

# Conception et implémentation de nouvelles fonctionnalités dans un prototype de fouille de données

Kévin Emamirad

Université du Québec en Outaouais  
Département d'informatique et d'ingénierie

*emak01@uqo.ca*

mercredi 31 mai 2017

# Plan de la présentation

- 1 Introduction
- 2 Rappels
- 3 Les outils de l'analyse formelle de concepts
- 4 Production d'implications avec négation

- L'analyse formelle de concepts (AFC) (Ganter et Wille 1999) est un formalisme de représentation des connaissances et de fouille de données qui est
  - ▶ utilisé dans divers domaines (informatique, linguistique, sociologie, biologie, etc.), et
  - ▶ produit des visualisations graphiques des structures inhérentes aux données sous forme de treillis de concepts (Galois).
- Au cours des deux dernières décennies, on a vu apparaître plusieurs d'outils d'analyse formelle de concepts tels *ToscanaJ*, *ConExp*, *Coron*, *Java Lattices*, et *Lattice Miner*.

- But : Enrichir *Lattice Miner* (Roberge 2007), un prototype de fouille de données qui exploite l'AFC avec les objectifs suivants :
  - ① production d'implications avec négation (Missaoui, Nourine et Renaud 2012),
  - ② enrichissement du module de génération des règles triadiques pour obtenir exhaustivement et précisément les trois formes d'implications triadiques définies par (Ganter et Obiedkov 2004),
  - ③ validation intensive de la procédure de production de la base d'implications de (Guigues et Duquenne 1986) et de la procédure de calcul des relations de flèches. Cette dernière est utile dans le processus de décomposition de contextes formels (Viaud et al. 2015), et
  - ④ amélioration de la convivialité de l'interface usager.

# Rappels

# Rappels I

## Analyse formelle de concepts

### Définition 1 : Contexte formel

Soit  $\mathbb{K} = (G, M, I)$  un contexte formel où  $G$ ,  $M$  et  $I$  sont respectivement un ensemble d'objets, une collection d'attributs et une relation binaire entre  $G$  et  $M$ . L'expression  $(g, m) \in I$  ou encore  $glm$  signifie que l'objet  $g$  possède l'attribut  $m$ .

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

Tableau 1: Exemple de contexte  $\mathbb{K}$

# Rappels II

## Analyse formelle de concepts

### Définition 2 : Concept formel

Un *concept formel*  $c$  est une paire d'ensembles  $c := (A, B)$  avec  $A \subseteq G$ ,  $B \subseteq M$ ,  $A = B'$  et  $B = A'$ , où  $A'$  est l'ensemble des attributs partagés par les objets dans  $A$  et  $B'$  est l'ensemble des objets ayant tous leurs attributs dans  $B$ . Les sous-ensembles  $A$  et  $B$  sont appelés respectivement l'extension et l'intention du concept  $c$ . Les valeurs de  $A'$  et  $B'$  sont obtenues comme suit :  $A' := \{m \in M \mid g \text{ l m } \forall g \in A\}$  et  $B' := \{g \in G \mid g \text{ l m } \forall m \in B\}$ .

### Définition 3 : Sous-ensemble fermé

Un sous-ensemble  $X$  est fermé si  $X'' = X$ . Un concept objet pour l'objet  $g$  est une paire de la forme  $\gamma(g) = (g'', g')$  alors que le concept attribut pour l'attribut  $m$  est  $\mu(m) = (m', m'')$ . Les sous-ensembles fermés de  $G$  sont les extensions alors que les sous-ensembles fermés de  $M$  sont les intentions de  $\mathbb{K}$ .

# Rappels III

## Analyse formelle de concepts

### Définition 4 : Treillis de concepts

Un *treillis de concepts*  $\mathfrak{B}(G, M, I)$  est un treillis résultant de l'ordre partiel existant entre les concepts du contexte  $\mathbb{K} = (G, M, I)$ .

$$(A, B) \leq (C, D) \Leftrightarrow A \subseteq C \text{ et } D \subseteq B$$

$(A, B)$  est alors un sous-concept ou prédécesseur immédiat de  $(C, D)$  alors que ce dernier est un successeur de  $(A, B)$ .

### Définition 5 : Les bornes du treillis

La borne inférieure  $\wedge$  (*meet*) d'un ensemble de concepts  $(X_i, Y_i)$  avec  $i = 1, \dots, k$  est le plus grand des prédécesseurs communs. De même, la borne supérieure  $\vee$  (*join*) d'un ensemble de concepts est le petit des successeurs communs.



# Rappels IV

## Analyse formelle de concepts

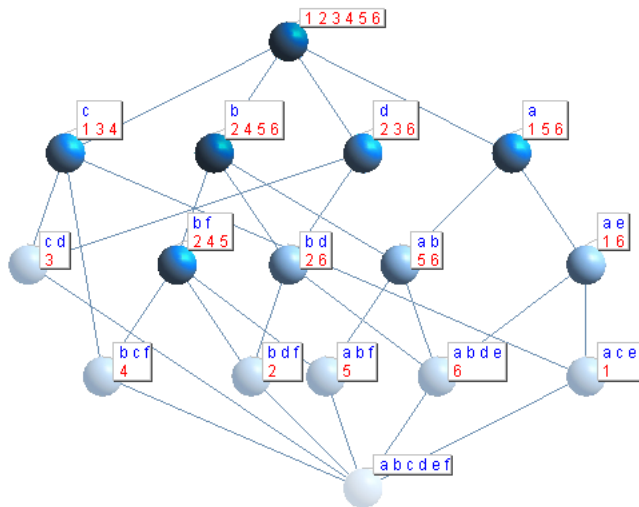


Figure 1: Treillis avec étiquetage complet pour le contexte du tableau 1

# Rappels V

## Analyse formelle de concepts

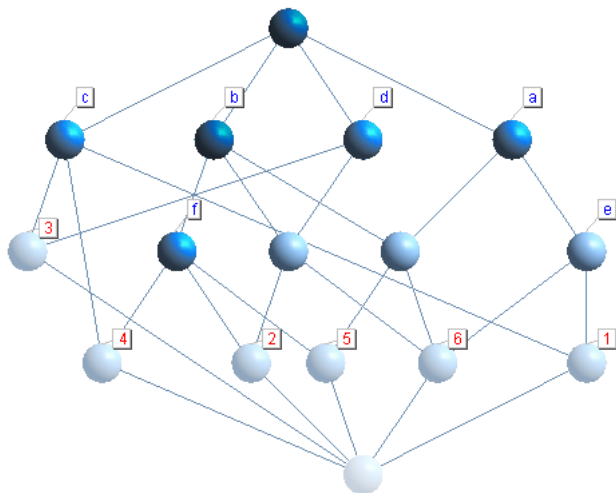


Figure 2: Treillis avec étiquetage réduit pour le contexte du tableau 1

# Rappels VI

## Analyse formelle de concepts

L'*infimum* est le plus petit concept d'un treillis de Galois alors que le *supremum* est son plus grand concept.

Lorsqu'un concept  $c_1$  est plus petit qu'un autre concept  $c_2$ , alors le nœud représentant  $c_1$  est placé plus bas que son successeur  $c_2$ . Lorsqu'une relation d'ordre existe entre les concepts  $c_1$  et  $c_2$ , on dit alors que ces concepts sont *comparables* ( $\leq$ ), sinon ils ne le sont pas ( $\not\leq$ ).

### Définition 6 : sup-irréductible et inf-irréductible

On rappelle qu'un concept est dit *sup-irréductible* (ou *join-irréductible*) si et seulement si il possède un unique prédécesseur et il est dit *inf-irréductible* (ou *meet-irréductible*) s'il admet un unique successeur.

# Rappels VII

## Règles d'association et implications

### Définition 7 : règle d'association

$$r : Y \rightarrow Z [sup, conf]$$

$r$  est une règle d'association avec

- $Y$  et  $Z$  sont des sous-ensembles d'attributs appelés *itemsets*,
- $Y \cap Z = \emptyset$ ,
- $sup$ , le support est  $Prob(Y \cup Z)$  (proportion des objets ayant simultanément les attributs  $Y$  et  $Z$ ), et
- $conf$ , la confiance est  $Prob(Y \cup Z)$  (probabilité d'avoir  $Z$  lorsque  $Y$  est présent dans le contexte  $\mathbb{K}$ ).

# Rappels VIII

## Règles d'association et implications

### Définitions 8 : itemsets et générateurs

- Un *itemset* est dit *fréquent* si la proportion d'objets le possédant est au moins égale au support minimum défini par l'utilisateur.
- Un *itemset*  $Y$  est dit *fermé* si  $Y = Y''$ . Cela signifie que  $Y$  est une intention d'un concept formel.
- Un *générateur*  $G$  (Pfaltz et Taylor 2002) d'un itemset fermé (intention)  $Y$  est un ensemble minimal de  $Y$  tel que  $G'' = Y$ .

# Rappels IX

## Règles d'association et implications

### Définition 9 : implication et base générique

$$r : g_i \rightarrow \text{Int}(c) \setminus g_i$$

$r$  est une implication ou règle exacte avec

- $c$  un concept formel,
- $G = \{g_1, \dots, g_i, \dots, g_n\}$  l'ensemble de ses générateurs,
- $\text{Int}(c)$  représente l'intention du concept  $c$ ,
- $\text{support}(c)$  est la proportion d'objets contenus dans l'extension de  $c$ ,
- $\text{support}(r) = \text{support}(c)$  et
- $\text{confiance}(r) = 100 \%$

Une base générique (Pasquier et al. 1999) d'implications est une représentation relativement concise d'implications de la forme précédente.

### Définition 10 : règle approximative

$$r : g_i \rightarrow \text{Int}(c_j) \setminus \text{Int}(c)$$

$r$  est une règle approximative avec

- $c$  un concept formel,
- $P = \{c_1, \dots, c_j, \dots\}$  l'ensemble des prédecesseurs de  $c$ ,
- $\text{Int}(c)$  représente l'intention du concept  $c$ ,
- $g_i$  un générateur de  $\text{Int}(c)$ ,
- $\text{support}(r) = \text{support}(c_j)$ , et
- $\text{confiance}(r) = \text{support}(c_j) / \text{support}(c)$

# Rappels XI

## Règles d'association et implications

### Définitions 11 : pseudo-intent et base de Guigues-Duquenne

Un ensemble  $P \subseteq M$  est un *pseudo-intent* de  $(G, M, I)$  (Ganter et Wille 1999 ; Guigues et Duquenne 1986) si et seulement si  $P \neq P''$  et si  $Q'' \subseteq P$  est vraie pour tout pseudo-intent  $Q \subseteq P$ ,  $Q \neq P$ .

L'ensemble des implications de la forme :

$$\mathcal{L} := \{ \mathcal{P} \rightarrow \mathcal{P}'' \setminus \mathcal{P} \mid \mathcal{P} \text{ pseudo-intent} \}$$

est appelé base de Guigues-Duquenne (*stem base*) et reconnue comme étant minimale.

À titre d'exemple,  $\{a, c\}$  est un pseudo-intent dont le fermé est  $\{a, c, e\}$ , d'où l'implication  $\{a, c\} \rightarrow \{e\}$



# Rappels XII

## Règles d'association et implications

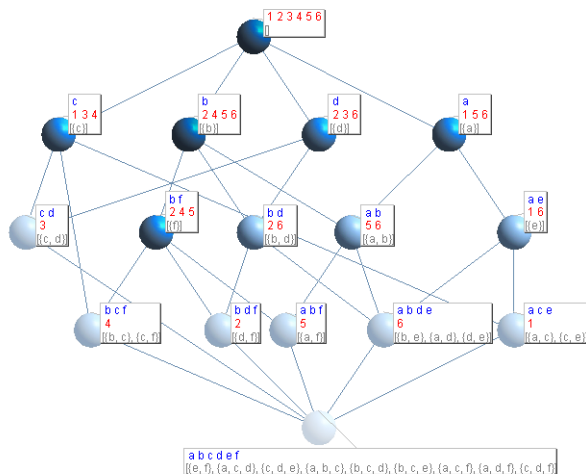


Figure 3: Treillis de concepts avec indication des générateurs

# Rappels XIII

## Règles d'association et implications

### Définition 12 : contexte complémentaire $\tilde{\mathbb{K}}$

Soit un contexte  $\mathbb{K} = (G, M, I)$  décrivant un ensemble  $G$  d'objets, un ensemble  $M$  de propriétés (attributs) et une relation binaire  $I$  entre  $G$  et  $M$ . Le contexte complémentaire de  $\mathbb{K}$  est  $\tilde{\mathbb{K}} = (G, \tilde{M}, G \times M \setminus I)$  avec  $\tilde{M}$  l'ensemble des attributs négatifs.

### Définition 13 : apposition du contexte $\mathbb{K}$ avec son complémentaire $\tilde{\mathbb{K}}$

Le contexte  $\mathbb{K}|\tilde{\mathbb{K}}$  est l'apposition du contexte  $\mathbb{K}$  avec son complémentaire  $\tilde{\mathbb{K}}$ . Cela signifie que  $\mathbb{K}|\tilde{\mathbb{K}} := (G, M \cup \tilde{M}, I \cup G \times M \setminus I)$ .

# Rappels XIV

## Règles d'association et implications

	a	b	c	d	e	f	$\tilde{a}$	$\tilde{b}$	$\tilde{c}$	$\tilde{d}$	$\tilde{e}$	$\tilde{f}$
1	×		×		×			×		×		×
2		×		×		×	×		×		×	
3			×	×			×	×			×	×
4		×	×			×	×			×	×	
5	×	×				×			×	×	×	
6	×	×		×	×				×			×

Figure 4:  $\mathbb{K}|\tilde{\mathbb{K}}$  : apposition du contexte  $\mathbb{K}$  avec son complémentaire  $\tilde{\mathbb{K}}$

### Définition 14 : Relations flèches

La relation entre l'objet  $g$  et l'attribut  $m$  dans un contexte formel se présente sous l'une des quatre formes suivantes :

- $g \updownarrow m$  si  $\gamma(g) \not\leq \mu(m)$ ,  $\gamma(g) \leq m^+$ , et  $g^- \leq \mu(m)$
- $g \uparrow m$  si  $\gamma(g) \not\leq \mu(m)$ ,  $\gamma(g) \leq m^+$ , et  $g^- \not\leq \mu(m)$
- $g \downarrow m$  si  $\gamma(g) \not\leq \mu(m)$ ,  $\gamma(g) \not\leq m^+$ , et  $g^- \leq \mu(m)$
- $g \circ m$  si  $\gamma(g) \not\leq \mu(m)$ ,  $\gamma(g) \not\leq m^+$ , et  $g^- \not\leq \mu(m)$

où  $g^-$  représente le prédécesseur immédiat du concept objet (sup-irréductible)  $\gamma(g)$  et  $m^+$  représente le successeur immédiat du concept attribut (inf-irréductible)  $\mu(m)$ .

# Rappels XVI

## Relations flèches

	a	b	c	d	e	f
1	×	↖	×	↖	×	↘
2	↖	×	↖	×	↘	×
3	↖	↖	×	×	↘	↘
4	↖	×	×	↖	↘	×
5	×	×	↖	↖	↖	×
6	×	×	↖	×	×	↖

Figure 5: Relation flèches pour le contexte  $\mathbb{K}$

Par exemple,  $3 \downarrow e$  car

- $\gamma(3) \not\leq \mu(e)$  avec  $\gamma(3) = (3, cd)$  et  $\mu(e) = (16, ae)$
- $\gamma(3) \not\leq e^+$  avec  $e^+ = (156, a)$
- $3^- \leq \mu(e)$  avec  $3^- = (\emptyset, abcdef)$

# Les outils de l'analyse formelle de concepts

# Les outils de l'analyse formelle de concepts

Galicia

- Galicia<sup>1</sup> a été développé en Java par Pekto Valchev et ses collaborateurs (Université de Montréal).

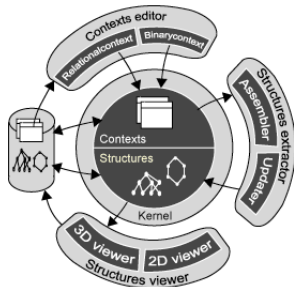


Figure 6: Cycle de vie du treillis  
Galicia

Quelques fonctionnalités de Galicia :

- Affichage du treillis en trois dimensions
- Manipulation du contexte et du treillis
- Plusieurs procédures de construction du treillis : Bordat, Nourine, Godin et *Next-Closure*
- Connection à des bases de données SQL pour sauvegarder les contextes

1. <http://www.iro.umontreal.ca/~galicia/>

# Les outils de l'analyse formelle de concepts

Galicia

Contexts Family name: KevinMemoire.rcf

File Edit Rules Generation Algorithms Database Console

KevinMemoire

A	B	C	D	E	F	G
KevinMem...	a	b	c	d	e	f
1	X	0	X	0	X	0
2	0	X	0	X	0	X
3	0	0	X	X	0	0
4	0	X	X	0	0	X
5	X	X	0	0	0	X
7	X	X	0	X	X	0

Open...

>

Figure 7: Éditeur de contexte

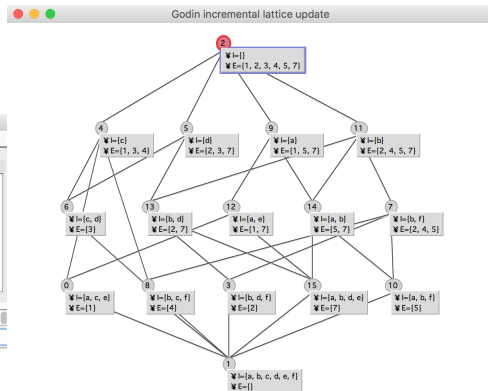


Figure 8: Treillis Galicia avec étiquetage complet



# Les outils de l'analyse formelle de concepts

## Le système Coron

Le système Coron<sup>2</sup> (Szathmary 2006) est une boîte à outils qui a été développée au laboratoire LORIA<sup>3</sup>

- Développé avec Java et Perl
- En ligne de commandes
- Version actuelle de ce système est 0.8 et date de janvier 2010
- Compatible avec les systèmes d'exploitation Unix, macOS et Windows
- Implémente plusieurs algorithmes de fouille de données tels qu'Apriori, Close, Aclose, Carpathia, Charm, Next-Closure ou NFI-Simple.
- Simple d'utilisation, modulaire et documentation riche
- Code source non disponible

---

2. <http://coron.loria.fr/site/index.php>.

3. Laboratoire lorrain de recherche en informatique et ses applications, Nancy, France

# Les outils de l'analyse formelle de concepts

## Librairie Java Lattices

La librairie *Java Lattices*<sup>4</sup> (Bertet et al. 2014) a été développée par le laboratoire L3i<sup>5</sup>.

- Permet de générer des treillis de Galois et des règles d'association
- Code source libre et disponible sur la plate-forme GitHub
- Entièrement écrit avec Java
- La librairie est compatible avec Maven (outil Java pour gérer les dépendances)
- Client en ligne de commandes disponible
- Algorithme *Limited Object Access (LOA)* (Demko et Bertet 2011)

---

4. <https://thegalactic.github.io/java-lattices>

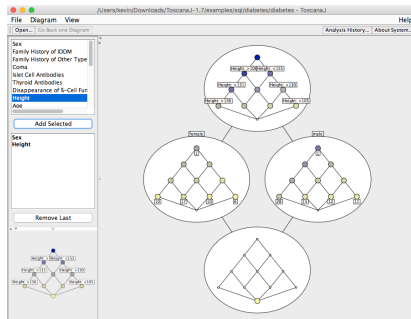
5. Laboratoire Informatique, Image et Interaction, université de La Rochelle, France

# Les outils de l'analyse formelle de concepts

## ToscanaJ

*ToscanaJ* (Becker, Hereth et Stumme 2002) est une réimplémentation avec le langage de programmation Java de l'outil d'AFC connu sous le nom de « *Toscana* ». Cet outil a la particularité d'être le premier outil à offrir l'affichage des diagrammes imbriqués (*nested line diagrams*) comme on peut le voir avec la figure 9.

Figure 9: ToscanaJ : Diagrammes imbriqués



*Concept Explorer*<sup>6</sup> (*ConExp*) est un outil souvent utilisé par les chercheurs en AFC.

- Interface graphique complète en Java Swing
- Édition et transformation de contextes
- Construction de treillis de concepts
- Code distribué librement
- Base Guigues-Duquenne
- Génération des règles d'association et l'exploration des attributs
- Génération des relations flèches

Dans le cadre de ce mémoire, nous allons utiliser *ConExp* à des fins de validation, notamment pour tester la procédure de production des relations flèches et pour la génération de la base de Guigues–Duquenne.

---

6. <http://conexp.sourceforge.net/>

# Les outils de l'analyse formelle de concepts

## Concept Explorer

2 ↓	1 < 1 > a b e = [100%] => < 1 > d;
1 ↓	2 < 1 > a c = [100%] => < 1 > e;
	3 < 1 > a d = [100%] => < 1 > b e;
	4 < 1 > b c = [100%] => < 1 > f;
	5 < 2 > e = [100%] => < 2 > a;
	6 < 3 > f = [100%] => < 3 > b;
	7 < 4 > b = [75%] => < 3 > f;
	8 < 6 > { } = [67%] => < 4 > b;
	9 < 3 > d = [67%] => < 2 > b;
	10 < 3 > a = [67%] => < 2 > e;
	11 < 3 > a = [67%] => < 2 > b;
	12 < 6 > { } = [50%] => < 3 > d;
	13 < 6 > { } = [50%] => < 3 > c;
	14 < 6 > { } = [50%] => < 3 > a;
	15 < 2 > b d = [50%] => < 1 > f;
	16 < 2 > b d = [50%] => < 1 > a e;
	17 < 2 > a b = [50%] => < 1 > f;
	18 < 2 > a e = [50%] => < 1 > c;
	19 < 3 > d = [33%] => < 1 > c;
	20 < 3 > c = [33%] => < 1 > b f;

7 < 4 > b = [75%] => < 3 > f;

Il s'agit de la 7-ème règle d'association laquelle indique que  $b$  implique  $f$  avec un support absolu de 3 parmi les six objets (et donc un support relatif de 50%) et une confiance de 75%.

Figure 10: ConExp : Règles d'association

# Les outils de l'analyse formelle de concepts

## Concept Explorer

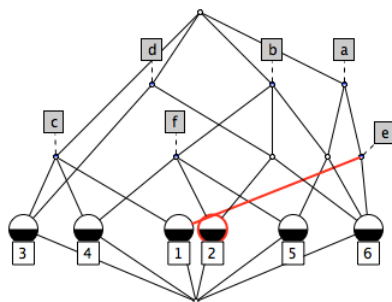


Figure 11: *ConExp* : Affichage du treillis

- L'affichage du treillis se fait selon un étiquetage réduit seulement.
- *ConExp* utilise un algorithme de construction du treillis avec le minimum d'intersections et les collisions sont affichées en rouge.

# Les outils de l'analyse formelle de concepts

## *Lattice Miner*

- Logiciel en Java pour la visualisation et la manipulation de treillis
- Initialement conçu par Geneviève Roberge dans le cadre de son mémoire de maîtrise.
- Continuellement enrichi par les membres de l'équipe et plusieurs stagiaires ingénieurs français.
- Aussi bien dans sa version initiale que dans sa version révisé, *Lattice Miner* est un logiciel libre disponible sur *SourceForge*<sup>7</sup> et *GitHub*<sup>8</sup>.

---

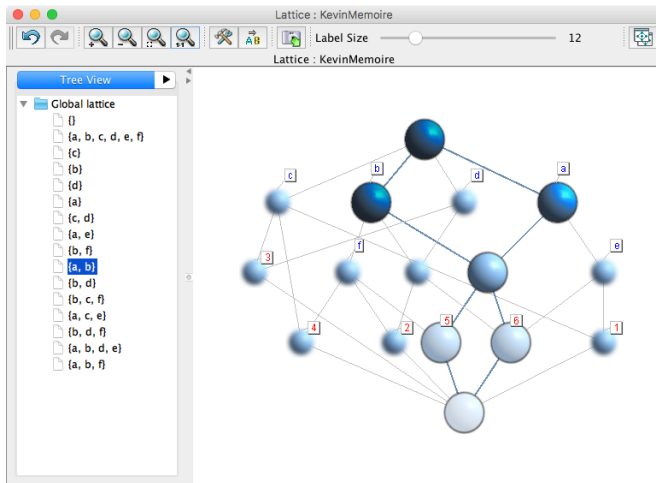
7. <https://sourceforge.net/projects/lattice-miner/>

8. <https://github.com/LarimUQO/lattice-miner>

# Les outils de l'analyse formelle de concepts

*Lattice Miner*

Figure 12: Génération du treillis de concepts dans *Lattice Miner*

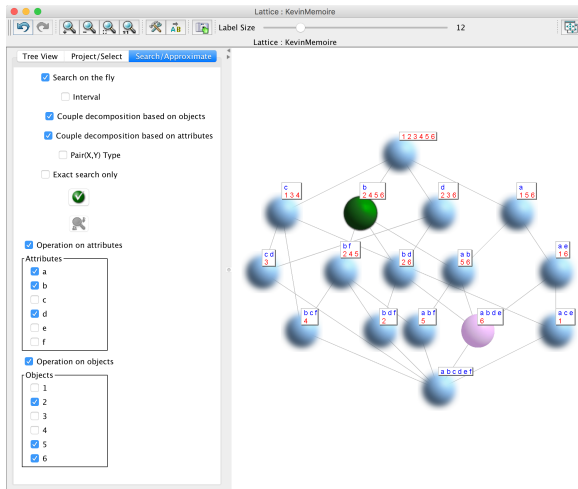




# Les outils de l'analyse formelle de concepts

*Lattice Miner*

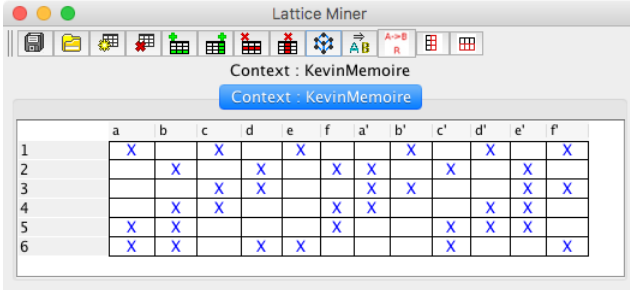
Figure 13: Approximation dans *Lattice Miner*



# Les outils de l'analyse formelle de concepts

*Lattice Miner*

Figure 14: Affichage de l'apposition d'un contexte avec son complémentaire



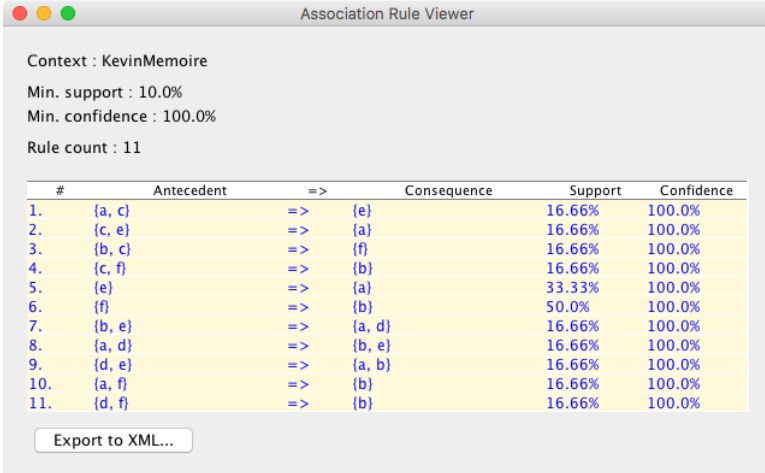
The screenshot shows the Lattice Miner application window. The title bar reads 'Lattice Miner'. Below the title bar is a toolbar with various icons for file operations and analysis. The main area displays the context 'KevinMemoire' and a button labeled 'Context : KevinMemoire'. Below this is a table with 6 rows and 12 columns. The columns are labeled a, b, c, d, e, f, a', b', c', d', e', f'. The rows are numbered 1 to 6. The table contains blue 'X' marks in the following cells: (1, a), (1, e), (1, f'), (2, b), (2, f), (2, a'), (2, c'), (2, e'), (3, c), (3, d), (3, a'), (3, f'), (4, b), (4, c), (4, f), (4, a'), (4, d'), (4, e'), (5, a), (5, b), (5, f), (5, c'), (5, d'), (5, e'), (6, a), (6, b), (6, d), (6, e), (6, c'), (6, f').

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

# Les outils de l'analyse formelle de concepts

*Lattice Miner*

Figure 15: Base générique d'implications avec possibilité de redondance



Association Rule Viewer

Context : KevinMemoire

Min. support : 10.0%

Min. confidence : 100.0%

Rule count : 11

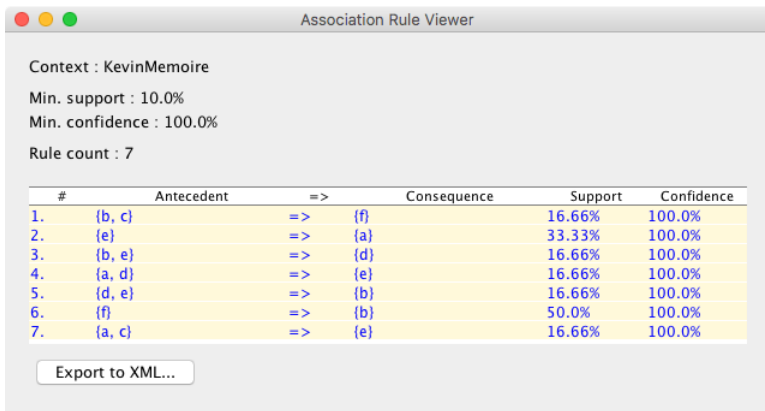
#	Antecedent	=>	Consequence	Support	Confidence
1.	{a, c}	=>	{e}	16.66%	100.0%
2.	{c, e}	=>	{a}	16.66%	100.0%
3.	{b, c}	=>	{f}	16.66%	100.0%
4.	{c, f}	=>	{b}	16.66%	100.0%
5.	{e}	=>	{a}	33.33%	100.0%
6.	{f}	=>	{b}	50.0%	100.0%
7.	{b, e}	=>	{a, d}	16.66%	100.0%
8.	{a, d}	=>	{b, e}	16.66%	100.0%
9.	{d, e}	=>	{a, b}	16.66%	100.0%
10.	{a, f}	=>	{b}	16.66%	100.0%
11.	{d, f}	=>	{b}	16.66%	100.0%

Export to XML...

# Les outils de l'analyse formelle de concepts

*Lattice Miner*

Figure 16: Base générique d'implications sans redondance



Association Rule Viewer

Context : KevinMemoire

Min. support : 10.0%

Min. confidence : 100.0%

Rule count : 7

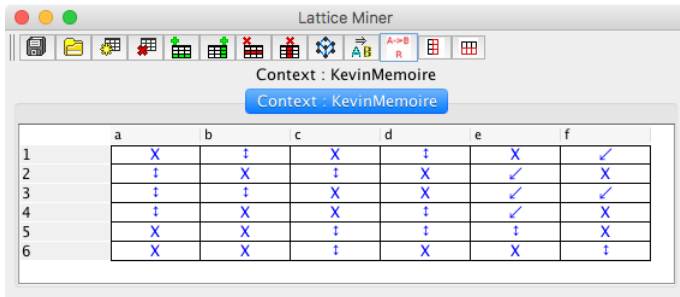
#	Antecedent	=>	Consequence	Support	Confidence
1.	{b, c}	=>	{f}	16.66%	100.0%
2.	{e}	=>	{a}	33.33%	100.0%
3.	{b, e}	=>	{d}	16.66%	100.0%
4.	{a, d}	=>	{e}	16.66%	100.0%
5.	{d, e}	=>	{b}	16.66%	100.0%
6.	{f}	=>	{b}	50.0%	100.0%
7.	{a, c}	=>	{e}	16.66%	100.0%

Export to XML...

# Les outils de l'analyse formelle de concepts

*Lattice Miner*

Figure 17: Relations flèches dans *Lattice Miner*



The screenshot shows the Lattice Miner application window. The title bar reads 'Lattice Miner'. Below the title bar is a toolbar with various icons for file operations, data manipulation, and logic. The main area displays the context 'KevinMemoire' and a table with 6 rows and 6 columns labeled a through f. The table contains blue symbols: 'X' for presence, '†' for absence, and '✓' for a specific relationship.

	a	b	c	d	e	f
1	X	†	X	†	X	✓
2	†	X	†	X	✓	X
3	†	†	X	X	✓	✓
4	†	X	X	†	✓	X
5	X	X	†	†	†	X
6	X	X	†	X	X	†

Production d'implications avec négation



Jean-Louis Guigues et Vincent Duquenne. « Familles minimales d'implications informatives résultant d'un tableau de données binaires ». In : *Mathématiques et Sciences Humaines* 95 (1986), p. 5–18.



Bernhard Ganter et Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., 1999.



Nicolas Pasquier et al. « Efficient Mining of Association Rules Using Closed Itemset Lattices ». In : *Information Systems* 24.1 (1999), p. 25–46.



Peter Becker, Joachim Hereth et Gerd Stumme. « ToscanaJ: An Open Source Tool for Qualitative Data Analysis ». In : *Advances in Formal Concept Analysis for Knowledge Discovery in Databases (FCAKDD'2002)*. Juil. 2002, p. 1–2.

## References II



J. Pfaltz et C. Taylor. « Scientific Discovery through Iterative Transformations of Concept Lattices ». In : *Proceedings of the 1st International Workshop on Discrete Mathematics and Data Mining*. Avr. 2002, p. 65–74.



Bernhard Ganter et Sergei A. Obiedkov. « Implications in Triadic Formal Contexts ». In : *ICCS*. 2004, p. 186–195.



Laszlo Szathmary. « Symbolic Data Mining Methods with the Coron Platform ». *Thèse de doctorat*. Univ. Henri Poincaré - Nancy 1, France, nov. 2006.



Geneviève Roberge. « Visualisation des résultats d'une fouille de données dans les treillis de concepts ». *Mém. de mast.* Université du Québec en Outaouais, 2007.





Christophe Demko et Karell Bertet. « Generation Algorithm of a Concept Lattice with Limited Object Access ». In : *Proceedings of The Eighth International Conference on Concept Lattices and Their Applications*. 2011, p. 239–250.



Rokia Missaoui, Lhouari Nourine et Yoan Renaud. « Computing Implications with Negation from a Formal Context ». In : *Fundam. Inf.* 115.4 (déc. 2012), p. 357–375.



Karell Bertet et al. *java-lattices: a Java library for lattices computation*. <http://thegalactic.org>. 2014.



Jean-François Viaud et al. « Décomposition sous-directe d'un treillis en facteurs irréductibles ». In : *Journées francophones d'Ingénierie des Connaissances IC 2015*. collection AFIA. Rennes, France, juin 2015.

Questions ?