$\emptyset$
4	$\{b,c\}$	$\{a,b,c,g,h\}$	$\{a,b,f,g,h\}$	$\emptyset$	$\{c\}$	$\{a,h\}$
5	$\{b,e\}$	$\{a,b,g,h\}$	$\{a,b,f,g,h\}$	$\{e\}$	$\emptyset$	$\{a,h\}$
6	$\{b,c,d,e\}$	$\{b,c,g,h\}$	$\{b,f,g\}$	$\{d,e\}$	$\{c,d\}$	$\emptyset$

Computing the intents and extents of the sub-hierarchy elements from the domination table.
For a given property, for example f, find from the table which properties f dominates, that is which have $\emptyset$ in the f column; this query yields a,b,f,g,h; $abfgh$ will be the intent; then compute $\mathcal{O} - N^+(f) = \{3\}$, which yields the extent of the element; we thus obtain element $abfgh \times 3$ as element of $\mathcal{L}(R)$, in a complexity proportional to the size of the result.

Computing the simplified Galois sub-hierarchy from the domination table.
When desirable, it is easy to reconstruct it from the property domination table:

For a given property or object, one can also deduce from the table the corresponding element of

When the entire Galois sub-hierarchy needs to be re-constructed, one can use the following

1. For each property $x \in \mathcal{P}$

$X \leftarrow$ list of properties which x dominates; // L(x) includes x.

If x shares its neighborhood with other properties, forming set $M(x)$, **then** Add $\{x, \emptyset\}$ to the list of elements of the simplified Galois sub-hierarchy;

$Y \leftarrow \overline{N^+(x)}$;

Add X to the list of already computed intents, keeping a pointer on $\{x, \emptyset\}$;

2. For each object $y \in \mathcal{O}$

$Y \leftarrow$ list of objects which y dominates;

If y shares its neighborhood with other objectss, forming set $M(y)$, **then**

$X \leftarrow \overline{N^+(y)}$;

If intent X has already been computed **then**

X points towards element $\{x, \emptyset\}$;

Replace element $\{x, \emptyset\}$ with element $\{x, y\}$;

This can be accomplished in roughly $O(n^2)$ time, though this complexity could be streamlined using the maximal number of crosses in a line.

Example 4.2

a shares its neighborhood with h and dominates no other property; $\overline{N^+(a)} = \{2, 3, 6\}$*; corresponding element of the Galois lattice:* $ah \times 236$*; temporarily store* $(ah, \emptyset)$ *as an element of the simplified sub-hierarchy.*

b dominates no other property: $\overline{N^+(b)} = \{1, 2, 3\}$*; corresponding element of the Galois lattice:* $b \times 123$*; tentatively store* $(b, \emptyset)$ *as an element of the simplified sub-hierarchy.*

c dominates no other property: $\overline{N^+(c)} = \{1, 2, 5\}$*; corresponding element of the Galois lattice:* $c \times 125$*; tentatively store* $(c, \emptyset)$ *as an element of the simplified sub-hierarchy.*

d dominates no other property: $\overline{N^+(d)} = \{1, 4, 5\}$*; corresponding element of the Galois lattice:* $d \times 145$*; tentatively store* $(d, \emptyset)$ *as an element of the simplified sub-hierarchy.*

e dominates d; $\overline{N^+(e)} = \{1, 4\}$*; corresponding element of the Galois lattice:* $de \times 14$*; tentatively store* $(e, \emptyset)$ *as an element of the simplified sub-hierarchy.*

f dominates a, b, f, g, h*;* $\overline{N^+(f)} = \{3\}$*; corresponding element of the Galois lattice:* $abfgh \times 3$*; tentatively store* $(f, \emptyset)$ *as an element of the simplified sub-hierarchy.*

g dominates a, b, g, h*;* $\overline{N^+(g)} = \{2, 3\}$*; corresponding element of the Galois lattice:* $abgh \times 23$*; tentatively store* $(g, \emptyset)$ *as an element of the simplified sub-hierarchy.*

h has already been processed as sharing its neighborhood with a;

1 dominates no other object: $\overline{N^+(1)} = \{b, c, d, e\}$*; corresponding element of the Galois lattice:* $bcde \times 1$*; store* $(\emptyset, 1)$ *as an element of the simplified sub-hierarchy.*

2 dominates no other object: $\overline{N^+(2)} = \{a, b, c, g, h\}$*; corresponding element of the Galois lattice:* $abcgh \times 2$*; store* $(\emptyset, 2)$ *as an element of the* *simplified sub-hierarchy.*

3 dominates no other object: $\overline{N^+(3)} = \{a, b, f, g, h\}$*; corresponding element of the Galois lattice:* $abfgh \times 3$*;* $abfgh$ *is an already computed intent, which points towards* $(f, \emptyset)$*; replace* $(f, \emptyset)$ *with* $(f, 3)$ *as an element of the simplified sub-hierarchy.*

4 dominates 1; $\overline{N^+(4)} = \{d, e\}$*; corresponding element of the Galois lattice:* $de \times 14$*;* de *is an already computed intent, which points towards* $(e, \emptyset)$*; replace* $(e, \emptyset)$ *with* $(e, 4)$ *as an element of the simplified sub-hierarchy.*

5 dominates 1; $\overline{N^+(5)} = \{c, d\}$*; corresponding element of the Galois lattice:* $cd \times 15$*; store* $(\emptyset, 5)$ *as an element of the simplified sub-hierarchy.*

6 dominates 2 and 3 ; $\overline{N^+(6)} = \{a, h\}$*; corresponding element of the Galois lattice:* $ah \times 236$*;* ah *is an already computed intent, which points towards* $(ah, \emptyset)$*; replace* $(ah, \emptyset)$ *with* $(ah, 6)$ *as an element of the simplified sub-hierarchy.*

Note how exactly all the elements of the the simplified sub-hierarchy have been computed.

4.2 Updating the Domination Table

1. When **adding an element** (x, z) to relation R, with $x \in \mathcal{P}$ and $z \in \mathcal{O}$, which means adding a cross in R at location (x, z), or, equivalently, removing edge xz from the corresponding coding graph G_R:

   - for each "non-cross" y of line z in R (i.e. $(y, z) \notin R$), **add** z to list (x, y);
   - for each "cross" y of line z in R (i.e. $(y, z) \in R$), **delete** z from list (y, x);

2. When **deleting an element** (x, z) from relation R, $x \in \mathcal{P}, z \in \mathcal{O}$:

   - for each "non-cross" y of line z in R (i.e. $(y, z) \notin R$), **delete** z from list (x, y);
   - for each "cross" y of line z in R (i.e. $(y, z) \in R$), **add** z to list (y, x);

Updating the table will cost time $O(|\mathcal{P}|)$.

Example 4.3 *Let us add element* $(b, 5)$ *to relation* R *of Example 2.1.*

New relation R' *obtained:*

	a	b	c	d	e	f	g	h
1		×	×	×	×			
2	×	×	×				×	×
3	×	×				×	×	×
4				×	×			
5		×	×	×				
6	×							×

Line 5 contains non-crosses a, e, f, g, h *and crosses* c, d. *5 must be added to the lists of* (b, a), (b, e), (b, f), (b, g) *and* (b, h), *and deleted from the lists of* (c, b) *and* (d, b).

New domination table obtained:

	a	b	c	d	e	f	g	h
a	$\emptyset$	$\{1, \boldsymbol{5}\}$	$\{1, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\emptyset$
b	$\{6\}$	$\emptyset$	$\{\not{5}\}$	$\{4, \not{5}\}$	$\{4\}$	$\emptyset$	$\emptyset$	$\{6\}$
c	$\{3, 6\}$	$\{3\}$	$\emptyset$	$\{4\}$	$\{4\}$	$\{3\}$	$\{3\}$	$\{3, 6\}$
d	$\{2, 3, 6\}$	$\{2, 3\}$	$\{2\}$	$\emptyset$	$\emptyset$	$\{3\}$	$\{2, 3\}$	$\{2, 3, 6\}$
e	$\{2, 3, 6\}$	$\{2, 3, \boldsymbol{5}\}$	$\{2, 5\}$	$\{5\}$	$\emptyset$	$\{3\}$	$\{2, 3\}$	$\{2, 3, 6\}$
f	$\{2, 6\}$	$\{1, 2, \boldsymbol{5}\}$	$\{1, 2, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\{2\}$	$\{2, 6\}$
g	$\{6\}$	$\{1, \boldsymbol{5}\}$	$\{1, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\{6\}$
h	$\emptyset$	$\{1, \boldsymbol{5}\}$	$\{1, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\emptyset$

As a result of the modification of R, c *now dominates* b. *Figure 5 shows the new concept lattice* $\mathcal{L}(R')$ *and the associated Galois sub-hierarchy; Figure 6 gives the new property domination relation.*

Let us now delete element $(b, 1)$ *from the previous relation* R'.

New relation R'' *obtained:*

	a	b	c	d	e	f	g	h
1			×	×	×			
2	×	×	×				×	×
3	×	×				×	×	×
4				×	×			
5		×	×	×				
6	×							×

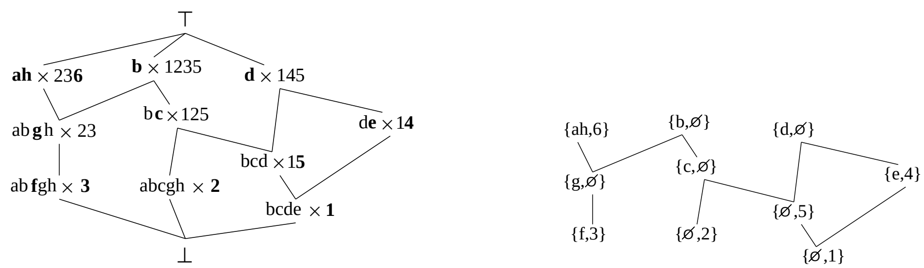

Figure 5: New concept lattice $\mathcal{L}(R')$ and the associated Galois sub-hierarchy obtained when adding $(b, 5)$ to relation R of Example 2.1.

Line 1 contains non-crosses a, f, g, h *and crosses* c, d, e. *1 must be deleted from the lists of* $(b, a), (b, f), (b, g)$ *and* (b, h), *and added to the lists of* $(c, b), (d, b)$ *and* (e, b).

New domination table obtained:

	a	b	c	d	e	f	g	h
a	$\emptyset$	$\{\not{1}, 5\}$	$\{1, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\emptyset$
b	$\{6\}$	$\emptyset$	$\{\boldsymbol{1}\}$	$\{\boldsymbol{1}, 4\}$	$\{\boldsymbol{1}, 4\}$	$\emptyset$	$\emptyset$	$\{6\}$
c	$\{3, 6\}$	$\{3\}$	$\emptyset$	$\{4\}$	$\{4\}$	$\{3\}$	$\{3\}$	$\{3, 6\}$
d	$\{2, 3, 6\}$	$\{2, 3\}$	$\{2\}$	$\emptyset$	$\emptyset$	$\{3\}$	$\{2, 3\}$	$\{2, 3, 6\}$
e	$\{2, 3, 6\}$	$\{2, 3, 5\}$	$\{2, 5\}$	$\{5\}$	$\emptyset$	$\{3\}$	$\{2, 3\}$	$\{2, 3, 6\}$
f	$\{2, 6\}$	$\{\not{1}, 2, 5\}$	$\{1, 2, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\{2\}$	$\{2, 6\}$
g	$\{6\}$	$\{\not{1}, 5\}$	$\{1, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\{6\}$
h	$\emptyset$	$\{\not{1}, 5\}$	$\{1, 5\}$	$\{1, 4, 5\}$	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\emptyset$

After this second modification of R, c *does not dominate* b *any longer. Figure 7 shows the new concept lattice and the associated Galois sub-relation.*

The property domination relation reverts to its original form, given by Figure 3, even though the lattice is larger and structurally significantly different.

5 Conclusion

We have shown a new approach to modifying significant information we need to extract from a concept lattice structure, with an efficient updating

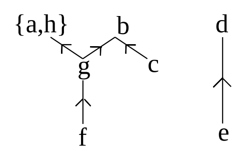

Figure 6: Domination relation obtained when adding $(b, 5)$ to relation R of Example 2.1.

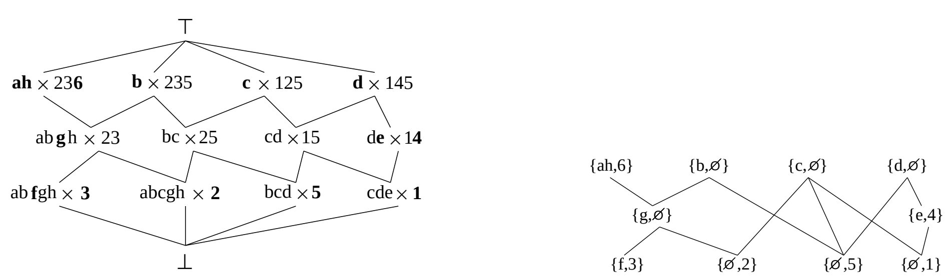

Figure 7: New concept lattice $\mathcal{L}(R'')$ and the associated Galois sub-hierarchy obtained when deleting $(b, 1)$ from the new relation R' of Example 4.3.

technique, based on a new graph-based data structure which enables us to avoid re-computing the entire structure.

This updating technique could be extended to efficiently constructing the initial domination table for a given relation which is very dense, by first considering the relation with only a diagonal of zeroes, which describes a graph with no domination, and then removing elements from this until the desired relation is obtained.

Conversely, the principle of domination table could also be used to model a relation into respecting a given sub-hierarchy, with particular desirable relationships between classes.

Acknowledgment

We thank the referees for their very interesting questions and suggestions.

References

[1] M. Barbut and B. Monjardet. Ordre et classification. *Classiques Hachette*, 1970.

[2] A. Berry and J.-P. Bordat. Orthotreillis et séparabilité dans un graphe non-orienté. *Mathématiques, Informatique et Sciences Humaines*, 146:5–17, 1999.

[3] A. Berry, J.-P. Bordat and O. Cogis. Generating all the minimal separators of a graph. *International Journal of Foundations of Computer Science*, 11:397-404, 2000.

[4] A. Berry and A. Sigayret. Representing a concept lattice by a graph. *Workshop on Discrete Mathematics for Data Mining, Proc. 2nd SIAM Workshop on Data Mining, Arlington (VA), April 11–13 2002.*

[5] G. Birkhoff. Lattice Theory. *American Mathematical Society*, 3rd Edition, 1967.

[6] J.-P. Bordat. Calcul pratique du treillis de Galois d'une correspondance. *Mathématiques, Informatique et Sciences Humaines*, 96:31–47, 1986.

[7] J.-B. Chen and S. C. Lee. Generation and Reorganization of Subtype Hierarchies. *Journal of Object Oriented Programming*, 8(8), 1996.

[8] R. Godin. Complexité de Structures de Treillis. *Annales des Sciences Mathématiques du Québec*, 13(1):19-38, 1989.

[9] R. Godin and H. Mili. Building and Maintaining Analysis-Level Class Hierarchies Using Galois Lattices. *Proceedings of ACM OOPSLA'93, Special issue of Sigplan Notice*, 28(10):394–410.

[10] R.Godin, R. Missaoui and A. April. Experimental Comparison of Navigation in a Galois Lattice with Conventional Information Retrieval Methods. *International Journal of Man-Machine Studies*, 38:747-767, 1993.

[11] R. Godin, E. Saunders and J. Gecsei. Lattice Model of Browsable Data Spaces. *Information Sciences*, 40:89-116,1986.

[12] M.C. Golumbic. Algorithmic Graph Theory and Perfect Graphs. *Academic Press*, New York, 1980.

[13] M. Hager. On Halin-Lattices in Graphs. *Discrete Mathematics*, 47:235–246, 1983.

[14] R. Halin. Lattices of cuts in graphs. *Abh. Math. Sem. Univ. Hamburg*, 61:217–230, 1991.

[15] M. Huchard, H. Dicky and H. Leblanc. Galois lattice as a framework to specify building class hierarchies algorithms. *Theoretical Informatics and Applications*, 34:521–548, 2000.

[16] J.L. Pfaltz and C.M. Taylor. Scientific Knowledge Discovery through Iterative Transformation of Concept Lattices. *Workshop on Discrete Mathematics for Data Mining, Proc. 2nd SIAM Workshop on Data Mining, Arlington (VA), April 11–13 2002.*

[17] N. Polat. Treillis de séparation des graphes. *Can. J. Math.*, vol. XXVIII, No 4, pp. 725–752, 1976

[18] G. Sabidussi. Weak separation lattices of graphs. *Can. J. Math.*, 28:691–734, 1976.

[19] P. Valtchef, R. Missaoui, and R. Godin. A Framework for Incremental Generation of Frequent Closed Item Sets. *Workshop on Discrete Mathematics for Data Mining, Proc. 2nd SIAM Workshop on Data Mining, Arlington (VA), April 2002.*

[20] A. Yahia, L. Lakhal and J.-B. Bordat. Designing Class Hierarchies of Object Database Schemes. *Proceedings 13e journées Bases de Données avancées (BDA'97)*, 1997.

[21] M. J. Zaki, S. Parthasarathy, M. Ogihara and W. Li. New Algorithms for Fast Discovery of Association Rules. *Proceedings of 3rd Int. Conf. on Database Systems for Advanced Applications, April 1997.*