WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Concise Representations of Frequent Itemsets

HUMAN CAPITAL

Project is co-financed by European Union within European Social Fund

Do We Need to Know All Frequent Itemsets?

- The number of frequent itemsets is usually huge.
- Time of their discovery can be significant.
- There are cases in which one needs to know only a small subset of frequent itemsets! (*Representative* and minimal *non-redundant rules* can be derived directly from concise representations of frequent itemsets called *closed itemsets* and *generators.*)

2

Lossless Representations of Frequent Itemsets

- Itemsets representation is meant *lossless* if it allows derivation and support determination of all frequent itemsets without accessing the database.

3

Lossless Representations of Frequent Itemsets

Lossless representations of frequent itemsets are based on the following **sets subsuming other sets**:

- *closed itemsets*
- (*key*) *generators*
- (*generalized*) *disjunctive sets*

4

Simple example of reasoning about suports of itemsets

- Let *sup*({*ac*}) = 3 and *sup*({*abcde*}) = 3.
- This information is sufficient to determine the support of {*abce*} as folows:

  3 = *sup*({*ac*}) ≥ *sup*({*abce*}) ≥ *sup*({*abcde*}) = 3.

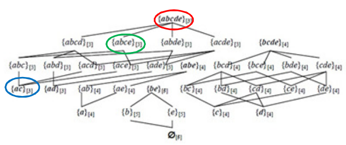
  Hence,

  sup({*ac*}) = *sup*({*abce*}) = *sup*({*abcde*}) = 3.
- **Property.** if $X \subseteq Y$ and $sup(X) = sup(Y) = k$, then for each itemset Z such that: $X \subseteq Z \subseteq Y$, its supports also equals k.

5

Reasoning about suports of itemsets…

Id	*Transaction*
T_1	{*abcde*}
T_2	{*abcdef*}
T_3	{*abcdehi*}
T_4	{*abe*}
T_5	{*bcdehi*}

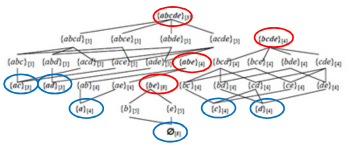


Example. *minSup*=2; *minConf*=77%.

6

Reasoning about suports of itemsets

Id	*Transaction*
T_1	{*abcde*}
T_2	{*abcdef*}
T_3	{*abcdehi*}
T_4	{*abe*}
T_5	{*bcdehi*}



Example. *minSup*=2; *minConf*=77%.

7

Supports and tid-lists of subsets/supersets

Lemma. Let $X \subseteq Y$. Then:

$$t(X) = t(Y) \Leftrightarrow sup(X) = sup(Y).$$

Id	*Transaction*
T_1	{*abcde*}
T_2	{*abcdef*}
T_3	{*abcdehi*}
T_4	{*abe*}
T_5	{*bcdehi*}

Proof

(⇒). Trivial.

(⇐). Let $X \subseteq Y$ and $sup(X) = sup(Y)$.

Then, $t(X) \supseteq t(Y)$ and $|t(X)| = |t(Y)|$.

Hence, $t(X) = t(Y)$.

8

Representation of Frequent Itemsets based on Closed Itemsets

9

Closures of Itemsets

- Let *I* be a set of all items that may occur in D.
- *A closure of itemset X* is denoted as $\gamma(X)$ and defined as:

$$\gamma(X) = \cap\{T \in D \cup \{I\} \mid T \supseteq X\}.$$

- **Note:** An itemset has exactly one itemset as its closure!
- **Property.** The closure of itemset X is the greatest superset $Y \supseteq X$ such that

$$sup(Y) = sup(X).$$

10

Supports of Proper Supersets of Closures

- **Property.** The closure of itemset X is the greatest superset $Y \supseteq X$ such that

$$sup(Y) = sup(X).$$

⇓

- **Note:** Supports of all proper supersets of an itemset being a closure of any itemset is different from (less than) than the support of this closure.

11

Example: Closures of Itemsets

Id	*Transaction*
T1	{*abcde*}
T2	{*abcdef*}
T3	{*abcdehi*}
T4	{*abe*}
T5	{*bcdehi*}

- $\gamma(\{abc\}_3) = \{abcde\}_3$
- $\gamma(\{abcde\}_3) = \{abcde\}_3$

Question: Can $sup(\{abcdehi\})$ be equal to 3?

Answer: No.

$\{abcde\}_3$ is the closure of $\{abc\}_3$. Thus, $sup(\{abcdehi\})$ must be less than $sup(\{abcde\})$. Indeed, $sup(\{abcdehi\}) = 1 < sup(\{abcde\}) = 3$.

12

Closed itemsets

- An itemset is defined as *closed* if it is equal to its closure.
- **Property.** Each closure is a closed itemset.
- **Property.** Supports of all proper supersets of a closed itemset are different from (less than) than the support of this closed itemset.

13

Example: Closed Itemsets

Id	Transaction
T1	{abcde}
T2	{abcdef}
T3	{abcdehi}
T4	{abe}
T5	{bcdehi}

- $\gamma(\{abc\}) = \{abcde\}$.
- Hence, $\{abc\}$ is not a closed itemset.
- $\gamma(\{abcde\}) = \{abcde\}$.
- Hence, $\{abcde\}$ is a closed itemset.

14

Important Property of Closed Itemsets

- **Important property of closed itemsets:** The set of all closed itemsets is sufficient to determine support of each itemset X in 2^I, namely:

  $sup(X) = \max\{sup(Y) \mid Y \text{ is closed} \wedge Y \supseteq X\}$.

15

Closed Itemsets Representation

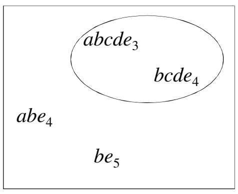
- *Closed itemsets representation* (CR) consists of all frequent closed itemsets and the information about their supports.

16

Example: Reasoning with CR

- $sup(\{bef\})$=?, $sup(\{bcd\})$=?

CR



- $\{bef\}$ has no superset in CR, so: $sup(\{bef\}) \leq minSup$.
- $sup(\{bcd\}) = \max(sup(\{bcde\}_4), sup(\{abcde\}_3)) = 4$.

17

Calculating *FC*s with CHARM

Id	Transaction
T1	{abcde}
T2	{abcdef}
T3	{abcdehi}
T4	{abe}
T5	{bcdehi}

$h(X \cup Y) = \Sigma_{i \in t(X \cup Y)}\, i$

- $t(X) = t(Y)$:
  - remove node Y from the tree;
  - replace each X with $X \cup Y$.
- $t(X) \subset t(Y)$:
  - replace each X with $X \cup Y$.
- $t(X) \supset t(Y)$:
  - remove node Y from the tree;
  - add $X \cup Y$ as a child of node X.
- otherwise:
  - add $X \cup Y$ as a child of node X.

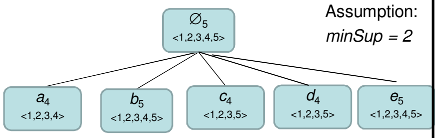
18

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Calculating *FC*s with CHARM…

Id Transaction
1 {abcde}
2 {abcdef}
3 {abcdehi}
4 {abe}
5 {bcdehi}

Assumption: *minSup* = 2



$h(X \cup Y) = \Sigma_{i \in t(X \cup Y)}\ i$

1	…	6	…	10	11	…	15

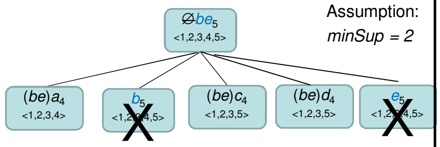
19

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Calculating *FC*s with CHARM…

Id Transaction
1 {abcde}
2 {abcdef}
3 {abcdehi}
4 {abe}
5 {bcdehi}

Assumption: *minSup* = 2



$h(X \cup Y) = \Sigma_{i \in t(X \cup Y)}\ i$

$h(be)$ = 1+2+3+4+5 = 15

1	…	6	…	10	11	…	15
							be_5

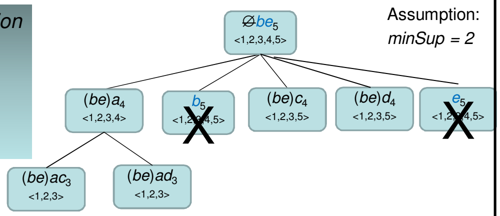
20

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Calculating *FC*s with CHARM…

Id Transaction
1 {abcde}
2 {abcdef}
3 {abcdehi}
4 {abe}
5 {bcdehi}

Assumption: *minSup* = 2



$h(X \cup Y) = \Sigma_{i \in t(X \cup Y)}\ i$

$h(be)$ = 1+2+3+4+5 = 15, $h(abe)$ = 1+2+3+4 = 10,

1	…	6	…	10	11	…	15
				abe_4			be_5

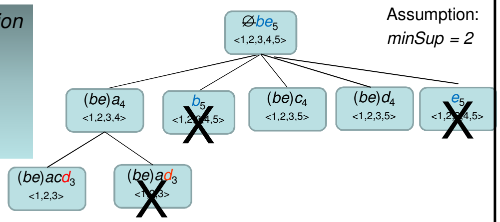
21

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Calculating *FC*s with CHARM…

Id Transaction
1 {abcde}
2 {abcdef}
3 {abcdehi}
4 {abe}
5 {bcdehi}

Assumption: *minSup* = 2



$h(X \cup Y) = \Sigma_{i \in t(X \cup Y)}\ i$

$h(be)$ = 1+2+3+4+5 = 15, $h(abe)$ = 1+2+3+4 = 10,
$h(abcde)$ = 1+2+3 = 6

1	…	6	…	10	11	…	15
		$abcde_3$		abe_4			be_5

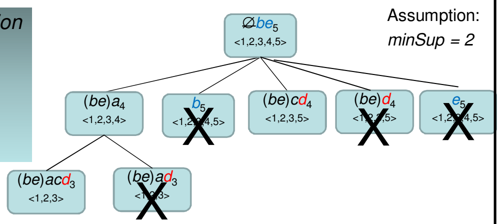
22

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Example: Calculating *FC*s with CHARM

Id Transaction
1 {abcde}
2 {abcdef}
3 {abcdehi}
4 {abe}
5 {bcdehi}

Assumption: *minSup* = 2



$h(X \cup Y) = \Sigma_{i \in t(X \cup Y)}\ i$

$h(be)$ = 1+2+3+4+5 = 15, $h(abe)$ = 1+2+3+4 = 10,
$h(abcde)$ = 1+2+3 = 6, $h(bcde)$ = 1+2+3+5 = 11

1	…	6	…	10	11	…	15
		$abcde_3$		abe_4	$bcde_4$		be_5

23

WARSAW UNIVERSITY OF TECHNOLOGY DEVELOPMENT PROGRAMME

Calculating *FC*s with dCHARM

Id Transaction
T1 {abcde}
T2 {abcdef}
T3 {abcdehi}
T4 {abe}
T5 {bcdehi}

- $d(X \cup Y) = d(Y) \setminus d(X)$
- $sup(X \cup Y) = sup(X) - |d(X \cup Y)|$
- $h(X \cup Y) = h(X) - (\Sigma_{i \in d(X \cup Y)}\ i)$

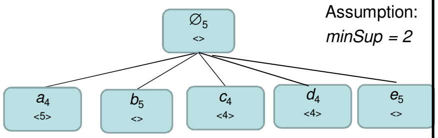
- $d(X) = d(Y)$:
  - remove node Y from the tree;
  - replace each X with $X \cup Y$.
- $d(X) \supset d(Y)$:
  - replace each X with $X \cup Y$.
- $d(X) \subset d(Y)$:
  - remove node Y from the tree;
  - add $X \cup Y$ as a child of node X.
- otherwise:
  - add $X \cup Y$ as a child of node X.

24

Example: Calculating *FC*s with dCHARM…

Id	*Transaction*
1	{*abcde*}
2	{*abcdef*}
3	{*abcdehi*}
4	{*abe*}
5	{*bcdehi*}

Assumption: *minSup* = 2



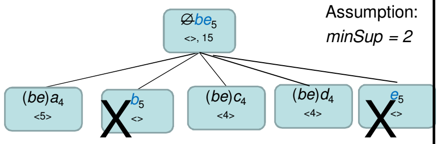
- $h(X \cup Y) = h(X) - (\Sigma_{i \in d(X \cup Y)}\ i)$

1	…	6	…	10	11	…	15

Example: Calculating *FC*s with dCHARM…

Id	*Transaction*
1	{*abcde*}
2	{*abcdef*}
3	{*abcdehi*}
4	{*abe*}
5	{*bcdehi*}

Assumption: *minSup* = 2



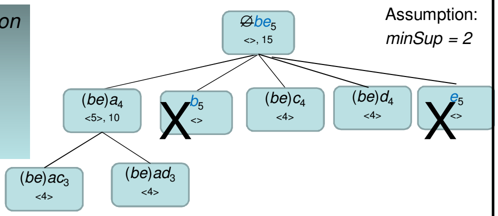
- $h(X \cup Y) = h(X) - (\Sigma_{i \in d(X \cup Y)}\ i)$
- $h(be) = 1+2+3+4+5 = 15$

1	…	6	…	10	11	…	15
							be_5

Example: Calculating *FC*s with dCHARM…

Id	*Transaction*
1	{*abcde*}
2	{*abcdef*}
3	{*abcdehi*}
4	{*abe*}
5	{*bcdehi*}

Assumption: *minSup* = 2



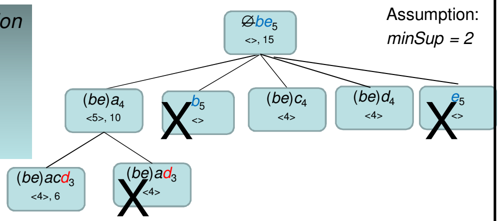
- $h(X \cup Y) = h(X) - (\Sigma_{i \in d(X \cup Y)}\ i)$
- $h(be) = 1+2+3+4+5 = 15$
- $h(abe) = h(be) - 5 = 15 - 5 = 10$

1	…	6	…	10	11	…	15
				abe_4			be_5

Example: Calculating *FC*s with dCHARM…

Id	*Transaction*
1	{*abcde*}
2	{*abcdef*}
3	{*abcdehi*}
4	{*abe*}
5	{*bcdehi*}

Assumption: *minSup* = 2



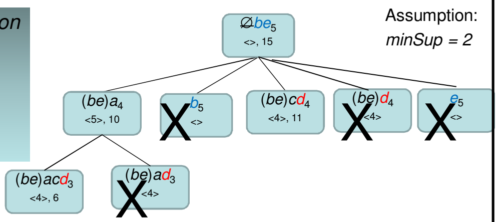
- $h(X \cup Y) = h(X) - (\Sigma_{i \in d(X \cup Y)}\ i)$
- $h(be) = 1+2+3+4+5 = 15$
- $h(abe) = h(be) - 5 = 15 - 5 = 10$
- $h(abcde) = h(be) - 4 = 6$

1	…	6	…	10	11	…	15
		$abcde_3$		abe_4			be_5

Example: Calculating *FC*s with dCHARM

Id	*Transaction*
1	{*abcde*}
2	{*abcdef*}
3	{*abcdehi*}
4	{*abe*}
5	{*bcdehi*}

Assumption: *minSup* = 2



- $h(X \cup Y) = h(X) - (\Sigma_{i \in d(X \cup Y)}\ i)$
- $h(be) = 1+2+3+4+5 = 15$
- $h(abe) = h(be) - 5 = 15 - 5 = 10$
- $h(abcde) = h(be) - 4 = 6$
- $h(bcde) = h(be) - 4 = 15 - 4 = 11$

1	…	6	…	10	11	…	15
		$abcde_3$		abe_4	$bcde_4$		be_5

Algorithm for Transforming *FC*s into *F*s

Id	*Transaction*
1	{*abcde*}
2	{*abcdef*}
3	{*abcdehi*}
4	{*abe*}
5	{*bcdehi*}

- *FC*: $\{be\}_5$, $\{abe\}_4$, $\{abcde\}_3$, $\{bcde\}_4$
- *F*: ?

- F_5: $\{abcde\}_3$
- F_4: $\{bcde\}_4$, $\{abcd\}_3$, $\{abce\}_3$, $\{abde\}_3$, $\{acde\}_3$
- F_3: $\{abe\}_4$, $\{bcd\}_4$, $\{bce\}_4$, $\{bde\}_4$, $\{cde\}_4$, $\{abc\}_3$, $\{abd\}_3$, $\{acd\}_3$, $\{ace\}_3$, $\{ade\}_3$
- F_2: $\{be\}_5$, $\{ab\}_4$, $\{ae\}_4$, $\{bc\}_4$, $\{bd\}_4$, $\{cd\}_4$, $\{ce\}_4$, $\{de\}_4$, $\{ac\}_3$, $\{ad\}_3$
- F_1: $\{b\}_5$, $\{e\}_5$, $\{a\}_4$, $\{c\}_4$, $\{d\}_4$
- F_0: $\emptyset_5$

Representation of Frequent Itemsets based on (Key) Generators

Generator of an Itemset

- Y is defined *a generator of itemset X* if it is a minimal subset of X such that

$$sup(Y) = sup(X)$$

(equivalently, $t(Y) = t(X)$).

- **Note 1:** Supports of all proper subsets of a generator of any itemset are different from (greater than) than the support of this generator.
- **Note 2:** An itemset may have more than one generator!

Example: Generators of Itemsets

Id	*Transaction*
T1	{*abcde*}
T2	{*abcdef*}
T3	{*abcdehi*}
T4	{*abe*}
T5	{*bcdehi*}

- $G(\{abcd\}_3) = \{\{ac\}_3, \{ad\}_3\}$
- $G(\{ac\}_3) = \{\{ac\}_3\}$
- $G(\{ad\}_3) = \{\{ad\}_3\}$
- $\{ac\}_3$ is a generator of (among other) $\{abcd\}_3$ and $\{ac\}_3$ itself. Thus, $sup(\{a\})$ must be greater than $sup(\{ac\})$. Indeed, $sup(\{a\}) = 4 > sup(\{ac\}) = 3$.

(Key) Generator

- An itemset X is defined as *a* (*key*) *generator* if X is a generator of X.
- (*Key*) *generator* has only one *generator* – itself.
- **Property.** Supports of all proper subsets of a (key) generator are different from (greater than) than the support of this (key) generator.

Example: (Key) Generators

Id	*Transaction*
T1	{*abcde*}
T2	{*abcdef*}
T3	{*abcdehi*}
T4	{*abe*}
T5	{*bcdehi*}

- $G(\{abcd\}_3) = \{\{ac\}_3, \{ad\}_3\}$.
- Hence, {*abcd*} is not a (key) generator

- $G(\{ac\}) = \{\{ac\}\}$.
- Hence, {*ac*} is a (key) generator.

- $G(\{ad\}) = \{\{ad\}\}$.
- Hence, {*ad*} is a (key) generator.

Supersets and Subsets of (Key) Generators

- **Theorem.**
  - All supersets of a non-(key) generator are not (key) generators.
  - All subsets of a (key) generator are (key) generators.

Example: Supersets of non-(Key) Generators

Id	Transaction
T1	{*abcde*}
T2	{*abcdef*}
T3	{*abcdehi*}
T4	{*abe*}
T5	{*bcdehi*}

- $G(\{abcd\}_3) = \{\{ac\}_3, \{ad\}_3\}$.
- Thus, {*abcd*} is not a (key) generator.
- Question: Can {*abcde*} be a (key) generator?
- Answer:

{*ac*}, which is a generator of {*abcd*}, has the same support (3) as {*abcd*}, so $t(\{ac\}) = t(\{abcd\})$.

- Now, $t(\{abcde\}) = t(\{abcd\}) \cap t(\{e\}) = t(\{ac\}) \cap t(\{e\}) = t(\{ace\})$.
- So, $sup(\{abcde\}) = sup(\{ace\})$ and $\{abcde\} \supset \{ace\}$.
- Therefore, {*abcde*} is not a key generator.

37

Supersets of Non-(Key) Generators

Theorem.

If $X \notin G$, then $\forall Y \supset X,\ Y \notin G$.

Proof. Let $X \notin G,\ Y \supset X$. Then:

$\exists X' \in G(X)$ such that $X' \subset X$ and

$\exists Z \neq \emptyset$ such that $Z = Y \setminus X$

$\Rightarrow t(X') = t(X)$ and

$t(Y) = t(X \cup Z) = t(X) \cap t(Z) = t(X') \cap t(Z) = t(X' \cup Z)$

$\Rightarrow sup(Y) = sup(X \cup Z) = sup(X' \cup Z)$ and

$Y = X \cup Z \supset X' \cup Z$

$\Rightarrow Y \notin G$.

38

Subsets of (Key) Generators

Theorem A. If $X \notin G$, then $\forall Y \supset X,\ Y \notin G$.

Theorem B.

If $X \in G$, then $\forall Y \subset X,\ Y \in G$.

Proof (by contradiction).

Let $X \in G,\ Y \subset X$ and $Y \notin G$. Then:

By Theorem A all proper supersets of Y (and thus also X) are not (key) generators.

Hence, $X \notin G$, which contradicts the assumption.

39

Properties of (Key) Generators

- **Important Property of Generators:**

The set of all (key) generators is sufficient to determine the support of each itemset X in 2^I, namely:

$$sup(X) = \min\{sup(Y) \mid Y \text{ is a (key) generator } \wedge Y \subseteq X\}.$$

40

Generators Representation (GR)

- *The generators representation* (GR) consists of:
  1) all frequent (key) generators and the information about their supports,
  2) the border of minimal infrequent (key) generators.

41

GR Border…

Theorem. X is a minimal infrequent (key) generator $\Leftrightarrow$ X is a minimal infrequent itemset.

Proof ($\Rightarrow$). X is a minimal infrequent (key) generator

$\Rightarrow$ X is a minimal infrequent (key) generator & all proper subsets of X are (key) generators

$\Rightarrow$ X is a minimal infrequent (key) generator & all proper subsets of X are frequent (key) generators

$\Rightarrow$ X is infrequent & all proper subsets of X are frequent

$\Rightarrow$ X is a minimal infrequent itemset.

42

GR Border

Theorem. X is a minimal infrequent (key) generator ⇔ X is a minimal infrequent itemset.

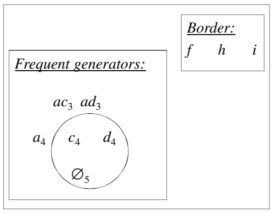
Proof (⇐). *X* is a minimal infrequent itemset
⇒ X is infrequent & proper subsets of *X* are frequent
⇒ X is infrequent & all proper subsets of *X* are frequent & have supports different from *sup*(*X*)
⇒ *X* is an infrequent (key) generator & all its proper subsets are frequent
⇒ X is an infrequent (key) generator & all proper subsets of X are frequent (key) generators
⇒ X is a minimal infrequent (key) generator.

43

Example: Reasoning with GR

- *sup*({*bef*})=?, *sup*({*bcd*})=?

GR



- {*bef*} is infrequent, as it is a superset of {*f*}, which is an infrequent (key) generator.
- *sup*({*bcd*}) = min(*sup*({*c*}$_4$), *sup*({*d*}$_4$), *sup*($\varnothing_5$)) = 4.

44

Calculating *GR* with GR-Apriori

Id	*Transaction*
*T*1	{*abcde*}
*T*2	{*abcdef*}
*T*3	{*abcdehi*}
*T*4	{*abe*}
*T*5	{*bcdehi*}

- The GR-Apriori algorithm discovers GR-representation.
- GR-Apriori uses the fact that GR contains only frequent (key) generators and minimal infrequent (key) generators (that is, minimal infrequent itemsets).
- In consequence, GR-Apriori uses the following properties to generate GR candidates:
  – No proper superset of a non-(key) generator belongs to GR;
  – No proper superset of an infrequent itemset belongs to GR.

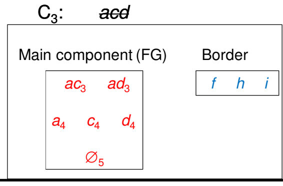
45

Calculating *GR* with GR-Apriori

Id	*Transaction*
*T*1	{*abcde*}
*T*2	{*abcdef*}
*T*3	{*abcdehi*}
*T*4	{*abe*}
*T*5	{*bcdehi*}

Assumption: *minSup* = 2

- Iteration 1:
  $C_0 \rightarrow FG_0$: $\varnothing_5$
  $C_1 \rightarrow FG_1$: a_4 b_5 c_4 d_4 e_5 f_1 h_2 i_2
- Iteration 2:
  $C_2 \rightarrow FG_2$: ac_3 ad_3 cd_4
- Iteration 3:
  C_3: ~~*acd*~~



46

References…

– Marzena Kryszkiewicz: Concise Representation of Frequent Patterns Based on Disjunction-Free Generators. ICDM 2001: 305-312
– Marzena Kryszkiewicz, Marcin Gajek: Concise Representation of Frequent Patterns Based on Generalized Disjunction-Free Generators. PAKDD 2002: 159-171
– Marzena Kryszkiewicz: Concise Representations of Frequent Patterns and Association Rules, Warsaw: Publishing House of Warsaw University of Technology (2002)

47

References

– Pasquier N.: Data mining: Algorithmes d'extraction et de Réduction des Règles d'association dans les Bases de Données. Thèse de Doctorat, Université Blaise Pascal – Clermont–Ferrand II (2000)
– Mohammed Javeed Zaki, Ching-Jui Hsiao: Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure. IEEE Trans. Knowl. Data Eng. 17(4): 462-478 (2005)

48