Representing a concept lattice by a graph

Anne Berry* Alain Sigayret*

Abstract

In this paper, we present a new relationship between concept lattices and graphs. Given a binary relation R, we define an underlying graph G_R, and establish a one-to-one correspondence between the set of elements of the concept lattice of R and the set of minimal separators of G_R.

We explain how to use the properties of minimal separators to define a sublattice, decompose a binary relation, and enumerate the elements of the lattice.

1 Introduction

One of the important challenges in data handling is generating or navigating the concept lattice of a binary relation.

Concept lattices are well-studied as a classification tool (see [1]), are used in several areas of Database Managing, as studying Object-Oriented Databases (see [28]), inheritance lattices (see [14]), and generating frequent sets, and are a promising aid for the rapidly emerging field of data mining for biological databases.

In this paper, we present a new paradigm for describing and understanding concept lattices, by equating the closed sets of the lattice with the set of minimal separators of an underlying graph.

The notion of minimal separator, introduced by Dirac in 1961 to characterize chordal graphs (see [10]), has been studied extensively during the past decade on non-chordal graphs (see [17], [16], [21], [3], [27]), and has yielded many new theoretical and algorithmical graph results.

We will apply some of these results to analyzing and decomposing a binary relation and the associated concept lattice.

Because of space restrictions, we will mostly limit ourselves to presenting our basic techniques and results, and to illustrating the mechanisms involved on a running example. The proofs of our results do not present great difficulties and are detailed in the full version.

The paper is organized as follows:

Section 2 gives some very brief preliminary notions on both concept lattices and graph separators, and presents our running example. For undefined notions, the reader is referred to the classical works of [8] and [12]. In Section 3, we define the underlying graph G_R of a binary relation R, describe some of its properties, and explain how it relates to the concept lattice $\mathcal{L}(R)$. In Section 4, we define a sublattice by making into a clique a minimal separator of the underlying graph. Section 5 shows how to use a clique minimal separator to decompose a binary relation. Section 6 addresses the issue of generating all the elements of the lattice efficiently.

*LIMOS FRE CNRS 2239, Ensemble Scientifique des Cézeaux, Université Blaise Pascal, F-63170 Aubière, France. e-mail: berry@isima.fr, sigayret@isima.fr

2 Preliminaries

2.1 Concept lattices

Given a set $\mathcal{P}$ of "properties" or "attributes" (which we will denote by lowercase letters) and a set $\mathcal{O}$ of "objects" or "tuples" (which we will denote by numbers), a binary relation R is a subset of the Cartesian product $\mathcal{P} \times \mathcal{O}$. A **closed set** of R, also called a **maximal rectangle** of R, is a sub-product $A \times B \subset R$ such that $\forall x \in \mathcal{O} - B, \exists y \in A | (y, x) \notin R$, and $\forall x' \in \mathcal{P} - A, \exists y' \in B | (x', y') \notin R$. For a given closed set $A \times B$, A is called the **intent** of the rectangle, B is called the **extent**. Element $\emptyset \times \mathcal{O}$ is called the **bottom element**, and $\mathcal{P} \times \emptyset$ is called the **top element**. A path from bottom to top is called a **maximal chain** of the lattice. Note that there is, in general, an exponential number of elements in a concept lattice.

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
1		x	x	x	x	
2	x	x	x			
3	x	x				x
4				x	x	
5			x	x		
6	x					

$bc \times 12$ and $abf \times 3$ are maximal rectangles of R. bc is the **intent** of rectangle $bc \times 12$, and 12 its **extent**.

The concept lattice (also called Galois lattice) $\mathcal{L}(R)$ of binary relation R is the Hasse diagram (transitivity arcs are omitted) of the partial ordering of all maximal rectangles (also called **elements** of the lattice) by inclusion on the intents. An element $A' \times B'$ is said to be a **successor** of element $A \times B$ if $A \subset A'$ and there is no intermediate element $A" \times B"$ such that $A \subset A" \subset A'$. The successors of the bottom element are called **atoms**.

On our example, $a \times 236 < ab \times 23 < abf \times 3$.

$ab \times 23$ and $bc \times 12$ are not comparable. $ab \times 23$ is a successor of $a \times 236$, $a \times 236$ is a predecessor of $ab \times 23$.

The atoms of $\mathcal{L}(R)$ are: $a \times 236, b \times 123, c \times 125$ and $d \times 145$.

$(\emptyset \times 123456, b \times 123, bc \times 12, abc \times 2, abcdef \times \emptyset)$ is a maximal chain of the lattice.

2.2 Graphs and graph separators

Let $G = (V, E)$ be an undirected graph (V is the vertex set, $|V| = n$ and $E \subset (V \times V)$ is the edge set, $|E| = m$). For $X \subset V$, $G(X)$ denotes the subgraph induced by X in G. The **neighborhood** of vertex x (the set of vertices y such that xy is an edge of E) is denoted by $N(x)$, and if xy is an edge of E, we say that x and y see each other. For $X \subset V$, $N(X) = \bigcup_{x \in X}(N(x)) - X$.

The basic notion we use is that of minimal separator.

A **separator** S of G is a subset of vertices such that subgraph $G(V - S)$ is disconnected. S is called an **xy-separator** if x and y lie in different connected components of $G(V - S)$; S is called a **minimal xy-separator** if S is an xy-separator and no proper subset of S separates x from y. Finally, S is called a **minimal separator** if there is some pair $\{x, y\}$ of vertices such that S is a minimal xy-separator.

Property 2.2 *Let $G = (V, E)$ be a graph, let S be a vertex set. S is a minimal separator of G iff there are at least two distinct connected components C_1 and C_2 of $G(V - S)$ such that $N(C_1) = N(C_2) = S$ (C_1 and C_2 are called* full components*).*

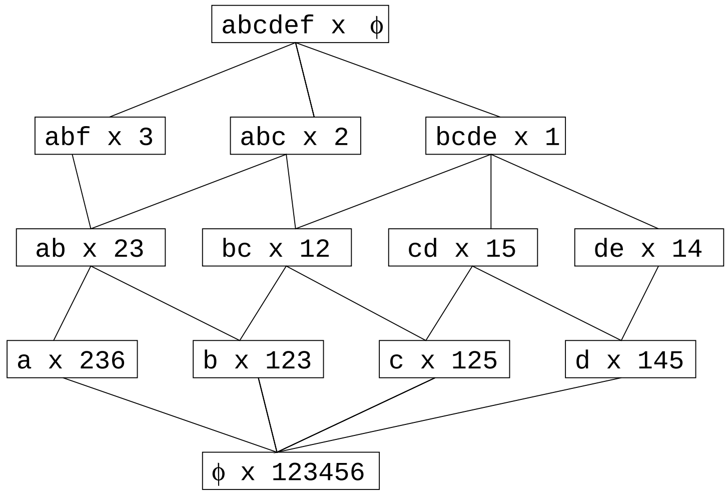


Figure 1: Concept lattice $\mathcal{L}(R)$ of relation R of Example 2.1.

There are several operations on graphs which will enable us to simplify the search for minimal separators.

The first one is the removal of the universal vertices (a vertex is said to be **universal** if it sees all the other vertices of the graph).

Property 2.3 *A vertex is universal iff it belongs to all the minimal separators of the graph.*

As a consequence of this property, if X is the set of universal vertices of G, then S is a minimal separator of $G(V - X)$ iff $S \cup X$ is a minimal separator of G. The set of universal vertices of a graph can be found in linear ($O(m)$) time.

The second simplification concerns maximal clique modules.

Definition 2.4 *A subset X of vertices is said to be a* ***clique module*** *iff $\forall x, y \in X, \{x\} \cup N(x) = \{y\} \cup N(y)$.*

Belonging to a maximal clique module defines an equivalence relation, and contracting a maximal clique module by replacing it by a single representing vertex does not modify the set of minimal separators of the graph (see [4]). In the rest of this paper, we will refer to maximal clique modules as if they were vertices. For example, the degree of a maximal clique module X will be $|N(X)|$.

The partition into maximal clique modules can be computed in linear time using Hsu and Ma's partition refinement algorithm (see [15]).

A separator S is called a **clique separator** if S is a separator and $G(S)$ is a clique; we will say that we **saturate** a non-clique separator if we add all missing edges necessary to make S into a clique.

We will also need the notion of minimal triangulation, which is the process of embedding a graph into a chordal graph by the addition of an inclusion-minimal set of edges.

Definition 2.5 *([23]) Let $G = (V, E)$ be a graph; $H = (V, E + F)$ is a* ***minimal triangulation*** *of G if H is chordal and $\forall F' \subset F$, graph $(V, E + F - F')$ fails to be chordal.*

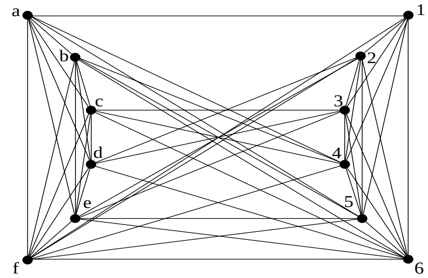


Figure 2: The underlying graph G_R of the concept lattice $\mathcal{L}(R)$ from Example 2.1.

3 Definition of the graph underlying a binary relation

3.1 Definition of G_R

We will use a binary relation R defined as in Section 2 as a subset of $\mathcal{P} \times \mathcal{O}$ to construct graph G_R as follows:

- The vertex set of G_R is $\mathcal{P} \cup \mathcal{O}$.
- $G_R(\mathcal{P})$ and $G_R(\mathcal{O})$ are cliques.
- For a vertex x of $\mathcal{P}$ and a vertex y of $\mathcal{O}$, there is an xy edge in G_R iff (x, y) is **not** in R.

Note that only the edges between a vertex of $\mathcal{P}$ and a vertex of $\mathcal{O}$ are relevant and need not be traversed when doing graph searches to find minimal separators. We will omit the internal clique edges in our complexity considerations; thus m will refer to the size of the complement of R, i.e. $|\mathcal{P} \cup \mathcal{O}| - |R|$.

3.2 Properties of G_R

By construction, the graph G_R we have just described belongs to the class of co-bipartite graphs. The graphs of this class have several remarkable properties, such as being AT-free and claw-free. This class is also hereditary: any subgraph of a co-bipartite graph is again co-bipartite.

This ensures several nice properties on the minimal separators, which make them easier to handle than on more general graphs:

Property 3.1 *Let G be a co-bipartite graph constructed on clique sets $\mathcal{P}$ and $\mathcal{O}$; then every minimal separator S of G has exactly 2 connected components, C_1 and C_2, the first of which contains only vertices of $\mathcal{P}$ and the second only vertices of $\mathcal{O}$.*

We can also derive the following from Property 2.2:

Property 3.2 *Let S be a separator of a co-bipartite graph $G = (V, E)$, let A and B be the components defined by S. Then S is a minimal separator iff $\forall x \in S, \exists a \in A, \exists b \in B, xa \in E$ and $xb \in E$.*

We are now ready to prove our main result:

Main Theorem 3.3 *Let R be a binary relation, $R \subset (\mathcal{P} \times \mathcal{O})$, let $G_R = (V, E)$ be the corresponding co-bipartite graph. Then $A \times B$ is a closed set of R iff $S = V - (A \cup B)$ is a minimal separator of G_R.*

Proof: Let R be a binary relation, $G_R = (V, E)$ the corresponding co-bipartite graph.

1. Let $A \times B$ be a closed set of R, let $S = V - (A \cup B)$. We claim that for each $a \in A, b \in B$, S is a minimal ab-separator of G_R. First of all, S is an ab-separator: if there was an edge ab in G_R, then by definition (a, b) would not be in R. Next we will prove that S is minimal: suppose that it is not; by Property 3.2, w.l.o.g. there must be some vertex $x \in S$ such that x sees no vertex of B, which means that $\forall y \in B, (x, y) \in R$; $Ax \times B$ would be a rectangle of R, which contradicts the minimality of $A \times B$.

2. Conversely, let S be a minimal separator of G_R, let A and B be the connected components of $G(V - S)$. Since $\forall x \in A, \forall y \in B, xy \notin E$, we can conclude that $(x, y) \in R$, and that $A \times B$ is a rectangle of R. Suppose $A \times B$ fails to be maximal: w.l.o.g. $\exists x \in \mathcal{O} - B, \forall y \in A, (y, x) \in R$. In G_R, x will see no vertex of A, so by Property 3.2, S fails to be minimal.

□

Using Property 2.2, it is easy to show that, in addition, the following holds:

Property 3.4 *Let $A \times B$ be a closed set of R, let $S = V - (A \cup B)$; then $S = N(A) = N(B)$.*

Definition 3.5 *Let $A \times B$ be a closed set of relation R, let $S = V - (A \cup B)$. We will say that minimal separator S* ***represents*** *closed set $A \times B$.*

We can now reformulate Property 3.1 with Property 2.2 as the following:

Property 3.6 *Let $A \times B$ be a closed set of relation R, let S be the minimal separator representing $A \times B$. Then in G_R, $N(A) = N(B) = S$.*

Example 3.7

In Figure 3, $S = \{a, d, e, f, 3, 4, 5, 6\}$ is a minimal separator of graph G_R of Figure 2, separating $C_1 = \{b, c\}$ from $C_2 = \{1, 2\}$, and $bc \times 12$ is a closed set of R and an element of $\mathcal{L}(R)$. In G_R, $N(\{b, c\}) = N(\{1, 2\}) = S$.

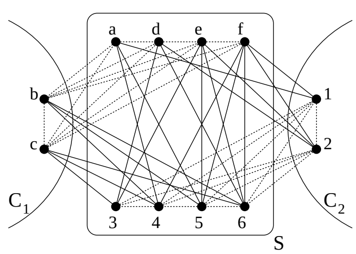


Figure 3: Separator $S = \{a, d, e, f, 3, 4, 5, 6\}$ of G_R.

Note that we can deduce from Theorem 3.3 that a co-bipartite graph may have an exponential number of minimal separators, since a concept lattice can have an exponential number of elements.

In [4], the notion of moplex was introduced as a general definition of the extremity of a graph (a moplex is defined as a maximal clique module, the neighborhood of which is a minimal separator of the graph; see [6] for an application of this concept). It is interesting to see that the moplexes of the underlying graph G_R correspond precisely to the non-trivial extremities of the lattices: its atoms and co-atoms:

Property 3.8 *Let R be a binary relationship, let $\mathcal{L}(R)$ be the corresponding concept lattice, let G_R be the underlying graph. If $A \times B$ is an atom of $\mathcal{L}(R)$ then A is a moplex of G_R; if $A \times B$ is a co-atom of $\mathcal{L}(R)$, then B is a moplex of G_R; there are no other moplexes in G_R.*

4 Selecting a sublattice by saturating a minimal separator

The process of saturating one minimal separator causes a number of other minimal separators to disappear from the graph; this process was first introduced by [17], is studied in [22] and its mechanism is described and used in [6]. In this Section, we will examine what happens to the lattice when a minimal separator of the underlying graph is saturated.

Definition 4.1 *([17]) Let S and T be two minimal separators of graph G; T is said to* **cross** *S if there are two different connected components C_1 and C_2 of $G(V-S)$ such that $T \cap C_1 \neq \emptyset$ and $T \cap C_2 \neq \emptyset$.*

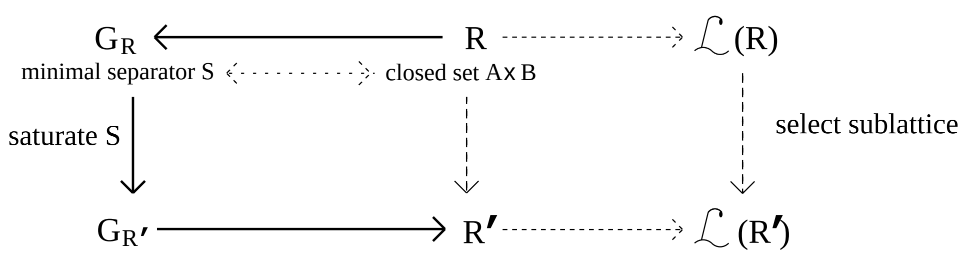
Property 4.2 *[22] Let G be a graph, let S be a minimal separator of G, let G_S denote the graph obtained from G by saturating S; then T is a minimal separator of G_S iff T is a minimal separator of G and T does not cross S in G.*

We will use this result on our underlying graph G_R: saturating a minimal separator S of G_R defines a new relation R', in which for each xy edge added to S the corresponding pair (x, y) is deleted from R.

Property 4.3 *Saturating a minimal separator of a co-bipartite graph results in a co-bipartite graph.*

Theorem 4.4 *Let R be a binary relation, and G_R the corresponding underlying co-bipartite graph. Let S be a minimal separator of G_R, representing the closed set $A \times B$ in the concept lattice $\mathcal{L}(R)$, let R' be the new relation obtained. Then the concept lattice $\mathcal{L}(R')$ is a sublattice of the original lattice $\mathcal{L}(R)$, and $\mathcal{L}(R')$ can be obtained from $\mathcal{L}(R)$ by removing all the elements which are not comparable to $A \times B$.*

We thus define a process which allows us to restrict a binary relation R to a smaller relation $R' \subset R$ such that $\mathcal{L}(R')$ is a sublattice of $\mathcal{L}(R)$. Note that restricting a relation in an arbitrary fashion will not, in general, yield a sublattice, and can even cause the lattice to be much bigger.



Example 4.5 Let us saturate separator $S = \{a, d, e, f, 3, 4, 5, 6\}$ of G_R, representing closed set $bc \times 12$. Edges $a3, a6, d4, d5, e4$ and $f3$ will be added.

New relation R' defined:

	a	b	c	d	e	f
1		x	x	x	x	
2	x	x	x			
3		x				
4						
5			x			
6						

Figure 4 gives the sublattice $\mathcal{L}(R')$ obtained.

Saturating S has caused closed sets $a \times 236, ab \times 23, abf \times 3, d \times 145$ and $cd \times 15$ to disappear from the lattice.

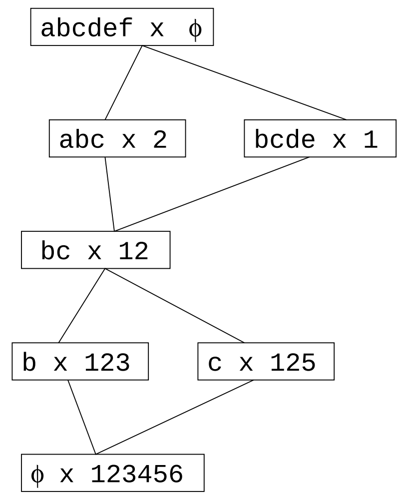


Figure 4: $\mathcal{L}(R')$

Remark 4.6 This process can be repeated on a second separator still represented in the sublattice obtained, and so forth. This may be very useful when the concept lattice is exponentially big, because it allows the user to examine only a part of it, which can be made as small as desirable by reiterating a minimal separator saturation step as many times as necessary.

We will conclude this section by discussing the minimal triangulations of G_R. We will first remark that it is easy to use the results in [19] and [21] to show that all the minimal triangulations of G_R are proper interval graphs, since G_R is both AT-free and claw-free.

Recent work has shown that minimal separators could be used to compute a minimal triangulation (see [17], [22], [2]).

Property 4.7 *[3] The process of repeatedly choosing a minimal separator which is not a clique, and saturating it, until all minimal separators are cliques, yields a minimal triangulation of the input graph in less than n steps.*

Property 4.8 *Computing a minimal triangulation of G_R by repeatedly saturating a non-clique minimal separator will result in a proper interval graph G" and a corresponding relation R" such that $\mathcal{L}(R")$ is a maximal chain of $\mathcal{L}(R)$.*

Remark 4.9 *A maximal chain has less than $min(|\mathcal{P}| + |\mathcal{O}|)$ elements, which ensures that the process described in Remark 4.6 can always result in a polynomial-sized sublattice.*

Property 4.10 *To each minimal triangulation of G_R uniquely corresponds a maximal chain of $\mathcal{L}(R)$ and vice-versa.*

5 Using clique separators to decompose a binary relation and its lattice

In [26], Tarjan introduces the decomposition by clique separators of a graph. This process is defined by repeatedly copying some clique separator S into each of the components it defines. This decomposition

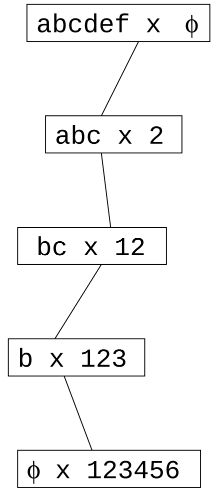


Figure 5: Lattice obtained by computing a minimal triangulation of graph G_R of Figure 2.

is proved to be unique and optimal when only clique minimal separators are used (see [18]), and can be described by the following general decomposition step:

General Decomposition Step 5.1 *Let G be a graph, let S be a clique minimal separator of G, defining components $C_1, C_2, ...C_k$. Replace G with $G_1 = G(C_1 \cup N(C_1))$ and $G_2 = G(C_2 \cup N(C_2))$.*

This decomposition has the remarkable property that it distributes the minimal separators into the subgraphs it defines.

Property 5.2 *Let G be a graph, let S be a clique minimal separator of G, let $\mathcal{S}(G)$ be the set of minimal separators of G. After a decomposition step by S, the elements of $\mathcal{S}(G) - \{S\}$ are partitioned into the subgraphs obtained.*

In the graph G_R defined by binary relation R, a clique minimal separator S defines only two components, $C_1 \subset \mathcal{P}$ and $C_2 \subset \mathcal{O}$, which are both full components, so that $N(C_1) = N(C_2) = S$. Moreover, the vertices of $S \cap \mathcal{P}$ are universal in graph $G_R(C_1 \cup S)$ and according to Property 2.3, they convey no information on minimal separators and can be removed from the graph. The vertices of $S \cap \mathcal{O}$ are likewise universal in $G_R(C_2 \cup S)$, and can be removed. We will define the following simplified decomposition step:

Decomposition Step 5.3 *Let G_R be the underlying graph of relation R, let S be a clique minimal separator of G_R, defining components $C_1 \subset \mathcal{P}$ and $C_2 \subset \mathcal{O}$. Replace G_R with $G_1 = G_R(C_1 \cup (S \cap \mathcal{O}))$ and $G_2 = G_R(C_2 \cup (S \cap \mathcal{P}))$.*

As a consequence of Property 5.2, computing the set of minimal separators of the original underlying graph G_R can be done separately on the smaller subgraphs defined by a decomposition step by a clique minimal separator: T_1 will be a minimal separator of G_1 iff $T_1 \cup (S \cap \mathcal{P})$ is a minimal separator of G_R, T_2 will be a minimal separator of G_2 iff $T_2 \cup (S \cap \mathcal{O})$ is a minimal separator of G_R. Thus the closed sets of R can be computed separately on the sub-relations defined.

Chances are the resulting sub-relations will be much smaller than the original one, and thus the queries on them much less costly.

Example 5.4 Let us use the relation R' of Example 4.5, the underlying graph G'_R of which contains clique minimal separator $S = \{a, d, e, f, 3, 4, 5, 6\}$. The corresponding lattice is given in Figure 4 in Section 4. S defines components $C_1 = \{b, c\}$ and $C_2 = \{1, 2\}$, thus representing the closed set $bc \times 12$. $S \cap \mathcal{O} = \{3, 4, 5, 6\}$ and $S \cap \mathcal{P} = \{a, d, e, f\}$.

A clique minimal separator decomposition step using S will yield $G_1 = G_R(C_1 \cup (S \cap \mathcal{O})) = G_{R'}(\{b, c, 3, 4, 5, 6\})$ and $G_2 = G_R(C_2 \cup (S \cap \mathcal{P})) = G_{R'}(\{a, d, e, f, 1, 2\})$.

The corresponding sub-relations obtained are the following:

	b	c
3	x	
4		
5		x
6		

	a	d	e	f
1		x	x	
2	x			

With a linear-time pass of G_1 it will become clear that vertices 4 and 6 have also become universal and can be removed. Figure 6 shows the very restricted graph G'_1 finally obtained. The minimal separators of G'_1 are $\{b, 3\}$ and $\{c, 5\}$, corresponding to closed sets $b \times 3$ and $c \times 5$. In the global graph, putting component $C_2 = \{1, 2\}$ back in will yield at no extra cost closed sets $b \times 123$ and $c \times 125$ of the original lattice. These are precisely the predecessors of $bc \times 12$.

In G_2, vertex f has become universal, and a linear-time pass using the algorithm of Hsu and Ma (see [15]) will show that vertices d and e now share the same neighborhood, and can be contracted without loss of information on the minimal separators of the graph.

The resulting graph G'_2 is also restricted to four vertices, and is shown in Figure 6. Its minimal separators are represented by $a \times 2$ and $de \times 1$, which, once we have put $C_1 = \{b, c\}$ back in, defines the closed sets $abc \times 2$ and $bcde \times 1$, which are the successors of $bc \times 12$.

Figure 6: Graphs G'_1 and G'_2 obtained after one decomposition step by clique minimal separator on G'_R.

Note that in a co-bipartite graph, the presence of a clique minimal separator can be tested for in linear time, and the decomposition can be computed in the same time.

Combining these results with those described in Section 4, we are able to choose a minimal separator representing a group of objects or properties, saturate it, and extract the sub-relation corresponding to the successors of this minimal separator.

6 Generating the closed sets

6.1 Generating and storing the closed sets

Recent work has been done on the efficient generation of the closed sets defined by a binary relation (see [11], [5], [20]), both when one wants to store all the closed sets, and when one simply wants to encounter all of them at least once.

In parallel, recent work has been done to generate all the minimal separators or all the minimal xy-separators of a graph (see [25], [16], [7], [24]).

As an illustration of the use that can be made of our new paradigm, we will show how we can easily match the complexity of generating and storing the closed sets obtained by [20], which is $O(n^2)$ per closed set, by using the work of [24], who claims a complexity of $O(n^2)$ time per minimal xy-separator to generate and store them.

Let us use our underlying graph G_R as described in Section 3, and add two simplicial vertices x and y: x is a neighbor of all vertices of $\mathcal{P}$, y of all vertices of $\mathcal{O}$.

The set of minimal separators of this new graph is exactly $\{\mathcal{P}\} \cup \{\mathcal{O}\} \cup \mathcal{S}(G_R)$, where $\mathcal{S}(G_R)$ is the set of minimal separators of G_R. Thus using Shen's algorithm and our graph representing relation R, we can easily generate and store all the closed sets of R in $O(n^2)$ time per closed set, noting that [24] claims a better space complexity than [20].

6.2 Generating the closed sets without storing them

We will now propose an algorithmic outline which computes the closed sets while storing only a polynomial number of them.

We will need to define the concept of **domination** in a graph:

Definition 6.1 *A vertex x is said to be **dominating** (or strongly dominating) in graph G if there is some vertex y such that $N(y) \subset N(x)$. We will say that a maximal clique module X is dominating if there is some vertex $y \notin X$ such that $\forall x \in X, N(y) \subset N(x)$. Conversely we will say that a vertex or maximal clique module is **non-dominating** if it is not dominating.*

For complexity considerations, we need to remark that a maximal clique module X of minimum degree is non-dominating, and that finding the set of vertices which dominate a given maximal clique module X can be done in linear time by checking for universal vertices in $N(X)$.

Our approach is based on the following property inspired from [5]:

Property 6.2 *Let G_R be the underlying graph of relation R, let $X \subset \mathcal{P}$ be a maximal clique module of G_R; then $N(X)$ represents an atom, with intent X, iff X is non-dominating.*

In Example 2.1, the set of maximal clique modules is exactly the set of vertices. Vertices e and f are dominating (e dominates d and f dominates both a and b). $N(a) = \{b, c, d, e, f, 1, 4, 5\}$, $N(b) = \{a, c, d, e, f, 4, 5, 6\}$, $N(c) = \{a, b, d, e, f, 3, 4, 6\}$, and $N(d) = \{a, b, c, e, f, 2, 3, 6\}$, which defines the atoms of $\mathcal{L}(R)$ as: $a \times 236, b \times 123, c \times 125$ and $d \times 145$.

Our strategy for finding the set of non-dominating maximal clique modules of G_R is the following:

1. Compute in linear time all the maximal clique modules of G_R and contract them.
2. Choose a vertex x of minimum degree.
3. Compute in linear time the set of vertices which dominate x.
4. Remove x and the vertices which dominate x from the graph and go back to Step 2.

This requires $O(m)$ time per non-dominated maximal clique module computed.

In order to compute the rest of the elements, we will then go on recursively to compute the sublattice of which each atom in turn is the bottom element, by saturating the corresponding minimal separator. This requires memorizing only the set of not yet processed non-dominating maximal clique modules and the corresponding graph's vertex set, which takes $O(n^3)$ space, as there are less than n atoms in any given graph, and as the height of a lattice cannot exceed n.

In the worse case, generating an element $A \times B$ requires $O(m|A|)$. The total time complexity required is in $O(nm)$ per element, which matches the corresponding best time algorithm for generating the elements without storing them, proposed by Ganter (see [11].

7 Conclusion

Though specific problems such as minimizing the number of times a database is accessed remain to be translated in terms of graph separators, we have presented a new approach to answering queries on the concept lattice of a binary relation, which uses a rapidly growing toolbox: the theory of minimal separation in undirected graphs.

We can expect that this approach will create a bridge between the two fields of concept lattice theory and undirected graph theory, and yield new results in both fields.

Moreover, we feel that since the minimal separators of a graph seem to describe the structure of the graph, we have contributed to show a strong semantic aspect behind the closed sets of a binary relation.

Several open questions arise from the issues discussed in this paper.

It is not known whether the set of non-dominant vertices can be computed in less than linear time per vertex, but improving this would also improve the complexity of the algorithmic process described in Subsection 6.2.

Likewise, efficiently computing the set of minimal separators which cross a given minimal separator S would result in a better generation algorithm for closed sets.

We have illustrated the use of clique minimal separator decomposition, but other minimal separator-preserving decompositions would directly yield decompositions of a binary relation and of the associated lattice. Conversely, other known decompositions of binary relations might lead to new hole and anti-hole preserving graph decompositions, an important open question in the context of perfect graphs.

Finally, it would be interesting to characterize the binary relations which define a polynomial number of closed sets, or the graphs which have a polynomial number of minimal separators; this might help the users of databases to maintain manageable binary relations.

Acknowledgment

The authors thank Marianne Huchard for many fruitful discussions on concept lattices.

References

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

[2] A. Berry. A Wide-Range Efficient Algorithm for Minimal Triangulation. *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99)*, Baltimore, Jan. 1999, pp. 860–861.

[3] A. Berry. Désarticulation d'un graphe. *PhD Dissertation, Université Montpellier II*, 1998.

[4] A. Berry and J.-P. Bordat. Separability generalizes Dirac's theorem. *Discrete Applied Mathematics*, 84:43–53, 1998.

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

[6] A. Berry and J.-P. Bordat. Asteroidal triples of moplexes. *Discrete Applied Mathematics*, 111: 219-229, 2001.

[7] 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.

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

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

[10] G.A. Dirac. On rigid circuit graphs. *Anh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.

[11] B. Ganter. Two basic algorithms in concept analysis. Preprint 831, Technische Hochschule Darmstadt, 1984.

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

[13] A. Guénoche. Construction du treillis de Galois d'une relation binaire. *Mathématiques, Informatique et Sciences Humaines*, 121:23–34, 1993.

[14] 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).

[15] W.-L. Hsu and T.-H. Ma. Substitution decomposition on chordal graphs and its applications. *SIAM J. Comput*, 28:1004-1020, 1999.

[16] T. Kloks and D. Kratsch. Listing all minimal separators of a graph. *SIAM Journal on Computing*, 27:605–613, 1998.

[17] T. Kloks, D. Kratsch and J. Spinrad. Treewidth and pathwidth of cocomparability graphs of bounded dimension. *Res. Rep. 93-46, Eindhoven University of Technology*, 1993.

[18] H.-G. Leimer. Optimal decomposition by clique separators. *Discrete Mathematics*, 113:99–123, 1993.

[19] R.H. Möhring. Triangulating graphs without asteroidal triples. *Discrete Applied Mathematics*, 64:281–287, 1996.

[20] L. Nourine and O. Raynaud. A Fast Algorithm for building Lattices. *Information Processing Letters*, 71:199–204, 1999.

[21] A. Parra. Structural and algorithmic aspects of chordal graph embeddings. *PhD Dissertation, Technische Universität Berlin*, 1996.

[22] A. Parra and P. Scheffler. How to use the minimal separators of a graph for its chordal triangulation. *Proceedings of the 22nd International Colloquium on Automata, Languages and Programming (ICALP '95), Lecture Notes in Computer Science*, 944:123–134, 1995.

[23] D. Rose, R.E. Tarjan, and G. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journ. Comput*, 5:146–160, 1976.

[24] H. Shen. Separators are as simple as cutsets. *Asian Computer Science Conference, Purket, Thailand, December 10-13, 1999, Lecture Notes in Computer Science*, 172:347–358.

[25] H. Sheng and W. Liang. Efficient enumeration of all minimal separators in a graph. *Theoretical Computer Science*, 180: 169–180, 1997.

[26] R.E. Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55:221–232, 1985.

[27] I. Todinca. Aspects algorithmiques des triangulations minimales des graphes. *PhD Dissertation, Ecole Normale Supérieure Lyon*, 1999.

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