Technische Universität Dresden

Prof. Dr. Sebastian Rudolph

Formal Concept Analysis

Exercise Sheet 7, Winter Semester 2015/16

Exercise 1 (frequent concept intents and closure systems)

Definition (frequent concept intent)**.** *Let* $\mathbb{K} = (G, M, I)$ *be a formal context.*

(a) The support *of a set* $B \subseteq M$ *of attributes in* $\mathbb{K}$ *is given by*

$$\operatorname{supp}(B) := \frac{|B'|}{|G|}.$$

(b) For a given minimal support minsupp the set of frequent concept intents *is given by*

$$\{B \subseteq M \mid \exists A \subseteq G : (A, B) \in \mathfrak{B}(G, M, I) \wedge \operatorname{supp}(B) \geq \mathit{minsupp}\}.$$

Show that the set of frequent concept intents together with the set M forms a closure system.

Solution:

Proof: We have to show that the intersection of frequent concept intents is again a frequent concept intent. We already know that the intersection of intents produces an intent. It remains to show that it is frequent. So let $\mathfrak{I}$ be a set of frequent intents. We pick one $B \in \mathfrak{I}$ and observe $\operatorname{supp}(B) = \frac{|B'|}{|G|} \geq \mathit{minsupp}$. Moreover, we have $\bigcap \mathfrak{I} \subseteq B$ and consequently $B' \subseteq (\bigcap \mathfrak{I})'$. Therefore $\operatorname{supp}(\bigcap \mathfrak{I}) = \frac{|(\bigcap \mathfrak{I})'|}{|G|} \geq \frac{|B'|}{|G|} \geq \mathit{minsupp}$, i.e., $\bigcap \mathfrak{I}$ is frequent.

Exercise 2 (support)

Show the validity of the properties of the support function that are employed by the Titanic algorithm:

Let (G, M, I) be a formal context $X, Y \subseteq M$. Then it holds:

1) $X \subseteq Y \implies \operatorname{supp}(X) \geq \operatorname{supp}(Y)$

2) $X'' = Y'' \implies \operatorname{supp}(X) = \operatorname{supp}(Y)$

3) $X \subseteq Y \wedge \operatorname{supp}(X) = \operatorname{supp}(Y) \implies X'' = Y''$

Solution:

1. Let $X \subseteq Y$, then $Y' \subseteq X'$ holds as we saw in Exercise Sheet 1. This implies, $\operatorname{supp}(Y) = \frac{|Y'|}{|G|} \leq \frac{|X'|}{|G|} = \operatorname{supp}(X)$
2. $X'' = Y'' \implies \operatorname{supp}(X) = \operatorname{supp}(Y)$
   $X'' = Y'' \iff X''' = Y''' \iff X' = Y' \implies \operatorname{supp}(X) = \frac{|X'|}{|G|} = \frac{|Y'|}{|G|} = \operatorname{supp}(|Y|)$.
3. $X \subseteq Y \wedge \operatorname{supp}(X) = \operatorname{supp}(Y) \implies X'' = Y''$
   $\operatorname{supp}(X) = \operatorname{supp}(Y) \implies |X'| = |Y'|$ and $X \subseteq Y \implies X' \supseteq Y'$. Hence $X' = Y'$, since X' and Y' are finite. It follows, $X'' = Y''$.

Exercise 3 (computing concept intents with TITANIC)

The following context contains transactions in a supermarket. Compute the closure system of all concept intents using the TITANIC algorithm. (hint: use the table structure from the example computation in the lecture slides)

	apples (a)	beer (b)	chips (c)	tv magazine (d)	toothpaste (e)
t_1	×	×	×		
t_2			×	×	
t_3		×	×	×	
t_4	×	×			×
t_5			×		×
t_6		×	×	×	
t_7	×	×			
t_8			×	×	

Solution:

In the first pass, the algorithm deals with the empty set and singletons, all 1-sets. It returns the results for k =0 and k=1:

k = 0:

step 1		step2
x	x.s	$x \in K_k$?
∅	1	yes

k = 1:

steps 4+5		step 7	step 9
X	$X.p\_s$	$X.s$	$X \in K_k$?
{a}	1	3/8	yes
{b}	1	5/8	yes
{c}	1	6/8	yes
{d}	1	4/8	yes
{e}	1	2/8	yes

Step 8 returns: ∅.closure ← ∅

k = 2:

steps 12		step 7	step 9
X	$X.p\_s$	$X.s$	$X \in K_k$?
{a,b}	3/8	3/8	no
{a,c}	3/8	1/8	yes
{a,d}	3/8	0	yes
{a,e}	2/8	1/8	yes
{b,c}	5/8	3/8	yes
{b,d}	4/8	2/8	yes
{b,e}	2/8	1/8	yes
{c,d}	4/8	4/8	no
{c,e}	2/8	1/8	yes
{d,e}	2/8	0	yes

Step 8 returns:
$\{a\}$.closure ← $\{a, b\}$
$\{b\}$.closure ← $\{b\}$
$\{c\}$.closure ← $\{c\}$
$\{d\}$.closure ← $\{c, d\}$
$\{e\}$.closure ← $\{e\}$

k = 3:

steps 12		step 7	step 9
X	$X.p\_s$	$X.s$	$X \in K_k$?
{a,c,e}	1/8	0	yes
{a,d,e}	0	0	no
{b,c,e}	1/8	0	yes
{b,d,e}	1/8	0	yes

Step 8 returns:
$\{a, c\}$.closure $\leftarrow \{a, b, c\}$
$\{a, d\}$.closure $\leftarrow \{a, b, c, d, e\}$
$\{a, e\}$.closure $\leftarrow \{a, b, e\}$
$\{b, c\}$.closure $\leftarrow \{b, c\}$
$\{b, d\}$.closure $\leftarrow \{b, c, d\}$
$\{b, e\}$.closure $\leftarrow \{a, b, e\}$
$\{c, e\}$.closure $\leftarrow \{c, e\}$
$\{d, e\}$.closure $\leftarrow \{a, b, c, d, e\}$

k = 4:

Step 12: returns the empty set. Hence there is nothing to **WEIGH** in Step 7. Step 9 sets $K_4 = \emptyset$; and in step 10, the loop is exited.
Step 8 returns:
$\{a, c, e\}$.closure $\leftarrow \{a, b, c, d, e\}$
$\{b, c, e\}$.closure $\leftarrow \{a, b, c, e, e\}$
$\{b, d, e\}$.closure $\leftarrow \{a, b, c, d, e\}$

Step 14: Collects all concept intents:
$\emptyset$, {a,b}, {b}, {c}, {c,d}, {e}, {a,b,c}, {b,c}, {b,c,d}, {a,b,e}, {c,e}, {a,b,c,d,e}

Algorithm 1 TITANIC

SUPPORT($\{\emptyset\}$);
$\mathcal{K}_0 \leftarrow \{\emptyset\}$;
$k \leftarrow 1$;
forall $m \in M$ **do** $\{m\}.p\_s \leftarrow \emptyset.s$;
$\mathcal{C} \leftarrow \{\{m\} \mid m \in M\}$;
loop begin
  SUPPORT($\mathcal{C}$);
  forall $X \in \mathcal{K}_{k-1}$ **do** X.closure $\leftarrow$ CLOSURE(X);
  $\mathcal{K}_k \leftarrow \{X \in \mathcal{C} \mid X.s \neq X.p\_s\}$;
  if $\mathcal{K}_k = \emptyset$ **then exit loop** ;
  $k++$;
  $\mathcal{C} \leftarrow$ TITANIC-GEN($\mathcal{K}_{k-1}$);
end loop ;
return $\bigcup_{i=0}^{k-1}\{X.\text{closure} \mid X \in \mathcal{K}_i\}$.

Algorithm 2 TITANIC-GEN

Input: $\mathcal{K}_{k-1}$, the set of key $(k-1)$-sets K with their weight $K.s$.

Output: $\mathcal{C}$, the set of candidate k-sets C
with the values $C.p\_s := \min\{s(C \setminus \{m\} \mid m \in C\}$.

The variables p_s assigned to the sets $\{m_1, \ldots, m_k\}$ which are generated in step 1 are initialized by $\{m_1, \ldots, m_k\}.p\_s \leftarrow s_{\max}$.

1) $\mathcal{C} \leftarrow \{\{m_1 < m_2 < \cdots < m_k\} \mid \{m_1, \ldots, m_{k-2}, m_{k-1}\}, \{m_1, \ldots, m_{k-2}, m_k\} \in \mathcal{K}_{k-1}\}$;
2) **forall** $X \in \mathcal{C}$ **do begin**
3)   **forall** $(k-1)$-subsets S of X **do begin**
4)   **if** $S \notin \mathcal{K}_{k-1}$ **then begin** $\mathcal{C} \leftarrow \mathcal{C} \setminus \{X\}$; **exit forall** ; **end**;
5) $X.p\_s \leftarrow \min(X.p\_s, S.s)$;
6)   **end**;
7) **end**;
8) **return** $\mathcal{C}$.

Algorithm 3 CLOSURE(X) for $X \in \mathcal{K}_{k-1}$

$Y \leftarrow X$;
forall $m \in X$ **do** $Y \leftarrow Y \cup (X \setminus \{m\})$.closure; **forall** $m \in M \setminus Y$ **do begin**
  if $X \cup \{m\} \in \mathcal{C}$ **then** $s \leftarrow (X \cup \{m\}).s$
      else $s \leftarrow \min\{K.s \mid K \in \mathcal{K},\ K \subseteq X \cup \{m\}\}$;
  if $s = X.s$ **then** $Y \leftarrow Y \cup \{m\}$
end;
return Y.