The dependence graph of a lattice

Karell Bertet

L3I - Université de La Rochelle - av Michel Crépeau - 17042 La Rochelle
kbertet@univ-lr.fr

Abstract: In this paper, we introduce the dependence graph of a lattice defined on the set of its join-irreducible elements. This graph, issued from the dependence relation on a lattice, is a nice structure encoding together the minimal generators and the canonical direct basis of a lattice. Then, we propose a new generation algorithm.

Keywords: lattice, closed set lattice, dependence graph, implicational system, closure system, canonical direct basis, minimal generators

1 Introduction

From the year 2000, the increasing interest into Formal Concept Analysis (FCA)[7] in various domains of computer science, such as data-mining and knowledge representation, as well as the fields of ontology or databases, has brought to light the structure of *concept lattice*. A concept lattice can be introduced as a directed acyclic graph with the lattice property, defined from data described by a *binary table* object x attribute, named a *context*. The nodes of the graph are concepts - a concept is a maximal subset of objects possessing common attributes. This lattice composed of concepts connected by a generalization- specialization relation, supplies a very intuitive representation of the data.

FCA's increasing importance has many reasons. For one, new scientific areas have recently begun to incorporate computer technologies on a large scale through data-bases, whence a large production of data and the need for handling them. Secondly, the increasing power of computers permits the automatization of tasks that might have, in the worst case, exponential time-space costs. Among them, the typical problems from FCA related to the representation of a lattice that could have exponential size relative to the size of the original data.

In data mining, the problem of classification is naturally related to the notion of concept from FCA. Unsupervised classification consists in grouping objects that have close attributes while separating those having distant attributes; supervised classification groups objects having a same label (called class), while it distinguishes those having different labels. The notion of concept is then repeatedly used in applications to classify data, in a supervised or unsupervised way. Dependencies between attributes are classically represented by rules. Associative rules, well-known in data mining, can be either exact or approximate rules. For relational data-bases, systems of rules are known under the name of functional dependencies. The combinatorial explosion of the number of rules and the growing volume of data to be handled have made the use of concise representations, called bases, necessary.

ISBN 978–84–695–5252–0, Universidad de Málaga (Dept. Matemática Aplicada)

In *lattice theory*, a fundamental representation theorem establishes that every lattice is the concept lattice of its *binary table* [1] defined from irreducible elements of the lattice - particular elements that are not a join or a meet of other elements. A second and less known representation theorem states that every finite lattice is isomorphic to a lattice of closed sets, where the closure operator can be defined by a basis of rules defined on the join-irreducible elements of the lattice.

There can be many equivalent bases, where two bases are equivalent if they give rise to the same closed set lattice. The canonical direct basis is the only one that is direct, meaning that the closure of an arbitrary set can be computed by just one application of the rules to the set, and that, at the same time, is minimal among the direct systems. Moreover, it has been shown in [2] that this basis is equivalent to five other bases whose rules are called *minimal functional dependency* in the domains of relational databases [10], and *proper implications* in the data-mining area research [13] whose premises are minimal generators of the lattice.

In this paper, we introduce the *dependence graph* as a representation of a lattice defined on the set of its join-irreducible elements. This graph, issued from the *dependence relation* on a lattice [1], is a nice structure encoding together the minimal generators and the canonical direct basis of a lattice. Representation of a lattice in the form of an edge-labeled graph was first suggested in [11]. This OD-Graph is closely associated to the *D-relation* on the set of join-irreducibles of a lattice, subset of the dependence relation, that was crucial in the study of free and lower bounded lattices [6].

After a first section of definitions, we define the dependence graph of a lattice. In the last section, we discuss about existing generation algorithms of the dependence graph, and we propose a new generation algorithm.

2 Definitions

Lattice. In lattice theory, the structure of lattice can been introduced either as an algebraic structure provided with two operators named lower and upper bounds, or as an ordered structure defined by the existence of particular elements called upper and lower bounds [3].

More formaly, a lattice is an order relation $\leq$ on a set S (i.e. a reflexiv, antisymmetic and transitiv relation) where every couple of elements has a a join and a meet. The *meet* (resp. *join*) of x and y, denoted $x \wedge y$ (resp. $x \vee y$), is the unique greatest lower bound (resp. least upper bound) of x and y.

Meet and join elements are defined in a more general but identical manner for a subset $X \subseteq S$: the meet of X, noted $\vee X$, is the unique greatest element of the predecessors of X, while the join of X, noted $\wedge X$, is the unique least element of the successors of X. As a direct consequence, any lattice admits a unique maximal element called *top* and denoted $\top$ or 1, and a unique minimal element called *bottom* and denoted $\bot$ or 0.

The *strict order relation* of any order relation $\leq$, denoted $<$, is an antisymmetric, transitive and irreflexive relation defined by $x < y$ if $x \leq y$ and $x \neq y$. It corresponds to the reflexive reduction of $\leq$. The *cover relation* of $\leq$, denoted $\prec$, is an antisymmetric relation defined by $x \prec y$ if $x < y$, and there is no z so that $x < z < y$. We then say that y *cover* x. It corresponds to the reflexive and transitive reduction of $\leq$. The *Hasse diagram* is a graphical representation of an order where only the arcs of the cover relation $\prec$ are represented since reflexivity and transitivity edges can be deduced.

Irreducibles elements. An element of a lattice is called *reducible* if it corresponds to a meet and a join of two distinct elements. Otherwise, it is called *irreducible*. More precisely, an element j is called a *join-irreducible* if for any subset X of elements, $j = \vee X$ implies that $j \in X$. An element m is called *meet-irreducible* if for any subset X of elements, $m = \wedge X$ implies that $m \in X$. The set of join-irreducibles of a lattice L is usually denoted J_L, and the set of meet-irreducibles M_L. In particular, we have $\bot = \vee\emptyset$ and $\top = \wedge\emptyset$ implying that $\bot$ is not a join-irreducible, and $\top$ is not a meet-irreducible.

A nice characterization establishes that an element is a join-irreducible if, and only if, it covers only one element, denoted j^-, while an element is a meet-irreducible if, and only if, it is covered by only one element, denoted m^+.

Any element $x \in S$ of a lattice L is the join of its predecessors, and the meet of its successors. The latticial property implies a reduction to join-irreducible predecessors and meet-irreducible successors:

$$x = \vee J_x = \vee\{y \in J_L \ : \ y \leq x\} \tag{1}$$

$$x = \wedge M_x = \wedge\{y \in M_L \ : \ y \geq x\} \tag{2}$$

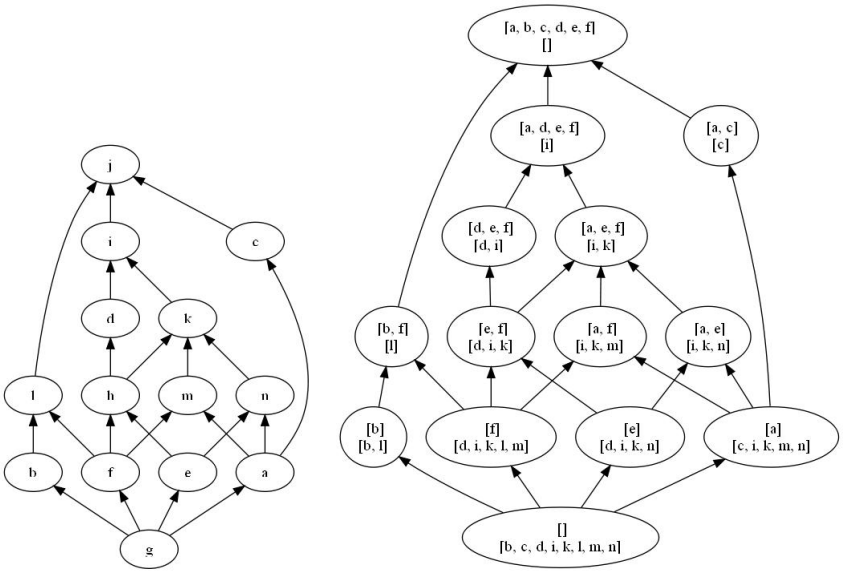
Therefore, irreducible elements are enough to build the lattice in its entirety, using either join irreducibles for reconstruction by upper bound, or meet-irreducibles for reconstruction by lower bound. Moreover, J_L and M_L are minimal set allowing reconstruction.

Minimal generators. Consider one element $x \in S$. Although x is the join of J_x, J_x is not always the minimal subset to define x as a join. A minimal subset to obtain x as a join, including in J_x, is named a *basis*, a *minimal generating set* or a *minimal generator* for x. More formally, a *minimal generator* of x is a subset B of J_x such that $x = \vee B$ and B is inclusion-minimal, i.e for all $A \subset B$, then $x \neq \vee A$. The family $\mathcal{B}_x$ of minimal generators of x is then:

$$\mathcal{B}_x = \{B \subseteq J_x \ : \ x = \vee B \text{ and } x \neq \vee A \text{ for all } A \subset B\} \tag{3}$$

The dual observation for M_x is valid for a reconstruction of x as meet. The number of minimal generators of x can be exponential in the cardinality of J_x in the worst case.

Consider the example of lattice in Figure 1(a). Six elements possess a single incoming arc, forming all join-irreducibles ; the meet-irreducibles, characterized



(a) A lattice

(b) The isomorphic lattice with the sets J_x and M_x inside each node x

Fig. 1. Example of lattice

by a single outcoming arc, are heigth:

$$J = \{a, b, c, d, e, f\} \tag{4}$$

$$M = \{b, c, d, i, k, l, m, n\} \tag{5}$$

These irreducible elements are used to describe more precisely the elements of the lattice (see Figure 1) where node of each element x contains its join-irreducible predecessors J_x and its meet-irreducible successors M_x. Minimal generators are given in Table 1. One can observe that each join-irreducible possesses itself as unique minimal generator ; the top element possesses 4 minimal generators.

x	a	b	c	d	e	f	g
J_x	a	b	ac	def	e	f	$\emptyset$
$\mathcal{B}_x$	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{e\}$	$\{f\}$	$\{\emptyset\}$

x	h	i	j	k	l	m	n
J_x	ef	adef	abcdef	aef	bf	af	ae
$\mathcal{B}_x$	$\{ef\}$	$\{ad\}$	$\{ab, bc, bd, dc\}$	$\{aef\}$	$\{bf\}$	$\{af\}$	$\{ae\}$

Table 1. Minimal generators of the lattice in Figure 1(a)

3 Dependence graph and canonical direct basis of a lattice.

Dependence graph. The dependence graph of a lattice is defined on the set J_L of join-irreducible elements.It is an edge-labeled directed graph whose edges corresponds to the dependence relation δ of a lattice [1], and are labeled by some minimal generators, thus a size that can be exponential in the cardinality of J_L. More precisely, the dependence graph of a lattice L is a pair (δ, ω) where:

- δ is the *dependence relation* [1] defined on J_L by $j\delta j'$ if there exists $x \in S$ such that $j \not\leq x$, $j' \not\leq x$ and $j < j' \vee x$. We note $j\delta_x j'$.
- ω is a label of the edges defined on $\mathcal{P}(J_L)$, for each relation $j\delta j'$, by:

$$\omega(j, j') = \{\text{minimal generators of } x \ : \ j\delta_x j' \text{ and } x \text{ minimal in the lattice}$$

A pair $(j, j') \in \delta$ can be denoted either $j\delta_x j'$ or $j\delta_B j'$, with B minimal generator of x. Figure 2 represents the dependence graph of lattice in Figure 1(a).

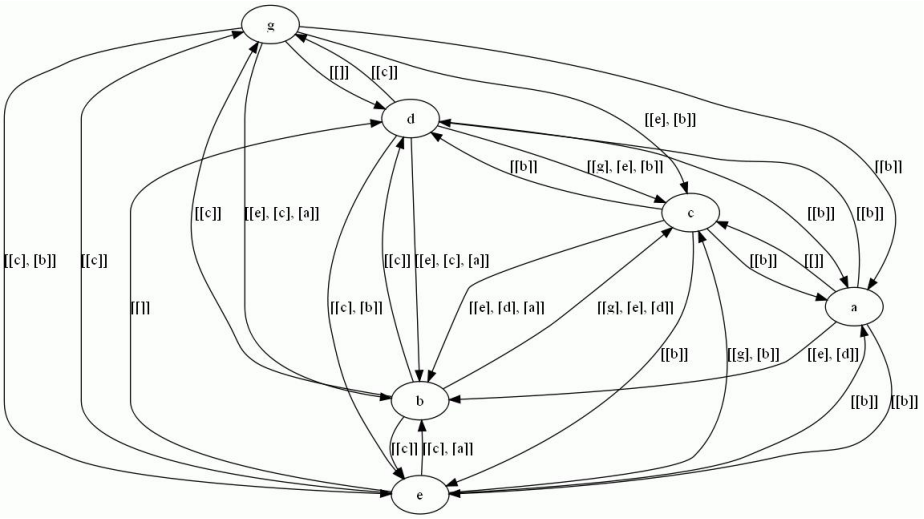


Fig. 2. Dependence graph of the lattice in Figure 1(a)

The subgraph $\delta_\emptyset$ of the dependence graph to the edges containing the empty set as label corresponds to the subgraph of the lattice induced by its join-irreducible, since $j < j'$ implies $j < j' \vee \perp$ and $\omega(j, j') = J_\perp = \{\emptyset\}$. As a direct consequence, a lattice is distributive when edge-labels of its dependence graph all are the empty set.

Canonical direct unit basis. A set of unit implication (or rules) Σ is a binary relation between $\mathcal{P}(S)$ and S where a rule (X, y), generaly denoted $X \to y$, means that X *"implies"* y, with X called the *premisse* and y the *conclusion*.

The dependence graph encodes a set of rules defined on the join-irreducibles J_L of a lattice, with a rule $B + j' \to j$ for each $(j, j') \in \delta_B$. This set of rules forms a particular basis of the lattice called the *canonical direct unit basis*, and denoted Σ_{cdb} [2].

Morover, an important result establishes that every lattice is isomorphic to the *closed set lattice* $(\mathcal{F}, \subseteq)$ of its canonical direct basis, where $\mathcal{F}$ is a family on J_L. This family contains all the *closures* of the basis where a closure is a subset X of J_L verifying all rules, i.e. for each rule $B \to y$, if $B \subseteq X$ then $y \in B$.

Therefore, the dependence graph of a lattice encodes the canonical direct basis from which the lattice reconstruction is possible as a closed set lattice.

4 Generation algorithm

Since the size of the dependence graph can be exponential in the size of the lattice, this generation problem belongs to the more general class of problems having an input of size n, and an output of size N bounded by 2^n. For this class of problems, a classical worst-case analysis makes them exponential, thus NP-hard, with an exponential space. However, a more precise information can be obtained by output-sensitive analysis techniques (see a survey in [9]). These analyzes are relevant since the recent improvements in storage and processing capacity increasingly often allow to handle some exponential data, what was not possible even some time ago. The idea is to consider the time complexity needed to generate only one element of the output (i.e. one rule or one minimal generator in our case).

The time complexity per Σ_{cdb}-rule has then to be considered. Although the most common algorithms have an exponential delay complexity, there exist some algorithms with a polynomial delay complexity.

The definition of minimal generators for an element x induces an exponential generation since any subset of J_x as to be tested. Another strategy is issued from the equivalence between minimal generators and minimal transversals of a closed set, problem known to be exponential. This strategy has been capitalized by Pfaltz's incremental algorithm [12], and by Jen's algorithm [5] used in data-mining to compute minimal generators. Jen's algorithm computes minimal generators from the *faces* of x defined by considering immediate predecessors of x in the lattice.

However, in logic area, the algorithm attributable to Ibaraki et al. ([8]) computes a Σ_{cdb}-rule - and thus the dependence graph - in polynomial time with a family $\mathcal{F}$ of closed set as input.

Algorithm 1 generates the dependence graph with the same polynomial complexity per rule or per minimal generator.

```
Name: dependenceGraph
Data  : A lattice L = (S, ≤)
Result : The dependence graph of the lattice
begin
    compute the set J_L of join-irreducibles of the lattice;
    initialize a graph G with join-irreducibles as nodes;
    compute a topological sort T of the lattice;
    foreach x ∈ T do
        foreach (j, j') ∈ J × J such that x ∨ j' ≥ j do
            add the edge (j, j') in G;
            compute the set J_x of join-irreducible predecessors of x;
            initalize the empty family GM_x;
            let G' subgraph of G induced by J_x on nodes and edges's labels;
            foreach edge (k, k') of G' do
                foreach valuation B of the edge (k, k') do
                    if B ∨ k' = x then add the set B + k' to the family GM_x
                end
            end
            if GM_x is empty then GM_x = {J_x};
        end
    end
    label the edge (j, j') with GM_x;
    return G;
end
```

Algorithm 1: Generation of the dependence graph of a lattice

Proposition 1. *Algorithm 1 generates the dependence graph, the minimal generators, or the canonical direct basis of a lattice* $L = (S, \leq)$ *in* $O(|\Sigma_{cdb}||S||J_L|^3)$*, i.e. in* $O(|S||J_L|^3)$ *per* Σ_{cdb}*-rule or per minimal generator.*

Proof. First, computation of the relation δ on the join-irreducibles can be done in $O(|S|^2|J_L|^2)$ by determining, for each $x \in S$, if x is a minimal element in the lattice such that $j\delta_x j'$.

Computation of edges's labels is more difficult. One can observe that minimal generators are recursively defined according to the relation $\leq$ in the lattice. Indeed, if we consider two join-irreducibles such that $j\delta_x j'$ - with x an element of the lattice - or equivalently $j\delta_B j'$ - with B minimal generator of x - then $B + j'$ is a minimal generator of $x \vee j'$, thus recursively defined from B. One can distinguish between two cases:

- When B is strictly included in J_x, then B can be deduced from a minimal generator of a predecessor of x.
- When $B = J_x$, then J_x is the only minimal generator of x.

Therefore, a travel of the lattice from the bottom to the top allows to recursively compute minimal generators in $O(|\Sigma_{cdb}||S||J_L|^3)$.

The dependence graph of a lattice, and thus its canonical direct basis and its minimal generators, can easily be generated with a binary table as input, after its concept lattice generation. In particular, Bordat's algorithm [4] generates the Hasse diagram of the concept lattice of a binary table from the botom to the top using a successor function. Therefore, another strategy would consists in computing the dependence graph along with the lattice generation.

5 Conclusion

In this paper, we introduced the dependence graph as a representation of a finite lattice encoding both its canonical direct basis and its minimal generators. We propose a new generation algorithm with an improvment complexity. This structure can be used in various domains of computer science, such as data-mining and knowledge representation. Indeed, the canonical direct basis of rules is a nice basis to represent dependencies between attributes, in a classification task for example. The use of minimal generators could gives raise to an attributs set reduction, usefull for data indexation for example.

References

1. M. Barbut and B. Monjardet. *Ordres et classifications : Algèbre et combinatoire.* Hachette, Paris, 1970. 2 tomes.
2. K. Bertet and B. Monjardet. The multiple facets of the canonical direct unit implicational basis. *Theoretical Computer Science*, 411(22-24):2155–2166, 2010.
3. G. Birkhoff. *Lattice theory.* American Mathematical Society, 1st edition, 1940.

4. J.P. Bordat. Calcul pratique du treillis de Galois d'une correspondance. *Math. Sci. Hum.*, 96:31–47, 1986.
5. A. Le Floch, C. Fisette, R. Missaoui, P. vatchev, and R. Godin. Jen: un algorithme efficace de construction de générateurs minimaux pour l'identification de règles d'association. *Nouvelles Technologies de l'Information (numéro spécial)*, 1(1):135–146, 2003.
6. R. Freeze, J. Jezek, and J.B. Nation. Free lattices. In Providence, editor, *Mathematical survey and monographs. Americal Mathematical Society*, volume 42, 1995.
7. B. Ganter and R. Wille. *Formal concept analysis, Mathematical foundations.* Springer Verlag, Berlin, 1999.
8. T. Ibaraki, A. Kogan, and K. Makino. Functional dependencies in Horn theories. *Artificial Intelligence*, 108:1–30, 1999.
9. E. Lawler, J.K. Lenstra, and A.H.G. Rinnoy kan. Generating all maximal independant sets: Np-hardness and polynomial time algorithms. *SIAM Journal on Computing*, 9:558–565, 1980.
10. D. Maier. *The Theory of Relational Databases.* Computer Sciences Press, 1983.
11. J. B. Nation. An approach to lattice varieties of finite height. *Algebra Universalis*, 27(4):521–543, 1990.
12. JL. Pfaltz and CM. Taylor. Scientific discovery through iterative transformations of concept lattices. In *Workhop on Discrete Applied Mathematics, in conjonction with the* 2^{nd} *SIAM International Conference on Data-Mining*, pages 65–74, 2002.
13. R. Taouil and Y. Bastide. Computing proper implications. In 9^{th} *International Conference on Conceptual Structures*, Stanford, USA, 2002.