# Concise Representation of Frequent Patterns based on Disjunction-free Generators

Marzena Kryszkiewicz
Institute of Computer Science
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
mkr@ii.pw.edu.pl

## Abstract

*Many data mining problems require the discovery of frequent patterns in order to be solved. Frequent itemsets are useful in the discovery of association rules, episode rules, sequential patterns and clusters. The number of frequent itemsets is usually huge. Therefore, it is important to work out concise representations of frequent itemsets. In the paper, we describe three basic lossless representations of frequent patterns in an uniform way and offer a new lossless representation of frequent patterns based on disjunction-free generators. The new representation is more concise than two of the basic representations and more efficiently computable than the third representation. We propose an algorithm for determining the new representation.*

## 1. Introduction

Many data mining problems require the discovery of frequent patterns in order to be solved. Frequent itemsets are useful in the discovery of association rules, episode rules, sequential patterns and clusters etc. (see [6] for overview). Nevertheless, the number of frequent itemsets is usually huge. Therefore, it is important to work out concise, preferably lossless, representations of frequent itemsets. Recently, there have been investigated the following interesting subsets of frequent itemsets: closed itemsets (see e.g. [2,8-9]), generators (see e.g. [2,8]), and the representation based on disjunction-free sets [5]. Both frequent closed itemsets and the disjunction-free sets representation are lossless representations in the sense that they allow derivation and support determination of all frequent itemsets without accessing the database. The frequent generators themselves do not possess this property unless augmented by the set of minimal infrequent generators. Applications of frequent closed itemsets and frequent generators have been demonstrated in the case of the discovery of association rules and their essential subsets (see e.g. [7-8,10-11]). In particular, in the case of representative association rules [7] and informative basis [8], the antecedent of any such rule is a generator, while the consequent is a closed itemset decreased by the items present in the rule's antecedent.

In this paper, we introduce yet another lossless representation of frequent itemsets that benefits both from the properties of generators and disjunction-free sets. We prove that the new representation constitutes a subset of the generators representation and the disjunction-free sets representation. On the other hand, the frequent closed itemsets representation may happen to be either more concise or less concise depending on particular data. Conciseness of the frequent closed itemsets representation has been proved experimentally (see e.g. [11]). The algorithms for computing this representation require the discovery of frequent generators first (see e.g. [2,8-9]). In [2,8], generators are treated as seeds of closed itemsets that are determined by intersecting database transactions. This makes the discovery of frequent closed itemsets inefficient. To the contrary, the new representation does not require such a computational overhead. In the paper, we propose an algorithm for determining the new representation.

The layout of the paper is as follows:

Section 2 introduces the notions and properties of frequent itemsets, closed itemsets and generators, as well as the representations based on closed itemsets, on generators, and on disjunction-free sets. In Section 3 we introduce the new representation based on disjunction-free generators and prove that all frequent itemsets and their supports can be derived from it. In Section 4 we prove that the new representation is not less concise than the ones based on generators and on disjunction-free sets. An algorithm for determining the new representation is presented in Section 5. Section 6 concludes the results.

0-7695-1119-8/01 $17.00 © 2001 IEEE

## 2. Basic notions and properties

### 2.1. Itemsets, frequent itemsets

Let $I = \{i_1, i_2, ..., i_m\}$, $I \neq \emptyset$, be a set of distinct literals, called *items*. In the case of a transactional database, a notion of an item corresponds to a sold product, while in the case of a relational database an item will be an (*attribute,value*) pair. Any non-empty set of items is called an *itemset*. An itemset consisting of $k$ items will be called *k-itemset*. Let D be a set of transactions (or tuples, respectively), where each transaction (tuple) $T$ is a subset of $I$. (Without any loss of generality, we will restrict further considerations to transactional databases.) By $I_D$ we will denote a subset of items in $I$ that occurred in D at least once. *Support* of an itemset $X$ is denoted by $sup(X)$ and defined as the number of transactions in D that contain $X$. The itemset $X$ is called *frequent* if its support is greater than some user-defined threshold *minSup*. The set of all frequent itemsets will be denoted by $F$:

$$F = \{X \subseteq I | \ sup(X) > minSup\}.$$

**Property 2.1.1 [1].**
a) Let $X,Y \subseteq I$. If $X \subset Y$, then $sup(X) \geq sup(Y)$.
b) If $X \in F$, then $\forall Y \subset X$, $Y \in F$.

**Property 2.1.2.** Let $X,Y,V \subseteq I$. If $X \subset Y$ and $sup(X)=sup(Y)$, then $sup(X \cup V)=sup(Y \cup V)$.

### 2.2. Closures, closed itemsets and generators

*Closure* of an itemset $X$ is denoted by $\gamma(X)$ and is defined as the greatest (w.r.t. set inclusion) itemset that occurs in all transactions in D in which $X$ occurs.

**Property 2.2.1.** Let $X \subseteq I$.
a) $sup(\gamma(X))=sup(X)$.
b) $\forall Y \subseteq I$, if $X \subset Y \subseteq \gamma(X)$, then $sup(Y)=sup(X)$.
**Proof:** Ad. a) Immediate by definition of a closure.
Ad. b). By Prop. 2.1.1a, $sup(X) \geq sup(Y) \geq sup(\gamma(X))$ and by Prop. 2.2.1a, $sup(\gamma(X))=sup(X)$. Thus, $sup(Y)=sup(X)$. □

The itemset $X$ is defined *closed* iff $\gamma(X)=X$. The set of all closed itemsets will be denoted by $C$, i.e.

$$C = \{X \subseteq I | \ \gamma(X)=X\}.$$

Let $X$ be a closed itemset. A minimal itemset $Y$ satisfying $\gamma(Y)=X$ is called a *generator* of $X$. By $G(X)$ we will denote the set of all generators of $X$. The union of generators of all closed itemsets will be denoted by $G$, i.e.

$$G = \bigcup\{G(X) | \ X \in C\}.$$

**Example 2.2.1.** Let D be the database from Table 1. The itemset $\{A,B,C,D,E\}$ is closed since $\gamma(\{A,B,C,D,E\})=\{A,B,C,D,E\}$. The itemsets $\{A,B,C\}$ and $\{A,B,C,D\}$ are not closed because $\gamma(\{A,B,C\})=\{A,B,C,D,E\} \neq \{A,B,C\}$ and $\gamma(\{A,B,C,D\})=\{A,B,C,D,E\} \neq \{A,B,C,D\}$, respectively. The support of $\{A,B,C\}$ and $\{A,B,C,D\}$ is the same as the support of their closure $\{A,B,C,D,E\}$, and is equal to 3. $\{A,B,C\}$ is a minimal subset the closure of which equals to $\{A,B,C,D,E\}$. Hence, $\{A,B,C\} \in G(\{A,B,C,D,E\})$. □

| *Id* | *Transaction* |
|---|---|
| $T_1$ | $\{A,B,C,D,E,G\}$ |
| $T_2$ | $\{A,B,C,D,E,F\}$ |
| $T_3$ | $\{A,B,C,D,E,H,I\}$, |
| $T_4$ | $\{A,B,D,E\}$ |
| $T_5$ | $\{A,C,D,E,H,I\}$ |
| $T_6$ | $\{B,C,E\}$ |

**Table 1. Example database D.**

**Property 2.2.2 [8].** Let $X \subseteq I$.

$$\gamma(X) = \bigcap\{T \in D | \ T \supseteq X\} = \bigcap\{Y \subseteq I | \ Y \in C \wedge Y \supseteq X\}.$$

Property 2.2.2 states that the closure of an itemset $X$ can be computed: 1) as the intersection of the transactions in D that are supersets of $X$, or 2) as the intersection of the closed itemsets that are supersets of $X$.

**Property 2.2.3 [2].** Let $X \subseteq I$.

$$X \in G \text{ iff } sup(X) \neq \min\{sup(X \setminus \{A\}) | \ A \in X\}.$$

**Property 2.2.4.** Let $X \subseteq I$. $X \in G$ iff $\forall Y \subset X$, $sup(X) \neq sup(Y)$.
**Proof:** By Property 2.2.3 and Property 2.1.1a. □

**Lemma 2.2.1.** Let $X,Y \subseteq I$. If $X \subset Y \subseteq \gamma(X)$ then $\forall Z \supseteq Y$, $Z \notin G$.
**Proof** (By contradiction): Let $X \subset Y \subseteq \gamma(X)$ and $Z \in G$, $Z=Y \cup V$, $Y \cap V=\emptyset$. Since $X \subset Y \subseteq \gamma(X)$, then $sup(Y)=sup(X)$ (by Property 2.2.1b) and $sup(Z)=sup(Y \cup V)=sup(X \cup V)$ (by Property 2.1.2). Hence, $X \cup V$, which is a proper subset of $Z=Y \cup V$, has the same support as $Z$. Then by Property 2.2.4, $Z \notin G$, which contradicts the assumption. □

**Lemma 2.2.2.** If $X \subset Y$ and $sup(Y)=sup(X)$, then $\gamma(Y)=\gamma(X)$.
**Proof:** Let $X \subset Y$ and $sup(Y)=sup(X)$. Then, the set of transactions in D in which $X$ occurs, say D', equals to the set of transactions in which $Y$ occurs. Thus, by definition of a closure, $\gamma(X)$ as well as $\gamma(Y)$ are the greatest itemset that occurs in all transactions in D'. Hence, $\gamma(Y)=\gamma(X)$. □

**Theorem 2.2.1.** Let $X \subseteq I$. If $X \in G$, then $\forall Y \subset X$, $Y \in G$.
**Proof:** (By contradiction): Let $X \in G$ and $Y \subset X$ such that $Y \notin G$. Then, by Property 2.2.4 there is some $Z \subset Y$ such that $sup(Z)=sup(Y)$. Hence, by Lemma 2.2.2, $\gamma(Z)=\gamma(Y)$ and thus $Z \subset Y \subseteq \gamma(Y)=\gamma(Z)$. By Lemma 2.2.1, we conclude that $X$, which is a superset of $Y$, is not a generator. This conclusion contradicts the assumption. □

Theorem 2.2.1 states that subsets of generators are generators (a different proof was provided in [2]).

**Property 2.2.5.** Let $X\subseteq I$.

a) $sup(X) = \max\{sup(Y)|\ Y\in C \wedge Y\supseteq X\}$.

b) $sup(X) = \min\{sup(Y)|\ Y\in G \wedge Y\subseteq X\}$.

**Proof:** Ad. a). Let $Z=\gamma(X)$. Then, $Z\supseteq X$ and $Z\in C$. Thus, $sup(X)=sup(Z)$ (by Prop. 2.2.1a) and $sup(X)\geq sup(Y)$ for every $Y\in C$ such that $Y\supseteq X$ (by Prop. 2.1.1a). Therefore, $sup(X)=\max\{sup(Y)|\ Y\in C \wedge Y\supseteq X\}$.

Ad. b). Let $Z\subseteq X$ such that $Z\in G(\gamma(X))$. Then, $Z\in G$ and $Z\subseteq X\subseteq\gamma(X)=\gamma(Z)$. Thus, $sup(X)=sup(Z)$ (by Prop. 2.2.1b) and $sup(X)\leq sup(Y)$ for every $Y\in G$, $Y\subseteq X$ (by Prop. 2.1.1a). Therefore, $sup(X)=\min\{sup(Y)|\ Y\in G \wedge Y\subseteq X\}$. □

Hence, in order to compute support of any itemset it is sufficient to know either supports of all closed itemsets or supports of all generators.

## 2.3. Closed itemsets representation

Most research on concise representations of frequent itemsets was devoted to closed itemsets. Here we will present this representation.

An itemset $X$ is defined to be *frequent closed* iff $X$ is closed and frequent. In the sequel, the set of all frequent closed itemsets will be denoted by $FC$, i.e.

$$FC = F\cap C.$$

*Closed itemsets representation* is defined as the set $FC$ enriched by the information on support for each $X\in FC$.

The property below is an immediate consequence of Property 2.2.5a and shows how to determine if an itemset is frequent and if so, how to determine its support based on the closed itemsets representation.

**Property 2.3.1.** Let $X\subseteq I$.

- If there is $Z\in FC$, such that $Z\supseteq X$, then $X\in F$ and $sup(X) = \max(\{sup(Y)|\ Y\in FC \wedge Y\supseteq X\})$.
- Otherwise, $X\notin F$.

## 2.4. Generators representation

Generators are commonly used as an intermediate step for the discovery of closed itemsets. However, the generators themselves can constitute a concise lossless representation of frequent itemsets. Below we introduce such a generators representation:

*Frequent generators*, denoted by $FG$, are defined as:

$$FG = F\cap G.$$

*Negative generators border*, denoted by $GBd^-$, is defined as follows:

$$GBd^- = \{X\in G|\ X\notin F \wedge (\forall Y\subset X,\ Y\in FG)\}.$$

$GBd^-$ consists of all minimal (w.r.t. set inclusion) infrequent generators.

*Generators representation* is defined as:

- the set $FG$ enriched by the information on support for each $X\in FG$,
- the border set $GBd^-$,
- the set $I_D$ of items that occurred in D.

It can be proved that the generators representation as introduced here is equivalent to the approximate $\delta$-free sets representation [3-4] for $\delta$=0, in which case the approximate representation becomes lossless.

The property below is an immediate consequence of Property 2.2.5b and shows how to determine if an itemset is frequent and if so, how to determine its support based on the generators representation.

**Property 2.4.1.** Let $X\subseteq I$.

- If $\neg(X\subseteq I_D)$ or $(\exists Z\in GBd^-, Z\subseteq X)$, then $X\notin F$.
- Otherwise, $X\in F$ and $sup(X) = \min(\{sup(Y)|\ Y\in FG \wedge Y\subseteq X\})$.

## 2.5. Disjunction-free sets representation

The notion of *disjunction-free sets* was introduced in [5]. Let us present this concept by means of an auxiliary notion called a 2-*disjunctive rule.*

$X\Rightarrow A_1\vee A_2$ is defined a 2-*disjunctive rule* if $X\subseteq I$, $A_1,A_2\in I$, $X\cap\{A_1,A_2\}=\emptyset$. Observe, that a 2-disjunctive rule $X\Rightarrow A_1\vee A_2$ can have an empty antecedent ($X=\emptyset$) and its consequents can be equal ($A_1=A_2$).

*Support of* $X\Rightarrow A_1\vee A_2$, denoted by $sup(X\Rightarrow A_1\vee A_2)$, is defined as the number of transactions in D in which $X$ occurs together with $A_1$ or $A_2$, that is:

$$sup(X\Rightarrow A_1\vee A_2) = sup(X\cup\{A_1\}) + sup(X\cup\{A_2\}) - sup(X\cup\{A_1,A_2\}).$$

*Confidence* of the rule $X\Rightarrow A_1\vee A_2$, denoted by $conf(X\Rightarrow A_1\vee A_2)$, is defined as follows:

$$conf(X\Rightarrow A_1\vee A_2) = sup(X\Rightarrow A_1\vee A_2)/sup(X).$$

$X\Rightarrow A_1\vee A_2$ is defined a *certain rule* if $conf(X\Rightarrow A_1\vee Y_2)=1$. Thus, $X\Rightarrow A_1\vee A_2$ is certain if each transaction containing $X$ contains also $A_1$ or $A_2$.

**Example 2.5.1.** Let us consider the database D from Table 1. To make the notation brief, we will write itemsets without brackets and commas (e.g. $AC$ instead of $\{A,C\}$).

Let us consider the 2-disjunctive rule $\emptyset\Rightarrow A\vee A$. The rule is not certain since there is a transaction ($T_6$) that contains $\emptyset$, and does not contain $A$. On the other hand, $\emptyset\Rightarrow A\vee C$ is a certain rule as each transaction in D contains $A$ or $C$. Similarly, $C\Rightarrow D\vee E$ is a certain rule since each transaction containing $C$ contains also $D$ or $E$. □

**Property 2.5.1 [5].** $X\Rightarrow A_1\vee A_2$ is certain iff $sup(X) = sup(X\cup\{A_1\}) + sup(X\cup\{A_2\}) - sup(X\cup\{A_1,A_2\})$.

**Property 2.5.2 [5].** If $X \Rightarrow A_1 \vee A_2$ is certain, then $\forall Z \supset X$, $Z \Rightarrow A_1 \vee A_2$ is also certain.

**Example 2.5.2.** Let us consider the database D from Table 1. The rule $C \Rightarrow D \vee E$ is certain, thus $AC \Rightarrow D \vee E$ and $ABC \Rightarrow D \vee E$ (and so forth) are also certain rules. □

An itemset $X$ is defined *disjunctive* iff there are $A,B \in X$ such that $X \backslash \{A,B\} \Rightarrow A \vee B$ is a certain rule. Otherwise, the itemset is called *disjunction-free*[1]. The set of all disjunction-free sets will be denoted by *DFree*.

**Example 2.5.3.** Let us consider the database D from Table 1 and the itemset *DE*. The only 2-disjunctive rules involving all items in *DE* are: $\emptyset \Rightarrow D \vee E$, $D \Rightarrow E \vee E$, $E \Rightarrow D \vee D$. The rule $E \Rightarrow D \vee D$ is not certain, however $\emptyset \Rightarrow D \vee E$ and $D \Rightarrow E \vee E$ are certain, thus *DE* is a disjunctive set (i.e. $DE \notin DFree$). Now, since $\emptyset \Rightarrow D \vee E$ is certain in D, then by Property 2.5.2, $A \Rightarrow D \vee E$ is also certain. Hence $ADE \notin DFree$. Similarly, we can conclude $ACDE \notin DFree$ (and so forth). The property below generalizes this observation. □

**Property 2.5.3 [5].**
a) If $X \notin DFree$, then $\forall Y \supset X$, $Y \notin DFree$.
b) If $X \in DFree$, then $\forall Y \subset X$, $Y \in DFree$.

*Frequent disjunction-free itemsets*, denoted by *FDFree*, are defined as:

$$FDFree = DFree \cap F.$$

*Negative border of FDFree* is denoted by $DFreeBd^-$ and defined as:

$$DFreeBd^- = \{X \subseteq I \mid X \notin FDFree \wedge (\forall Y \subset X,\ Y \in FDFree)\}.$$

*Disjunction-free sets representation* is defined as:

- the set *FDFree* enriched by the information on support for each $X \in FDFree$,
- the border set $DFreeBd^-$ enriched by the information on support for each $X \in DFreeBd^-$,
- the set $I_D$ of items that occurred in D.

The disjunction-free sets representation is sufficient to determine all frequent itemsets and their supports [5].

## 3. New representation of frequent itemsets based on disjunction-free generators

In this section we will introduce a new representation of frequent itemsets based on frequent generators that are disjunction-free sets. We will prove that the new representation is sufficient to derive all frequent itemsets.

*Disjunction-free generators*, denoted by *DFreeG*, are defined as follows:

$$DFreeG = DFree \cap G.$$

**Property 3.1.** If $X \in DFreeG$, then $\forall Y \subset X$, $Y \in DFreeG$.
**Proof:** By Theorem 2.2.1 and Property 2.5.3b. □

*Frequent disjunction-free generators*, denoted by *FDFreeG*, are defined as:

$$FDFreeG = DFree \cap F \cap G.$$

**Property 3.2.** If $X \in FDFreeG$, then $\forall Y \subset X$, $Y \in FDFreeG$.
**Proof:** By Property 2.1.1b and Property 3.1. □

*Negative infrequent generators border*, denoted by $IDFreeGBd^-$, is defined as follows:

$$IDFreeGBd^- = \{X \in G \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFreeG)\}.$$

$IDFreeGBd^-$ consists of all minimal (w.r.t. set inclusion) infrequent generators the subsets of which are disjunction-free generators.

*Negative frequent generators border*, denoted by $FDFreeGBd^-$, is defined as:

$$FDFreeGBd^- = \{X \in G \mid X \in F \wedge X \notin DFreeG \wedge (\forall Y \subset X,\ Y \in FDFreeG)\}.$$

$DFreeGBd^-$ consists of all minimal (w.r.t. set inclusion) frequent disjunctive generators.

Let us note that $IDFreeGBd^- \cap FDFreeGBd^- = \emptyset$.

*Disjunction-free generators representation* is defined as:

- the set *FDFreeG* enriched by the information on support for each $X \in FDFreeG$,
- the border set $FDFreeGBd^-$ enriched by the information on support for each $X \in FDFreeGBd^-$,
- the border set $IDFreeGBd^-$,
- the set $I_D$ of items that occurred in D.

**Theorem 3.1.** The disjunction-free generators representation is sufficient to determine for any itemset if it is frequent and if so, to determine its support.

**Proof** (constructive): Any itemset $X$ that is not a subset of $I_D$ is infrequent. In the sequel of the proof, we assume $X \subseteq I_D$. The proof will be made by induction on $|X|$.

*Induction hypothesis. For every itemset $V \subset X$, we can determine if it is frequent or not, and if V is frequent then we can determine its support.*

One can distinguish the following five cases:

- If $X \in FDFreeG$, then $X \in F$ and $sup(X)$ is known.
- If $X \in FDFreeGBd^-$ then $X \in F$ and $sup(X)$ is known.
- If $\exists Y \in IDFreeGBd^-$, $Y \subseteq X$, then $X \notin F$.
- If $\neg \exists Z \in IDFreeGBd^-$, $Z \subseteq X$, and $\exists Y \in FDFreeGBd^-$, $Y \subset X$, then $X$ is a disjunctive set as a superset of some disjunctive itemset in $FDFreeGBd^-$ (by Property 2.5.3a). Let $Y \in FDFreeGBd^-$ and $Y \subset X$. Hence, there are some items $A,B \in Y$ such that the

[1] For the original definition of a disjunction-free set see [5]. Based on Lemma 3 in [5], we propose an equivalent definition that is more suitable for further presentation.

rule $Y\setminus\{A,B\}\Rightarrow A\vee B$ is certain. Let $A$ and $B$ be such items. Then, by Property 2.5.2, $X\setminus\{A,B\}\Rightarrow A\vee B$ is also certain and $sup(X)=sup(X\setminus\{A\})+sup(X\setminus\{B\})-sup(X\setminus\{A,B\})$. By induction hypothesis, we can determine if $X\setminus\{A\}$, $X\setminus\{B\}$, and $X\setminus\{A,B\}$ are frequent, and if so, we can determine their supports. If any of these itemsets is not frequent, then $X\notin F$. Now, if all the three itemsets are frequent, then $sup(X)$ can be determined according to the formula above. If $sup(X)>minSup$, then $X\in F$; otherwise $X\notin F$.

- Let $X\notin FDFreeG$ and $\neg\exists Z\in FDFreeGBd^- \cup IDFreeGBd^-$, $Z\subseteq X$. Then no generator being a subset of $X$ is a superset of any $Z\in FDFreeGBd^- \cup IDFreeGBd^-$. Hence, all generators being subsets of $X$ are contained in $FDFreeG$. By Property 2.2.5b, $sup(X)=\min(\{sup(Y)|\ Y\in G \wedge Y\subseteq X\})$. In our case, this equation is equivalent to: $sup(X)=\min(\{sup(Y)|\ Y\in FDFreeG \wedge Y\subseteq X\})$. Clearly, $X\in F$ as $sup(X)$ is equal to the support of some frequent disjunction-free generator. □

The proof of Theorem 3.1 can be treated as a naive algorithm for determining frequent itemsets and their supports.

**Example 3.1.** Given $minSup=1$, the following disjunction-free generators representation will be discovered in the database D from Table 1 (The information on supports of the itemsets is provided in the form of a subscript.):

- $FDFreeG = \{\emptyset_6, A_5, B_5, C_5, D_5, H_2, I_2\}$,
- $FDFreeGBd^- = \{AB_4, AC_4, BC_4, BD_4, CD_4\}$,
- $IDFreeGBd^- = \{F, G, BH, BI\}$,
- $I_D = ABCDEFGHI$.

Thus, the disjunction-free generators representation consists of 17 itemsets. Below we illustrate how to use this representation for evaluating the itemsets: $ACDF$ and $ACD$.

- The itemset $ACDF$ is infrequent, as it is a superset of the itemset $F$ in $IDFreeGBd^-$;
- The itemset $ACD$ is a superset of $AC\in FDFreeGBd^-$, so $ACD$ is disjunctive. The following 2-disjunctive rule is certain for $AC$: $\emptyset\Rightarrow A\vee C$. Hence, $D\Rightarrow A\vee C$ is a certain 2-disjunctive rule for $ACD$. Thus, $sup(ACD)=sup(AD)+sup(CD)-sup(D)=sup(AD)+4-5$. We note that $AD\notin FDFreeG$ and there is no subset of $AD$ in the border $FDFreeGBd^-\cup IDFreeGBd^-$. Hence, $sup(AD) = \min(\{sup(Y)|\ Y\in FDFreeG \wedge Y\subseteq AD\}) = \min\{sup(\emptyset),sup(A),sup(D)\}=5$. Finally, $sup(ACD)=5+4-5=4$. □

In the disjunction-free generators representation all infrequent items are kept in $IDFreeGBd^-$. An alternative more concise representation of frequent itemsets will not contain this information. Below, we specify such a *reduced disjunction-free generators representation*:

- $FDFreeG' = FDFreeG$,
- $FDFreeGBd^{-\prime} = FDFreeGBd^-$,
- $IDFreeGBd^{-\prime} = IDFreeGBd^- \setminus \{\{A\}|\ A\in I_D \wedge \{A\}\notin F\}$,
- $I_D' = I_D \setminus \{A\in I_D|\ \{A\}\notin F\}$.

We observe that only $IDFreeGBd^-$ and $I_D$ are reduced in this representation. The reduced disjunction-free generators representation can be used for retrieving frequent itemsets the same way as the original one.

**Example 3.2.** Given $minSup=1$, the following reduced disjunction-free generators representation will be obtained for the database D from Table 1:

- $FDFreeG' = \{\emptyset_6, A_5, B_5, C_5, D_5, H_2, I_2\}$,
- $FDFreeGBd^{-\prime} = \{AB_4, AC_4, BC_4, BD_4, CD_4\}$,
- $IDFreeGBd^{-\prime} = \{BH, BI\}$,
- $I_D' = ABCDEHI$.

Let us note that the infrequent items $F$ and $G$ do not occur in the reduced representation. □

## 4. Disjunction-free generators versus generators and disjunction-free sets

In this section we investigate the relationship between generators and disjunction-free sets and compare the disjunction-free generators representation with the generators and the disjunction-free sets representations.

**Theorem 4.1.** Let $X\subseteq I$.

a) If $X\notin G$, then $X\notin DFree$.
b) If $X\in DFree$, then $X\in G$.
c) $DFreeG = DFree$.

**Proof**: Ad. a) If $X\notin G$, then $\exists A\in X$, $sup(X\setminus\{A\})=sup(X)$ (by Property 2.2.3). Thus $X\setminus\{A\}\Rightarrow A\vee A$ is a certain disjunctive rule. So, $X\notin DisFree$.

Ad. b) By Theorem 4.1a: $X\notin G$ implies $X\notin DisFree$. Now, $X\notin G$ implies $X\notin DisFree$ iff $\neg X\notin G \vee X\notin DisFree$ iff $\neg X\in DisFree \vee X\in G$ iff $X\in DisFree$ implies $X\in G$.

Ad. c) Follows immediately from Theorem 4.1b. □

Theorem 4.1 states an interesting fact that each disjunction-free set is a generator. The proposition below compares the disjunction-free generators representation with the generators representation.

**Proposition 4.1.**

a) $FDFreeG \cup FDFreeGBd^- \subseteq FG$,
b) $IDFreeGBd^- \subseteq GBd^-$,
c) $FDFreeG \cup FDFreeGBd^- \cup IDFreeGBd^- \cup \{I_D\} \subseteq FG \cup GBd^- \cup \{I_D\}$.

**Proof:** By definitions of the disjunction-free generators representation and the generators representation. □

It follows from Proposition 4.1 that the disjunction-free generators representation constitutes a subset of the generators representation.

**Example 4.1.** Let us assume *minSup*=1. The following generators representation will be discovered in the database D from Table 1:

- $FG = \{\emptyset_6, A_5, B_5, C_5, D_5, H_2, I_2, AB_4, AC_4, BC_4, BD_4, CD_4, \underline{ABC}_3, \underline{BCD}_3\}$,
- $GBd^- = \{F, G, BH, BI\}$,
- $I_\mathrm{D} = ABCDEFGHI$.

The generators representation consists of 19 itemsets. In comparison with the disjunction-free generators representation (see Example 3.1), the generators representation contains 2 more itemsets (the underlined ones). □

In order to compare the new representation with the disjunction-free sets one, below we specify properties of sets characteristic for these representations.

**Lemma 4.1.**

a) $IDFreeGBd^- = \{X \in G \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$,

b) $FDFreeGBd^- = \{X \in G \mid X \in F \wedge X \notin DFree \wedge (\forall Y \subset X,\ Y \in FDFree)\}$,

c) $DFreeBd^- \setminus F = \{X \subseteq I \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$,

d) $DFreeBd^- \cap F = \{X \subseteq I \mid X \in F \wedge X \notin DFree \wedge (\forall Y \subset X,\ Y \in FDFree)\}$.

**Proof:** Ad. a) By Theorem 2.2.1, if an itemset $X$ is a generator, then all its subsets are generators. Thus, $\{X \in G \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\} =$
$\{X \in G \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFreeG)\} = IDFreeGBd^-$.
Ad. b) Similar to that for the case a).
Ad. c, d) Immediate by definition of $DFreeBd^-$. □

Now, we are able to compare both representations:

**Proposition 4.2.**

a) $FDFreeG = FDFree$,

b) $FDFreeGBd^- \subseteq DFreeBd^- \cap F$,

c) $IDFreeGBd^- = DFreeBd^- \setminus F$,

d) $FDFreeG \cup FDFreeGBd^- \cup IDFreeGBd^- \cup \{I_\mathrm{D}\} \subseteq FDFree \cup DFreeBd^- \cup \{I_\mathrm{D}\}$.

**Proof:** Ad. a) Follows immediately from Theorem 4.1c.
Ad. b) Immediate from Lemma 4.1b, d.
Ad. c) We will prove that $IDFreeGBd^- = DFreeBd^- \setminus F$ by showing the equivalence of the following sets $\{X \in G \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$ and $\{X \subseteq I \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$ that are equal to $IDFreeGBd^-$ and $DFreeBd^- \setminus F$, respectively (by Lemma 4.1a,c).

Let $X$ be an infrequent itemset whose all proper subsets are frequent. Then, $\forall Y \subset X,\ sup(Y) > sup(X)$. By Property 2.2.4, each such itemset $X$ is a generator.

The set $\{X \subseteq I \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$ consists of infrequent itemsets whose proper subsets are frequent. Thus, each itemset in $\{X \subseteq I \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$ is a generator. Hence, $\{X \subseteq I \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$ $= \{X \in G \mid X \notin F \wedge (\forall Y \subset X,\ Y \in FDFree)\}$.
Ad. d) Immediate from Proposition 4.2a-c. □

As follows from Proposition 4.2, the disjunction-free generators representation constitutes a subset of the disjunction-free sets representation. Surprisingly, the both representations differ solely on the parts of the respective negative borders that contain frequent itemsets.

**Example 4.2.** Let us assume *minSup*=1. The following disjunction-free sets representation will be discovered in the database D from Table 1:

- $FG = \{\emptyset_6, A_5, B_5, C_5, D_5, H_2, I_2\}$,
- $DFreeBd^- \cap F = \{\underline{E}_6, AB_4, AC_4, \underline{AD}_5, \underline{AH}_2, \underline{AI}_2, BC_4, BD_4, CD_4, \underline{CH}_2, \underline{CI}_2, \underline{DH}_2, \underline{DI}_2, \underline{HI}_2\}$,
- $DFreeBd^- \setminus F = \{F, G, BH, BI\}$,
- $I_\mathrm{D} = ABCDEFGHI$.

The disjunction-free sets representation consists of 26 itemsets. In comparison with the disjunction-free generators representation (see Example 3.1), the disjunction-free generators representation contains 9 more itemsets (the underlined ones). In accordance with Proposition 4.2, all the redundant itemsets belong to $DFreeBd^- \cap F$. □

## 5. Computing disjunction-free generators representation

### 5.1. Algorithmic properties of disjunction-free generators representation

In this subsection, we provide properties that will be used in the algorithm determining the disjunction-free generators representation.

**Lemma 5.1.1.** Let $X \subseteq I$. The following statements are equivalent.

- $\exists A \in X$ such that $X \setminus \{A\} \Rightarrow A \vee A$ is a certain rule.
- $\exists A \in X$ such that $sup(X) = sup(X \setminus \{A\})$.
- $X \notin G$.

**Proof:** $\exists A \in X$ such that $X \setminus \{A\} \Rightarrow A \vee A$ is a certain rule iff $sup(X) = sup(X \setminus \{A\}) + sup(X \setminus \{A\}) - sup(X \setminus \{A, A\})$ iff $sup(X) = sup(X \setminus \{A\})$ iff $X \notin G$ (by Property 2.2.3). □

**Lemma 5.1.2.**

a) $\emptyset \in DFree$.

b) $\emptyset \in G$.

c) Let $A \in I$. $\{A\} \in DFree$ iff $\{A\} \in G$.

**Proof:** Ad. a) There is no 2-disjunctive rule involving only $\emptyset$ and no more items. Hence, $X \in DFree$.
Ad. b) Immediate from Lemma 5.1.2a and Theorem 4.1b.
Ad. c) ($\Rightarrow$) Immediate from Theorem 4.1b.
($\Leftarrow$) If $\{A\} \in G$, then $\emptyset \Rightarrow A \vee A$, which is the only

2-disjunctive rule that can be built from $\{A\}$, is not certain (by Lemma 5.1.1). Thus $\{A\} \in DFree$. □

**Lemma 5.1.3.** Let $X \in G$. The following statements are equivalent.

- $X$ is a disjunctive set.
- $\exists A,B \in X$ such that $A \neq B$ and $X \setminus \{A,B\} \Rightarrow A \vee B$ is a certain rule.
- $\exists A,B \in X$ such that $A \neq B$ and $sup(X) = sup(X \setminus \{A\}) + sup(X \setminus \{B\}) - sup(X \setminus \{A,B\}$.

**Proof:** Immediate by definition of a disjunctive set and Lemma 5.1.1. □

## 5.2. Algorithm for determining disjunction-free generators representation

The outline of the *DFreeGenApriori* algorithm we propose is similar to that of *Apriori* (see [1]). It differs from the original algorithm by additional constraints that guarantee the resultant set to be restricted to the frequent disjunction-free generators and their border instead of the whole set of frequent itemsets.

In the algorithm we use the following notation:

- $FDFreeG_k$, $FDFreeGBd^-_k$, $IDFreeGBd^-_k$, – $k$-itemsets in the respective components of the disjunction-free generators representation;
- $C_k$ – candidate frequent disjunction-free $k$-generators.

The itemsets are assumed to be kept in an ascending order. With each itemset $c$ there are associated the following fields:

- *sup* – support of $c$;
- *minSubSup* – minimum of the supports of the proper subsets of $c$.

The *DFreeGenApriori* algorithm starts with checking if the number of transactions in D is greater than *minSup*. If so, then $\emptyset$ is frequent. By Lemma 5.1.2a-b, $\emptyset$ is a disjunction-free generator. Hence, $\emptyset$ is included in $FDFreeG_0$ provided $\emptyset$ is frequent. Next, all items in D are identified and stored as 1-candidates in $C_1$. Their union determines $I_D$. By Property 2.2.3, each itemset in $C_1$ is a generator if its support differs from $sup(\emptyset)$. In addition, Lemma 5.1.2c guarantees that each generator in $C_1$ is a disjunction-free set. Hence, each generator in $C_1$ is added to the set of frequent disjunction-free generators $FDFreeG_1$, if its support is sufficiently high. Otherwise, it is included in the negative infrequent generators border $IDFreeGBd^-_1$. Next, the 2-candidates $C_2$ are created from $FDFreeG_1$ by the *AprioriGGen* algorithm (see Subsection 5.3). Now, the following steps are performed level-wise for all $k$-candidates, for $k \geq 2$:

1. Supports for the candidate $k$-itemsets $C_k$ are determined by a pass over the database (see proc. *SupportCount*)
2. The $k$-candidates $C_k$ the support of which differs from the supports of their proper subsets ($c.sup \neq c.minSubSup$) are found generators (by Property 2.2.3).
3. Infrequent $k$-generators in $C_k$ are added to the negative infrequent generators border $IDFreeGBd^-_k$. The *IsDis* function determines for each frequent $k$-generator if it is disjunctive (see Subsection 5.4). Frequent disjunctive $k$-generators are added to the negative frequent generators border $FDFreeGBd^-_k$. The remaining frequent $k$-generators are disjunction-free and hence, they are added to $DFreeG_k$.
4. The *AprioriGGen* function is called to generate the candidate $(k+1)$-itemsets $C_{k+1}$ from the frequent disjunction free $k$-generators $FDFreeG_k$ and to initialize the *minSubSup* field for each new candidate (see Subsection 5.3). *AprioriGGen* follows Property 3.2 to guarantee that the $(k+1)$-candidates include all itemsets having all their subsets in $FDFreeG_k$.

The algorithm ends when there are no more candidates.

```
Algorithm DFreeGenApriori(var FDFreeG, FDFreeGBd⁻,
                                    IDFreeGBd⁻, $I_D$);
FDFreeG = {}; FDFreeGBd⁻ = {}; IDFreeGBd⁻ = {}; $I_D = \emptyset$;
if |D| > minSup then begin
    $\emptyset.sup = |D|$; $FDFreeG_0 = \{\emptyset\}$;
    $C_1$ = {1-itemsets in D with minSubSup initialized to $\emptyset.sup$};
    $I_D = \cup C_1$;
    forall candidates $c \in C_1$ do begin
        SupportCount($C_1$);
        if $c.sup \neq \emptyset.sup$ then                    // c is a generator
            if $c.sup \leq minSup$ then add c to $IDFreeGBd^-_1$
/*or remove c from $I_D$ if computing the reduced representation*/
            else add c to $FDFreeG_1$
            endif;
        endif;
    endfor;
    $C_2$ = AprioriGGen($FDFreeG_1$);
    for ($k = 2$; $C_k \neq \emptyset$; $k$++) do begin
        SupportCount($C_k$);
        forall candidates $c \in C_k$ do
            if $c.sup \neq c.minSubSup$ then          // c is a generator
                if $c.sup \leq minSup$ then add c to $IDFreeGBd^-_k$
                elseif IsDis(c, $FDFreeG_{k-1}$, $FDFreeG_{k-2}$) then
                    add c to $FDFreeGBd^-_k$
                else add c to $FDFreeG_k$
                endif;
            endif;
        endfor;
        $C_{k+1}$ = AprioriGGen($FDFreeG_k$);
    endfor;
    $FDFreeG = \cup_k FDFreeG_k$;
    $FDFreeGBd^- = \cup_k FDFreeGBd^-_k$;
    $IDFreeGBd^- = \cup_k IDFreeGBd^-_k$;
endif;
return <FDFreeG, FDFreeGBd⁻, IDFreeGBd⁻, $I_D$>;
```

```
procedure SupportCount(var $C_k$);
forall transactions $t \in D$ do
    forall candidates $c \in C_k$ do
        if $c \subseteq t$ then c.count++;
        endif;
    endfor;
endfor;
endproc;
```

Let us observe that an algorithm for computing the reduced disjunction-free generators representation would differ only slightly from the presented *DFreeGenApriori* algorithm. The only change would occur for candidate infrequent 1-generators. Such candidates should be discarded from $I_D$ instead of being added to $IDFreeGBd^-_1$.

### 5.3. Generating candidates

The *AprioriGGen* function is similar to *AprioriGen* (see [1] for details). The difference consists in additional computing the value of *minSubSup* field. For each new candidate *c*, *minSubSup* is assigned the minimum from the supports of the proper subsets of *c*.

```
function AprioriGGen($G_k$);
forall $f, h \in G_k$ do
    if $f[1]=h[1] \wedge \ldots \wedge f[k-1]=h[k-1] \wedge f[k]<h[k]$ then begin
        $c = f[1] \bullet f[2] \bullet \ldots \bullet f[k] \bullet h[k]$;
        add $c$ to $C_{k+1}$
    endif;
endfor;
/* Pruning */
forall $c \in C_{k+1}$ do
    forall $k$-itemsets $s \subset c$ do
        if $s \notin G_k$ then delete $c$ from $C_{k+1}$
        else $c.minSubSup = \min(c.minSubSup, s.sup)$
        endif;
    endfor;
endfor;
return $C_{k+1}$;
```

### 5.4. Checking if generator is disjunctive

The *IsDis* function checks if an itemset *c* provided as the first argument is disjunctive or not. It is assumed that *c* is a frequent generator of the size $k \geq 2$. The second and third arguments: $FDFreeG_{k-1}$, $FDFreeG_{k-2}$, contain all frequent disjunction-free generators of the size *k*-1 and *k*-2, respectively. Let us note that for every pair $(g_1,g_2)$ of different (*k*-1)-subsets of *c*, $g_1 \cap g_2$ is a (*k*-2)-subset of *c*. *IsDis* checks if there is a pair $(g_1,g_2)$ of different (*k*-1)-subsets of *c* satisfying the equation: $sup(c)=sup(g_1) +sup(g_2)-sup(g_1 \cap g_2)$. If so, then by Lemma 5.1.3 the itemset *c* is disjunctive and the function returns **true**. Otherwise, *c* is not disjunctive and the function returns **false**.

```
function IsDis($k$-itemset $c$, $FDFreeG_{k-1}$, $FDFreeG_{k-2}$);
/* Assert: $c$ is a frequent generator of the size $k \geq 2$ */
forall ($k$-1)-itemsets $g_1, g_2 \subset c$ such that $g_1 \neq g_2$ do begin
    determine supports of $g_1$ and $g_2$ based on $FDFreeG_{k-1}$;
    determine support of $(g_1 \cap g_2)$ based on $FDFreeG_{k-2}$;
    if $c.sup = g_1.sup + g_2.sup - (g_1 \cap g_2).sup$ then return true;
    endif;
endfor;
return false;
```

## 6. Conclusions

An overview of concise lossless representations of frequent itemsets was provided. The new lossless disjunction-free generators representation was offered. It was proved that the new representation constitutes a subset of the generators representation and the disjunction-free sets representation. It was also proved that each disjunction-free set is a generator. The algorithm for determining the new representation was offered.

## References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307-328. AAAI Press, Menlo Park, California, 1996.

[2] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, L. Lakhal. Mining frequent patterns with counting inference. *ACM SIGKDD Explorations*, Vol. 2(2):66-75, December 2000.

[3] J-F. Boulicaut, A. Bykowski, C. Rigotti. Approximation of frequency queries by means of free-sets. In *Proc. of PKDD '00*, pages 75-85, Springer, September 2000.

[4] J-F. Boulicaut, A. Bykowski, C. Rigotti. Free-Sets: a condensed representation of Boolean data for the approximation of frequency queries. Research Report, LISI, INSA-Lyon, June 2001.

[5] A. Bykowski, C. Rigotti. A condensed representation to find frequent patterns. In *Proc. of the 12th ACM SIGACT-SIGMOD-SIGART PODS' 01*, May 2001.

[6] J. Han, M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2000.

[7] M. Kryszkiewicz. Closed set based discovery of representative association rules. In *Proc. of IDA '01*, Springer, September 2001.

[8] N. Pasquier. 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, January 2000.

[9] J. Pei, J. Han, R. Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *Proc. of the ACM-SIGMOD DMKD '00*, pages 21-30, Dallas, May 2000.

[10] J. Saquer, J. S. Deogun. Using closed itemsets for discovering representative association rules. In *Proc. of ISMIS '00*, pages 495-504, Springer, October 2000.

[11] M. J. Zaki. Generating non-redundant association rules. In *Proc. of the 6th ACM SIGKDD Intl Conf. on Knowledge Discovery and Data Mining*, pages 34-43, August 2000.