Parallel Algorithms for general Galois lattices building

Fatma BAKLOUTI , Gérard LEVY
CERIA, University PARIS IX Dauphine
FatmaBaklouti@dauphine.fr
GerardLevy@dauphine.fr

0. Abstract- Standard Galois Lattices are very useful tools in data mining. They allow us to structure data sets, by extracting concepts and rules to deduce concepts from other concepts. They concern binary data arrays, called *contexts*. Several algorithms are well known and used to determine the Galois lattice *GL(C)* which is associated to a given context *C*, when the size of C is not too large. Today, we need to treat contexts which are large and not necessarily binary. Since it is difficult to determine GL(C) for large contexts C on one only computer, we propose to partition C depending on its rows or columns, to build on different computers Galois lattices associated to sub-contexts, and then to determine the global lattice from these lattices. Moreover, since it is not much more difficult to deal with general contexts than with standard contexts, we shall work on general contexts and Galois lattices.

Our paper is presented as follows:

- Terminology of Galois lattices;
- Algebraic tools to share the context *C* into contexts C_1, C_2, depending on rows or columns;
- Algorithms to determine *GL(C)* from $GL(C_1)$, $GL(C_2)$.
- Computer implementation.
- Conclusion
- References.
- Appendixes: Examples.

1. General Galois Lattices.

1.1: By definition, a <u>lattice</u> is a mathematical structure $F = < F, \leq, \vee, \wedge, 0_F, 1_F >$, where *F* is a partially ordered set by the relationship $\leq$, with a largest element 1_F , a smallest element 0_F , and $\vee$, $\wedge$ are internal composition laws of *sup* (or supremum), and *inf* (or infimum).

In many situations *F* is the Cartesian product of several lattices $F_j = < F_j , \leq_j, \vee_j, \wedge_j, 0_{Fj}, 1_{Fj} >$,

for $j \in J=\{1,..., n\}$. We write this $F = F_1 x...x F_j x...x F_n = \prod_{j=1}^{n} F_j$.

The relationship $\leq$ on *F* is defined by $z= (z_1 ,..., z_j , ..., z_n) \leq t= (t_1, ..., t_j , ..., t_n)$ iff $z_j \leq_j t_j$ for each *j* of *J* .

And we put $z \vee t = (..., z_j \vee_j t_j ,....)$, $z \wedge t= (..., z_j \wedge_j t_j , ...)$, $O_F = (...., O_{Fj} , ...)$, $1_F = (..., 1_{Fj} , ...)$.

(For standard Galois lattices, we have for each j : $F_j = \{0, 1 \}$, $0 < 1$, $0 \vee_j 0 = 0$, $0 \vee_j 1 = 1 \vee_j 0 = 1 \vee_j 1 = 1$, $0 \wedge_j 0 = 0 \wedge_j 1 = 1 \wedge_j 0 = 0$, $1 \wedge_j 1 = 1$. So $O_F = (...., 0,...)$, and $1_F = (..., 1,...)$.)

1.2 : <u>Contexts and descriptions:</u>

Let *m* be a finite positive integer, $I = \{1,..., m \} = [1.. m]$, and *F* any lattice. Let $d: I \rightarrow F$ be any mapping from *I* to *F*. By definition, the array with rows $d(i)$, $i = 1, ..., m$, is a context *C*. It gives for each *individual*, or *object i* of *I*, its *description* $d(i) = (d_1(i),..., d_j(i),..., d_n(i)) \in F = F_1 x .. x F_n$, according to the *attributes* or *properties* $j \in J$. (In standard case, $d_j(i)=1$ means that *i* has property *j*, and $d_j(i) = 0$ means that *i* has not the property *j* .) So, in general case, C is an *m x n* array of elements $d_j(i)$ of *F*, and for each individual *i* and each property *j*, $d_j(i)$ is the value of this property for *i* .

1.3 : <u>Galois connexion</u>:

Let $C = <I, F, d >$ be a context. We define $E = 2^I = P(I)$ and $f: E \rightarrow F$ as follows: for each subset *X* of *I*, $f(X) = \bigwedge \{ d(i) : i \in X \}$ if $X \neq \emptyset$, and $f(\emptyset) = 1_F$.

So, $f(X)$ is the infimum of the descriptions $d(i)$ of elements i of X, and in standard case $f(X)$ is an element $z=(...z_j,...)$ of F, and $z_j = \bigwedge_j \{d_j(i) : i \in X\} = 1$, iff $d_j(i) = 1$, for each i of X. This means that $z_j= 1$ iff j is a property which belongs to all i in X. And for this reason, we call $f(X)$ the *intention* of X.

Remark: for each i of I, we have $f(\{i\}) = d(i)$, and f is decreasing (if $X \subset X' \subset I$ then $f(X') \leq f(X)$) .

Now, we define $g: F \rightarrow E$ by $g(z) = \{i \in I : z \leq d(i)\}$, for each z of F .

We say that $g(z)$ is the *extension* of z. (In standard case, $g(z)$ is the set of all individuals who have all properties of z , $z_j = 1$, .).

We can see that g is also a decreasing mapping.

The ordered pair (f, g) is called a *Galois connexion* . From it we define two other mappings:

$h : P(I) \rightarrow P(I)$, by $h = g \circ f$, and $k = F \rightarrow F$ by $k = f \circ g$.

So, for each subset X of I, we have $h(X) = g(f(X)) = \{i \in I : f(X) \leq d(i)\}$, and for each z of F , we have $k(z) = f(g(z)) = \bigwedge \{ d(i) : i \in g(z) \}$.

We can see that h and k are *closure operators*. This means that each of them is an increasing, extensive, and idempotent operator. More explicitly, for each X, X' of E, and z , z' of F :

$X \subset X'$ implies that $h(X) \subset h(X')$, $z \leq z'$ implies that $k(z) \leq k(z')$;

$X \subset h(X)$, $z \leq k(z)$;

$h(h(X)) = h(X)$, $k(k(z)) = k(z)$.

Any subset X of I such that $X = h(X)$ is called a *I - closed set*, and each z of F such that $z = k(z)$ is called a *F-closed element* .

Let us define $H = \{X \subset I: h(X) = X\}$ the set of all closed subsets of I , and $K = \{z \in F : z = k(z)\}$, the set of all closed elements of F .

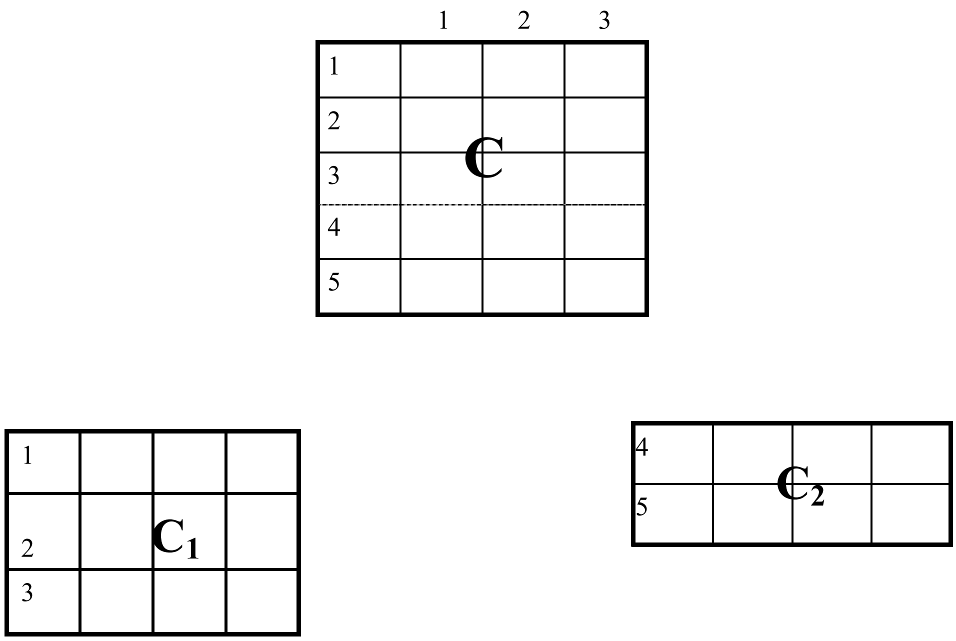
One can proof that there is a bijection between H and K. The ordered pairs $(X, z) \in H \times K$ buch that

$f(X)= z$, and therefore such that $g(z) = X$, are called the *concepts* associated with the context C.

The set of all such concepts constitutes the *Galois lattice* $GL(C)$ associated with this context C. (The order relationship on $GL(C)$ is defined by $(X, z) \leq (X', z')$ iff $X \subset X'$ and $z' \leq z$.)

2. **Row-sharing of the array C.**

We suppose that we have an algorithm to determine the Galois lattice $GL(C)$ associated with a general context C, when the array C is not to large. Now we suppose that C is too large to be solved with one only computer. We partition the array associated with $C = < I, F, d >$ into two sub-arrays.



Let us write $I = [1...m] = [1... m_1] \cup [m_1+1... m] = I_1 \cup I_2$. And let us define the two contexts $C_1 = <I_1, F, d>$, and $C_2 = < I_2, F, d >$. (We have shared the array associated with C into two arrays of $m_1 \; x \; n$ and $m_2 \; x \; n$ cells respectively, $m = m_1 + m_2$.)

Let us call $T_1 = GL(C_1)$ and $T_2 = GL(C_2)$ the Galois lattices associated with those contexts.

We want to build $T = GL(C)$ from T_1 and T_2. We use the following propositions:

<u>Proposition 2.1</u> : For each ordered pair (X, z) of $GL(C)$, let us define $X_1 = X \cap I_1$ and $X_2 = X \cap I_2$.

Then we have $X = X_1 \cup X_2$, and $(X_1, f_1(X_1))$ belongs to T_1 and $(X_2, f_2(X_2))$ belongs to T_2.

Therefore to determine the (X, z) *that* belong to T, we have necessarily $X = X_1 \cup X_2$, and $(X_1, f_1(X_1)) \in T_1$ and $(X_2, f_2(X_2)) \in T_2$.

<u>Proposition 2.2</u> : If $(X_1, z_1) \in T_1$ and $(X_2, z_2) \in T_2$, let $X = X_1 \cup X_2$ and $z = z_1 \wedge z_2$.

Thus we have $(X, z) \in T$ if and only if $g_1(z) = g_1(z_1) = X_1$, and $g_2(z) = g_2(z_2) = X_2$.

(by definition, $g_j(z) = \{i \in I_j : z \leq d(i)\}$, and $f_j(X)= f(X)$, for $j = 1, 2$.)

3.Column-sharing of C.

Here we examine the situation of splitting the array C $(m \; x \; n)$ into two subsets $C_1 \; (m \; x \; n_1)$ and $C_2 \; (m \; x \; n_2)$, with $n= n_1+ n_2$. Let us write $J = [1..n]= [1..n_1] \cup [n_1+1..n] = J_1 \cup J_2$.

From $F = (F_1 x...x F_{n1}) \; x \; (F_{n1+1} x ... x F_n)$, we define $\boldsymbol{F_1} = F_1 x... x F_{n1}$, $\boldsymbol{F2} = F_{n1+1} x...x F_n$, and for each $z = (z_1,..., z_{n1}, z_{n1+1},..., z_n)$ of F, we write $\boldsymbol{z_1}= (z_1,...,z_{n1}) \in \boldsymbol{F_1}$, and $\boldsymbol{z_2} = (z_{n1+1},..., z_n) \in \boldsymbol{F_2}$.

In such a way that we have $F = \boldsymbol{F_1} \; x \; \boldsymbol{F_2}$, $z = (\boldsymbol{z_1} , \boldsymbol{z_2})$, and that $z \leq t$ iff ($\boldsymbol{z_1} \leq \boldsymbol{t_1}$ and $\boldsymbol{z_2} \leq \boldsymbol{t_2}$) .

For each i of I , we define $\boldsymbol{d_1}(i) = (d_1(i),..., d_{n1}(i)) \in \boldsymbol{F_1}$, and $\boldsymbol{d_2}(i) = (d_{n1+1}(i),..., d_n(i)) \in \boldsymbol{F_2}$.

So, we can therefore define the two contexts $C_1 = < I, \boldsymbol{F_1}, \boldsymbol{d_1} >$ and $C_2 = < I, \boldsymbol{F_2}, \boldsymbol{d_2} >$ from context C.

Now, we can establish the two following properties:

<u>Proposition 3.1</u> :

For each (X, z) of $T = GL (C)$, let us write $z = (\boldsymbol{z_1}, \boldsymbol{z_2})$, and let us define $X_1 = g_1(\boldsymbol{z_1}) = \{i \in I : \boldsymbol{z_1} \leq \boldsymbol{d_1}(i)\}$, $X_2 = g_2(\boldsymbol{z_2}) = \{i \in I : \boldsymbol{z_2} \leq \boldsymbol{d_2}(i)\}$. Then we have $X = X_1 \cap X_2$, and $(X_1, \boldsymbol{z_1}) \in T_1 = GL(C_1)$, *and* $(X_2, \boldsymbol{z_2}) \in T_2 = GL(C_2)$.

This proposition shows that in order to obtain the elements z witch are F- closed, we need to find those $z = (\boldsymbol{z_1}, \boldsymbol{z_2})$ with $\boldsymbol{z_1}$ $\boldsymbol{F_1}$ - closed and $\boldsymbol{z_2}$ $\boldsymbol{F_2}$ - closed .

Conversely, we have:

<u>Proposition 3.2</u> :

Let $(X_1, \boldsymbol{z_1}) \in T_1$ and $(X_2, \boldsymbol{z_2}) \in T_2$. Let us define $z = (\boldsymbol{z_1}, \boldsymbol{z_2})$, and $X = X_1 \cap X_2$.

Then we have $(X, z) \in T = GL(C)$, iff $f_1(X) = \boldsymbol{z_1}$ and $f_2(X) = \boldsymbol{z_2}$.

(By definition $f_1(X) = \bigwedge \{\boldsymbol{d_1}(i) : i \in X\}$, and $f_2(X) = \bigwedge \{\boldsymbol{d_2}(i) : i \in X\}$.)

4. Parallel implementation of row-sharing of context C.

We suppose that we have three workstations S, S_1 and S_2.

Each S_j, $j=1, 2$, can communicate with S ;

For $j = 1, 2$, S_j contains context C_j and has determined Galois lattice $T_j = GL(C_j)$.

We suppose that $T_j = \{(tX_j(i_j), tz_j(i_j))\}$, with $i_j = 1,..., nf_j$ (nf : number of closed pairs).

Now station S is able to determine $T= GL(C) =\{(tX(k), tz(k))\}, k=1,..., nf$, as follows.

k :=0 ;

For i_1:= 1 to nf_1 do

begin

S_1 sends X_1= $tX_1(i_1)$, and $z_1 = tz_1(i_1)$ to S ;

For i_2 := 1 to nf_2 do

begin

S_2 sends $X_2 = tX_2(i_2)$, and $z_2 = tz_2(i_2)$ to S ;

S calculates $X = X_1 \cup X_2$ and $z = z_1 \wedge z_2$;

S sends z to S_1, and asks it to calculate $U_1= g_1(z)= \{i \in I_1 : z \leq d(i) \}$;

S sends z to S_2, and asks it to calculate $U2 = g_2(z)= \{i \in I_2 : z \leq d(i) \}$;

S_1 returns U_1 to S;

S_2 returns U_2 to S;

On station S: If $(X_1= U_1)$ and $(X_2= U_2)$ then S does:

k :=k+1 ; tX(k) :=X ; tz(k) :=z ;

end ;

nf :=k ;

end ;

At the end of the session, lattice *T = GL(C)* is located on station *S*.

5. **Implementation of column- sharing of context C**.

In order to implement a column-sharing of context *C*, a quite similar algorithm to the precedent one is obtained, using the same configuration of three stations S, S_1, S_2.

Each S_j, $j=1, 2$, can communicate with *S* ;

For $j=1, 2$, S_j contains context *Cj* and has determined Galois lattice $T_j = GL(C_j)$.

We suppose that $T_j = \{(tX_j(i_j), tz_j(i_j))\}$, with $i_j = 1,..., nf_j$.

Now, station *S* is able to determine $T= GL(C) =\{(tX(k), tz(k))\}, k=1,..., nf$, as follows.

k :=0 ;

For i_1: =1 to nf_1 do

begin

S_1 sends X_1= $tX_1(i_1)$, and $z_1 = tz_1(i_1)$ to S ;

For i_2: =1 to nf_2 do

begin

S_2 sends X2 = $tX_2(i_2)$, and $z_2 = tz_2(i_2)$ to S ;

S calculates $X = X_1 \cap X_2$ and $z = (z_1 , z_2)$;

S sends X to S_1, and asks it to calculate $t_1= f_1(X)= \bigwedge \{ d_1(i) : i \in X \}$;

S sends X to S_2, and asks it to calculate $t_2 = f_2(X)= \bigwedge \{ d_2(i) : i \in X \}$;

S_1 returns t_1 to S;

S_2 returns t_2 to S;

On station S: If $(z_1= t_1)$ and $(z_2= t_2)$ then S does:

k :=k+1 ; tX(k) :=X ; tz(k) :=z ;

end ;

nf :=k ;

end ;

At the end of the session, lattice *T= GL(C)* is located on station *S*.

6. **Conclusion**

The aim of this paper was to propose a way to parallelise large contexts of general lattices.

Supposing the existence of an algorithm to compute lattices in the case of general contexts, we have implemented more general algorithms.

The idea of sharing contexts into two subsets could be easily extended to multipartitions and to many workstations; one could imagine building different architectures for networks of stations to manage very complex situations.

This feature is now experimented on SDDS systems which have been developed at CERIA Center, University Paris IX- Dauphine.

7. **References:**

[1] E.Diday, R.Emilion . Treillis de Galois maximaux et capacités de Choquet. C.R.Acad.Sci. Paris, t.325,Série I, p.261-266, 1997.

[2] P.Njiwoua, E. Mephu Nguifo. A parallel algorithm to build concept lattice. In Proceedings of 4th Groningen Intl. Information Tech. Conf. For Students, pp. 103-107, 1997.

[3] R. Godin. Complexité de structures de treillis. Ann. SC. Math. Quebec, 13(1) : 19-38, 1989.

[4] R. Concept Lattices and Knowledge Systems. Computer Mathematic Applied, 23(6-9) :493-515, 1992.

[5] A. Guénoche. Construction du treillis de Galois d'une relation binaire. Math. Inf. Si. Hum.(28ème année, n° 109, 1990, p 41-53)

[6] Ganter B. Two basic algorithms in concept analysis. Preprint 831, Technische Hochschule Darmstadt, 1984, 28 p .

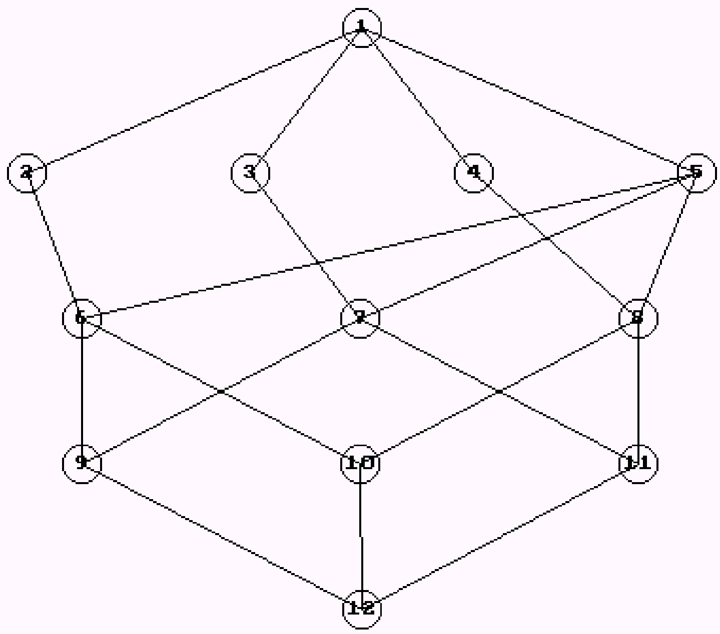
[7] R. Godin, G.W. Mineau, R. Missaoui. Incremental structuring of knowledge bases. Proceedings of the International knowledge retrevial, use, and storage for efficiency Symposium (KRUSE' 95), Santa Cruz : pp 179-198.

8. Appendice 1: An example of horizontal product of lattices

Let us consider the context C which is given by the following array, where
m = 7 , n = 3, I = [1...7], $b_1 = 2$, $b_2 = 3$, $b_3 = 3$. (b = 1_F)

j → / i ↓	1	2	3
1	1	0	2
2	2	1	0
3	0	3	1
4	1	1	1
5	0	1	3
6	0	0	2
7	2	0	0

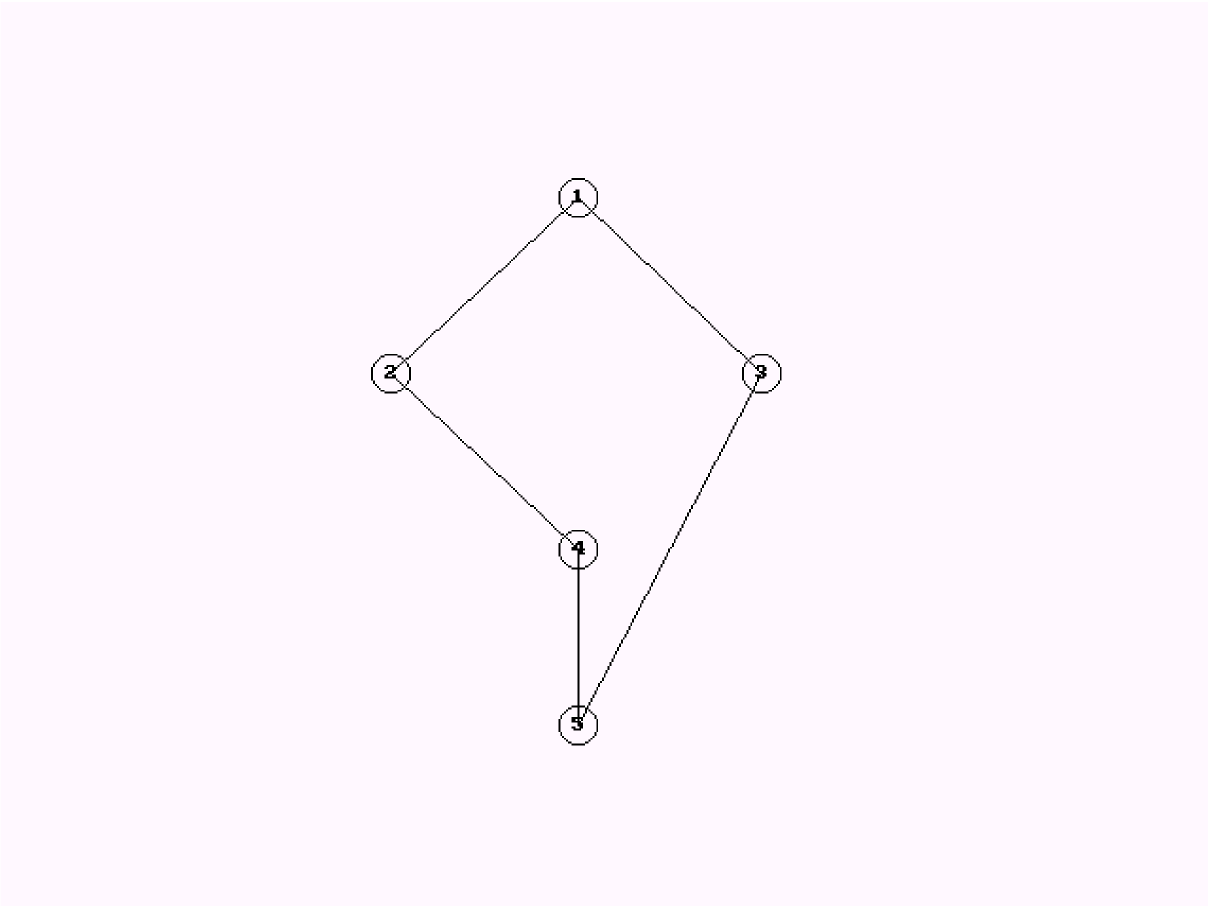
The first 4 rows constitute context C_1, and the 3 last ones constitute context C_2.
We shall determine $T_1 = GL(C_1)$, $T_2 = GL(C_2)$, T = GL(C), and then show that T is egal to the horizontal product of T_1 and T_2.



Graph of lattice T_1 = GL (C_1)

We apply an algorithm (here Bordat's algorithm) to context C_1 to build lattice T_1 = GL (C_1). We obtain:
Total number of closed pairs (X, z) of lattice T_1 =GL (C_1) =12.

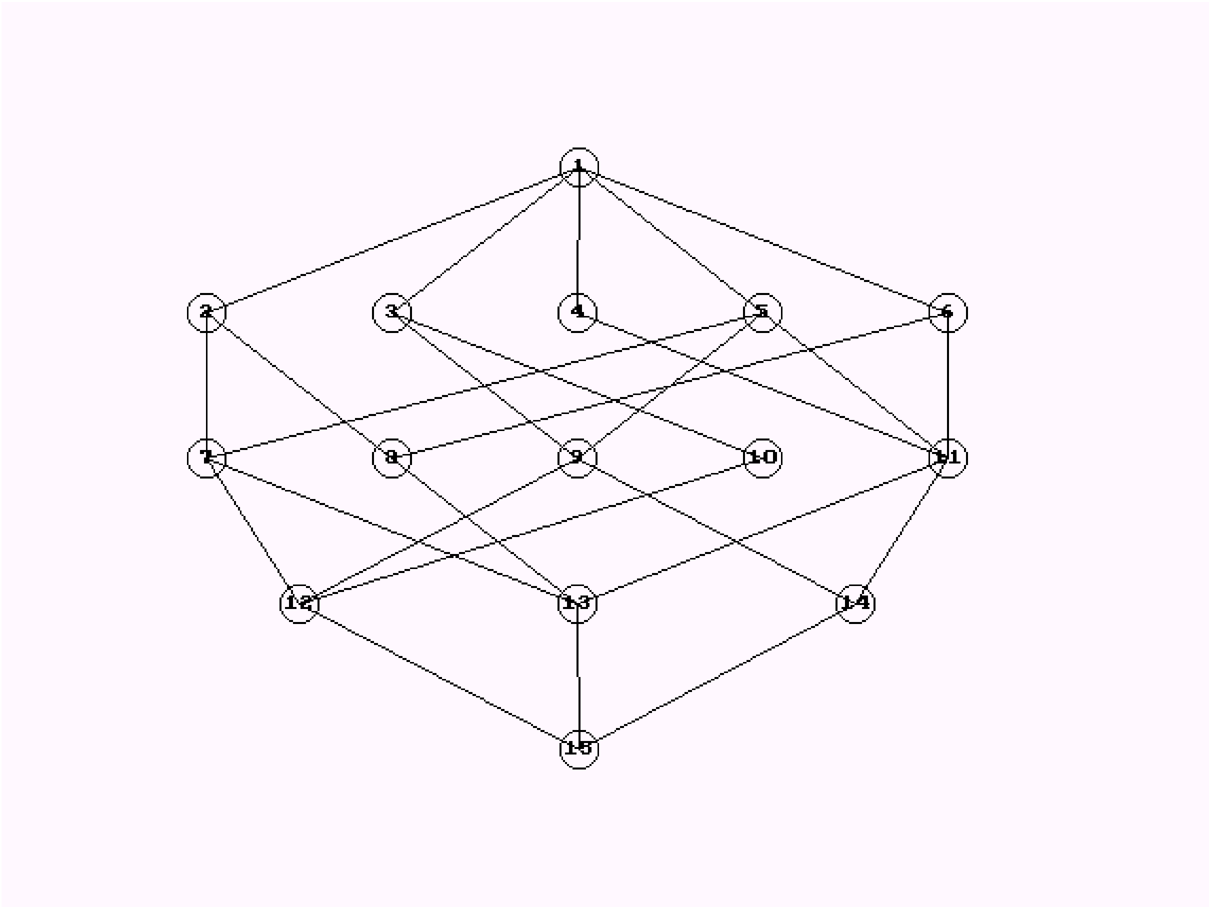
pair(1)= X={}, z=(2,3,3)/ pair(2)= X={1}, z=(1,0,2)/ pair(3)= X={2}, z=(2,1,0),
pair(4)= X={3}, z=(0,3,1)/ pair(5)= X={4}, z=(1,1,1)/pair(6)= X={1,4}, z=(1,0,1),
pair(7)= X={2,4}, z=(1,1,0)/ pair(8)= X={3,4}, z=(0,1,1)/ pair(9)= X={1,2,4}, z=(1,0,0),
pair(10)= X={1,3,4}, z=(0,0,1)/ pair(11)= X={2,3,4}, z=(0,1,0)/ pair(12)= X={1,2,3,4}, z=(0,0,0).



Graph of lattice T_2 = GL (C_2)

Now we apply the same algorithm to context C_2 to build lattice T_2 = GL (C_2). We obtain:
Total number of closed pairs of T_2 =5

pair (1)= X={}, z=(2,3,3) /pair(2)= X={5}, z=(0,1,3) /pair(3)= X={6}, z=(2,0,0),
pair (4)= X={5,6}, z=(0,0,2) / pair(5)= X={5, 6, 7}, z=(0,0,0).



<u>Graph of lattice T = GL(C)</u>

Applying BORDAT's algorithm to the full context C, we obtain
Total number of closed pairs (X, z) of T = GL(C) =15

pair(1)= X={}, z=(2,3,3) /pair(2)= X={1}, z=(1,0,2) /pair(3)= X={2}, z=(2,1,0),
pair (4)= X={3}, z=(0,3,1) /pair(5)= X={4}, z=(1,1,1)/ pair(6)= X={5,}, z=(0,1,3),
pair(7)= X={1,4}, z=(1,0,1) /pair(8)= X={1,5,6}, z=(0,0,2)/pair(9)= X={2,4}, z=(1,1,0),
pair(10)= X={2,7}, z=(2,0,0)/pair(11)= X={3,4,5}, z=(0,1,1)/pair(12)= X={1,2,4,7}, z=(1,0,0),
pair(13)= X={1,3,4,5,6}, z=(0,0,1)/pair(14)= X={2,3,4,5}, z=(0,1,0)/pair(15)= X={1,2,3,4,5,6,7}, z=(0,0,0).

X1↓ | X2→	{}	{5}	{5,6}	{5,6,7}	{7}
z1 ↓ | z2→	(2,3,3)	(0,1,3)	(0,0,2)	(0,0,0)	(2,0,0)
{}	{}	{5}	{5,6}	{5,6,7}	{7}
(2,3,3)	(2,3,3)	(0,1,3)	(0,0,2)	(0,0,0)	(2,0,0)
{1}	{1}	{1,5}	{1,5,6}	{1,5,6,7}	{1,7}
(1,0,2)	(1,0,2)	(0,0,2)	(0,0,2)	(0,0,0)	(1,0,0)
{1,2,4}	{1,2,4}	{1,2,4,5}	{1,2,4,5,6}	{1,2,4,5,6,7}	{2,4,7}
(1,0,0)	(1,0,0)	(0,0,0)	(0,0,0)	(0 ,0,0)	(1,0,0)
{1,2,3,4}	{1,2,3,4}	{1,2,3,4,5}	{1,2,3,4,5,6}	{1,2,3,4,5,6,7}	{1,2,3,4,7}
(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
{1,3,4}	{1,3,4}	{1,3,4,5}	{1,3,4,5,6}	{1,3,4,5,6,7}	{1,3,4,7}
(0,0,1)	(0,0,1)	(0,0,1)	(0,0,1)	(0,0,0)	(0,0,0)
{1,4}	{1,4}	{1,4,5}	{1,4,5,6}	{1,4,5,6,7}	{1,3,4,7}
(1,0,1)	(1,0,1)	(0,0,1)	(0,0,2)	(0,0,0)	(0,0,0)
{2}	{2}	{2,5}	{2,5,6}	{2,5,6,7}	{2,7}
(2,1,0)	(2,1,0)	(0,1,0)	(0,0,0)	(0,0,0)	(2,0,0)
{2,3,4}	{2,3,4}	{2,3,4,5}	{2,3,4,5,6}	{2,3,4,5,6,7}	{2,3,4,7}
(0,1,0)	(0,1,0)	(0,1,0)	(0,0,0)	(0,0,0)	(0,0,0)
{2,4}	{2,4}	{2,4,5}	{2,4,5,6}	{2,4,5,6,7}	{2,4,7}
(0,1,0)	(1,1,0)	(0,1 ,0)	(0,0,0)	(0,0,0)	(1,0,0)
{3}	{3}	{3,5}	{3,5,6}	{3,5,6,7}	{3,7}
(0,3,1)	(0,3,1)	(0,1,1)	(0,0,1)	(0,0,0)	(0,0,0)
{3,4}	{3,4}	{3,4,5}	{3,4,5,6}	{3,4,5,6,7}	{3,4,7}
(0,1,1)	(0,1,1)	(0,1,1)	(0,0,1)	(0,0,0)	(0,0,0)
{4}	{4}	{4,5}	{4,5,6}	{4,5,6,7}	{4,7}
(1,1,1)	(1,1,1)	(0,1,1)	(0,0,1)	(0,0,0)	(1,0,0)

T = GL(C) is the horizontal product of lattices T_1 = GL (C_1) and T_2 = GL (C_2)

The first column of the array contains the closed pairs (X_1, z_1) of T_1, and the first row contains the closed pairs of T_2.

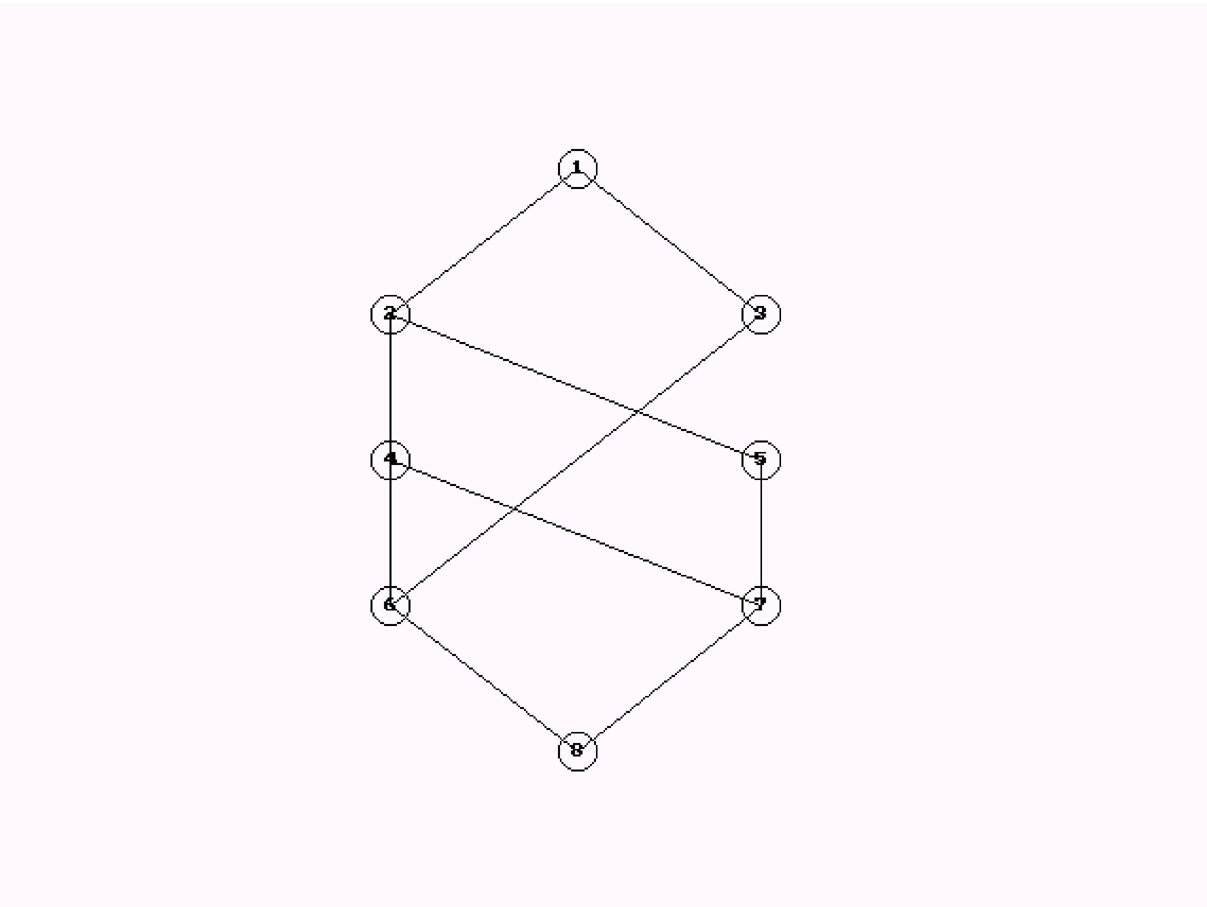
The cell at the intersection of row i and column j corresponds to the pairs (X_i , z_i) of T_1 and (X_j , z_j) of T_2.

When it gives a closed pair of T, this box is drawn with a thick line.

Here we find 15 closed pairs for the horizontal product, and they are effectively all the pairs of T, as we can see it.

$$\begin{array}{ccc} & & (X_2 , z_2) \\ & & \downarrow \\ (X_1 , z_1) & \rightarrow & X = X_1 \cup X_2 \ , z = z_1 \wedge z_2 . \end{array}$$

9. Appendice 2: An example of vertical product of lattices



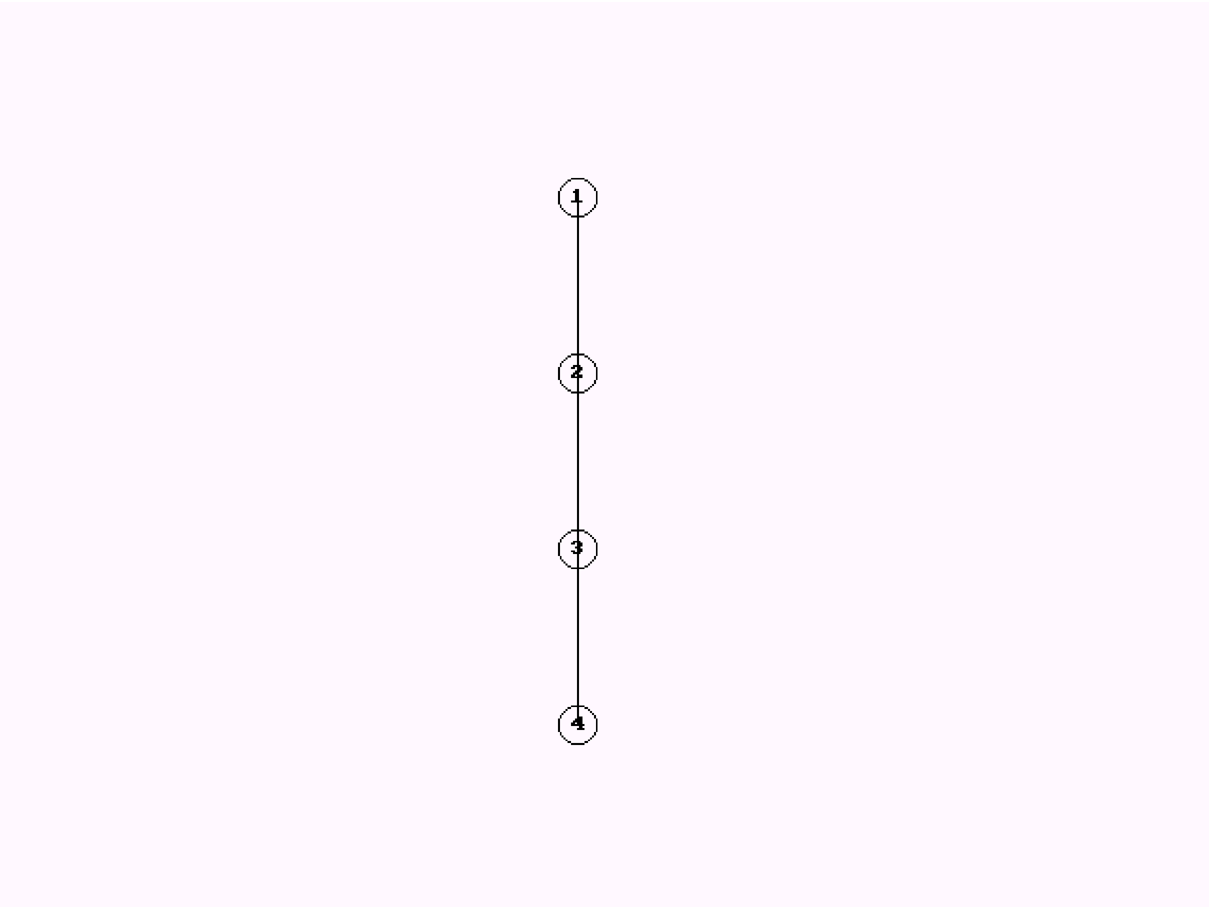
Graph of the Galois T3 lattice which is associated to the first 2 columns of C

Total number of closed pairs=8

Pair(1) : X={}, z =(2,3) / pair(2) : X={2}, z = (2,1) / pair(3) : X ={3}, z = (0,3),

Pair(4) : X = {2,4}, z = (1,1) / pair(5) : X= {2,7}, z =(2,0) / pair(6) : X= {2,3,4,5}, z=(0,1),

pair(7) : X = {1,2,4,7}, z=(1,0) / pair(8) : X ={1,2 ?3,4,5,6,7}, z=(0,0).



<u>Graph of the Galois G4 lattice which is associated to the last column of C</u>

Total number of closed pairs = 4.

Pair(1) : X = {5}, z=(3) / pair(2) : X= {1,5,6}, z=(2)
Pair(3) : X= {1,3,4,5,6}, z=(1) / pair(4) : X ={1,2 ,3,4,5,6,7}, z= (0).

X1↓ X2→	{5}	{1,5,6}	{1,3,4,5,6}	[1..7]
Z1↓ z2→	(3)	(2)	(1)	(0)
{}	{}	{}	{}	{}
(2,3)	(2,3,3)	(2,3,2)	(2,3,1)	(2,3,0)
{2}	{}	{}	{}	{2}
(2,1)	(2,1,3)	(2,1,2)	(2,1,1)	(2,1,0)
{3}	{}	{}	{3}	{3}
(0,3)	(0,3,3)	(0,3,2)	(0,3,1)	(0,3,0)
{2,4}	{}	{}	{4}	{2,4}
(1,1)	(1,1,3)	(1,1,2)	(1,1,1)	(1,1,0)
{2 ,7}	{}	{}	{}	{2,7}
(2,0)	(2,0,3)	(2,0,2)	(2,0,1)	(2,0,0)
{2,3,4,5}	{5}	{5}	{3,4,5}	{2,3,4,5}
(0,1)	(0,1,3)	(0,1,2)	(0,1,1)	(0,1,0)
{1,2,4,7}	{}	{1}	{1,4}	{1,2,4,7}
(1,0)	(1,0,3)	(1,0,2)	(1,0,1)	(1,0,0)
[1..7]	{5}	{1,5,6}	{1,3,4,5,6}	[1..7]
(0,0)	(0,0,3)	(0,0,2)	(0,0,1)	(0,0,0)

T = GL(C) is the vertical product of lattices T3 = GL (C_3) and T_4 = GL (C_4)

The first column of the array contains the closed pairs (X_1 , z_1) of T_3 , and the first row contains the closed pairs of T_4 .

The cell at the intersection of row i and column j corresponds to the pairs (X_i , z_i) of T_3 and (X_j , z_j) of T_4.

When it gives a closed pair of T, this box is drawn with a thick line.

Here we find 15 closed pairs for the horizontal product, and they are effectively all the pairs of T, as we can see it.

$$\begin{array}{c} (X_2 , z_2) \\ \downarrow \\ (X_1 , z_1) \rightarrow X = X_1 \cap X_2 , z = (z_1 , z_2) . \end{array}$$