Generalized Domination in Closure Systems

Anne Berry [a] Eric SanJuan [b] Alain Sigayret [c]

[a] *LIMOS (CNRS UMR 6158), Université Blaise Pascal, Ensemble Scientifique des Cézeaux, 63173 Aubière cedex, France.* `berry@isima.fr`

[b] *IUT de Metz, LITA (EA 3097), Université de Metz, Ile du Saulcy, 57045 Metz cedex 1, France.* `eric.sanjuan@iut.univ-metz.fr`

[c] *LIMOS (CNRS UMR 6158), Université Blaise Pascal, Ensemble Scientifique des Cézeaux, 63173 Aubière cedex, France.* `sigayret@isima.fr`

Abstract

In the context of extracting maximal item sets and association rules from a binary data base, the graph-theoretic notion of *domination* was recently used to characterize the neighborhood of a concept in the corresponding lattice.

In this paper, we show that the notion of domination can in fact be extended to any closure operator on a finite universe and be efficiently encoded into propositional Horn functions. This generalization enables us to endow notions and algorithms related to Formal Concept Analysis with Horn minimization and minimal covers of functional dependencies in Relational Databases.

1 Introduction

The massive amounts of data which are currently being accumulated worldwide make it important to find fast algorithms to sift through the databases, or new techniques to avoid scanning the whole base. One of the approaches is to factorize the data, in order to minimise the size occupied by relevant information as well as the time required for searches.

In this data mining context, recent works by Wille (32) and Ganter (17) use Formal Concept Analysis to investigate *concepts*, which are maximal rectangles of a binary relation and correspond to a maximal factorization of item sets; this is used in a combinatorial approach for extracting patterns from a database. Concept lattices stem from Galois lattices, which have been studied for a long time (7), for example in the context of Social Sciences (1), but Wille and Ganter's work has introduced new perspectives and applications. The use of concept lattices is rapidly emerging in many areas related to Artificial Intelligence and Data Mining, such as Database Management (see e.g. (23)), organization of object hierarchies (see e.g. (22)), machine learning (see e.g. (26)) and frequent set generation (see e.g. (19)).

Equally important, the related problem of rule generation, which corresponds to finding functional dependencies in databases, is of major importance in data analysis, for wide-spread applications such as behavioral prediction, artificial intelligence, modelization of genomic phenomena, and so forth. Recent work has been done by Maier (25) in the theory of Relational Databases to define a minimal set of functional dependencies and simultaneously by Guigues and Duquenne (20) in Formal Concept Analysis to define a canonical basis of exact association rules. Mathematical investigation has shown that concepts as well as rules are associated with several mutually inclusive closure lattices. These lattices are potentially of exponential size, and as there may be even more rules than there are concepts, efficient algorithmic techniques are actively being sought to deal with these problems. An interesting breakthrough was initiated by Bordat (8) when remarking that in order to generate the neighbors of a given concept in the lattice, no information on other concepts is required. However, state of the art rule generation algorithms require, in order to generate one rule, information on *all* previously generated rules, a set which it is not always feasible to handle.

Our general purpose in this mathematical-oriented paper is to study various relationships between different formal approaches, in view of using mathematical and/or algorithmic results which stem from various fields of discrete mathematics. Several approaches have been proposed very recently in this direction. The Rough Set approach explored the relationships between functional dependencies and mining of prime implicants of discernability functions. Discernability functions are based on approximation operators which are special cases of disjunctive closure operators. SanJuan in (31) used Heyting algebras to modelize and generalize this concept of approximation operators. In the finite case, Bioch and Ibaraki in (6) use generalized monotone Boolean functions for the same purpose. However, all the algebraic structures defined to deal with approximation operators are based on distributive lattices. On the other hand, functional dependencies induce general (i.e. not necessarily disjunctive) closure operators and arbitrary lattices, as the class of Horn functions in Logical Analysis of Data and Galois Lattices in Formal Concept Analysis. The Rough Sets and Formal Concept Analysis approaches are compared in (30).

In this paper, we focus on general finite closure operators and their underlying finite lattices. Berry and Sigayret in (3) proposed a representation of a concept lattice by a graph, where the graph-related notions of domination and maxmods were used, as well as that of minimal separation. Bordat's results (8) were explained and extended, the cover of a concept characterized using only local information. This work established a relationship between graph theory and concept lattices, and was rewarded by immediate algorithmic results in terms of concept generation analysis at least as good as that of the best such algorithms (5).

In this paper, we show that we can extend the notion of domination to any closure operator defined on a finite universe U. This develops into new interesting algorithmic approaches for generating lattices related to implicational systems or canonical covers of functional dependencies in Relational Database.

The paper is organized as follows: Section 2 gives preliminaries on Galois and concept lattices, Section 3 explains previous work on the relationship between graphs and lattices, Section 4 extends the corresponding results to a general closure system, Section 5 interprets our results from Section 4 in a logical-based fashion, and Section 6 deals with the logical aspects of rule generation.

2 Preliminaries

We will first give some preliminaries on binary relations and the associated lattice. In this field, there are two main approaches with many notions in common: Galois lattices and concept lattices. We will present both aspects in this preliminary section, although in the rest of the paper we will refer to concept lattices.

2.1 Maximal rectangles, contexts and concepts

Given a finite set $\mathcal{P}$ of "properties" or "attributes" (which we will denote by lowercase letters) and a finite set $\mathcal{O}$ of "objects" or "tuples" (which we will denote by numbers), we will 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 subset X of $\mathcal{P}$ and a subset X' of $\mathcal{O}$, the set $R \cap (X \times X')$ is a *subrelation* of R, which we will denote by $R(X, X')$.

Definition 2.1 *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 subproduct $A \times B \subseteq R$ 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*.*

Example 2.2 *Binary relation R for our running example:*

Property set:

$\mathcal{P} = \{a, b, c, d, e, f, g, h\}$

Object set:

$\mathcal{O} = \{1, 2, 3, 4, 5, 6\}$

$R \subseteq \mathcal{P} \times \mathcal{R}$

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

In this relation, $ah \times 236$ and $bc \times 125$ are maximal rectangles (or concepts) of R. bc is the intent *of rectangle $bc \times 125$, and 125 its* extent.

2.2 Concept lattices

A lattice is a partially ordered set in which every pair $\{X, Y\}$ of elements has both a lowest upper bound (denoted by $join(X, Y)$) and a greatest lower bound, (denoted by $meet(X, Y)$). We represent a lattice by the Hasse diagram of the partial ordering on the elements: transitivity and reflexivity edges are omitted. The reader is referred to the classical work of (7) for basic results on lattices. An element Y is said to *cover* an element X if $X < Y$ and there is no intermediate element Z such that $X < Z < Y$. The set of elements which cover an element X is called the *cover* of X.

Given a context $C = (\mathcal{P}, \mathcal{O}, R)$, the concepts of C, ordered by inclusion on the intents, define a lattice, called a *Galois lattice* or *concept lattice*, which is usually represented with an ordering on the intents from bottom to top. We will denote this lattice by $\mathcal{L}(R)$. An element $B \times B'$ is said to be a *descendant* of element $A \times A'$ if $A \subset B$. $B \times B'$ is said to *cover* (to be a successor of) $A \times A'$ if $A \subset B$ and there is no element $C \times C'$ such that $A \subset C \subset B$.

This lattice may be of exponential size, as it may scan the power set of $\mathcal{P}$ or of $\mathcal{O}$. Such a lattice, sometimes referred to as a complete lattice, has a smallest element, called the *bottom element*, and a greatest element, called the *top element*. The elements which cover the bottom element are called *atoms*.

This lattice has special properties:

Property 2.3 *Each element X is the bottom element of a sublattice which contains its descendants.*

Property 2.4 *(5) For each element $x \in \mathcal{P}$, the subset of elements containing* *x defines a sublattice of $\mathcal{L}(R)$; we will call the bottom element of this sublattice the* introducer *of x.*

Example 2.5 *The lattice $\mathcal{L}(R)$ of relation R in Example 2.2 is given in Figure 1. In elements which are introducers of a property, this property is represented in bold. The atoms of $\mathcal{L}(R)$ are: $ah \times 236$, $b \times 1235$ and $d \times 145$. The introducer of c is element $bc \times 125$. The sublattice defined by the elements containing c is given in Figure 2.*

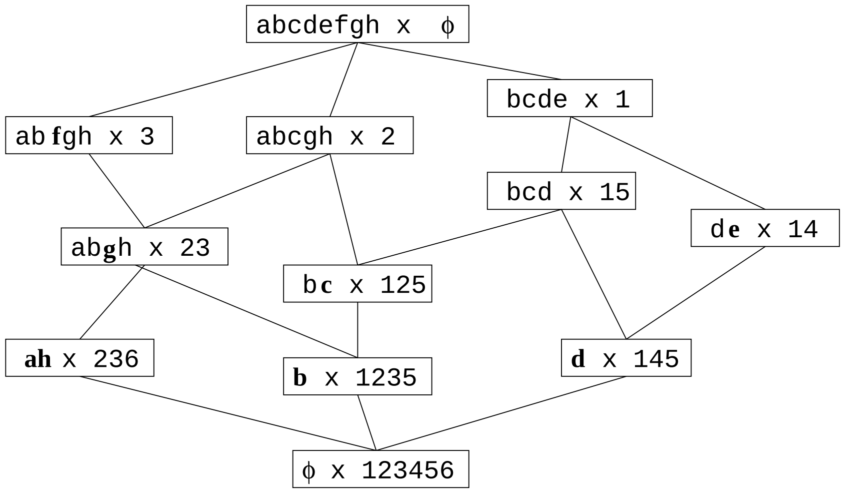

Fig. 1. Concept lattice $\mathcal{L}(R)$ of relation R of Example 2.2. In elements which are property introducers, the introduced properties are represented in bold.

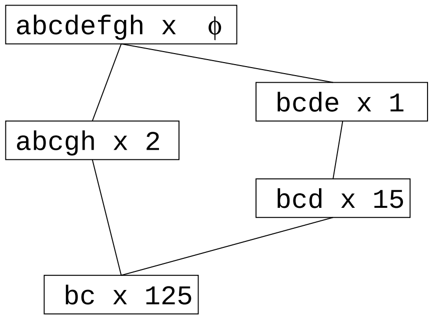

Fig. 2. Sublattice of the elements containing c of lattice $\mathcal{L}(R)$ of Figure 1.

3 Relationships between domination and concepts

3.1 An underlying graph

Our approach to handling a concept lattice (see (3)) is to encode the relation by 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 ($xy \in E$ if $x, y \in \mathcal{P}$ or if $x, y \in \mathcal{O}$).
(2) external edges: for $x \in \mathcal{P}$ and $y \in \mathcal{O}$, $xy \in E$ iff $(x, y) \notin R$.

Example 3.1 *Figure 3 gives the graph which corresponds to the relation from Example 2.2.*

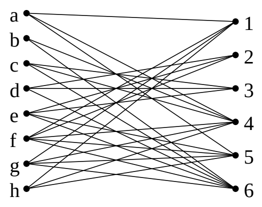

Fig. 3. Graph G_R coding relation R from Example 2.2.

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 A' if and only if $A \times A'$ is a concept defined by relation R (see (3) for details on minimal separators and this relationship). This leads to interesting results, because much recent work on graphs has been done on minimal separation, with results on efficient separator generation and on separator decomposition.

Although in this paper we do not need to go into details about these graph results, we will use some vocabulary such as 'neighborhood', 'domination' and 'maxmods' which stems from graph theory; we thus usually denoted by $N^+(x)$ the external neighborhood of vertex x in graph G_R: for $x \in \mathcal{P}, N^+(x) = \{y \in \mathcal{O} | (x, y) \notin R\}$, and for $x \in \mathcal{O}, N^+(x) = \{y \in \mathcal{P} | (y, x) \notin R\}$.

In this paper, we will only need to use the neighborhood of the complement, thus, instead of the graph notation $\overline{N}^+(X)$, we will use notation $R[X]$:

Definition 3.2 *Given a context $(\mathcal{P}, \mathcal{O}, R)$, for any subset X of $\mathcal{P}$ or $\mathcal{O}$, we will define:*

- $R[X] = \{y \in \mathcal{O} : \forall x \in X, (x, y) \in R\}$ *if* $X \subseteq \mathcal{P}$,
- $R[X] = \{y \in \mathcal{P} : \forall x \in X, (y, x) \in R\}$ *if* $X \subseteq \mathcal{O}$.

We will denote $R[R[X]]$ by $R^2[X]$. $R[\{x\}]$ will be denoted $R[x]$ for short.

Using this notation, we can describe the maximal rectangles as: $R^2[X] \times R[X]$, for $X \subseteq \mathcal{P}$.

A concept $A \times A'$ is uniquely defined by its intent A, since $A' = R[A]$; in the rest of this section, we will accordingly refer only to intents, i.e. to subsets of $\mathcal{P}$.

One of the related graph notions which turns out to be of primary importance for the study of concept lattices is that of vertex domination: in a graph, a vertex x is said to dominate another vertex y if $N^+(y) \subset N^+(x)$. In this paper, we will transpose this definition using notation $R[\ ]$:

Definition 3.3 *Let $(\mathcal{P}, \mathcal{O}, R)$ be a context, let x, y be in $\mathcal{P}$; we say that x* dominates *y if $R[x] \subseteq R[y]$*

Example 3.4 *In our example, $R[\{b, c\}] = \{1, 2, 5\}$.*

In (3) domination is used to define a pre-order on $\mathcal{P}$. With this pre-order are associated equivalence classes called maxmods (a short for the graph term 'maximal clique module'), which led to the quotient order of this pre-order defining domination between maxmods:

Definition 3.5 *(3) Let $(\mathcal{P}, \mathcal{O}, R)$ be a context; we will say that $X \subseteq \mathcal{P}$ is a* maxmod *of R if $\forall x, y \in X, R[x] = R[y]$ and X is maximal for this property. We say that a maxmod X* dominates *a maxmod $Y \neq X$ if $R[X] \subset R[Y]$.*

Property 3.6 *(3) Let X and Y be maxmods; then $X \subset Y$ iff Y dominates X. Domination between maxmods defines a partial order.*

Example 3.7 *In Example 2.2, the maxmods are: $\{a, h\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$, $\{f\}$ and $\{g\}$. $\{b\}$ is a non-dominating maxmod; $\{c\}$ dominates $\{b\}$; $\{d\}$ is non-dominating; $\{e\}$ dominates $\{d\}$; $\{a, h\}$ is non-dominating; $\{g\}$ dominates $\{a, h\}$ and $\{b\}$; $\{f\}$ dominates $\{a, h\}$, $\{b\}$ and $\{g\}$.*

The maxmods can be computed in $O(|\mathcal{P} \cup \mathcal{O}| - |R|)$ time (see (3)).

One of the ways of computing the partition into maxmods is to use a partition refinement technique, based on a famous graph algorithm called LexBFS (29) which was originally designed to recognize chordal graphs: start with $\mathcal{P}$ and repeatedly choose an object i, and use $R[i]$ to split the classes of the current partition into neighbors and non-neighbors of i; if at each step the subclass of elements in $R[i]$ is put to the left of the subclass of non-elements, then at the end, a partition into maxmods is obtained, with the interesting property that a given maxmod X can dominate only maxmods which lay to the left of X in the partition. This process is described in detail in (5).

Example 3.8 *Figure 4 illustrates the partition refinement based on LexBFS from Example 2.2.*
The ordered partition into maxmods obtained is ($\{b\}$, $\{c\}$, $\{d\}$ $\{e\}$ $\{ah\}$ $\{g\}$ $\{f\}$). With Example 3.7 we can verify that a maxmod dominates no maxmod which is after it in this list.

Properties can be used in a similar fashion to split the partition, this time using the intent of the introducer corresponding to a given property, as shown in Figure 5.

$$abcdefgh$$
$$\downarrow R[1] = \{b, c, d, e\}$$
$$bcde \mid afgh$$
$$\downarrow R[2] = \{a, b, c, g, h\}$$
$$bc \mid de \mid agh \mid f$$
$$\downarrow R[3] = \{a, b, f, g, h\}$$
$$b \mid c \mid de \mid agh \mid f$$
$$\downarrow R[4] = \{d, e\}$$
$$b \mid c \mid de \mid agh \mid f$$
$$\downarrow R[5] = \{b, c, d\}$$
$$b \mid c \mid d \mid e \mid agh \mid f$$
$$\downarrow R[6] = \{a, h\}$$
$$b \mid c \mid d \mid e \mid ah \mid g \mid f$$

Fig. 4. Partition refinement based on LexBFS (see Example 3.8).

The maxmods turn out to be closely related to the introducers: the partial ordering on maxmods has the same structure as the suborder defined by the concept lattice restricted to the introducers.

Property 3.9 *(4) A concept with intent $A \subseteq \mathcal{P}$ is an introducer iff there is a maxmod $X \subseteq \mathcal{P}$ such that $X \subseteq A$ and $A - X$ is the union of all the maxmods dominated by X.*

A similar result holds for extents and object maxmods.

Example 3.10 *Figure 6 gives the domination ordering on maxmods corresponding to the relation of Example 2.2. Concept $abgh \times 23$ is the introducer of g. $\{g\}$ is a maxmod and dominates $\{a, h\}$ and $\{b\}$. Concept $abfgh \times 3$ is the introducer of f. $\{f\}$ is a maxmod and dominates $\{g\}$, $\{a, h\}$, and $\{b\}$.*

$$abcdefgh$$

$$\downarrow a : \{a, h\}$$

$$ah \mid bcdefgh$$

$$\downarrow b : \{b\}$$

$$ah \mid b \mid cdefg$$

$$\downarrow c : \{b, c\}$$

$$ah \mid b \mid c \mid defg$$

$$\downarrow d : \{d\}$$

$$ah \mid b \mid c \mid d \mid efg$$

$$\downarrow e : \{d, e\}$$

$$ah \mid b \mid c \mid d \mid e \mid fg$$

$$\downarrow f : \{a, b, f, g, h\}$$

$$ah \mid b \mid c \mid d \mid e \mid fg$$

$$\downarrow g : \{a, b, g, h\}$$

$$ah \mid b \mid c \mid d \mid e \mid g \mid f$$

$$\downarrow h : \{a, h\}$$

$$ah \mid b \mid c \mid d \mid e \mid g \mid f$$

Fig. 5. Partition refinement based on the intents of the introducers (see Example 3.8).

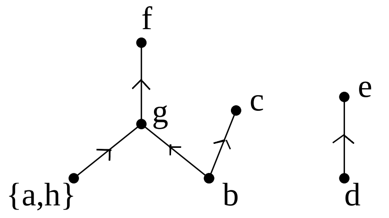

Fig. 6. The domination ordering on maxmods for Relation R from Example 2.2.

This ordering has interesting applications: when the number of elements of the lattice is exponential, several authors ((18), (13), (22)) have found it useful to further simplify this lattice into a *Galois subhierarchy*, by using the suborder induced by the introducers (using introducers for both properties and objects), which has a polynomial number of elements; its properties have been investigated in several applications such as UML representations and handling object-oriented hierarchies. In (4), Berry and Sigayret show how to efficiently maintain such a subhierarchy by decomposing it into two partially ordered sets of introducers: one for introducers of properties and one for introducers of objects.

Bordat in (8) used the fact that, for any concept $A \times A'$, the sublattice of which $A \times A'$ is the bottom element is isomorphic to the concept lattice of a subrelation of R:

Theorem 3.11 *Let $(\mathcal{P}, \mathcal{O}, R)$ be a context, let $A \times A'$ be a concept. The elements which contain A in their intent define a sublattice of $\mathcal{L}(R)$ which is isomorphic to the lattice of relation $R(\mathcal{P}-A, A')$. We will refer to $R(\mathcal{P}-A, A')$ as* Bordat's subrelation related to *A.*

We use the notion of maxmod and the results from (8) to present the following theorem, which uses the Bordat's subrelation to define the cover of an arbitrary element of the lattice.

Theorem 3.12 *(3) Concept $B \times B'$ covers a concept $A \times A'$ iff $B - A$ is a non-dominating maxmod in $R((\mathcal{P} - A), R[A])$.*

This is algorithmically interesting, because it enables a local approach. However, when generating all the concepts, the idea that domination is inherited as one moves up in the lattice avoids a complete re-computation of the domination order, thus yielding an interesting time and space complexity (5):

Property 3.13 *Let A and B be concepts, with $A \subset B$, let x and y be properties which are not in B. Then if x dominates y in Bordat's subrelation related to element A, x also dominates y in Bordat's subrelation related to element B.*

4 General closure systems

In the previous section, we have discussed various aspects of a concept lattice. However, in several applications, other lattices are used, for example for dealing with functional dependencies in databases; another such application is rule generation, which, as we will see in Section 6, is associated with two different superlattices of the concept lattice.

Thus, a more general definition of lattices built on a family of subsets of properties, attributes, or, more generally, on a family of subsets of any finite universe U is needed. This corresponds to *closure systems*, which we will discuss in this section. We will see that the notion of domination between maxmods can be usefully extended to this more general case.

Definition 4.1 *A unary operator φ on a universe U is called a* closure operator *on U if for $A, B \subseteq U$:*

(1) $A \subseteq \varphi(A)$ *(extensivity)*

(2) $\varphi(\varphi(A)) = \varphi(A)$ *(idempotence)*

(3) if $A \subseteq B$*, then* $\varphi(A) \subseteq \varphi(B)$ *(isotony)*

A subset *A of U is said to be* closed *if $\varphi(A) = A$.*

Property 4.2 *Let $(\mathcal{P}, \mathcal{O}, R)$ be a context. R^2 is a closure operator on $\mathcal{P}$.*

Definition 4.3 *Given a family $\mathcal{E}$ of subsets of a finite set U, the* closure by intersection *$\mathcal{E}^*$ of $\mathcal{E}$ is defined inductively as follows:*

(1) U and every element of $\mathcal{E}$ are in $\mathcal{E}^*$.
(2) If X and Y are in $\mathcal{E}^*$, then $X \cap Y$ is in $\mathcal{E}^*$.

Example 4.4 *Let $U = \{a, b, c, d, e, f\}$, let $\mathcal{E} = \{\{a, c, d, e, f\}, \{b, d, e, f\}, \{a, c, d\}, \{a, c, e\}\}$. Then $\mathcal{E}^* = \mathcal{E} \cup \{U, \{d, e, f\}, \{d\}, \{e\}, \{a, c\}, \emptyset\}$.*

Definition 4.5 *A family $\mathcal{F}$ of subsets of a finite set U is said to be a* closure system *or a* Moore family *if: $\forall\, \mathcal{E} \subseteq \mathcal{F}$, $(\bigcap_{X \in \mathcal{E}} X) \in \mathcal{F}$.*

Each closure operator φ on U can be associated with the family $\mathcal{F}_\varphi = \{\varphi(A) : A \subseteq U\}$, which is a closure system such that for any $A \subseteq U$, $\varphi(A)$ is the smallest element of $\mathcal{F}_\varphi$ which includes A. Conversely, each closure system $\mathcal{F} \subseteq 2^U$ (where 2^U is the power set of U) can be associated with a closure operator φ defined for any $X \in U$ by $\varphi(X) = \bigcap_{Y \in \mathcal{F}, Y \supseteq X} Y$

Property 4.6 *If $\mathcal{F}$ is a closure system, then $(\mathcal{F}, \subseteq)$ is a lattice with top element U.*

Example 4.7 *The lattice associated with the closure system given in Example 4.4 is given in Figure 7.*

From this lattice stem the notions of cover and atom:

Definition 4.8 *Given a closure operator φ on U, and the corresponding closure system $\mathcal{F}_\varphi$.*
A closed set B is said to cover *a closed set A in $\mathcal{F}_\varphi$ if $A \subset B$ and there is no closed set C such that $A \subset C \subset B$.*
B is said to be an atom *of $\mathcal{F}_\varphi$ if it covers the closed set $\varphi(\emptyset)$.*

Thus, B covers A if for any $X \subseteq U$, $(A \subset X \subset B) \Rightarrow \varphi(X) = B$.

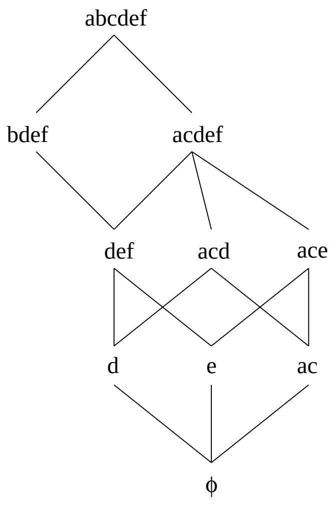

Fig. 7. Lattice associated with the closure system defined in Example 4.4.

The following property links the notion of covers and atoms and will be used to generalize Theorem 3.12 and recursive approaches to generating lattices of closed sets.

Property 4.9 *Let φ be a closure operator on a finite set U. For every closed set $A \in \mathcal{F}_\varphi$, the map $\varphi_A : X \subseteq U \mapsto \varphi(X \cup A)$ is a closure operator on U such that for every closed set $B \in \mathcal{F}_\varphi$, B covers A iff B is an atom of $\mathcal{F}_{\varphi_A}$.*

PROOF. φ_A is clearly a closure operator on U: for every $X \subseteq U$,

(1) $X \subseteq \varphi_A(X)$, since $X \subseteq X \cup A \subseteq \varphi(X \cup A) = \varphi_A(X)$,
(2) if $X \subseteq Y$ then $\varphi_A(X) \subseteq \varphi_A(Y)$, since $\varphi_A(X) = \varphi(X \cup A) \subseteq \varphi(Y \cup A) = \varphi_A(Y)$,
(3) $\varphi_A(\varphi_A(X)) = \varphi_A$, since:

$$\begin{aligned}\varphi_A(\varphi_A(X)) &= \varphi(\varphi(X \cup A) \cup A)\\ &= \varphi(\varphi(X \cup A)) \text{ since } A \subseteq \varphi(X \cup A)\\ &= \varphi_A(X)\end{aligned}$$

Let us now show that B covers A iff B is an atom of $\mathcal{F}_{\varphi_A}$.

If B covers A in $\mathcal{F}_\varphi$, then $A \subset B$. Moreover, for any $X \subseteq B, X \neq \emptyset$, we have $\varphi_A(X) = \varphi(X \cup A) = B$, since $A \subset X \cup A \subseteq B$. Thus, B covers $\varphi_A(\emptyset)$ and consequently is an atom of $\mathcal{F}_{\varphi_A}$.

Conversely, if B is an atom of $\mathcal{F}_{\varphi_A}$ then B covers $\varphi_A(\emptyset) = \varphi(A)$.

We will now explain how we can extend the results from Section 3 from R^2 to an arbitrary closure operator.

We have seen in Section 3.2 that by virtue of Theorem 3.12, it is possible to compute the cover of any element of the lattice by simply restricting the relation and by computing the corresponding minimal elements of the order defined by the maxmods. This is based on Bordat's subrelation, but since in this more general context no relation is given to work with, we will need to define domination as related to a given closed set A.

Definition 4.10 *Given a closure operator φ and a closed set A, we will define a binary relation on $U - A$, which we will denote by $dom_\varphi(A)$, by setting for any $x, y \in U - A$:*

$$(x, y) \in dom_\varphi(A) \Longleftrightarrow y \in \varphi(A \cup \{x\})$$

We will say that x dominates *y in A.*

This extension of the notion of domination as studied in Section 3 preserves many of the original results: for any closed set A, $dom_\varphi(A)$ is a pre-order (i.e. $dom_\varphi(A)$ is reflexive and transitive). As a result, $U - A$ can be partitioned into equivalence classes which we will call *maxmods*; this results in a quotient order, which is a partial order on the maxmods.

Clearly, a subset $M \subseteq U - A$ is a *maxmod* of $dom_\varphi(A)$ if and only if it is a maximal set such that for any $x \in M$, $M \subseteq \varphi(A \cup \{x\})$.

The notion of domination is naturally extended to maxmods:

Definition 4.11 *We denote by $Dom_\varphi(A)$ the binary relation defined on the maxmods of $dom_\varphi(A)$: for $X, Y \subseteq U - A$*

$$(X, Y) \in Dom_\varphi(A) \Longleftrightarrow (\exists x \in X)(\exists y \in Y)\ (x, y) \in dom_\varphi(A)$$

We will say that maxmod X dominates *maxmod Y.*

Let us remark that the existential quantifiers in previous definition can be replaced by universal quantifiers:

$$(X, Y) \in Dom_\varphi(A) \Longleftrightarrow (\forall x \in X)(\forall y \in Y)\ (x, y) \in dom_\varphi(A)$$

In the rest of this paper the relation $dom_\varphi(\varphi(\emptyset))$ ($Dom_\varphi(\varphi(\emptyset))$ resp.) will be denoted in short by dom_φ (Dom_φ resp.).

The notion of introducer also extends to closure systems:

Definition 4.12 *For $x \in U$, $\varphi(\{x\})$ is called the* introducer *of x.*

It is easy to see that this definition, when applied to R^2, is the same as the one given in Section 3. In fact, for the lattices defined by closure systems, for each element $x \in U$, the subset of elements containing x defines a sublattice, the bottom element of which is called the *introducer* of x. This can be extended to defining the introducer of a maxmod:

Property 4.13 *Each maxmod X of dom_φ defines an introducer $\varphi(X)$ which is :*
$\varphi(X) = \bigcup\{Y : (X, Y) \in Dom_\varphi\}$.

PROOF. If $M \subseteq U$ is a maxmod of dom_φ, then for any $x \in M$, $M \subseteq \varphi(\{x\})$. Since φ is an isotone operator, $\varphi(\{x\}) \subseteq \varphi(M) \subseteq \varphi(\{x\})$; thus $\varphi(M)$ is the introducer of x.

Moreover, for any $y \in U$ such that $(x, y) \in dom_\varphi$, $\varphi(\{y\}) \subseteq \varphi(\{x\}) = \varphi(M)$. This shows that: $\bigcup\{Y : (M, Y) \in Dom_\varphi\} \subseteq \varphi(M)$. The converse inclusion follows by minimality of M.

Example 4.14 *Let us consider the closure system from Example 4.4; let us compute the domination relation with respect to the bottom element $\emptyset$ of the associated lattice shown in Figure 7. $\varphi(a) = \{a, c\}$; $\varphi(b) = \{b, d, e, f\}$; $\varphi(c) = \{a, c\}$; $\varphi(d) = \{d\}$; $\varphi(e) = \{e\}$; $\varphi(f) = \{d, e, f\}$. By definition, $(x, y) \in dom_\varphi(A)$ iff $y \in \varphi(A \cup \{x\})$, here with $A = \emptyset$. The elements of dom_φ are: (a, a), (b, b), (c, c), (d, d), (e, e), (f, f), (a, c), (b, d), (b, e), (b, f), (c, a), (f, d), (f, e). So a dominates c and c dominates a; b dominates f, and f dominates d and e. Maxmods: $\{a, c\}, \{b\}, \{d\}, \{e\}, \{f\}$. Non-dominating maxmods (which are thus atoms): $\{a, c\}, \{d\}$ and $\{e\}$. Other maxmods (which also define introducers): $\{f\}$ which defines introducer $\{d, e, f\}$, and $\{b\}$, which defines introducer $\{b, d, e, f\}$.*

In a fashion quite similar to that described in Section 3, the partition into maxmods can be computed by using partition refinement, as illustrated in the following example.

Example 4.15 *Using the closure system from Example 4.4, Figure 8 gives the details of the computation of the partition into maxmods related to closed set $\emptyset$.*

The result is: $\{a, c, d\}$ *is non-dominating,* $\{d\}$ *and* $\{e\}$ *are non-dominating,* $\{f\}$ *dominates* $\{d\}$ *and* $\{e\}$*,*$\{b\}$ *dominates* $\{f\}$*,* $\{d\}$ *and* $\{e\}$*.*

To compute the partition related to a closed set A*, one would replace* $\varphi(x)$ *by* $\varphi(\{x\} \cup A)$ *and use each of the elements of* U *which is not in* A*.*

$$\begin{array}{l}
abcdef \\
\qquad \downarrow \varphi(a) = \{a, c\} \\
ac \mid bdef \\
\qquad \downarrow \varphi(b) = \{b, d, e, f\} \\
ac \mid bdef \\
\qquad \downarrow \varphi(c) = \{a, c\} \\
ac \mid bdef \\
\qquad \downarrow \varphi(d) = \{d\} \\
ac \mid d \mid bef \\
\qquad \downarrow \varphi(e) = \{d, e, f\} \\
ac \mid d \mid e \mid bf \\
\qquad \downarrow \varphi(f) = \{d, e, f\} \\
ac \mid d \mid e \mid f \mid b
\end{array}$$

Fig. 8. Partition refinement into maxmods from Example 4.4 (see Example 4.15).

Using Definition 4.11, we can now reformulate Theorem 3.12 into a general statement:

Theorem 4.16 *Given a closure operator* φ *on a finite set* U*, and two closed sets* A, B*, then* B *covers* A *iff* $A \subseteq B$ *and* $B - A$ *is a non-dominating maxmod of* $dom_\varphi(A)$ *(or, equivalently, a minimal element of* $Dom_\varphi(A)$*).*

PROOF. From Property 4.9, follows that a closed set B covers a closed set A if A is a proper subset of B and B is an atom of the closure system: $\mathcal{F}_{\varphi_A}$. By Property 4.13, B is a non-dominating maxmod of dom_{φ_A}. For any $x, y \in U - A$, we have the following equivalent statements:

$$\begin{aligned}
(x, y) \in dom_{\varphi_A} &\Longleftrightarrow y \in \varphi_A(\{x\}) \\
&\Longleftrightarrow y \in \varphi(A \cup \{x\}) \\
&\Longleftrightarrow (x, y) \in dom_\varphi(A)
\end{aligned}$$

Thus, $B - A$ is a maxmod of $dom_\varphi(A)$.

The converse proof is similar.

The inheritance mechanisms also extend readily:

Property 4.17 *Let A and B be closed sets, with $A \subset B$, let x and y be elements of U which are not in B. Then if x dominates y in A (i.e. $(x, y) \in dom_\varphi(A)$) , then x also dominates y in B (i.e. $(x, y) \in dom_\varphi(B)$).*

PROOF. By definition of $dom_\varphi(X)$, the following equivalences hold for any $(x, y) \notin B$:

$$(x, y) \in dom_\varphi(A) \Longleftrightarrow y \in \varphi(A \cup \{x\})$$
$$(x, y) \in dom_\varphi(B) \Longleftrightarrow y \in \varphi(B \cup \{x\})$$

Moreover, $A \cup \{x\} \subseteq B \cup \{x\}$ implies $\varphi(A \cup \{x\}) \subseteq \varphi(B \cup \{x\})$.
Consequently, $dom_\varphi(A) \cap (U - B)^2 \subseteq dom_\varphi(B)$.

Thus, even in this more general context, we are able, given a closure operator, to compute the cover of an element, with the same algorithmic advantages: possibility of a cheap local investigation of the lattice, efficient recursive generation of all closed sets, quick generation of all the introducers.

5 Logical representation of generalized domination

Horn functions are used in relational databases theory (14) and logic programming (24). In order to efficiently compute generalized domination, we will now similarly consider Horn functions associated with closure operators.

5.1 Preliminary notions

This section deals with Boolean functions that map 2^U into $\{0, 1\}$. Given such a Boolean function f, we call *model* (*counter model* resp.) any subset $X \subseteq U$ such that $f(X) = 1$ ($f(X) = 0$ resp.). We identify every $x \in U$ with the Boolean function such that $x(X) = 1$ iff $x \in X$. f is said to be a *literal* if $f = x$ or $f = \neg x$ for some $x \in X$. Literals of the form x are said to be *positive*, and *negative* otherwise.

We now introduce the necessary notations and basic concepts on Horn functions which we will need throughout the rest of this paper. We refer the reader to (21; 12; 9) for general statements and proofs of main results in this theory.

A *propositional clause* is a finite disjunction of literals that do not contain both a function x and its negation $\neg x$. A (proper) sub-disjunction of a clause is called a *(proper) subclause*. A clause is said to be a *Horn clause* if it has at most one positive literal. The empty clause is the constant Boolean function 0. The set of negative literals of a clause is called the *support* of this clause. A non-empty clause $\bigvee_{a \in A} \neg a$ with no positive literal is said to be *negative*, and is usually denoted by $A \rightarrow$, or sometimes by $A \rightarrow U$. A non-empty Horn clause $\bigvee_{a \in A} \neg a \vee b$ with exactly one positive literal b is said to be *pure* and will be denoted by $A \rightarrow b$. Moreover, we will sometimes write the conjunction $\bigwedge \{A \rightarrow b : b \in B, b \notin A\}$ of a set of pure Horn clauses having the same support A, simply as $A \rightarrow B$.

A set $\mathcal{H}$ of Horn clauses is said to be:

- *unsatisfiable* if $\bigwedge \mathcal{H} = 0$.
- a *Horn representation* of f if $\bigwedge \mathcal{H} = f$; f is then said to be a *Horn function.*
- *irredundant* if for any proper subset $\mathcal{H}'$ of $\mathcal{H}$, $\bigwedge \mathcal{H}' \neq \bigwedge \mathcal{H}$.
- *equivalent* to another set $\mathcal{H}'$ of clauses if $\bigwedge \mathcal{H}' = \bigwedge \mathcal{H}$.

Finally, a clause g is an *implicate* of a Boolean function of f if $f \leq g$. It is *prime* if no proper subclause is an implicate. We denote by $\mathcal{P}_f$ the set of prime implicates of a given Boolean function f. It is well known that f is a Horn function if and only if $\mathcal{P}_f$ is a Horn representation of f. Any Horn representation $\mathcal{H}$ of f such that $\mathcal{H} \subseteq \mathcal{P}_f$ is said to be a *prime representation* of f.

5.2 Horn functions associated with closure operators

We will now how we can associate a Boolean function with a closure operator φ.

Definition 5.1 *Let φ be a closure operator on U; we denote by f_φ the Boolean function that maps 2^U onto $\{0, 1\}$ defined by:*

$$f_\varphi(X) = 1 \Longleftrightarrow \varphi(X) = X \text{ and } X \neq U$$

Definition 5.2 *Let H be a set of clauses. We will denote by* ABS(H) *the minimal equivalent set of clauses obtained from H by dropping clauses by absorption (i.e. by dropping all clauses that have a subclause in $\mathcal{H}$).*

To clarify the relationship between closure systems and prime implicates of a Horn function, we need to associate a set of propositional Horn clauses with the subsets of U.

Definition 5.3 *Let φ be a closure operator on U; let A be a subset of U. Then A can be associated with the following set of propositional Horn clauses $H_\varphi(A)$:*

$$H_\varphi(A) = \begin{cases} \{A \rightarrow\} & \text{if } \varphi(A) = U \\ \{A \rightarrow b : b \in \varphi(A) - A\} & \text{if } A \subset \varphi(A) \neq U \\ \emptyset & \text{otherwise} \end{cases}$$

For every $\mathcal{X} \subseteq 2^U$, then $H_\varphi(\mathcal{X})$ is defined as the set of clauses:

$$H_\varphi(\mathcal{X}) = \mathrm{ABS}\left(\bigcup_{A \in \mathcal{X}} H_\varphi(A)\right)$$

We apply this to define a Horn representation of f_φ and show the connection with Boolean functions usually associated with Functional Dependencies in theory of Relational Databases (14).

Lemma 5.4 *For every closure operator φ on U, $H_\varphi(2^U)$ is a Horn representation of f_φ.*

PROOF. The lemma follows from the following equivalent statements:

(1) X is a counter-model of f_φ.
(2) X is a subset of U such that $X \neq \varphi(X)$ or $X = U$.
(3) X is a counter-model of $\bigwedge H_\varphi(2^U)$

The last two statements are equivalent because $H_\varphi(2^U)$ contains either a subclause of $X \rightarrow$ if $\varphi(X) = U$, or a subclause of $X \rightarrow x$ for some $x \in \varphi(X) - X$.

It is worth mentioning that this Horn representation of f_φ is not pure, as it contains negative clauses. Any Horn function on n variables can be encoded into a unique positive Horn function on $n+1$ variables. We will not consider such translations in this paper, since the positive component of $\mathcal{P}_{f_\varphi}$ plays an important role in rule generation, as we will see in Subsection 6.3.

Theorem 5.6 below characterizes the prime implicates of f_φ. It could be deduced from well-known results in relational databases (14) or Boolean analysis (16), But for the sake of self-containment, we will give a direct proof.

Definition 5.5 *Let φ be a closure operator on U, we denote by $\mathcal{J}_\varphi$ the family of subsets X of U such that:*

(1) $X \neq \varphi(X)$,
(2) for any proper subset Y *of* X, $\varphi(Y) \neq \varphi(X)$.

In the terminology of Relational Databases, an element J of $\mathcal{J}_\varphi$ such that $\varphi(J) = F \in \mathcal{F}_\varphi$ (i.e. $\varphi(J)$ is an element of closure system $\mathcal{F}_\varphi$) is called a *generator* of F. If $F = U$ then J is said to be a *key*.

Note that, by Item 1 of Definition 5.5, $\mathcal{J}_\varphi \cap \mathcal{F}_\varphi = \emptyset$ and that, by Item 2, each X in $\mathcal{J}_\varphi$ is a minimal element of $\{Y \subseteq U : \varphi(Y) = \varphi(X)\}$. Thus a subset $A \subset U$ is closed if and only if for any negative clause $X \rightarrow$ we have $X \subsetneq A$, and for any clause $X \rightarrow \alpha \in H_\varphi(\mathcal{J}_\varphi)$ such that $X \subseteq A$, we have $\alpha \in A$.

Theorem 5.6 *Let* φ *be a closure operator on* U, *then* $H_\varphi(\mathcal{J}_\varphi \cup \{U \rightarrow\})$ *is the set of prime implicates of* f_φ.

PROOF. First, we show that any clause of $H_\varphi(\mathcal{J}_\varphi)$ is a prime implicate of f_φ. By Lemma 5.4, since $H_\varphi(\mathcal{J}_\varphi) \subseteq H_\varphi(2^U)$, $H_\varphi(\mathcal{J}_\varphi)$ is a set of implicates of f_φ. Let $g \in H_\varphi(\mathcal{J}_\varphi)$. We consider two cases:

(1) Suppose $g = J \rightarrow$ is a negative clause of $H_\varphi(\mathcal{J}_\varphi)$. If J was not prime, we would have $J' \subset J$ such that $f_\varphi \leq J' \rightarrow < J \rightarrow$.
Since $J \in \mathcal{J}_\varphi$ and $J' \subset J$ we have $\varphi(J') \neq U$. Then $\varphi(J')$ is a model of f_φ and a counter-model of $J' \rightarrow$. This contradicts the hypothesis that J' is an implicate of f_φ. Thus no subclause of $J \rightarrow$ is an implicate of f_φ, which shows that $J \rightarrow$ is prime.
(2) Suppose $g = J \rightarrow j$ is a pure Horn clause of $H_\varphi(\mathcal{J}_\varphi)$. By absorption, we have:

$$j \in \varphi(J) - \bigcup_{S \subset J, S \in \mathcal{J}_\varphi} \varphi(S)$$

Let us suppose that there exists a proper subclause h of $J \rightarrow j$ that is an implicate. Therefore there exists $J' \subset J$ such that $h = J' \rightarrow j$. Consequently, there exists $K \in \mathcal{J}_\varphi$ with: $K \subseteq J' \subset J$ such that $j \in \varphi(K)$, which contradicts (1).

Conversely, let h be a prime implicate of f_φ. Since f_φ is a Horn function, h is a Horn clause. We again examine two cases:

(1) If h is negative, $h = A \rightarrow$ for some proper subset A of U. Then for any subset $X \subseteq U$ such that $A \subseteq X$, $f(X) = 0$. Then $\varphi(A) = U$ and consequently A is a key in $\mathcal{J}_\varphi$.
(2) If h is pure, then $h = A \rightarrow a$ for some proper subset A of U and some $a \in U - A$. Since $A \rightarrow a$ is a prime implicate of f, $f(A) = 0$ and for any subset X of U such that $A \subset X$, $f(X) = 1$ implies $a \in X$. thus $a \in \varphi(A)$ and A is a generator of $\varphi(A)$, as if A was not a generator, there would

exist a proper subset A' of A such that $f \leq A' \rightarrow a$, which contradicts the assumption that $A \rightarrow a$ is prime.

Example 5.7 *We will use the following relation, from (20)*

$\mathcal{P} = \{a, b, c, d, e\}$

$\mathcal{O} = \{1, 2, 3, 4\}$

R	a	b	c	d	e
1	×	×			
2	×		×		
3		×	×	×	
4				×	×

The associated Concept Lattice is shown in Figure 9.

Let us use the concepts to define a closure system on $U = \{a, b, c, d, e\}$*:*

$$\begin{aligned}
\mathcal{F}_\varphi &= \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,c\}, \{b,c,d\}, \{d,e\}, U\} \\
\mathcal{J}_\varphi &= \{\{e\}, \{a,d\}, \{a,e\}, \{b,c\}, \{b,d\}, \{b,e\}, \{c,d\}, \{c,e\}, \{a,b,c\}\} \\
H_\varphi(\mathcal{J}_\varphi) &= \{\{a,d\} \rightarrow, \{a,e\} \rightarrow, \{c,e\} \rightarrow, \{b,e\} \rightarrow, \{a,b,c\} \rightarrow, \\
&\quad \{e\} \rightarrow d, \{b,c\} \rightarrow d, \{b,d\} \rightarrow c, \{c,d\} \rightarrow b\} \\
f_\varphi &= (\neg a \vee \neg d) \wedge (\neg a \vee \neg e) \wedge (\neg c \vee \neg e) \wedge (\neg b \vee \neg e) \\
&\quad \wedge (\neg a \vee \neg b \vee \neg c) \wedge (\neg e \vee d) \wedge (\neg b \vee \neg c \vee d) \wedge (\neg b \vee c \vee \neg d) \\
&\quad \wedge (b \vee \neg c \vee \neg d)
\end{aligned}$$

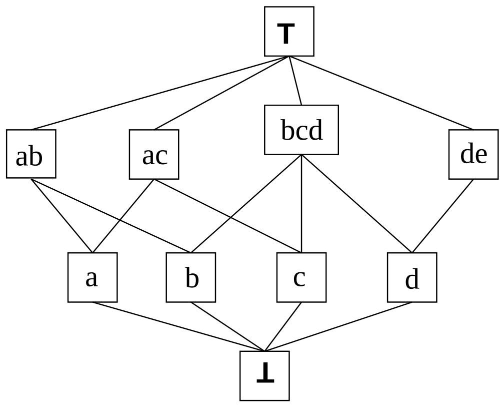

Fig. 9. Concept lattice of the relation from Example 5.7.

5.3 Horn representation of domination for closed sets

We will now translate the domination relations into logical form.

Definition 5.8 *Let φ be a closure operator on U, $\mathcal{X} \subseteq 2^U$, $A \subseteq U$, and $(x,y) \subseteq (U-A)^2$. We will define $K_\varphi(\mathcal{X}, A, x, y)$ as the set of clauses:*

$$H_\varphi(\mathcal{X}) \cup \{\rightarrow a : a \in A\} \cup \{\rightarrow x, y \rightarrow\}$$

Theorem 5.9 *Let φ be a closure operator on U, $\mathcal{H}$ a Horn representation of f_φ, A a closed set, and $(x,y) \in (U-A)^2$. Then $(x,y) \in dom_\varphi(A)$ iff $K_\varphi(\mathcal{H}, A, x, y)$ is unsatisfiable.*

PROOF. From Lemma 5.4, we can deduce that $\mathcal{H} = H_\varphi(2^U)$, and by definition, $(x,y) \in dom_\varphi(A)$ iff $y \in \varphi(A \cup \{x\})$.

Clearly, we again have to consider two cases:

(1) $\varphi(A \cup \{x\}) = U$. In this case, $(x,y) \in dom_\varphi(A)$ for every $y \in U - A$ by definition of $dom_\varphi(A)$. Moreover, there exists a least one subclause of $(A \cup \{x\}) \rightarrow$ in $\mathcal{H}$, thus $\mathcal{H} \cup \{\rightarrow a : a \in A\} \subseteq K_\varphi(\mathcal{H}, A, x, y)$ is obviously unsatisfiable for every $y \in U - A$.

(2) $\varphi(A \cup \{x\}) \neq U$. In this case, if $(x,y) \in dom_\varphi(A)$, then a subclause $A \cup \{x\} \rightarrow y$ is in $\mathcal{H}$ and $K_\varphi(\mathcal{H}, A, x, y)$ is unsatisfiable. Conversely, if $K_\varphi(\mathcal{H}, A, x, y)$ is unsatisfiable, then for any model M of f such that $A \cup \{x\} \subseteq M$, we have $y \in M$ and therefore, $y \in \varphi(A \cup \{x\})$.

As the Horn SAT problem can be solved in linear time (see, for example, (15) or (27)), we can deduce from Theorem 5.9 that $dom_\varphi(A)$ can be computed in $O(|\mathcal{H}|.|U-A|^2)$ time, for any closed set A. Moreover, we can suppose that $\mathcal{H}$ is an irredundant subset of $H_\varphi(\mathcal{J}_\varphi)$, since an irredundant and prime representation of $\bigwedge \mathcal{H}$ can be computed in $O(|\mathcal{H}|^2)$ time (21).

Example 5.10 *Let $\mathcal{F}_\varphi$ be the closure system defined in Example 5.7.*

$H_\varphi(\mathcal{J}_\varphi) \cup \{\rightarrow d\} = \{\{a,d\} \rightarrow, \{a,e\} \rightarrow, \{c,e\} \rightarrow, \{b,e\} \rightarrow, \{a,b,c\} \rightarrow, \{b,d\} \rightarrow c, \{c,d\} \rightarrow b, \rightarrow d\}$.

and thus:

- $dom_\varphi(\{d\}) = \{(b,c),(c,b),(a,b),(a,c),(a,e)\}$
- $Dom_\varphi(\{d\}) = \{(\{a\},\{b,c\}),(\{a\},\{e\})\}$
- *Cover of* $\{d\}$*:* $\{\{b,c,d\},\{d,e\}\}$

6 Closure systems associated with rule generation

One of the most crucial problems in Data Mining using Formal Concept Analysis is rule extraction. In Example 5.7, e will imply d, because there is no concept where e appears without d. Finding these rules, called *exact association rules*, is of major importance in practise, and clearly there are a great number of them.

Work by Guigues and Duquenne (20) and by Ganter (17) show that the set of such rules can be represented by a basis of rules, from which all other rules can be easily inferred, a process which can drastically reduce the number of rules which need to be computed and memorized. Computing this basis is equivalent to computing the canonical cover of functional dependencies in a relational database (25).

In relation to the work in this paper, existing rule generation algorithms in Formal Concept Analysis are based on the definition of two closure systems, corresponding to pseudo-closed sets and quasi-closed sets associated with the initial closure system corresponding to concepts.

In this section, we will apply our results to these two other closure systems, and in particular we will accordingly transpose Theorem 5.9.

6.1 Dependency relations and basis

Any closure system is associated with a dependency relation corresponding to the set of association rules (28). Generators and basis can thus be used in the context of closure systems.

Definition 6.1 *A binary relation D on 2^U is said to be a* dependency relation *if the following properties hold for all $Y_1, Y_2, Y_3 \subseteq U$:*

D1) *D is transitive,*
D2) *if $Y_2 \subseteq Y_1$ then $(Y_1, Y_2) \in D$,*
D3) *if $(Y_1, Y_2) \in D$ then $(Y_1 \cup Y_3, Y_2 \cup Y_3) \in D$.*

Note that conditions D1) and D3) imply that if $(Y_1, Y_2) \in D$ and $(Y_3, Y_4) \in D$, then $(Y_1 \cup Y_3, Y_2 \cup Y_4) \in D$. Consequently, the binary relation Θ_D defined on 2^U by $(X, Y) \in \Theta_D$ iff $(X, Y) \in D$ and $(Y, X) \in D$ is a congruence on the semi-lattice $(2^U, \cup)$. The structure (U, Θ_D) is called a dependence space in (28).

Definition 6.2 *If R is a relation on 2^U, we will denote by R^+ the minimal*

relation on 2^U including R which is a dependency relation.
Let D be a dependency relation on 2^U. A subrelation $R \subseteq D$ is said to be a generator *of D iff $R^+ = D$.*
If there is some proper subrelation S of R such that $S^+ = D$, then R is said to be redundant.

Definition 6.3 *Let φ be a closure operator U. We define a binary relation $\rightarrow_\varphi$ on $(2^U)^2$ by the following equivalent conditions for $X, Y \subseteq U$:*

$$\begin{aligned} X \rightarrow_\varphi Y &\Longleftrightarrow (\forall Z \subseteq U)\ \ (X \subseteq \varphi(Z) \Rightarrow Y \subseteq \varphi(Z)) \\ &\Longleftrightarrow \varphi(Y) \subseteq \varphi(X) \\ &\Longleftrightarrow Y \subseteq \varphi(X) \end{aligned}$$

where $(X, Y) \in \rightarrow_\varphi$ is denoted by the infix notation $X \rightarrow_\varphi Y$.

A pair of subsets X, Y such that $X \rightarrow_\varphi Y$ is called an exact association rule *in Data Mining or a* (functional) dependency *in the theory of relational databases.*

From (28) hold the following results.

Property 6.4 *An operator φ on U is a closure iff $\rightarrow_\varphi$ is a dependency relation on 2^U.*

From a formal point of view, $X \rightarrow_\varphi Y$ denotes a pair of sets, while $X \rightarrow Y$ denotes a set of propositional clauses. However, we will see that one holds if and only if the other holds.

We can now define generators and basis for an association relation:

Definition 6.5 *Let φ be a closure operator on U, let $\mathcal{X} \subseteq 2^U$ be a family of non-closed sets. We will denote by $R_\varphi(\mathcal{X})$ the relation $\{(X, \varphi(X)) : X \in \mathcal{X}\} \subseteq (2^U)^2$.*
We will say that $\mathcal{X}$ is a generator *of $\rightarrow_\varphi$ if $R_\varphi(\mathcal{X})^+ = \rightarrow_\varphi$. If in addition $R_\varphi(\mathcal{X})$ is minimal, then $\mathcal{X}$ is called a* basis *of $\rightarrow_\varphi$.*

As it has been pointed out in (20), $\mathcal{J}_\varphi$ is a generator of $\rightarrow_\varphi$.

Definition 6.6 *Let φ be a closure operator on U. Then a subset $X \in 2^U - \mathcal{F}_\varphi$ is said to be* quasi-closed *iff for any $Y \in \mathcal{F}_\varphi$, $X \cap Y \in \mathcal{F}_\varphi \cup \{X\}$. We will denote by $\mathcal{Q}_\varphi$ the family of quasi-closed sets.*

Because of Definition 4.5, for any quasi-closed set X, $\mathcal{F}_\varphi \cup \{X\}$ is a closure system. This leads to the following theorem, proved in (20; 10).

Theorem 6.7 *Let φ be a closure operator on U, then:*

(1) $\mathcal{Q}_\varphi$ *is a generator of* $\rightarrow_\varphi$.
(2) $\mathcal{F}_\varphi \cup \mathcal{Q}_\varphi$ *is a closure system.*

(20) showed that all the basis of $\rightarrow_\varphi$ have the same cardinality, and they define a unique (canonical) basis by using the closure system which we will now describe.

Definition 6.8 *Let* φ *be a closure operator on* U. *The family* $\mathcal{B}_\varphi$ *of* pseudo-closed sets *of* φ *is defined by:*

$$B \in \mathcal{B}_\varphi \text{ iff } \varphi(B) \neq B \text{ and } (\forall A \in \mathcal{B}_\varphi)\ A \subset B \Rightarrow \varphi(A) \subseteq B$$

Theorem 6.9 *(20) Let* φ *be a closure operator on a finite set* U*, then:*

(1) $\mathcal{B}_\varphi$ *is a basis of* $\rightarrow_\varphi$.
(2) $\mathcal{B}_\varphi \subseteq \mathcal{Q}_\varphi$.
(3) $\mathcal{F}_\varphi \cup \mathcal{B}_\varphi$ *is a closure system.*

We refer the reader to the original paper (20) or to (11) for the proof of Theorems 6.7 and 6.9.

Example 6.10 *Figure 10 gives the lattice of* $\mathcal{F}_\varphi \cup \mathcal{B}_\varphi$ *corresponding to Example 5.7.*

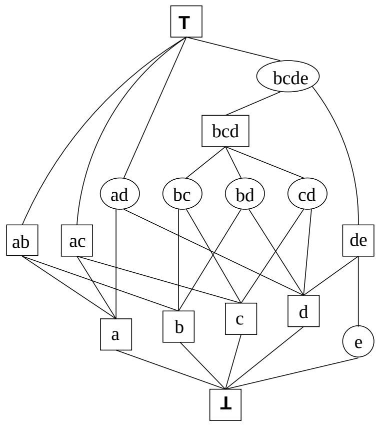

Fig. 10. Lattice of concepts and pseudo-closed sets of the relation from Example 5.7.

In the rest of this work, we will call family $\mathcal{B}_\varphi$ the *canonical basis* of $\rightarrow_\varphi$; we will denote by θ_φ the closure operator associated with the closure system $\mathcal{F}_\varphi \cup \mathcal{Q}_\varphi$, and by β_φ the closure operator associated with $\mathcal{F}_\varphi \cup \mathcal{B}_\varphi$.

As we have generalized domination to any closure system, there will be a domination for closure system $\mathcal{F}_{\beta_\varphi} = \mathcal{B}_\varphi \cup \mathcal{F}_\varphi$ and a domination for closure system $\mathcal{F}_{\theta_\varphi} = \mathcal{Q}_\varphi \cup \mathcal{F}_\varphi$. Characterization 4.16 can thus be applied to generating the closed sets of β_φ and θ_φ.

6.2 Canonical basis and minimal support Horn representation

We will now give a logical representation of the canonical basis, and correspondingly express the domination relations associated with $\mathcal{F}_{\beta_\varphi}$ and $\mathcal{F}_{\theta_\varphi}$.

This logical representation is based on the following theorem well-known in Relational Databases (14, Ch.11, §2.2). We restate it here using the notations introduced in the previous subsection.

Theorem 6.11 *Let R be a binary relation on 2^U, X a subset of U and x an element of U. Then, $(X, \{x\}) \in R^+$ iff we have:*

$$\bigwedge\{V \rightarrow W : (V, W) \in R\} \wedge \neg(X \rightarrow x) = 0$$

From this theorem, the two following corollaries follows directly. The first one translates in terms of Horn functions the notion of generator of a dependency relation, the second one translates the notion of non redundancy of a set of functional dependencies.

Corollary 6.12 *Let φ be a closure operator; $\mathcal{X}$ is a generator of $\rightarrow_\varphi$ iff $H_\varphi(\mathcal{X})$ is equivalent to $H_\varphi(2^U)$.*

PROOF. $\mathcal{X}$ is a generator of $\rightarrow_\varphi$ iff $R_\varphi(\mathcal{X})^+ = \rightarrow_\varphi$. From Theorem 6.11 follows that, for any $A \subseteq U$ and $a \in U - A$,

$$(A, \{a\}) \in R_\varphi(\mathcal{X})^+ \Longleftrightarrow \bigwedge H_\varphi(\mathcal{X}) \wedge \neg(A \rightarrow a) = 0$$

Consequently, $R_\varphi(\mathcal{X})^+ = \rightarrow_\varphi$ iff $\bigwedge H_\varphi(2^U)$ and $\bigwedge H_\varphi(\mathcal{X})$ have the same implicates iff $H_\varphi(\mathcal{X})$ is equivalent to H_φ.

This corollary shows the well-known connection between the closure of a set of functional dependencies using rules D1), D2) and D3) and the forward chaining closure defined by a set of clauses. However, there is a slight difference between the notion of a non-redundant set of functional dependencies and a non redundant set of clauses. The following definition and corollary give the exact connections between these concepts, as it is done in (9).

Definition 6.13 *Given a set $\mathcal{H} = \{A_p \rightarrow a_p : p \in P\} \cup \{B_n \rightarrow : n \in N\}$ of Horn Clauses, we denote by $Su(\mathcal{H})$ the set of its supports: $\{A_p : p \in P\} \cup \{B_n : n \in N\}$.*

We shall say that $\mathcal{H}$ is support -non-redundant *if $\{B_n : n \in N\}$ is irredundant and for any $p \in P$,*

$$\bigwedge(\mathcal{H} - \{A_i \rightarrow a_i : A_i = A_p, i \in P\}) \neq \bigwedge \mathcal{H}$$

Otherwise $\mathcal{H}$ is said to be support-redundant.

Finally, $\mathcal{H}$ is said to be a minimal-support *Horn representation of a function f if $\mathcal{H}$ is a Horn representation of f such that the number $|Su(\mathcal{H})|$ of supports in $\mathcal{H}$ is minimal.*

Corollary 6.14 *Let φ be a closure operator on U, and $\mathcal{X} \subseteq \mathcal{J}_\varphi$, then:*

(1) $R_\varphi(\mathcal{X})$ is non-redundant iff $H_\varphi(\mathcal{X})$ is support-non-redundant.
(2) $\mathcal{X}$ is the canonical basis of φ iff $H_\varphi(\mathcal{X})$ is a minimal-support Horn representation of f_φ with supports of maximal cardinality.

PROOF. We first prove Item 1. Let us suppose that there exists $J \in \mathcal{J}_\varphi$ such that $R_\varphi(\mathcal{X} - \{J\})^+ = \rightarrow_\varphi$. By Corollary 6.12, $H_\varphi(\mathcal{X} - \{J\})$ is equivalent to $H_\varphi(2^U)$. Since J is a generator, the set $\mathcal{X}_J$ of clauses in $H_\varphi(\mathcal{X})$ such that $Su(\mathcal{X}_J) = \{J\}$ is non-empty. Consequently, $H_\varphi(\mathcal{X}) - \mathcal{X}_J \subset H_\varphi(\mathcal{X})$ and $\bigwedge(H_\varphi(\mathcal{X}) - \mathcal{X}_J) = \bigwedge H_\varphi(\mathcal{X})$. The converse can be proved in a similar fashion.

Item 1 together with Corollary 6.12 imply that $\mathcal{X}$ is a basis if and only if $H(\mathcal{X})$ is a minimal support Horn representation of f_φ. Given such a representation, it follows from Theorem 6.9 that $\bigwedge H(\mathcal{X}) = \bigwedge H(\mathcal{B}_\varphi)$ and $|Su(H(\mathcal{X}))| = |\mathcal{B}_\varphi| = |Su(H(\mathcal{B}_\varphi))|$. Moreover, the closure operator β_φ maps each support $S \in Su(H(\mathcal{X}))$ into the smallest $B \in \mathcal{B}_\varphi$ such that $S \subseteq B$. This defines a one-to-one correspondence that maps any element of $|Su(H(\mathcal{X}))|$ into a larger element in $Su(H(\mathcal{B}_\varphi))$.

The following corollary derives (25), where algorithms to compute canonical covers of functional dependencies in a relational database are presented. A complete and direct proof can now be found in (9).

Corollary 6.15 *Let φ be a closure operator on a finite set U. Given the generator $\mathcal{J}_\varphi$, the problem of finding a basis of φ is polynomially solvable.*

We will now translate dominations for closure β_φ into logical form, as we did in Theorem 5.9 for closure φ.

Let φ be a closure operator on U, A a closed set, $(x, y) \in (U - A)^2$ and $[\mathcal{B}_\varphi]_{A,x}$ the following subset of $\mathcal{B}_\varphi$:

$$[\mathcal{B}_\varphi]_{A,x} = \{X \in \mathcal{B}_\varphi : |X| < |\beta_\varphi(A \cup \{x\})|\}$$

By Definition 6.8 and by Theorem 6.11,

$$(x, y) \in dom_{\beta_\varphi}(A) \Longleftrightarrow K_\varphi([\mathcal{B}_\varphi]_{A,x}\,,\ A,\ x, y)$$

is unsatisfiable.

6.3 Horn representations of domination and quasi-closed sets

Another approach to finding the canonical basis is to generate the quasi-closed sets by the method presented in (20). In the context of propositional Horn clauses, the relationship between the representation of f_φ based on quasi-closed sets and the one based on pseudo-closed sets is quite simple since $H_\varphi(\mathcal{B}_\varphi) = H_\varphi(\mathcal{Q}_\varphi)$.

Domination for θ_φ can be computed for any $A \in \mathcal{F}_\varphi$ using any generator $G \subseteq \mathcal{J}_\varphi$ of $\rightarrow_\varphi$, as we will see in Theorem 6.20. To state this theorem we need to consider finite ideals of closure systems and pure components of Horn representations.

Definition 6.16 *For any closed set A, we will denote by $\varphi|A$ the closure defined on A by $(\varphi|A)(X) = \varphi(X)$ for any $X \subseteq A$. Let $\mathcal{H}$ be a prime representation of f_φ, we will denote by $\mathcal{H}|A$ the set of clauses:*

$$\mathcal{H}|A = \{g \in \mathcal{H} : Su\{g\} \subseteq A\} \cup \{A \rightarrow\}$$

where $Su\{g\}$ is the support of g.

Property 6.17 *Let φ be a closure operator and $\mathcal{H}$ a prime representation of f_φ, then for every closed set A, $\mathcal{H}|A$ is a Horn representation of $f_{\varphi|A}$.*

PROOF. Clearly, $\bigwedge \mathcal{H}|A = f_\varphi \wedge \bigwedge\{x \rightarrow : x \notin A\} \wedge A \rightarrow = f_{\varphi|A}$

Given a set of clauses $\mathcal{H}$, we denote by $\mathcal{H}^P$ the subset of pure Horn clauses. We shall call this subset the *pure component* of $\mathcal{H}$. The following property has been shown in (21).

Property 6.18 *The pure components of prime Horn representations of a given Boolean function are equivalent.*

We can now state the Horn representation of domination for quasi-closed sets based on the following lemma from (20).

Lemma 6.19 ((20)) *$X \in \mathcal{Q}_\varphi$ iff for any $Y \subset X$, $\varphi(Y) \neq \varphi(X) \Rightarrow \varphi(Y) \subset X$.*

Theorem 6.20 *Let φ be a closure operator on U, A a closed set, $(x, y) \in (U - A)^2$, $\mathcal{H}$ a prime representation of f_φ and $\mathcal{P}$ a prime representation of $\bigwedge(\mathcal{H}|\varphi(A \cup \{x\})) = f_{\varphi|\varphi(A\cup\{x\})}$.*

Then $(x, y) \in dom_{\theta_\varphi}(A)$ iff $K_\varphi\left(\mathcal{P}^P, A, x, y\right)$ is unsatisfiable.

PROOF. Let X be a subset of U and $\mathcal{J}_\varphi(A, x) = \{J \in \mathcal{J}_\varphi : \varphi(J) \subset \varphi(A \cup \{x\})\}$. From Lemma 6.19 and Theorem 6.11, for any subset $Z \subseteq U$, the smallest quasi-closed set $Q_\varphi(Z)$ containing Z is the smallest solution of the equation:

$$\bigwedge H_\varphi(\mathcal{J}_\varphi(A, x)) \wedge \bigwedge\{\to z : z \in Z\} = 1$$

However, $H(\mathcal{J}_\varphi(A, x))$ is a prime Horn representation of the pure component of $f_{\varphi|\varphi(A\cup\{x\})}$. By Property 6.18 we have:

$$\bigwedge H_\varphi(\mathcal{J}_\varphi(A, x)) = \bigwedge \mathcal{P}^P$$

Consequently, $y \in \theta_\varphi(A \cup \{x\})$ iff $\bigwedge\{\to z : z \in A \cup \{x\}\} \wedge y \to = 0$, which proves the theorem.

As recalled in Section 5.3, a prime cover of $\mathcal{H}$ can be computed in $O(|\mathcal{H}|^2)$, so $dom_{\theta_\varphi}(A)$ can be computed in $O(|\mathcal{H}|^2.|U - A|)$ if $|U - A| \leq |\mathcal{H}|$.

However, using the fact that, given a closure operator φ, any generator $\mathcal{G} \subseteq \mathcal{J}_\varphi$ of $\to_\varphi$ induces a prime representation of f_φ, it is also possible to compute $dom_{\theta_\varphi}(A)$ in $O(|\mathcal{G}|.|U - A|^2)$ time if the relation $R_\varphi(\mathcal{G})$ is known. We illustrate this in the following example.

Example 6.21 *Consider again the closure system defined in Example 5.7.*

$$\begin{aligned}\mathcal{J}_\varphi(\{d, e\}, a) &= \{\{e\}, \{b, c\}, \{b, d\}, \{c, d\}\}\\ &= \mathcal{J}_\varphi(\{d, e\}, b) = \mathcal{J}_\varphi(\{d, e\}, c)\end{aligned}$$

where $\mathcal{J}_\varphi(A, a)$ has been defined for any closed set and any $a \in U$ in the proof of Theorem 6.20. For $x \in \{a, b, c\}$:

$$H_\varphi(\mathcal{J}_\varphi(\{d, e\}, x) \cup \{\rightarrow d, \rightarrow e\}) = \{bd \rightarrow c, cd \rightarrow b, \rightarrow d, \rightarrow e\})$$

and $dom_{\theta_\varphi}(\{d, e\}) = \{(b, c), (c, b)\}$. Therefore, the elements of $\mathcal{Q}_\varphi$ that cover $\{d, e\}$ are $\{a, d, e\}$ and $\{b, c, d, e\}$. Among them, only the last one is in $\mathcal{B}_\varphi$.

7 Conclusion and open questions

In this paper, we use the relationship between concept lattices and domination in graphs to extend existing graph-oriented results on concept lattices to a general closure system and to Horn clauses.

Though there obviously remains much work to be done in this direction, our results are interesting not only from a possible algorithmic point of view, but also because they can lead to a better understanding of the canonical basis of rules; moreover, it is important to find new ways of modeling these results so that a variety of non-specialists can achieve a better grasp on these problems.

Our results are algorithmically promising because the notion of domination allows a local approach to generating closed sets: first, one can easily examine a subproblem related to a particular area of the underlying lattice, without generating the entire lattice structure starting from the bottom element. Second, this can allow a very efficient recursive generation technique of the lattice, as we have shown is the case for concept lattices (5). The same technique applies to generating other lattices of closed sets; rule generation for example should be an interesting application of this.

Another question of great current interest is that of generating approximate association rules. As an example, an interesting recent approach by J-M. Bernard and S. Poitrenaud (2) works by first approximating the binary relation according to coherent probabilistic models which must be compatible with logical rules; the logical interpretation we introduce in this paper could be combined with this approach in future work.

References

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

[2] Bernard J-M., Poitrenaud S.: L'analyse implicative Bayésienne multivariée d'un questionnaire binaire : quasi-implications et treillis de Galois simplifié. *Mathématiques Informatique et Sciences humaines*, **147** (1999) 25-46.

[3] Berry A., Sigayret A.: Representing a concept lattice by a graph. *Workshop on Discrete Math. and Data Mining, Proc. 2nd SIAM Conf. on Data Mining (SDM'02), Arlington (VA, USA), April 2002.* Discrete Applied Mathematics, special issue on Discrete Math. and Data Mining, **144:1-2** (2004) 27–42.

[4] Berry A., Sigayret A.: Maintaining class membership information. *Workshop on MAnaging of SPEcialization/Generalization HIerarchies (MASPEGHI), Montpellier (France), Sept. 2002.* LNCS Proc. Int. Conf. OOIS'02 (Object-Oriented Information Systems), (2002).

[5] Berry A., Bordat J-P., Sigayret A.: Concepts can't afford to stammer. *INRIA Proc. Int. Conf.* Journées de l'Informatique Messine (JIM'03), Metz (France), (Sept. 2003). Submitted as 'A local approach for concept generation'.

[6] Bioch J.C., Ibaraki T.: Version Spaces and Generalized Monotone Boolean Functions. *ERIM report (www.erim.eur.nl)*, ERS-2002-34-LIS (2002).

[7] Birkhoff G.: *Lattice Theory.* American Mathematical Society, 3rd Edition, (1967).

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

[9] Boros E., private communication on *Horn functions* (2003), to appear.

[10] Burosch G., Demetrovics J., Katona G.O.H.: The POSet of Closures as a Model of Changing Databases. *Order*, D. Reidel Publ. Co., **4** (1987) 127–142.

[11] Caspard N., Monjardet B.: The lattices of closure systems, closure operators and implicational systems on a finite set: a survey. *Discrete Applied Mathematics*, **127:2** (2003) 241– 269.

[12] Cepek O.: *Structural properties and minimization of Horn Boolean functions.* PhD thesis, RUTCOR, Rutgers University, Piscataway (NJ, USA), 08854, (Oct. 1995).

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

[14] Delobel C., Adiba M.: *Bases de données et systèmes relationnels.* Dunod, (1982).

[15] Dowling W.F., Gallier J.H.: Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulas. *J. Log. Program.*, **1:3** (1984) 267–284.

[16] Flament C.: *L'analyse booléenne de questionnaires.* Mouton, Paris (France), (1976).

[17] Ganter B.: Two basic algorithms in concept analysis. *Preprint 831, Technische Hochschule Darmstad*, (1984).

[18] Godin R., Mili H.: Building and Maintaining Analysis-Level Class Hierarchies Using Galois Lattices. *ACM Proc. OOPSLA'93, Special issue of Sigplan Notice*, **28:10** (1993) 394–410.

[19] Gouda K., Zaki M.J.: Efficiently Mining Maximal Frequent Itemsets. *Proc. 1st IEEE Conf. on Data-mining*, (Nov. 2001).

[20] Guigues J-L., Duquenne V.: Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences humaines*, **95** (1986) 5–18.

[21] Hammer P.L., Kogan A.: Horn functions and their DNF's. *Information Processing Letters*, **44** (1992) 23–29.

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

[23] Yahia A., Lakhal L., Cicchetti R., Bordat J-P.: iO2, An Algorithmic Method for Building Inheritance Graphs in Object Database Design. *LNCS Proc. 15th Int. Conf. on Conceptual Modeling (ER'96), Cottbus (Germany)*, **1157** (1996) 422–437.

[24] Lloyd J.W.: *Foundations of Logic Programming*, Springer - Verlag, (1984).

[25] Maier D.: *The Theory of Relational Databases*. Md.: Computer Science, (1983).

[26] Mephu Nguifo E., Njiwoua P.: Using Lattice-based Framework as a Tool for Feature Extraction. *LNCS Proc. European Conf. on Machine Learning (ECML), Chemnitz (Germany), April 1998*, Springer Verlag, **1398** (1998).

[27] Minoux M.: LTUR: A Simplified Linear-Time Unit Resolution Algorithm for Horn Formulae and Computer Implementation. *Inf. Process. Lett.*, **29:1** (1988) 1–12.

[28] Novotný M.: Dependence Spaces of Information Systems. *in. Incomplete Information : Rough Set Analysis*, E. Orlowska (ed.), Physica-Verlag, (1998).

[29] Rose D.J., Tarjan R.E., Lueker G.S.: Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, **5** (1976) 266–283.

[30] SanJuan E.: On Rough Set Analysis applied to Questionnaires. *STSM Report, COST Action 15*, (1999).

[31] SanJuan E.: Heyting Algebra for Modeling Information Retrieval Based on Thematic Clustering. *Workshop on Discrete Mathematics and Data Mining, Proc. 2nd SIAM Conf. on Data Mining (SDM'02)*, Arlington (VA, USA), (April 2002).

[32] Wille R.: Restructuring Lattice Theory: an approach based on hierarchies of concepts. *Ordered Sets*. I. Rival (ed.), D. Reidel Publ. Co., **83** (1982) 445–470.