

CryptoLat - a Pedagogical Software on Lattice Cryptomorphisms and Lattice Properties

Florent Domenach

Computer Science Department, University of Nicosia
46 Makedonitissas Ave., 1700 Nicosia, Cyprus
`domenach.f@unic.ac.cy`

Abstract. Although lattice theory is a rich field dating from Dedekind, Birkhoff and Öre, few studies in FCA use lattice properties to enhance their results. Moreover, out of the many cryptomorphisms associated with lattices, only the ones associating context, lattice and implicational system are effectively studied.

CryptoLat is a software implemented in C# which provides an intuitive view on different cryptomorphisms of lattices as well as on different properties of lattices. Its purpose is pedagogical, for students and researchers likewise, and work by showing incremental changes in the lattice and the associated cryptomorphisms.

Keywords: Lattice Cryptomorphisms, Lattice Properties, Pedagogy

1 Introduction

This article, together with the software, originates from an observation about the current state of research in the lattice and the FCA communities. Lattice theory dates back to the works of Dedekind, Birkhoff [7] and Öre [19], and is a rich field in which FCA is deeply and fundamentally rooted. Despite that fact, relatively few studies in FCA takes advantage of lattice properties to enhance their results. This dichotomy of the two communities is apparent in the research interests: on the one hand, researchers from the graph and ordered set communities are focusing on equivalence theorem and characterizations, while on the other hand data mining researchers are more focused on practical or practically oriented results.

Although it is natural that this dichotomy exists because of the nature of each community's focus of interest, it is the author's suggestion that improvement of the communication between the two communities may produce a synergistic effect. The study of lattice cryptomorphisms and lattice properties is important for many reasons: other than giving a better understanding of the tools being used, it was shown to lead to the discovery of efficient algorithms in counting the number of Moore families [16] or calculating the stem basis [6] for example.

There are two main purposes for this software: first, to promote and to give a better understanding of the many shapes and forms of lattices thanks to an incremental approach - any change made in any equivalent cryptomorphism is

transcribed to the other cryptomorphisms. Second, to help familiarize users to fundamental latticial properties that are recalculated after any modification. Its purpose is pedagogical for both students and researchers. Students can learn and get a better understanding on what is happening during the construction of the lattice and the implicational basis, while researchers can test hypotheses and find small illustrating examples. The program is written in C \sharp and is freely available (executable and source code) at <http://www.cs.unic.ac.cy/florent/software.htm>

This paper is organized as follows: we start in Sec. 2 by describing the software itself. Sec 3 will recall the well-known definitions of lattices and some of the existing cryptomorphisms implemented in `CryptoLat`, together with a short explanation on how the equivalence is done. Finally, we compare in Sec. 5 `CryptoLat` with already existing software.

2 Description of `CryptoLat`

`CryptoLat` software is structured around a model view controller type of software architecture: each cryptomorphism described in Sec. 3 is a view with which the user can interact. The model is the reduced context associated with a Moore family on which all the calculations are based. When a view is updated, by adding or deleting a Sperner family for example, then the Moore family associated is calculated, the associated concept lattice, together with all the properties, is computed, and the other views get updated.

Depending on which view getting updated, the process may involve different steps. If the overhanging list or the Sperner village change, then first the Moore family associated with the new list, and the reduced context is then calculated. Similarly, if the input is an arbitrary set system, then the closure system is calculated as in Sec. 3.2. If the input is on the set of implications, since implications are usually on the set of attributes, the corresponding Moore family will be dual to the previous ones. From the reduced context, we are using Closed by One algorithm to calculate the concept lattice.

Fig. 1 shows a screen shot of the software with the five different views: the Moore family on top, the context as a datagrid, the list of implications, the list of overhanging and the list of Sperner families. The drawing of the lattice is done with a simple algorithm that optimise the position of each concept depending on the number (and position) of concepts covered.

3 Cryptomorphisms of Lattices

In this section, we present the different cryptomorphisms of lattices implemented in `CryptoLat` in details; we do not claim to be exhaustive, so we refer the readers to [9] for a survey on those cryptomorphisms. Since we are describing implementations, the following descriptions needs only to be for the finite case.

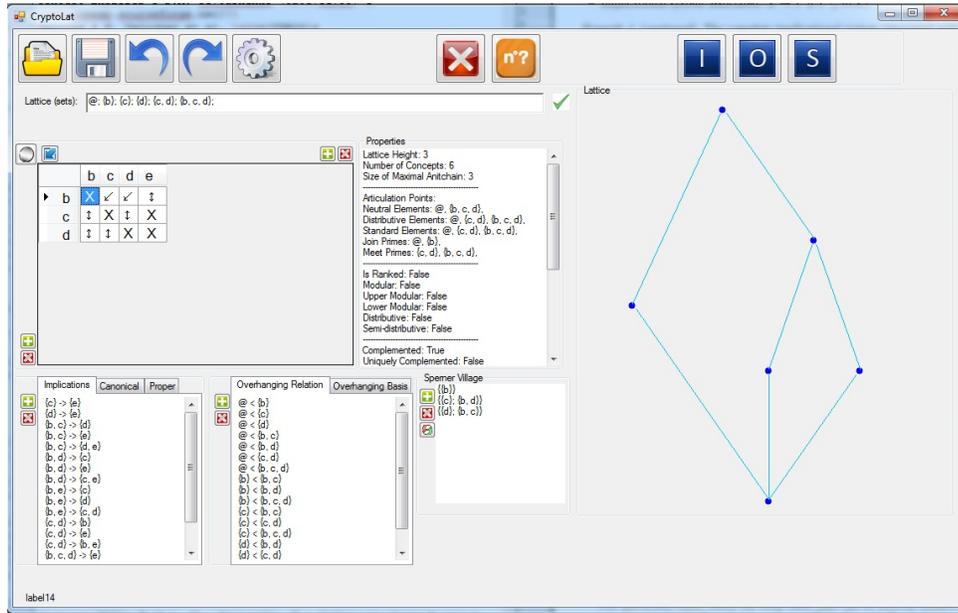


Fig. 1. CryptoLat software with the different views displaying our running example

3.1 Lattice and Irreducible Elements

Different yet equivalent definitions for lattices exist: here, we will define a *lattice* (L, \wedge, \vee) as a set L with two binary operators satisfying the following four properties:

- Associativity: $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ and $x \vee (y \vee z) = (x \vee y) \vee z$
- Commutativity: $x \wedge y = y \wedge x$ and $x \vee y = y \vee x$
- Idempotence: $x \wedge x = x$ and $x \vee x = x$
- Absorption: $x \wedge (y \vee x) = x$ and $x \vee (y \wedge x) = x$

One can associate an order on the elements of L to these operators as follow: $\forall x, y \in L, x \leq y$ if $x \wedge y = x$ or, equivalently, $x \vee y = y$. Fig. 2 shows a lattice having six elements using its Hasse diagram representation: the elements of L are depicted by nodes in the plane, and a line segment connects the nodes corresponding to a covering pair $x \prec y$ (i.e. $x \leq y$ and $\forall z, x \leq z \leq y$ implies $x = z$ or $y = z$), where the node representing y is placed 'higher' in the plane than the node representing x .

Some elements of lattices are of particular interest: in any lattice, we have a minimal element, usually denoted by 0_L or \perp , and a maximal element 1_L or \top . Irreducible elements of the lattice are also the subject of studies since they constitute minimal generating sets, and are often used to characterize classes of lattices. An element j of L is *join-irreducible* if $j = \vee X$ then $j \in X$, or,

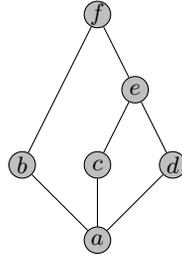


Fig. 2. Example of a lattice with 6 elements

equivalently, j covers only one element in L . Dually, an element m of L is *meet-irreducible* if $m = \bigwedge X$ then $m \in X$, or m is covered by only one element of L . Let J_L and M_L be the sets of join and meet-irreducible of L .

3.2 Moore Family and Closure Operator

Let S be a finite set. A *closure system* (or *Moore family*) \mathcal{C} on S is a family of subsets of S satisfying $S \in \mathcal{C}$ and, for all $A, B \in \mathcal{C}$, $A \cap B \in \mathcal{C}$. A closure system is a lattice with the following operations:

- $C_1 \wedge C_2 = C_1 \cap C_2$
- $C_1 \vee C_2 = \bigcap \{C \in \mathcal{C} : C_1 \cup C_2 \subseteq C\}$

Let \mathcal{A} be an arbitrary set system on a set S . Then one can associate a closure system \mathcal{C}_A to \mathcal{A} :

$$\mathcal{C}_A = \{\bigcap \{A_i : A_i \in \mathcal{A}\}\} \cup \{S\}$$

Example 1 (continued). The family $\{\emptyset, \{G_0\}, \{G_1\}, \{G_2\}, \{G_1, G_2\}, \{G_0, G_1, G_2\}\}$, with the operators previously defined, constitute a lattice isomorphic to the lattice of Fig. 2.

Another fundamental cryptomorphism of lattice concerns closure operators. A *closure operator* on a set S is a map $\sigma : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ satisfying $\forall X, Y \subseteq S$:

- Isotone: $X \subseteq Y \Rightarrow \sigma(X) \subseteq \sigma(Y)$
- Extensive: $X \subseteq \sigma(X)$
- Idempotent: $\sigma(\sigma(X)) = \sigma(X)$

These three properties together are equivalent to extensivity and path independence property [20]: $\sigma(X \cup Y) = \sigma(\sigma(X) \cup \sigma(Y))$.

The well-known equivalence between closure operators and closure systems is the following: given the closure system \mathcal{C} , the associated closure operator is $\sigma_{\mathcal{C}}(X) = \bigcap \{C \in \mathcal{C} : X \subseteq C\}$. Conversely, given a closure operator σ , the associated closure system is the set of fixed points of σ , i.e. $\mathcal{C}_{\sigma} = \{X \subseteq S : X = \sigma(X)\}$.

3.3 Context Associated with a Lattice

One of the most celebrated cryptomorphism in the FCA community is the one between a lattice and a binary relation, also known as the basic theorem of Formal Concept Analysis [3, 13].

Formally, a context (G, M, I) is a binary relation I between a set of objects G and a set of attributes M . One can associate the Galois connection between $\mathcal{P}(G)$ and $\mathcal{P}(M)$ to this binary relation as $\forall A \subseteq G, B \subseteq M$,

$$A' = \{m \in M : \forall g \in A, (g, m) \in I\}$$

$$B' = \{g \in G : \forall m \in B, (g, m) \in I\}$$

The lattice associated with the context (G, M, I) , also called the concept lattice or Galois lattice, is the set of all concepts (A, B) , with $A' = B$ and $B' = A$, together with the order $(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2$ ($\iff B_1 \supseteq B_2$). Conversely, any lattice L is isomorphic to the Galois lattice of the context (J_L, M_L, \leq) , also called the reduced context.

Two other binary relations, the arrows relations, on $\mathcal{P}(J) \times \mathcal{P}(M)$ are often used for characterization theorems. Let $j \in J, m \in M$, we define:

- $j \uparrow m \iff j \not\leq m$ and $j < m^+$
- $j \downarrow m \iff j \not\leq m$ and $j^- < m$
- $j \updownarrow m \iff j \uparrow m$ and $j \downarrow m$

Example 1 (continued). The lattice associated with the context of Fig. 3 is isomorphic to the lattice of Fig. 2.

	b	c	d	e
b	X	✓	✓	†
c	†	X	†	X
d	†	†	X	X

Fig. 3. The reduced context, together with the arrows relations, associated with our running example.

3.4 Implications

An implicational system \mathcal{I} on S is a binary relation on $\mathcal{P}(S)$, and if $(A, B) \in \mathcal{I}$, we denote it $A \rightarrow_{\mathcal{I}} B$ or $A \rightarrow B$ if no confusion is possible. We say that A implies B or $A \rightarrow B$ is an implication (of \mathcal{I}). A *complete implicational system* \mathcal{I} is an implicational system on $\mathcal{P}(S)$ satisfying, $\forall A, B, C, D \subseteq S$:

- $B \subseteq A$ implies $A \rightarrow B$;

- $A \rightarrow B$ and $B \rightarrow C$ imply $A \rightarrow C$;
- $A \rightarrow B$ and $C \rightarrow D$ imply $(A \cup C) \rightarrow (B \cup D)$.

[2] showed that a one-to-one correspondence between closure operators and complete implicational system exists. The closure system associated to a complete implicational system is $\mathcal{C}_{\mathcal{I}} = \{C \subseteq S : \forall X \subseteq S, [X \subseteq C \text{ and } X \rightarrow Y] \Rightarrow Y \subseteq C\}$, while the implicational system associated to a closure mapping is $\{X \rightarrow Y : Y \subseteq \sigma(X)\}$.

Example 1 (continued). The complete implicational system associated with the lattice in Fig. 2 is the following:

$$\begin{array}{l}
c \rightarrow e; \quad d \rightarrow e; \quad b, c \rightarrow d; \\
b, c \rightarrow e; \quad bc \rightarrow d, e; \quad b, d \rightarrow c; \\
b, d \rightarrow e; \quad b, d \rightarrow c, e; \quad b, e \rightarrow c; \\
b, e \rightarrow d; \quad b, e \rightarrow c, d; \quad c, d \rightarrow b; \\
c, d \rightarrow e; \quad c, d \rightarrow b, e; \quad c, d, e \rightarrow b; \\
b, d, e \rightarrow c; \quad b, c, e \rightarrow d; \quad b, c, d \rightarrow e;
\end{array}$$

As the above example shows, many implications in a complete implicational system can be redundant: in our example, it is evident that if we have $c \rightarrow e$ then we have $b, c \rightarrow e$. So smaller implicational system allowing a smaller representation of a complete implicational system is of interest. We say that an implicational system Σ is a *generating system* for \mathcal{I} if any implication of \mathcal{I} can be obtained from Σ by applying recursively the rules defining a complete implicational system. A minimal generating system for \mathcal{I} is called a *basis* for \mathcal{I} , and a minimum basis is called a *canonical basis*.

One particular basis, called the *stem basis* [15], can be directly defined: the stem basis is made of all implications $X \rightarrow \sigma(X) - X$, with X a critical set. A set $Q \subseteq S$ is called a *quasi-closed set* if $Q \notin \mathcal{C}$ and $\mathcal{C} + \{Q\}$ is a closure system, and Q is a *F-quasi-closed set* if Q is quasi-closed and $\sigma(Q) = F$. So a set $C \subseteq S$ is *critical* if there exists $F \in \mathcal{C}$ such that C is a minimal *F-quasi-closed set*.

Example 1 (continued). Canonical basis associated with the lattice of Fig.2.

$$b, e \rightarrow c, d; \quad c \rightarrow e; \quad d \rightarrow e; \quad c, d, e \rightarrow b;$$

3.5 Overhanging Relation

An *overhanging relation* \mathcal{O} is a binary relation on $\mathcal{P}(S)$ that was originated in [1] in consensus theory on trees under the term of nesting, and was generalized in [12] to any lattice. We will indifferently write $(A, B) \in \mathcal{O}$ or $A \mathcal{O} B$ to signify that a set A is overhanged in B . An overhanging relation satisfies the following properties:

- for all $A, B \in S$, $A \mathcal{O} B$ implies $A \subset B$;
- for all $A, B, C \in S$, $A \subset B \subset C$ implies $[A \mathcal{O} C \iff A \mathcal{O} B \text{ or } B \mathcal{O} C]$;
- for all $A, B \in S$, $A \mathcal{O} (A \cup B)$ implies $(A \cap B) \mathcal{O} B$.

We showed in [12] that overhanging relations and closure operators (and so lattices) are in a one-to-one correspondence by associating the overhanging relation defined as $A \mathcal{O} B \iff A \subset B$ and $\sigma(A) \subset \sigma(B)$ to any closure operator, and, conversely, associating the closure operator $\sigma(X) = \{x \in S : x \mathcal{O}^c X \cup \{x\}\}$ to any overhanging relation. In fact, an overhanging relation can be seen as a kind of negative implication, as given by the equivalence that $A \rightarrow B$ if and only if $(A, A \cup B) \notin \mathcal{O}$

Example 1 (continued). The overhanging relation associated with the lattice of Fig.2 is as follows:

$$\begin{array}{lll}
 \emptyset \mathcal{O} b & \emptyset \mathcal{O} c & \emptyset \mathcal{O} d \\
 \emptyset \mathcal{O} b, c & \emptyset \mathcal{O} b, d & \emptyset \mathcal{O} c, d \\
 \emptyset \mathcal{O} b, c, d & b \mathcal{O} b, c & b \mathcal{O} b, d \\
 b \mathcal{O} b, c, d & c \mathcal{O} b, c & c \mathcal{O} c, d \\
 c \mathcal{O} b, c, d & d \mathcal{O} b, d & d \mathcal{O} c, d \\
 d \mathcal{O} b, c, d & c, d \mathcal{O} b, c, d &
 \end{array}$$

3.6 Sperner Family

A *Sperner family* \mathbb{F} on S is a family s.t. $\forall X, Y \in \mathbb{F}$, X and Y are incomparable for set inclusion. A *Sperner village* $\mathcal{V} = (\mathbb{F}_1, \dots, \mathbb{F}_n)$ on S is a set of n Sperner families. The equivalence between a closure operator and a Sperner village is obtained as follows: given σ closure operator, for any $x \in S$ we create the Sperner family $\mathcal{C}_x = \{X \subseteq S : x \in \sigma(X) \text{ and } X \text{ minimal for that property}\}$. Conversely, the closure operator associated with \mathcal{V} is defined as $\sigma(X) = \{x \in S : X \text{ contains a set of } \mathcal{C}_x\}$.

Example 1 (continued). The Sperner village equivalent with our running example is the following Fig. 4.

When we add to the existing Sperner village the Sperner family $\{\{e\}, \{b, c\}\}$,

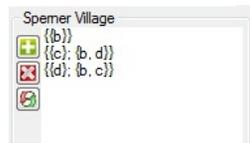


Fig. 4. Sperner village associated with our running example.

the software computes the smallest Sperner village possible from the existing village together with the new family, and from this Sperner village finds the associated concept lattice. Fig. 5 shows the results.

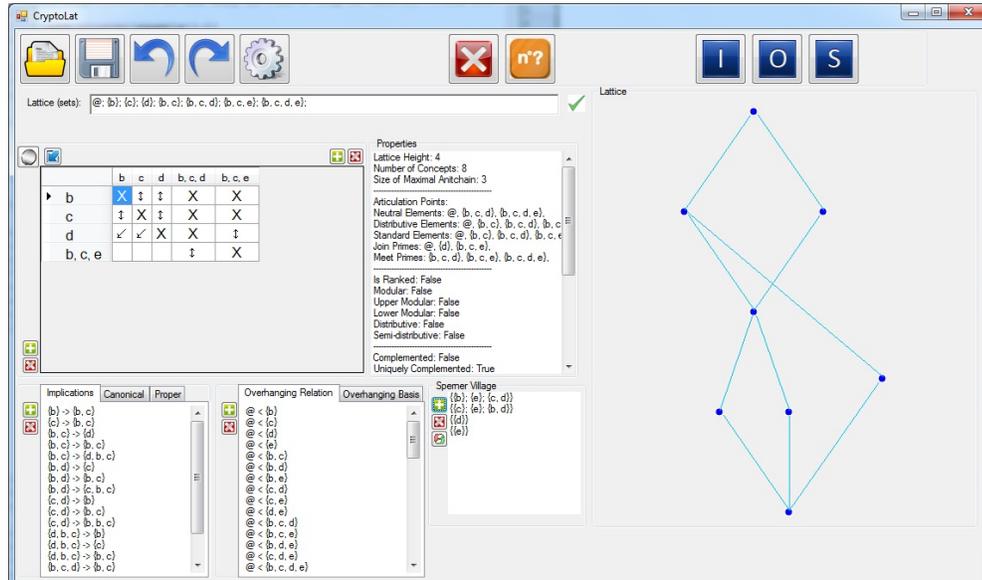


Fig. 5. Screenshot of the software after adding a new Sperner family

4 Properties of lattices

The main focus of the software concerns the dependency between the different cryptomorphisms associated with lattices. In the same pedagogical spirit, we also implemented different properties of lattices and elements of lattices, briefly described in Tab. 1 and in Tab. 2 for completeness sake. We refer the readers to [11, 14, 22] for more information on lattices and lattice properties.

5 Existing Software

All of the software mentioned in Table 3 have different goals. In terms of ease of use, Concept Explorer is the closest to our goal, but offers limited functionalities. OpenFCA is similar to Concept Explorer, offering an interesting, web based and intuitive approach to context creation, lattice visualization and attribute exploration. FCAstone's main purpose is file formats conversion to improve interoperability between FCA and graph editing. The other software (Galicia, ToscanaJ, Lattice Miner, FCART) were designed to push the boundaries of the scalability of analysis and lattice drawing. Research wise, Formal Concept Analysis Research Toolbox (FCART) is a particularly interesting approach of a universal

Table 1. List of lattice properties

Types of Lattices	
Atomistic	Every join-irreducible element of L is an atom
Co-atomistic	Every meet-irreducible element is covered by 1_L
Ranked	There is a ranking function $r : L \rightarrow \mathbb{N}$ such that $\forall x, y \in L, x \prec y$ implies $r(y) = r(x) + 1$
Upper semi-modular	$\forall x, y \in L: x \wedge y \prec x \Rightarrow y \prec x \vee y$
Lower semi-modular	$\forall x, y \in L: y \prec x \vee y \Rightarrow x \wedge y \prec x$
Modular	Upper and lower semi-modular
Distributive	$\forall x, y, z \in L, x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
Semi-distributive	$\forall x, y, z \in L$, we have both $x \wedge y = x \wedge z \Rightarrow x \wedge y = x \wedge (y \vee z)$ and $x \vee y = x \vee z \Rightarrow x \vee y = x \vee (y \wedge z)$
Families of Lattices	
Chain	$\forall x \in L, x \neq 0_L, x \in J_L$
Hierarchy	$\forall x \in S, \{x\} \in L$ and $\forall A, B \in L, A \cap B \in \{\emptyset, A, B\}$
Weak hierarchy	$\forall x \in S, \{x\} \in L$ and $\forall A, B, C \in L, A \cap B \cap C \in \{A \cap B, A \cap C, B \cap C\}$
Boolean	Isomorphic to the power set of some set
Complemented Lattices	
Complemented	$\forall x \in L, \exists y \in L: x \wedge y = 0_L$ and $x \vee y = 1_L$
Uniquely complemented	The complement is unique
Pseudo-complemented	$\forall x \in L, \exists y \in L: x \wedge y = 0_L$ and $\forall z \in L, x \wedge z = 0_L \Rightarrow z \leq y$
Relatively complemented	Any $[u, v]$ is relatively complemented, i.e. $\forall x$ in $[u, v], \exists y$ such that $x \wedge y = u$ and $x \vee y = v$
Uniquely relatively complemented	The complement in $[u, v]$ is unique
Sectionally complemented	Any $[0_L, v]$ is complemented
Other Lattices Properties	
δ -hard	$\forall j, j' \in J_L, j \delta j'$, i.e. there exists $x \in L$ such that $j \not\leq x, j' \not\leq x$ and $j < j' \vee x$
δ -strong	The transitive graph of the δ relation is complete
Sperner poset	L is ranked, and the size of the maximal antichain is equal to the maximal size rank

Table 2. List of elements properties

x is distributive	$\forall y, z \in L, x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
x is standard	$\forall y, z \in L, y \wedge (x \vee z) = (y \wedge x) \vee (y \wedge z)$
x is neutral	$\forall y, z \in L, (x \wedge y) \vee (y \wedge z) \vee (x \wedge z) = (x \vee y) \wedge (x \vee z) \wedge (y \vee z)$
x is an articulation point	$\forall y \in L$, either $x \leq y$ or $y \leq x$
x is join-prime	$x \leq a \vee b$ implies $x \leq a$ or $x \leq b$
x is meet-prime	$a \wedge b \leq x$ implies $a \leq x$ or $b \leq x$

integrated environment to lattice construction and Formal Concept Analysis techniques, where one can test new algorithms for drawing lattice for example.

Table 3. Some FCA software tools (taken from [18])

Program title	Authors	Web-site
Concept Explorer	S.A. Yevtushenko [25]	conexp.sourceforge.net
Galicja	P. Valtchev et al. [23]	www.iro.umontreal.ca/~galicia
ToscanaJ	U. of Queensland, TU Darmstadt [24]	toscanaj.sourceforge.net
FcaStone	U. Priss et al. [21]	fcastone.sourceforge.net
Lattice Miner	B. Lahcen et al. [17]	lattice-miner.sourceforge.net
Conexp-clj	TU-Dresden, D. Brochman	daniel.kxpq.de/math/conexp-clj
OpenFCA	P. Borza, O. Sabou et al. [8]	code.google.com/p/openfca
FCART	A. Neznanov and D. Ilvovsky [18]	

6 Concluding Remarks

CryptoLat is a pedagogical and research oriented software in active development phase, and it is open source and freely available. It is still in alpha testing, and the next milestone of development is to make it cloud based with a web interface.

Many other cryptomorphisms of concept lattices exist but are not implemented yet. For example, [4] proved that we can associate a minimal separator of the co-bipartite graph to every concept of the concept lattice, and [16] showed a bijection between Moore families and ideal colour sets of the coloured poset. Similarly we are planning to increase the number of properties available, adding for example the dismantlable property (one can repeatedly remove a doubly irreducible element until the lattice becomes a chain) or being a convex geometry ($\emptyset \in L$ and $\forall x \in L$, there is a unique minimal (for inclusion) generator of x). the lattice drawing component is also under consideration, to be enhance and more user friendly.

References

1. Adams III, E.N.: N-trees as nestings: complexity, similarity and consensus. *Journal of Classification* 3, 299–317 (1986)
2. Armstrong, W.W.: Dependency structures of data base relationships. *Information Processing* 74, 580583 (1974)
3. Barbut, M., Monjardet, B.: *Ordres et classification: Algèbre et combinatoire (tome II)*. Hachette, Paris (1970)
4. Berry, A., Sigayret, A.: Representing a concept lattice by a graph. *Discrete Applied Mathematics* 144 (12), 27–42 (2004)
5. Berry, A., Sigayret, A.: A Peep through the Looking Glass: Articulation Points in Lattices. In: Domenach, F., Ignatov, D., Poelmans, J. (Eds.) *Formal Concept Analysis*, Springer Berlin Heidelberg, 7278, 45-60 (2012)

6. Bertet, K.: The dependence graph of a lattice. In Szathmary, L., Priss, U. (Eds.): CLA 2012, 223–231 (2012)
7. Birkhoff, G.: Lattice Theory (Third ed.). Providence, R.I.: Amer. Math.Soc.(1967)
8. Borza, P.V., Sabou, O., Sacarea, C.: OpenFCA, an open source formal concept analysis tool. Proc. of IEEE International Conference on Automation Quality and Testing Robotics (AQTR), 1–5 (2010)
9. Caspard, N., Monjardet, B.: The lattices of Moore families and closure operators on a finite set: a survey. Disc. App. Math. 127, 241–269 (2003)
10. Caspard, N., Leclerc, B., Monjardet, B.: Finite Ordered Sets: Concepts, Results and Uses. Cambridge University Press (2012)
11. Davey, B.A., Priestley, H. A.: Introduction to Lattices and Order, 2nd ed. Cambridge University Press (2002)
12. Domenach, F., Leclerc, B.: Closure systems, Implicational Systems, Overhanging Relations and the case of Hierarchical Classification. Math. Soc. Sci. 47 (3), 349–366 (2004)
13. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations, Springer Verlag (1999)
14. Grätzer, G.: General lattice theory. Academic Press (1978).
15. Guigues, J.-L., Duquenne, V.: Familles minimales d’implications informatives résultant d’un tableau de données binaires. Math. Sci. Hum. **95** , 5-18 (1986)
16. Habib, M., Nourine, L.: The number of Moore families on $n=6$. Disc. Math. 294(3), 291–296 (2005)
17. Lahcen, B., Kwuida, L.: Lattice Miner: A Tool for Concept Lattice Construction and Exploration. In Boumedjout, L., Valtchev, P., Kwuida, L., Sertkaya, B. (eds.): Supplementary Proceeding of International Conference on Formal concept analysis (ICFCA’10), 59-66 (2010)
18. Neznanov, A.A., Ilvovsky, D.A., Kusnetsov, S.O.: FCART: A New FCA-based System for Data Analysis and Knowledge Discovery. In Cellier, P., Distel, F., Ganter, B. (eds.): Contributions to the 11th International Conference on Formal Concept Analysis, 65–78 (2013)
19. Öre, O.: Galois connexions. Trans. Amer. Math. Soc. 55(3), 494-513 (1944)
20. Plott, C.R.: Path independence, rationality and social choice. Econometrica 41 (6), 1075-1091 (1973)
21. Priss, U.: FCA Software Interoperability. In: Belohlavek, R., Kuznetsov, S.O. (eds.), CLA 2008, 331-44, (2008)
22. Roman, S: Lattices and Ordered Sets. Springer (2000)
23. Valtchev, P., Grosser, D., Roume, C., Rouane Hacene, M.: Galicia: An Open Platform for Lattices. In Using Conceptual Structures: Contributions to the 11th Intl. Conference on Conceptual Structures (ICCS’03), 241–254 (2003)
24. Vogt, F., Wille, R.: TOSCANA - a graphical tool for analyzing and exploring data. In Tamassia R., Tollis, I.G. (eds.), Graph Drawing, LNCS 894, 226-233 (1994)
25. Yevtushenko, S.A.: System of data analysis ”Concept Explorer”. Proceedings of the 7th national conference on Artificial Intelligence KII-2000, Russia (2000) 127-134