Algorithmic Knowledge Discovery with Concept Lattices

2.Formal Concept Analysis and FCA-based Knowledge Discovery

Sergei O. Kuznetsov

School of Applied Mathematics and Information Science
National Research University Higher School of Economics, Moscow, Russia

International Summer School "Information and Uncertainty"
Palacky University, Olomouc, June 4, 2012

FCA. Short (pre)history.

- Logique de Port-Royal (1662)
- G. Birkhoff, starting from 1930s
- O. Øre, starting from 1930x
- M. Barbut, B. Monjardet, Ordre et classification, Hachette, Paris, 1970
- R. Wille, Restructuring lattice theory: An approach based on hierarchies of concepts, 1982
- B. Ganter, R. Wille, Formale Begriffsanalyse, Springer, 1996
- B. Ganter, R. Wille, Formal Concept Analysis, Springer, 1999
- Chapter in B. Davey, H. Priestly, Introduction to Order and Lattices, 1990.
- Chapter in G. Grätzer (Ed.), General Lattice Theory.
- Concept Data Analysis, C.Carpineto, G. Romano, 2004.

FCA. Main conferences

- International Conference on Conceptual Structures (ICCS), FCA participation starting from 1996 (Proceedings in LNAI, Springer)

- International Conference on Formal Concept Analysis (ICFCA), from 2003 года (Proceedings in LNAI, Springer)

- International Conference on Concept Lattices and Their Applications (CLA), from 2006, special issues

FCA-based Knowledge Discovery: An Outline

1. **An Introduction to FCA**
2. Attribute Exploration
3. Learning JSM-hypotheses
4. Relational Learning with Pattern Structures
5. FCA in Data Mining: Learning Association Rules
6. Algorithms and Complexity of FCA-based Knowledge Discovery

Formal Concept Analysis. 1

[Wille 1982], [Ganter, Wille 1996]

Let two sets G and M be given. Elements of G are called **objects**, elements of M are called **attributes**.
Let $I \subseteq G \times M$ be a binary relation. If $(g, m) \in I$, one says that object g has attribute m. Triple $\mathbb{K} := (G, M, I)$ is called a **(formal) context**.

Formal Concept Analysis. 3

[Wille 1982], [Ganter, Wille 1996]

Let $\mathbb{K} := (G, M, I)$ be a context. In FCA, instead of two notations φ and ψ, a unified notation $(\cdot)'$ is used, so for arbitrary $A \subseteq G$, $B \subseteq M$

$$A' \stackrel{\text{def}}{=} \{m \in M \mid gIm \text{ for all } g \in A\}, \quad B' \stackrel{\text{def}}{=} \{g \in G \mid gIm \text{ for all } m \in B\}.$$

(Formal) concept is a pair (A, B):

$$A \subseteq G,\ B \subseteq M,\ A' = B, \ \text{ and } \ B' = A.$$

A is called a **(formal) extent**, and B is called a **(formal) intent** of a concept (A, B).

Concepts are partially ordered by relation
$(A_1, B_1) \geq (A_2, B_2) \iff A_1 \supseteq A_2 \quad (B_2 \supseteq B_1)$.

Example. Context

	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

Objects:

1 – equilateral triangle,

2 – rectangle triangle,

3 – rectangle,

4 – square

Attributes:

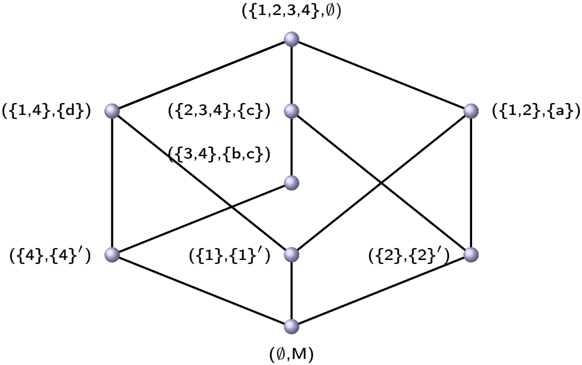
a – has 3 vertices,

b – has 4 vertices,

c – has a direct angle,

d – equilateral

Example. Diagram of the ordered set of concepts



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

a – has 3 vertices,

b – has 4 vertices,

c – has a direct angle,

d – equilateral

Properties of operation $(\cdot)'$

Let (G, M, I) be a formal context, $A, A_1, A_2 \subseteq G$ be subsets of objects, $B \subseteq M$ be subsets of attributes, then

1. If $A_1 \subseteq A_2$, then $A_2' \subseteq A_1'$;
2. If $A_1 \subseteq A_2$, then $A_1'' \subseteq A_2''$
3. $A \subseteq A''$
4. $A''' = A'$ (hence, $A'''' = A''$);
5. $(A_1 \cup A_2)' = A_1' \cap A_2'$;
6. $A \subseteq B' \Leftrightarrow B \subseteq A' \Leftrightarrow A \times B \subseteq I$.

Similar properties hold for subsets of attributes.

Closure operator on a set

A **closure operator** on set G is a mapping $\varphi\colon \mathcal{P}(G) \to \mathcal{P}(G)$ with the following properties:

1. $\varphi\varphi X = \varphi X$ (**idempotency**)
2. $X \subseteq \varphi X$ (**extensity**)
3. $X \subseteq Y \Rightarrow \varphi X \subseteq \varphi Y$ (**monotonicity**)

For a closure operator φ the set φX is called **closure** of X.
A subset $X \subseteq G$ is called **closed** if $\varphi X = X$.
Example. Let (G, M, I) be a context, then operators $(\cdot)''\colon 2^G \to 2^G$, $(\cdot)''\colon 2^M \to 2^M$ are closure operators.

Basic Theorem of Formal Concept Analysis

[Wille 1982], [Ganter, Wille 1996]

Concept lattice $\underline{\mathfrak{B}}(G, M, I)$ is a complete lattice. For arbitrary sets of formal concepts

$$\{(A_j, B_j) \mid j \in J\} \subseteq \underline{\mathfrak{B}}(G, M, I)$$

infimums and supremums are given in the following way:

$$\bigwedge_{j \in J} (A_j, B_j) = (\bigcap_{j \in J} A_j, (\bigcup_{j \in J} B_j)''),$$
$$\bigvee_{j \in J} (A_j, B_j) = ((\bigcup_{j \in J} A_j)'', \bigcap_{j \in J} B_j).$$

A complete lattice V is isomorphic to a lattice $\underline{\mathfrak{B}}(G, M, I)$ iff there are mappings $\gamma\colon G \to V$ and $\mu\colon M \to V$ such that $\gamma(G)$ is supremum-dense in V, $\mu(M)$ is infimum-dense in V, and $gIm \Leftrightarrow \gamma g \leq \mu m$ for all $g \in G$ and all $m \in M$.
In particular, $V \cong \underline{\mathfrak{B}}(V, V, \leq)$.

*Reducing attributes

Attribute $m \in M$, $\mathbb{K} = (G, M, I)$ is **reducible** if

$$m' = G \text{ or } m' = \bigcap\{n' \mid n \in M \;\&\; n' \supset m'\}.$$

If m is reducible, then $\underline{\mathfrak{B}}(G, M, I) \cong \underline{\mathfrak{B}}(G, M \setminus \{m\}, I \cap (G \times (M \setminus \{m\})))$

Dually for objects. An irreducible attribute corresponds to a meet-irreducible lattice element, an irreducible object corresponds to a join-irreducible lattice element.

Example. Attribute m_k is reducible, since $m_k' = m_i' \cap m_j'$

$G \setminus M$	...	m_i	...	m_j	...	m_k	...
g_1	...	×	...		...		...
g_2	...	×	...	×	...	×	...
g_3	...	×	...	×	...	×	...
g_4	...		...	×	...		...

Implications on subsets of attributes

Implication $A \to B$, where $A, B \subseteq M$ holds in context (G, M, I) if $A' \subseteq B'$, i.e., each object having all attributes from A also has all attributes from B.

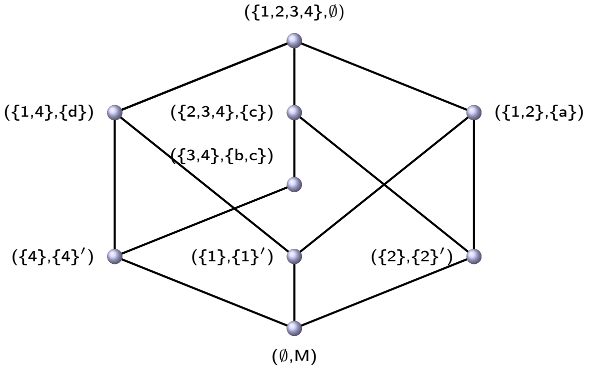
Implications and concept lattice: If $A \to B$, then meet of all attribute concepts for attributes from A in the lattice diagram lies below the meet of all attribute concepts of attributes in B.

Implications satisfy **Armstrong rules**:

$$\frac{}{X \to X}\ ,\quad \frac{X \to Y}{X \cup Z \to Y}\ ,\quad \frac{X \to Y, Y \cup Z \to W}{X \cup Z \to W},$$

An **implication cover** is a subset of implications from which all other implications can be derived by means of Armstrong rules.
An **implication base** is a minimal (by inclusion) implication cover.

Concept lattice and implications



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

a – exactly 3 vertices,
b – exactly 4 vertices,
c – has a direct angle,
d – equilateral

Implications:

$abc \rightarrow d$

$b \rightarrow c$

$cd \rightarrow b$

Generator-based implication cover

A subset of attributes $D \subseteq M$ is a **generator** of a closed subset of attributes $B \subseteq M$, $B'' = B$ if $D \subseteq B$, $D'' = B = B''$.
A subset $D \subseteq M$ is a **minimal generator** if for any $E \subset D$ one has $E'' \neq D'' = B''$.
Generator $D \subseteq M$ is called **nontrivial** if $D \neq D'' = B''$.
Denote the set of all nontrivial minimal generators of B by nmingen(B).

Generator implication cover looks as follows:
$\{F \to (F'' \setminus F) \mid F \subseteq M, F \in \text{ nmingen } (F'')\}$.

Minimum implication basis

Duquenne-Guigues base is an implication base where each implication is a pseudo-intent.
A subset of attributes $P \subseteq M$ is called a **pseudo-intent** if $P \neq P''$ and for any pseudo-intent Q such that $Q \subset P$ one has $Q'' \subset P$.

Duquenne-Guigues base looks as follows:
$\{P \rightarrow (P'' \setminus P) \mid P$ - pseudo-intent $\}$.

Duquenne-Guigues base is a minimum (cardinality minimal) implication base.

Implications and functional dependencies. 1

Functional dependency (e.g., Meier 1983) in terms of FCA:
$X \to Y$ is a **functional dependency** in a complete many-valued context (G, M, W, I) if the following holds for every pair of objects $g, h \in G$:

$$(\forall m \in X \ m(g) = m(h)) \Rightarrow (\forall n \in Y \ n(g) = n(h)).$$

The reduction of functional dependencies to implications:
Proposition A. For a many-valued context (G, M, W, I), one defines the context $K_N := (\mathcal{P}_2(G), M, I_N)$, where $\mathcal{P}_2(G)$ is the set of all pairs of different objects from G and I_N is defined by

$$\{g, h\} I_N m :\Leftrightarrow m(g) = m(h).$$

Then a set $Y \subseteq M$ is functionally dependent on the set $X \subseteq M$ if and only if the implication $X \to Y$ holds in the context K_N.

Implications and functional dependencies. 2

There is an inverse reduction:

Proposition B. For a context $K = (G, M, I)$ one can construct a many-valued context K_W such that an implication $X \to Y$ holds if and only if Y is functionally dependent on X in K_W.

FCA-based Knowledge Discovery: An Outline

Attribute exploration

- Start with any (possibly empty) set of objects.
- Generate an implication valid in the current subcontext.
- If the implication is not valid in the entire context, provide an object that violates it.
- Go to the next implication, etc.

Follow the Duquenne-Guigues basis to ask no more questions than is strictly necessary.

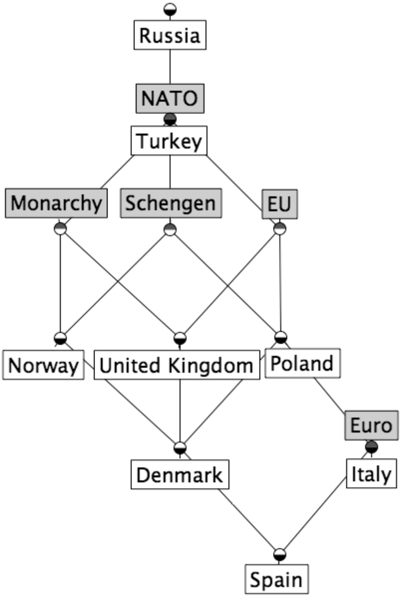
Attribute exploration

- Start with any (possibly empty) set of objects.
- Generate an implication valid in the current subcontext.
- If the implication is not valid in the entire context, provide an object that violates it.
- Go to the next implication, etc.

Follow the Duquenne-Guigues basis to ask no more questions than is strictly necessary.

Attribute exploration

European states



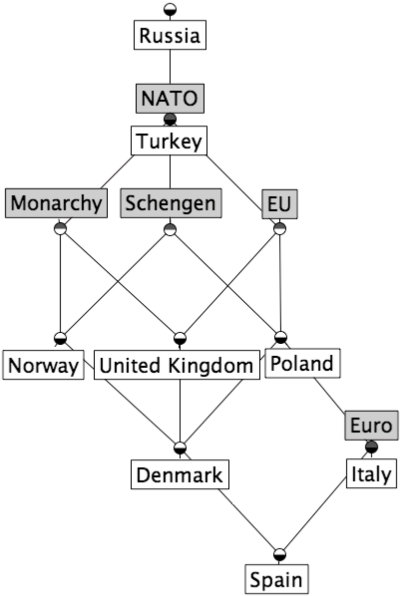
Question Is every European monarchy in NATO?

Answer No: Sweden is not.

	EU	Euro	Schengen	NATO	Monarchy
Sweden	×		×		×

Attribute exploration

European states



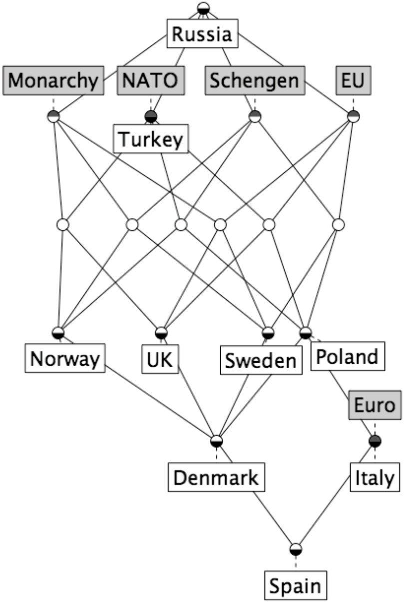
Question Is every European monarchy in NATO?

Answer No: Sweden is not.

	EU	Euro	Schengen	NATO	Monarchy
Sweden	×		×		×

Attribute exploration

European states



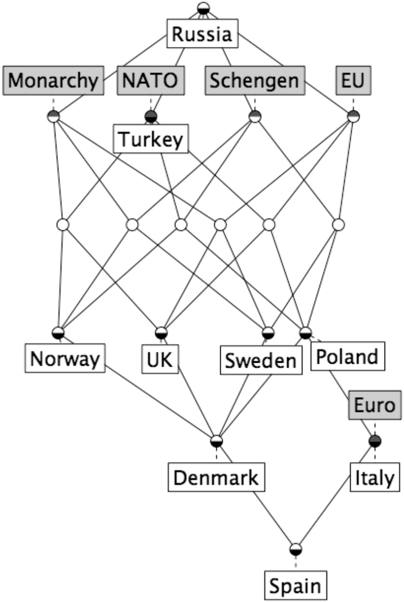
Question Is every Eurozone country in EU, Schengen, and NATO?

Answer No: Ireland is not.

	EU	Euro	Schengen	NATO	Monarchy
Ireland	×	×			

Attribute exploration

European states



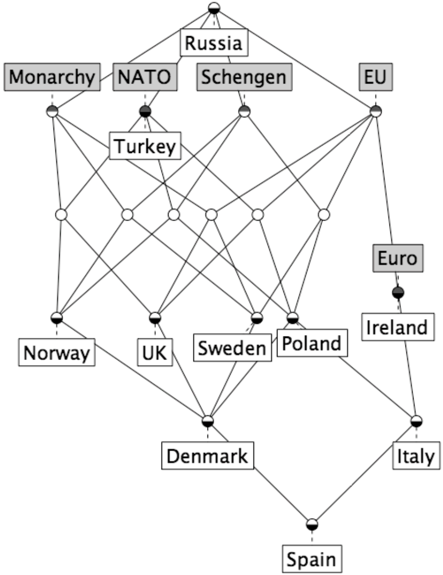
Question Is every Eurozone country in EU, Schengen, and NATO?

Answer No: Ireland is not.

	EU	Euro	Schengen	NATO	Monarchy
Ireland	×	×			

Attribute exploration

European states

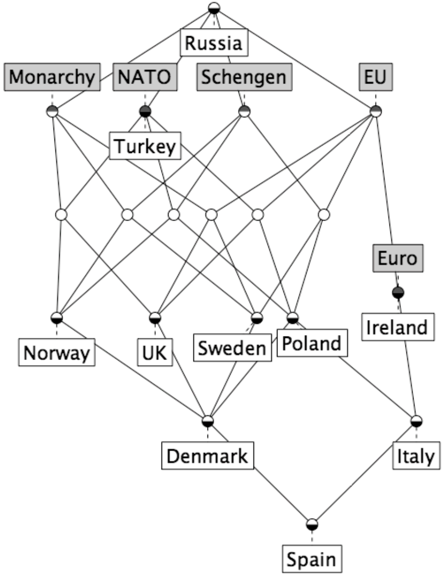


Question Is every Eurozone country in EU?

Answer No: Montenegro is not. . .

Attribute exploration

European states



Question Is every Eurozone country in EU?

Answer No: Montenegro is not...

FCA-based Knowledge Discovery: An Outline

1. An Introduction to FCA
2. Attribute Exploration
3. **Learning JSM-hypotheses**
4. Relational learning with pattern structures
5. FCA in Data Mining: Learning Association Rules
6. Algorithms and Complexity of FCA-based Knowledge Discovery

JSM-method of hypothesis generation

A **target attribute** $w \notin M$,

- **positive examples**: Set $G_+ \subseteq G$ of objects known to have w,
- **negative examples**: Set $G_- \subseteq G$ of objects known not to have w,
- **undetermined examples**: Set $G_\tau \subseteq G$ of objects for which it is unknown whether they have the target attribute or do not have it.

Three subcontexts of $\mathbb{K} = (G, M, I)$: $\mathbb{K}_\varepsilon := (G_\varepsilon, M, I_\varepsilon)$, $\varepsilon \in \{-, +, \tau\}$ with respective derivation operators $(\cdot)^+$, $(\cdot)^-$, and $(\cdot)^\tau$.

A **positive hypothesis** $H \subseteq M$ is an intent of $\mathbb{K}_+$ not contained in the intent g^- of any negative example $g \in G_-$: $\forall g \in G_- \quad H \not\subseteq g^-$. Equivalently,

$$H^{++} = H, \quad H' \subseteq G_+ \cup G_\tau.$$

Example of a learning context

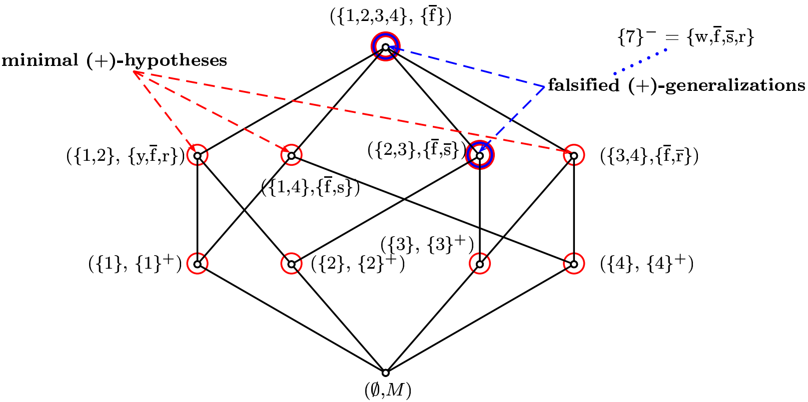
	G \ M	color	firm	smooth	form	**fruit**
1	apple	yellow	no	yes	round	+
2	grapefruit	yellow	no	no	round	+
3	kiwi	green	no	no	oval	+
4	plum	blue	no	yes	oval	+
5	toy cube	green	yes	yes	cubic	−
6	egg	white	yes	yes	oval	−
7	tennis ball	white	no	no	round	−
8	mango	green	no	yes	oval	τ

Natural scaling of the context

	G \ M	w	y	g	b	f	$\bar{\text{f}}$	s	$\bar{\text{s}}$	r	$\bar{\text{r}}$	**fruit**
1	apple		×				×	×		×		$+$
2	grapefruit		×				×		×	×		$+$
3	kiwi			×			×		×		×	$+$
4	plum				×		×	×			×	$+$
5	toy cube			×		×		×			×	$-$
6	egg	×				×		×			×	$-$
7	tennis ball	×					×		×	×		$-$
8	mango			×			×	×			×	τ

Abbreviations:
"g" for green, "y" for yellow, "w" for white, "f" for firm, "$\bar{\text{f}}$" for nonfirm,
"s" for smooth, "$\bar{\text{s}}$" for nonsmooth, "r" for round,
"$\bar{\text{r}}$" for nonround.

Positive Concept Lattice



	G \ M	w	y	g	b	f	$\overline{f}$	s	$\overline{s}$	r	$\overline{r}$	fruit
1	apple		×				×	×		×		+
2	grapefruit		×				×		×	×		+
3	kiwi			×			×		×		×	+
4	plum				×		×	×			×	+
5	toy cube			×		×		×			×	−
6	egg	×				×		×			×	−
7	tennis ball	×					×		×	×		−
8	mango			×			×	×			×	τ

Hypotheses vs. implications

A positive hypothesis h corresponds to an implication $h \to \{w\}$ in the context $K_+ = (G_+, M \cup \{w\}, I_+ \cup G_+ \times \{w\})$.
A negative hypothesis h corresponds to an implication $h \to \{\bar{w}\}$ in the context $K_- = (G_-, M \cup \{\bar{w}\}, I_- \cup G_- \times \{\bar{w}\})$.
Hypotheses are special implications: their premises are closed (in K_+ or in K_-).

	G \ M	w	y	g	b	f	$\bar{\text{f}}$	s	$\bar{\text{s}}$	r	$\bar{\text{r}}$	**fruit**	**nonfruit**
1	apple		×				×	×		×		×	
2	grapefruit		×				×		×	×		×	
3	kiwi			×			×		×		×	×	
4	plum				×		×	×			×	×	
5	toy cube			×		×		×			×		×
6	egg	×				×		×			×		×
7	tennis ball	×					×		×	×			×

FCA-based Knowledge Discovery: An Outline

Learning with labeled graphs: A motivation

- Structure-Activity Relationship problems for chemicals given by molecular graphs
- Learning semantics from graph-based (XML, syntactic tree) text representation

Starting point

To proceed with graphs like it was done for objects described by binary sets of attributes (i.e., for contexts), one should define for graphs an operation $\sqcap$ similar to that of set-theoretic $\cap$ (since then a closure operator $''$ can be defined).

The first natural attempt to do this, like introducing an operation "take the largest common subgraph of two graphs" fails, since there can be several subgraphs of this type.

Perhaps operation should be defined not for graphs, but for sets of graphs?

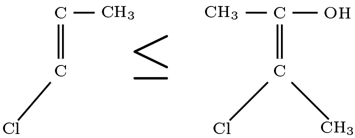
The attempt even fails if we take all largest (in the number of vertices) common subgraphs of two graphs.

Order on labeled graphs

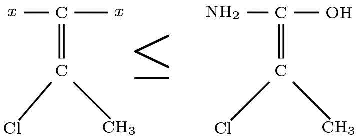
Let $(\mathcal{L}, \preceq)$ be an ordered set of vertex labels.
$\Gamma_1 := ((V_1, l_1), E_1)$ **dominates** $\Gamma_2 := ((V_2, l_2), E_2)$ or $\Gamma_2 \leq \Gamma_1$
if there exists a one-to-one mapping $\varphi \colon V_2 \to V_1$ such that

- respects edges: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- fits under labels: $l_2(v) \preceq l_1(\varphi(v))$.

Example: $\mathcal{L} = \{x, NH_2, Cl, CH_3, C, OH\}$



vertex labels are unordered



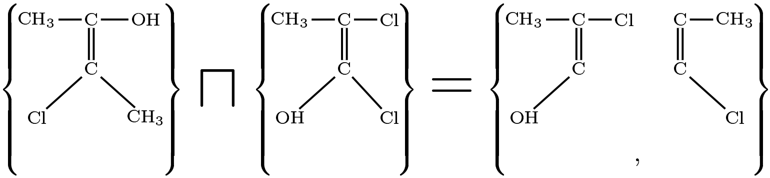
$x \preceq A$ for any vertex label $A \in \mathcal{L}$

Semilattice on graph sets

$$\{X\} \sqcap \{Y\} := \{Z \mid Z \leq X, Y, \quad \forall Z_* \leq X, Y \quad Z_* \not< \ Z\}$$

$= $ The set of all maximal common subgraphs of X and Y.

Example:



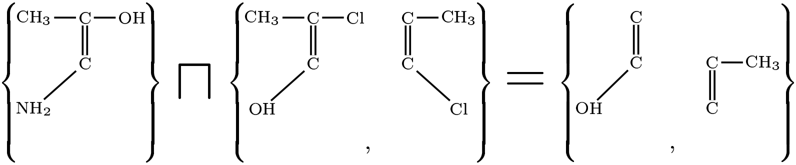
Meet of graph sets

For sets of graphs
$\mathcal{X} = \{X_1, \ldots, X_k\}$ and $\mathcal{Y} = \{Y_1, \ldots, Y_n\}$

$\mathcal{X} \sqcap \mathcal{Y} :=$ MAX $_{\leq}(\bigcup_{i,j}(\{X_i\} \sqcap \{Y_j\}))$

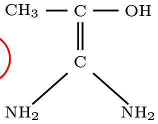
$\sqcap$ is idempotent, commutative, and associative.

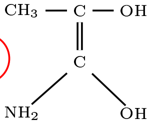
Example:

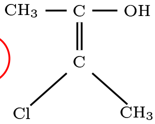


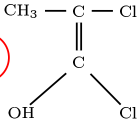
Examples

Positive examples:

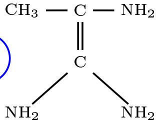
G_1: 

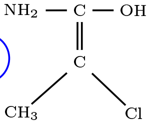
G_2: 

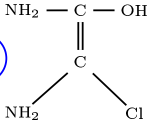
G_3: 

G_4: 

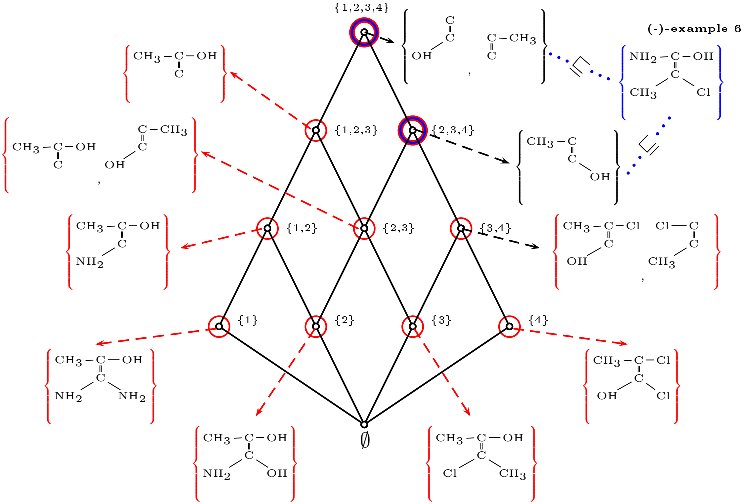
Negative examples:

G_5: 

G_6: 

G_7: 

Positive lattice



positive examples 1, 2, 3, 4

*Pattern Structures

[Ganter, Kuznetsov 2001]

$(G, \underline{D}, \delta)$ is a **pattern structure** if

- G is a set ("set of objects");
- $\underline{D} = (D, \sqcap)$ is a meet-semilattice;
- $\delta : G \to D$ is a mapping;
- the set $\delta(G) := \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$.

Possible origin of $\sqcap$ operation:

- A set of objects G, each with description from P;
- Partially ordered set $(P, \leq)$ of "descriptions" ($\leq$ is a "more general than" relation);
- The (distributive) lattice of order ideals of the ordered set $(P, \leq)$.

FCA-based Knowledge Discovery: An Outline

1. An Introduction to FCA
2. Attribute Exploration
3. Learning JSM-hypotheses
4. Relational Learning with Pattern Structures
5. **FCA in Data Mining: Learning Association Rules**
6. Algorithms and Complexity of FCA-based Knowledge Discovery

Lattices in Data Mining. Association rules

In mid 1990s in papers of R. Agrawal et al. on association rules "partial implications" from FCA were rediscovered.

A partial implication (association rule) of context (G, M, I) is an expression $A \to_{c,s} B$, where

- $c, s \in [0, 1]$;
- $c = \frac{|(A \cup B)'|}{|A'|}$, called **confidence**, $\mathrm{conf}(A \to B)$;
- $s = \frac{|(A \cup B)'|}{|G|}$, called **support**, $\mathrm{supp}(A \to B)$.

Covers of association rules

What is a minimal representation of the set of association rules, from which one can obtain all association rules of a context using "admissible transformations"?
Consider an association rule $A \rightarrow_{c,s} B$. Under fixed confidence $c = \frac{|(A \cup B)'|}{|A'|}$ and support $s = \frac{|(A \cup B)'|}{|G|}$ we try to reduce its premise and increase its conclusion.

1. **Decreasing premise.** For fixed c and s one can decrease premise from A to a certain subset $D \subseteq A$ such that $(D \cup B)' = (A \cup B)' = A' \cap B' = D' \cap B'$, that is $D' = A' = A'''$. Thus, minimal D is by definition a minimal generator of A'', i.e. $D \in$ mingen(A'').

Recall that a subset of attributes $D \subseteq M$ is a **generator** of a closed subset of attributes $B \subseteq M$, $B'' = B$ if $D \subseteq B$, $D'' = B = B''$. A subset $D \subseteq M$ is a **minimal generator** if for any $E \subset D$ one has $E'' \neq D'' = B''$.

Covers of association rules

2. **Increasing conclusion.** Conclusion B can be increased by a set Δ such that $(A \cup B)' = (A \cup B \cup \Delta)' = (A \cup B)' \cap \Delta'$, which is possible only when $(A \cup B)' \subseteq \Delta'$, which is equivalent to $A \cup B \to \Delta$ and to $\Delta \subseteq (A \cup B)''$. Thus, conclusion of the association rule can be increased up to $(A \cup B)''$.
Thus, the rules from the set

$$CP(K) = \{D \to (A \cup B)'' \mid D \in \text{ mingen } (A'')\}$$

make a cover of the set of all association rules. We can obtain all other rules by admissible transformations – increasing premises and decreasing conclusions (these operations do not decrease confidence and support) – of rules from $CP(K)$. In terms of these admissible transformations, $CP(K)$ makes a cover of association rules.

Base of association rules

Consider an association rule of the form $D \to (A \cup B)''$, where $D \in \text{mingen}(A'')$. In the concept lattice diagram this rule corresponds to a path from the concept (A', A'') to the concept $((A \cup B)', (A \cup B)'')$. If $(A', A'') \not\succ ((A \cup B)', (A \cup B)'')$, i.e., if the vertex of the diagram corresponding to the concept (A', A'') is not an upper neighbor of the vertex corresponding to $((A \cup B)', (A \cup B)'')$, then there is a concept (E', E'') such that $(A', A'') \succ (E', E'') > ((A \cup B)', (A \cup B)'')$.
Consider $D \to E''$, where $D \in \text{mingen}(A'')$ and $F \to (A \cup B)''$, where $F \in \text{mingen}(E'')$. The confidence of the first rule is $c_1 = \frac{|E'|}{|A'|}$, and the confidence of the second rule is $c_2 = \frac{|(A \cup B)'|}{|E'|}$. The confidence of the rule $D \to (A \cup B)''$, where $D \in \text{mingen}(A'')$ is

$$c = \frac{|(A \cup B)'|}{|A'|} = \frac{|E'|}{|A'|} \cdot \frac{|(A \cup B)'|}{|E'|} = c_1 \cdot c_2.$$

Base of association rules

Hence, the cover of the set of association rules can be made even smaller by restricting to the set of rules

$$\{F \to ('' \setminus F'') \mid F \subseteq M, F \in \text{ mingen } (F''), (F', F'') \succ (E', E'')\},$$

which correspond to the arcs of the diagram. Supports and confidence of other rules from the cover can be obtained by multiplying supports along the respective paths in the diagram.

To minimize the cover of association rules, making it a basis, one can retain only those rules from $CP(K)$ that correspond to edges from a spanning tree of the lattice diagram.

General task of finding association rules

Find all "frequent" (with support greater than a threshold) association rules with confidence greater than a threshold.

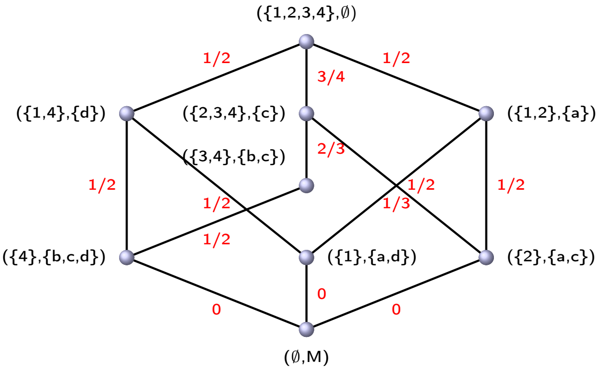
Solution stages

- Find all frequent "closed itemsets" (frequent intents)
- For each frequent intent B find all its maximal subintents $A_1, \ldots, A_n$
- Retain only those A_i for which $\mathrm{conf}(A_i \to B) \geq \theta$, where θ is confidence threshold
- Find minimal generators of the remaining A_i, compose rules of the form $mingen(A_i) \to B$.

Luxenburger basis

- Spanning tree of the concept lattice diagram
- Duquenne-Guigues implication base

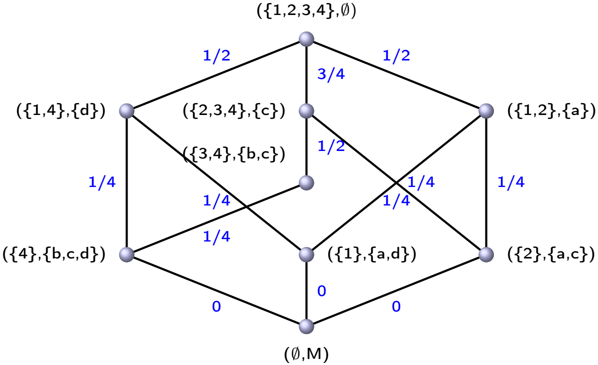
Example. Confidence of association rules



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

Good rules with *supp* $\geq 1/2$ and *minconf* $\geq 3/4$
1. $\emptyset \rightarrow c$, sup($\emptyset \rightarrow c$) = conf($\emptyset \rightarrow c$) = 3/4;
2. $c \rightarrow b$, sup($c \rightarrow b$)= 1/2, conf($c \rightarrow b$) = 2/3.

Example. Support of association rules



	G \ M	a	b	c	d
1		×			×
2		×		×	
3			×	×	
4			×	×	×

Good rules with *supp* $\geq 1/2$ and *minconf* $\geq 3/4$
1. $\emptyset \to c$, sup$(\emptyset \to c)$ = conf$(\emptyset \to c) = 3/4$;
2. $c \to b$, sup$(c \to b)$= $1/2$, conf$(c \to b)$ = $2/3$.

FCA-based Knowledge Discovery: An Outline

1. An Introduction to FCA
2. Attribute Exploration
3. Learning JSM-hypotheses
4. Relational Learning with Pattern Structures
5. FCA in Data Mining: Learning Association Rules
6. **Algorithms and Complexity of FCA-based Knowledge Discovery**

Concept lattice of size 2^n

Consider context $K = (G, G \neq)$ for an arbitrary finite set G. Then $\underline{\mathfrak{B}}(K)$ is isomorphic to the Boolean lattice 2^G.

D.-G. implication base of size 2^n

$G \setminus M$	m_0	$m_1 \dots m_n$	$m_{n+1} \dots m_{2n}$
g_1			
$\vdots$		$\neq$	$\neq$
g_n			
g_{n+1}	$\times$		
$\vdots$	$\vdots$		
$\vdots$	$\vdots$	$\neq$	
$\vdots$	$\vdots$		
g_{3n}	$\times$		

The set $\{m_1, \dots, m_n\}$ is a pseudo-intent. Replacing m_i with m_{n+i} independently for each i, one obtains all 2^n pseudo-intents.

An example: $K_{\exp,3}$

$G \setminus M$	m_0	m_1	m_2	m_3	m_4	m_5	m_6
g_1			×	×		×	×
g_2		×		×	×		×
g_3		×	×		×	×	
g_4	×		×	×	×	×	×
g_5	×	×		×	×	×	×
g_6	×	×	×		×	×	×
g_7	×	×	×	×		×	×
g_8	×	×	×	×	×		×
g_9	×	×	×	×	×	×	

Here, we have $2^3 = 8$ pseudo-intents: $\{m_1, m_2, m_3\}$, $\{m_1, m_2, m_6\}$, $\{m_1, m_5, m_3\}$, $\{m_1, m_5, m_6\}$, $\{m_4, m_2, m_3\}$, $\{m_4, m_2, m_6\}$, $\{m_4, m_5, m_3\}$, $\{m_4, m_5, m_6\}$.

#P and #P-completeness

Definition: #P is the class of counting problems associated with the decision problems in NP. More formally, a problem is in #P if there is a non-deterministic, polynomial time Turing machine that, for each instance I of the problem, has a number of accepting computations that is exactly equal to the number of distinct solutions for instance I.

A problem is #P-**complete** if it is in #P and it is
A problem is #P-**hard**, i.e., any problem in #P can be reduced by Turing to it.
In particular, a problem in #P is #P-complete if a #P-complete problem can be reduced to it. Obviously, $\#\text{P} = \text{P} \Longrightarrow \text{NP} = \text{P}$.

Examples of #P-complete problems:

- Given a matrix, output its permanent
- Given a bipartite graph, output the number of its perfect matchings
- Given a CNF, output the number of its satisfying assignments
- Given a graph, output the number of its vertex covers
- Given a context, output the number of its concepts

#P-hardness of counting pseudo-intents

Proposition 2. The following problem is $\#P$-hard.
INPUT A formal context $K = (G, M, I)$
OUTPUT The number of pseudo-intents of K

Proof: by reduction from the problem of counting all (inclusion) minimal covers proved to be $\#P$-complete in

L. G. Valiant, The Complexity of Enumeration and Reliability Problems, SIAM J. Comput. 8, 3 (1979), 410–421.

For a graph (V, E) a subset $W \subseteq V$ is a **vertex cover** if every edge $e \in E$ is incident to some $w \in W$.

Preprocessing. Reducing attributes

Attribute $m \in M$, $\mathbb{K} = (G, M, I)$ is **reducible** if

$$m' = G \text{ or } m' = \bigcap \{n' \mid n \in M \ \& \ n' \supset m'\}.$$

If m is reducible, then $\underline{\mathfrak{B}}(G, M, I) \cong \underline{\mathfrak{B}}(G, M \setminus \{m\}, I \cap (G \times (M \setminus \{m\})))$

Dually for objects. An irreducible attribute corresponds to a meet-irreducible lattice element, an irreducible object corresponds to a join-irreducible lattice element.

Example. Attribute m_k is reducible, since $m'_k = m'_i \cap m'_j$

$G \setminus M$	...	m_i	...	m_j	...	m_k	...
g_1	...	×	...		...		...
g_2	...	×	...	×	...	×	...
g_3	...	×	...	×	...	×	...
g_4	...		...	×	...		...

Removing reducible attributes can be performed in time $O(|G| \cdot |M|^2)$.

Close-by-One Algorithm. Notation

- $\min(X)$ ($\max(X)$) return elements of set X with the minimal (maximal) number
- (A, i) denotes $(A \cup \{i\})''$,
- $suc(A)$ is the set of all *heirs* of set A: concepts of the form $(A \cup \{i\})''$ such that $\min((A \cup \{i\})'' \setminus A) = i$. The pairs $(X, suc(X))$ are edges of the tree, vertices of which are concepts (extents).
- $prev(A)$ returns the *parent* (in the tree) of a concept with extent A.
- $nexti(A)$ returns number of the subsequent element i to check whether (A, i) is a heir of A.

Close-by-One Algorithm (CbO)

$A{:=}\emptyset$, $nexti(A){:=}1$, $prev(A){:=}\emptyset$, $suc(A){:=}\emptyset$.

until $A = \emptyset$ and $nexti(A) > |G|$ **do**

begin until $nexti(A) > |G|$ **do**

  begin $i{:} = nexti(A)$

    if $min((A \cup \{i\})'' \setminus A) \geq i$ **then**

    begin $suc(A) \leftarrow address(A, i)$

      $prev(A, i){:=}A$

      $nexti(A){:=}\ nexti(A) + 1$

      $nexti((A, i)){:=}\ min\ (\{j \mid i < j \& j \notin (A, i)\})$

      $A{:=}\ (A, i)$, **output** $(A.A')$

    end

    else $nexti(A){:=}nexti(A) + 1$

  end

  $A{:=}prev(A)$

end

Canonical generations

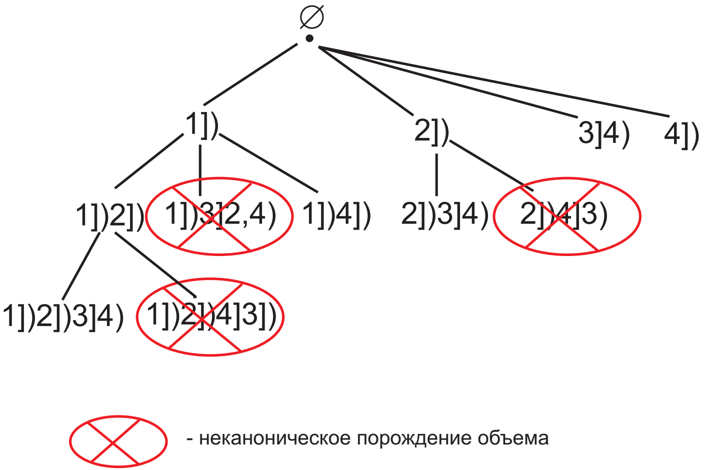
Definition of canonical generation. $\emptyset$ is a canonically generated extent. $(A \cup \{i\})''$ is a canonically generated formal extent if $A \subseteq G$ is a canonically generated extent, $i \in G \setminus A$, and $\min((A \cup \{i\})'' \setminus A) = i$.
In terms of bracket notation: extent generation is canonical if for any subsequence $\ldots x]Y)\ldots$ of characters, $x \in M$, $Y \subseteq M$, the number of any $y \in Y$ is greater than the number of x.

Proposition. An arbitrary extent A has a unique canonical generation.

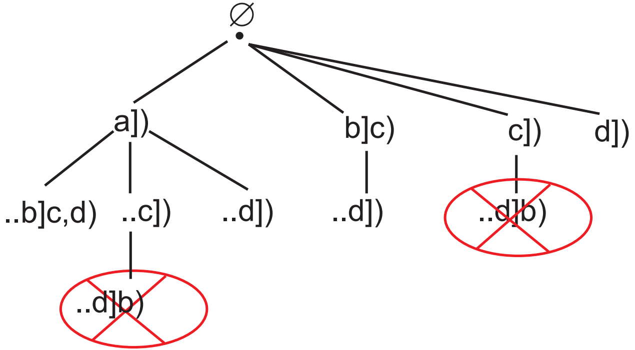
Proof. An algorithm for canonical generation of extent A.

0. $C := \emptyset$, $i := 0$
1. **from** $C = \emptyset$ **until** $C = A$ **do**
2. **begin**
3. $\quad i := min(A \setminus C)$, $C := (C \cup \{i\})''$
4. **end**

CbO in the strategy "bottom-up" (object-wise)



CbO in the strategy "top-down" (attribute-wise)



Modifications of CbO

- FCbO [J.Outrata, V.Vychodil, 2009]
- PCbO [J.Outrata, V.Vychodil, 2010]
- FPCbO