# Subset-generated complete sublattices as concept lattices

Martin Kauer and Michal Krupka

Palacký University
Olomouc
Czech Republic

14. 10. 2015

# Introduction

- Construction of substructures is a fundamental task.
- Computing a complete sublattice generated by given set is computationally hard.
- We are aware of only one algorithm, which we believe is not correct.

# Closed subrelations

**Definition (closed subrelation)**

A relation $J \subseteq I \subseteq X \times Y$ is called a closed subrelation of the context $\langle X, Y, I \rangle$ if every concept of the context $\langle X, Y, J \rangle$ is also concept of $\langle X, Y, I \rangle$.

- There is a 1-1 correspondence between closed subrelations and complete sublattices.

# Substructures

- Let $U$ be a complete lattice.

**Definition (substructures)**

A subset $V \subseteq U$ is a $\bigvee$-subsemilattice (resp. $\bigwedge$-subsemilattice, resp. complete sublattice) of U, if for each $P \subseteq V$ it holds $\bigvee P \in V$ (resp. $\bigwedge P \in V$, resp. $\{\bigvee P, \bigwedge P\} \subseteq V$).

**Definition (generated substructures)**

For a subset $P \subseteq U$ denote by $\mathrm{C}_{\bigvee} P$ (resp. $\mathrm{C}_{\bigwedge} P$, resp. $\mathrm{C}_{\bigvee\bigwedge} P$) the $\bigvee$-subsemilattice (resp. $\bigwedge$-subsemilattice, resp. complete sublattice) of U generated by $P$.
$\mathrm{C}_{\bigvee} P$ always exists and is equal to intersection of all $\bigvee$-subsemilattices of $U$ containing $P$.
Similarly for operators $\mathrm{C}_{\bigwedge}$ and $\mathrm{C}_{\bigvee\bigwedge}$.

- The operators $\mathrm{C}_{\bigvee}$, $\mathrm{C}_{\bigwedge}$, $\mathrm{C}_{\bigvee\bigwedge}$ are closure operators on the set $U$.

# Substructures cont.

- Let $\mathcal{B}(X, Y, I)$ be finite.
- Set $V_1 = \mathrm{C}_{\bigvee} P$, $V_2 = \mathrm{C}_{\bigwedge} V_1$, $V_3 = \mathrm{C}_{\bigvee} V_2$, $\ldots$.
- Once $V_i = V_{i-1}$, we obtained the complete sublattice $V \subseteq \mathcal{B}(X, Y, I)$ generated by $P$.
- $V$ can be computed by alternating applications of operators $C_{\bigvee}$, $C_{\bigwedge}$.
- For each $i > 0$, $V_i$ is a complete lattice but not a complete sublattice of $\mathcal{B}(X, Y, I)$.

# Problem statement

- Let $\langle X, Y, I\rangle$ be a formal context, $\mathcal{B}(X, Y, I)$ its concept lattice.
- Denote $V$ the complete sublattice of $\mathcal{B}(X, Y, I)$ generated by $P \subseteq \mathcal{B}(X, Y, I)$.
- $V$ can be obtained using the subsemilattices $V_i$, $i > 0$.
- We are not computing $V_i$ instead we work with contexts.
- For each $i > 0$ we compute a formal context with the concept lattice isomorphic to $V_i$.
- There exist a closed subrelation $J \subseteq I$ such that

$$\mathcal{B}(X, Y, J) = V.$$

- The interesting part is how to construct $J$.
- $\mathcal{B}(X, Y, J)$ can then be constructed by any known efficient algorithm.

# The first step

- Recall $V_1 = \mathrm{C}_{\bigvee} P$.
- For $V_1$ the corresponding formal context is $\langle P, Y, K_1 \rangle$ where $K_1$ is given by

$$\langle \langle A, B \rangle, y \rangle \in K_1 \quad \text{iff} \quad y \in B.$$

- The rows in $\langle P, Y, K_1 \rangle$ are exactly intents of concepts from $P$.

## Proposition

*The concept lattice $\mathcal{B}(P, Y, K_1)$ and the complete lattice $V_1$ are isomorphic. The isomorphism assigns to each concept $\langle B^{\downarrow_{K_1}}, B \rangle \in \mathcal{B}(P, Y, K_1)$ the concept $\langle B^{\downarrow_I}, B \rangle \in \mathcal{B}(X, Y, I)$.*

## Proof.

Concepts from $V_1$ are exactly those with intents equal to intersections of intents of concepts from $P$. The same holds for concepts from $\mathcal{B}(P, Y, K_1)$. $\square$

# General step

- For all $V_i$ where $i > 1$ the corresponding formal contexts are of the form $\langle X, Y, K_i \rangle$ where $K_i$ is given by

$$\langle x, y \rangle \in K_i \quad \text{iff} \quad \begin{cases} x \in \{y\}^{\downarrow_{K_{i-1}} \uparrow_{K_{i-1}} \downarrow_I} & \text{for even } i, \\ y \in \{x\}^{\uparrow_{K_{i-1}} \downarrow_{K_{i-1}} \uparrow_I} & \text{for odd } i. \end{cases}$$

- Useful to think about concepts as maximal rectangles.
- For even $i > 0$ we obtain $K_i$ by "extending extents of attribute concepts of $K_{i-1}$ in $I$".
- Similarly for odd $i$.

# General step cont.

Following can be easily proven:

- If $i$ is even then for each $y \in Y$, $\{y\}^{\downarrow_{K_i}} = \{y\}^{\downarrow_{K_{i-1}}\uparrow_{K_{i-1}}\downarrow_I}$.
- If $i$ is odd then for each $x \in X$, $\{x\}^{\uparrow_{K_i}} = \{x\}^{\uparrow_{K_{i-1}}\downarrow_{K_{i-1}}\uparrow_I}$.
- For each $i > 1$ it holds $K_i \subseteq I$ and $K_i \subseteq K_{i+1}$.

**Proposition**

*If $i$ is even then each extent of $K_i$ is also an extent of $I$.*
*If $i$ is odd then each intent of $K_i$ is also an intent of $I$.*

**Proof.**

Previous observation implies that each attribute extent of $K_i$ is an extent of $I$. Thus, the proposition follows from the fact that each extent of $K_i$ is an intersection of attribute extents of $K_i$. Similarly for intents. □

# General step conclusion

## Proposition

*For each $i > 0$, the concept lattice $\mathcal{B}(P, Y, K_i)$ (for $i = 1$) resp. $\mathcal{B}(X, Y, K_i)$ (for $i > 1$) and the complete lattice $V_i$ are isomorphic. The isomorphism is given by $\langle B^{\downarrow_{K_i}}, B\rangle \mapsto \langle B^{\downarrow_I}, B\rangle$ if $i$ is odd and by $\langle A, A^{\uparrow_{K_i}}\rangle \mapsto \langle A, A^{\uparrow_I}\rangle$ if $i$ is even.*

- If $X$ and $Y$ are finite then $K_i \subseteq K_{i+1}$ (for $i > 1$) implies there is a number $n > 1$ such that $K_{n+1} = K_n$. Denote this relation by $J$.
- From previous we know there are two isomorphism of $\mathcal{B}(X, Y, J)$ and $V_n = V_{n+1} = V$.
- Those two isomorphism coincide and $\mathcal{B}(X, Y, J) = V$.

**Algorithm 1** Computing the closed subrelation $J$.

**Input:** formal context $\langle X, Y, I\rangle$, subset $P \subseteq \mathcal{B}(X, Y, I)$
**Output:** the closed subrelation of $J \subseteq I$ whose concept lattice is equal to $\mathrm{C}_{\bigvee\bigwedge} P$

$J \leftarrow$ relation $K_1$
$i \leftarrow 1$
**repeat**
  $L \leftarrow J$
  $i \leftarrow i + 1$
  **if** $i$ is even **then**
    $J \leftarrow \{\langle x, y\rangle \in X \times Y \mid x \in \{y\}^{\downarrow_L \uparrow_L \downarrow_I}\}$
  **else**
    $J \leftarrow \{\langle x, y\rangle \in X \times Y \mid y \in \{x\}^{\uparrow_L \downarrow_L \uparrow_I}\}$
  **end if**
**until** $i > 2 \;\&\; J = L$
**return** $J$

## Proposition

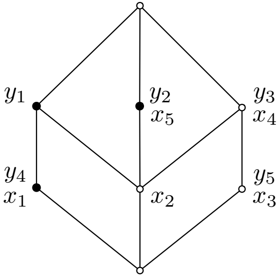
*Algorithm 1 is correct and terminates after at most* $\max(|I| + 1, 2)$ *iterations.*

- The algorithm cannot stop after one iteration, since comparison of $K_1 \subseteq P \times Y$ and $K_2 \subseteq X \times Y$ does not make sense.
- Each iteration adds at least one incidence.
- Once nothing new gets added algorithm terminates.
- We ran number of experiments on *Mushrooms* dataset, the maximum recorded number of iterations was 11.
- There was apparent decreasing trend of number of iteration for increasing size of $P$.

# Example

$\langle X, Y, I \rangle$ (left) and concept lattice $\mathcal{B}(X, Y, I)$, together with a subset $P \subseteq \mathcal{B}(X, Y, I)$, depicted by filled dots (right).

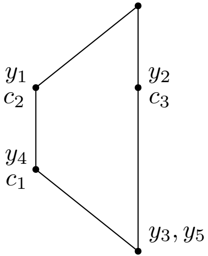
| $I$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | × | | | × | |
| $x_2$ | × | × | × | | |
| $x_3$ | | | × | | × |
| $x_4$ | | | × | | |
| $x_5$ | | × | | | |

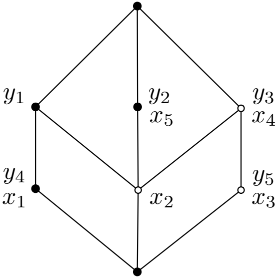


# Example cont.

$\langle P, Y, K_1 \rangle$ (left), the concept lattice $\mathcal{B}(P, Y, K_1)$ (center) and the $\bigvee$-subsemilattice $\mathrm{C}_{\bigvee} P \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(P, Y, K_1)$, depicted by filled dots (right).

| $K_1$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $c_1$ | $\times$ | | | $\times$ | |
| $c_2$ | $\times$ | | | | |
| $c_3$ | | $\boxed{\times}$ | | | |

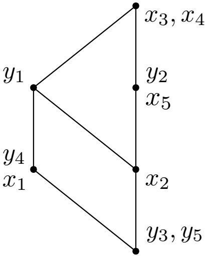


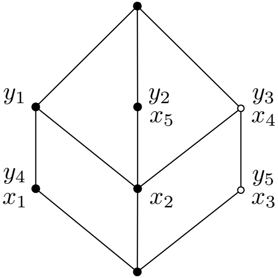



# Example cont.

$\langle X, Y, K_2 \rangle$ (left), the concept lattice $\mathcal{B}(X, Y, K_2)$ (center) and the $\bigwedge$-subsemilattice $V_2 = \mathrm{C}_{\bigwedge} V_1 \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(X, Y, K_2)$, depicted by filled dots (right).

| $K_2$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | × | | | × | |
| $x_2$ | × | × | · | | |
| $x_3$ | | | · | | · |
| $x_4$ | | | · | | |
| $x_5$ | | × | | | |

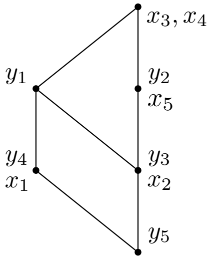


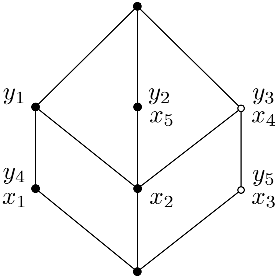



# Example conclusion

$\langle X, Y, K_3 \rangle$ (left), the concept lattice $\mathcal{B}(X, Y, K_3)$ (center) and the $\bigvee$-subsemilattice $V_3 = \mathrm{C}_{\bigvee} V_2 \subseteq \mathcal{B}(X, Y, I)$, isomorphic to $\mathcal{B}(X, Y, K_3)$, depicted by filled dots (right). As $K_3 = K_4 = J$, it is a closed subrelation of $I$ and $V_4 = \mathrm{C}_{\bigwedge} V_3 = V_3$ is a complete sublattice of $\mathcal{B}(X, Y, I)$.

| $K_3$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | $\times$ | | | $\times$ | |
| $x_2$ | $\times$ | $\times$ | $\times$ | | |
| $x_3$ | | | $\cdot$ | | $\cdot$ |
| $x_4$ | | | $\cdot$ | | |
| $x_5$ | | $\times$ | | | |



# Open problems

Closed subrelations

- Set of all closed subrelations of given context is not a closure system.
- There is no closure operator that would assign to a subrelation the smallest closed subrelation.
- But our algorithm can be easily changed to compute a closed subrelation to any subrelation.
- This closed subrelation seems to be minimal in some sense.
- Dual computation might give different results.

Complexity

- We believe presented bound on number of iteration is loose.
- We didn't prove a stronger statement yet.
- But we were also unable to construct any example that would require more than $min(|X|,|Y|)$ iterations.

# Conclusion

- We present a method for computing sublattices generated by sets of elements.
- Our approach is based only on work with contexts.
- Can be very efficient especially if the size of the sublattice is small compared to the whole lattice.
- The actual construction of sublattices can be done using any know algorithm for computing concept lattices.

Thank you for your attention!

# Algorithm from the paper by K. Bertet, M. Morvan (1999)

- Authors cite a book (by Davey and Priestley) with the definition of sublattice as was presented here.
- The output of algorithm from the paper need not to be sublattice in this sense.
- If we would admit a different definition for generated sublattice (such that the algorithm would be correct with respect to this definition), then the claim that computed result would be the smallest one would be false (it would not be a closure system).
- The algorithm is based on the following false claim: Given an element $e \in U$ the smallest element $s \geq e$ such that $s \in \mathrm{C}_{\bigvee\bigwedge} P$ can be expressed as $s = \bigwedge\{p \in P \mid p \geq e\}$.