**Acta Informatica manuscript No.**
(will be inserted by the editor)

**Toon Calders**

# The Complexity of Satisfying Constraints on Databases of Transactions

Received: date / Revised: date

**Abstract** Computing frequent itemsets is one of the most prominent problems in data mining. Recently, a new related problem, called `FREQSAT`, was introduced and studied: given some itemset-interval pairs, does there exist a database such that for every pair, the frequency of the itemset falls in the interval? In this paper, we extend this `FREQSAT`-problem by further constraining the database by giving other characteristics as part of the input as well. These characteristics are the maximal transaction length, the maximal number of transactions, and the maximal number of duplicates of a transaction. These extensions and all their combinations are studied in depth, and a hierarchy w.r.t. complexity is given. To make a complete picture, also the cases where the characteristics are constant; i.e., bounded and the bound being a fixed constant that is not a part of the input, are studied.

## 1 Introduction

The *frequent itemset mining problem* [1] is one of the core problems in data mining. We are given a database $\mathcal{D}$ of sets, called *transactions*, and a threshold *minfreq*. The *frequency* of a set $I$ in $\mathcal{D}$ is the number of transactions in $\mathcal{D}$ that contain all items of $I$ divided by the total number of transactions in $\mathcal{D}$. The frequent itemset problem is to compute all sets $I$ such that the frequency of $I$ in $\mathcal{D}$ is at least *minfreq*.

The problem `FREQSAT` [7,12] was introduced in this context: given a collection of expressions $\mathrm{freq}\,(I) \in [a,b]$, does there exist a database of transactions that satisfies them? For example, $\{\mathrm{freq}\,(\{a\}) \in [0,0.5], \mathrm{freq}\,(\{a,b\}) \in [0.6,1]\}$ is not satisfiable, because of the monotonicity of frequency. As pointed out

Toon Calders
Eindhoven University of Technology, Departement of Mathematics and Computer Science, Den Dolech 2, 5600 MB Eindhoven, The Netherlands
E-mail: t.calders@tue.nl

in [7,12], the study of the FREQSAT-problem is interesting in the context of condensed representations [11], privacy preserving data mining, and optimizing the pruning in frequent itemset mining algorithms: in these three application areas, the question of what can be derived from some given frequencies is important. In condensed representations, this information can be used to see whether the frequency of a certain itemset in a collection is uniquely determined by the other itemsets in that collection. If that is the case, such a redundant itemset can be removed without loosing information. This approach has been applied successfully in the Non-Derivable Itemsets representation of the frequent itemsets [9]. For the privacy-preserving data mining, FREQSAT and its variants can be used to assess to what extent released frequency information can lead to the disclosure of the frequencies of other itemsets. Last, but not least, FREQSAT can help mining algorithms for pruning itemsets. Based on some frequency information, gathered in previous iterations, often it can be seen that a candidate itemset can be pruned because there cannot exist a database that satisfies the already found frequencies together with the constraint that the frequency of the candidate is above the threshold. To some extent, all frequent mining algorithms already use this information when they apply the Apriori-principle. With FREQSAT this pruning can be extended. In the context of the Non-Derivable Itemsets, this extended pruning, in combination with the derivation of frequencies for the redundant sets has been applied [9].

In this paper, we extend the original FREQSAT-problem of [7] as follows. Besides bounds on the frequency of itemsets, also other constraints on the database are given. These constraints change the FREQSAT problem considerably. Consider, e.g., the following set $\mathcal{C}$ of constraints:

$$\left\{\mathrm{freq}\left(\{a\}\right)=\frac{1}{2}, \mathrm{freq}\left(\{b\}\right)=\frac{1}{2}, \mathrm{freq}\left(\{c\}\right)=\frac{1}{2}, \mathrm{freq}\left(\{a,b,c\}\right)=0\right\}$$

$\mathcal{C}$ is satisfiable by the database $\{a,b\},\{a,c\},\{b,c\},\{\}$. If we, however, require that the number of transactions is 2, or that every transaction contains at most 1 item, $\mathcal{C}$ is no longer satisfiable. This simple example already shows that a seemingly small adaptation of the original problem can have a large influence. Another important difference is in the entailment. $\mathrm{ENT}_I(\mathcal{C})$ will denote the set of all possible frequency values for $I$ given that $\mathcal{C}$ holds. For FREQSAT, $\mathrm{ENT}_I(\mathcal{C})$ is always an *interval* of the rational numbers. If we, however, fix the number of transactions, the set $\mathrm{ENT}_I(\mathcal{C})$ can be *any finite subset* of rational numbers between 0 and 1.

The characteristics we consider are: the *maximal transaction size*, the *number of transactions*, and the *maximal number of duplicates* of a transaction. The complexity of the problem depends on the additional characteristics. We show that the extension of FREQSAT where, besides a set of frequency constraints, also an upper bound on the length of the transactions is part of the input, has the complexity as plain FREQSAT. When an upper bound on the number of transactions is added as part of the input, the properties of FREQSAT change drastically, but it is left open whether it increases the complexity. When the upper bound on the number of duplicate transactions

| Restrictions on $\mathcal{D}$ | | | Complexity | |
|---|---|---|---|---|
| ltrans | ntrans | ndup | lower | upper |
| cte | | | **P** | |
| | 1 | | **P** | |
| | | | **NP** | |
| ● | | | **NP** | |
| | cte≥ 2 | | **NP** | |
| cte≥ 3 | ● | | **NP** | **PSPACE** |
| | ● | | **NP** | **PSPACE** |
| ● | ● | | **NP** | **PSPACE** |
| | | ● | **PP** | **PSPACE** |
| | | cte | **PP** | **PSPACE** |
| ● | | ● | **PP** | **PSPACE** |
| | ● | ● | **PP** | **PSPACE** |
| ● | ● | ● | **PP** | **PSPACE** |

●: the characteristic is part of the input
cte: the characteristic is a constant expression

**Table 1** Summary of the complexity results; every row represents one variant of FREQSAT. Problems in the same partition (i.e., no horizontal line separates them), are proven to be equivalent w.r.t. logspace reductions. Problems separated by a double horizontal line, are proven not to be equivalent(assuming $\mathbf{P} \neq \mathbf{NP} \neq \mathbf{PP}$).

is added to the input, the problem provably becomes more complex (assuming $\mathbf{PP} \neq \mathbf{NP}$). To make a complete picture, we also study the case where the extra constraints on the allowable databases are fixed; i.e., they are not counted as a part of the input. The complexities of the FREQSAT-variants that are proven in the paper, have been summarized in Table 1.

The organization of the paper is as follows. In Section 2, we formally introduce important notions, and define the problems studied in the paper. In Section 3, many important properties of FREQSAT without the extensions, that will be needed throughout this paper, are revisited. Then, in Sections 4, 5, and 6, FREQSAT is gradually extended with bounds on the transaction length, on the number of transactions, and on the number of duplicates. Section 7 discusses applications and gives connections between on the one hand FREQSAT and its extensions and on the other hand, related works in data mining and probabilistic logics. Section 8 summarizes the most important results and concludes the paper.

## 2 Preliminaries

In this section we revisit the definition of the FREQSAT-problem, and we formalize the extensions studied in this paper.

### 2.1 Itemsets, Frequencies, and Databases

Let $\mathcal{I}$ be a finite set, called the set of items. A *transaction* over $\mathcal{I}$ is a pair $(tid, J)$, with $tid$ an identifier, and $J$ a subset of $\mathcal{I}$. A *database* over $\mathcal{I}$ is a finite set of such transactions where no two transactions have the same identifier. In the following, we assume that the transaction identifiers are strictly positive integers. Hence, a transaction is a pair $(tid, I)$, with $tid \in \{1, 2, 3, \ldots\}$, and $I \subseteq \mathcal{I}$.

Let $I$ be some set of items. We say that the transaction $(tid, J)$ *contains* $I$, denoted $I \subseteq (tid, J)$, if $I \subseteq J$. The *support* of $I$ in $\mathcal{D}$, denoted $\mathrm{supp}(I, \mathcal{D})$, is the absolute number of transactions in $\mathcal{D}$ that contain $I$. The *frequency* of $I$ in $\mathcal{D}$, denoted $\mathrm{freq}\,(I, \mathcal{D})$, is $\mathrm{supp}(I, \mathcal{D})$ divided by the number of transactions in $\mathcal{D}$.

In all what follows, $\mathcal{D}$ is a database of transactions over $\mathcal{I}$.

## 2.2 Frequency Constraints

A *Frequency Constraint* is an expression $\mathrm{freq}\,(I) \in [l, u]$, with $I$ an itemset, and $0 \leq l, u \leq 1$ rational numbers. We say that $\mathcal{D}$ *satisfies* this expression, denoted $\mathcal{D} \models \mathrm{freq}\,(I) \in [l, u]$, if the frequency of $I$ in $\mathcal{D}$ is in the interval $[l, u]$. $\mathcal{D}$ satisfies a set of frequency constraints, if it satisfies all of them.

A set of frequency constraints $\mathcal{C}$ *entails* a constraint $\mathrm{freq}\,(I) \in [l, u]$, denoted $\mathcal{C} \models \mathrm{freq}\,(I) \in [l, u]$, if every database $\mathcal{D}$ that satisfies $\mathcal{C}$, satisfies $\mathrm{freq}\,(I) \in [l, u]$ as well. The entailment is said to be *tight*, denoted $\mathcal{C} \models_{tight} \mathrm{freq}\,(I) \in [l, u]$, if for every smaller interval $[l', u'] \subset [l, u]$, $\mathcal{C}$ does not entail $\mathrm{freq}\,(I) \in [l', u']$. That is, if $[l, u]$ is the best interval that can be derived for $I$, based on $\mathcal{C}$.

For notational convenience, we use the shorthand $\mathrm{freq}\,(I) = f$ to denote $\mathrm{freq}\,(I) \in [f, f]$.

*Example 1* Consider the following set of frequency constraints:

$$\mathcal{C} = \left\{ \begin{array}{ll} \mathrm{freq}\,(\{a\}) \in [0.75, 1], & \mathrm{freq}\,(\{b\}) \in [0.5, 0.75], \\ \mathrm{freq}\,(\{c\}) = 0.75, & \mathrm{freq}\,(\{a, b\}) = 0.5 \end{array} \right\} .$$

This set of constraints is satisfied by the database

$$\mathcal{D} = \{(1, \{a, b\}), (2, \{a, c\}), (3, \{c\}), (4, \{a, b, c\})\} .$$

The constraint $\mathrm{freq}\,(\{a, b, c\}) = 0.5$ is not entailed by the constraints in $\mathcal{C}$. The database $\mathcal{D}$ is a counter example; it satisfies $\mathcal{C}$, but it does not satisfy $\mathrm{freq}\,(\{a, b, c\}) = 0.5$.

The constraint $\mathrm{freq}\,(\{a, b, c\}) = [0, 0.5]$ is entailed by $\mathcal{C}$. Indeed, because of the monotonicity of frequency, the frequency of $\{a, b, c\}$ must always be less than or equal to the frequency of $\{a, b\}$. Therefore, in every database that satisfies $\mathrm{freq}\,(\{a, b\}) = 0.5$, the frequency of $\{a, b, c\}$ will be less than 0.5. The entailment is not tight, however, because the interval $[0, 0.5]$ can be made smaller; in every database that satisfies $\mathcal{C}$, the frequency of $\{a, b, c\}$ must be at least 0.25. This can easily be seen as follows: because of the constraints $\mathrm{freq}\,(\{c\}) = 0.75$ and $\mathrm{freq}\,(\{a, b\}) = 0.5$, 75% of the transactions of a satisfying database for $\mathcal{C}$ contains item $c$, and 50% contains items $a$ and $b$. Therefore, there must be an overlap of at least 25% transactions that contain item $c$ and that contain items $a$ and $b$.

The entailed interval $[0.25, 0.5]$ for $\{a, b, c\}$ from $\mathcal{C}$ is tight. We can prove this by showing, with examples, that the lower and upper bound are indeed both feasible. The tightness of the lower bound follows from the database $\mathcal{D}$. For the upper bound, the following database shows the tightness:

$$\{(1, \{a, b, c\}), (2, \{a, b, c\}), (3, \{a, c\}), (4, \{b\})\} \ .$$

## 2.3 Other Database Constraints

In realistic situations, often, more characteristics of a database of transactions are known than only the frequencies of some sets. We now describe what extra information we will consider in this paper.

**Transaction Length** The number of items is, of course, always an upper bound for the maximal number of items in a transaction. Often, however, the maximal size of the transactions is given. Moreover, it is a common practice in frequent itemset mining to start from a relational table $R(A_1, \ldots, A_n)$, and to encode this table as a database of transactions before mining. This transformation is carried out as follows: for every attribute-value pair $(A, v)$ of $R$, an item $I_{(A,v)}$ is introduced. A tuple $(v_1, \ldots, v_n)$ is represented by the transaction $\{I_{(A_1,v_1)}, \ldots, I_{(A_n,v_n)}\}$. In such a situation, if the original schema is known, also the maximal transaction size is known.

**Number of Transactions** The size of the database $|\mathcal{D}|$ is often known to the user. Knowing the number of transactions seriously affects the properties of `FREQSAT`.

**Number of Duplicates** In our definition of frequent set mining we did not require that the set of items in a transaction is unique; due to the identifier, two different transactions can have the same set of items. In many practical situations, however, duplicates cannot occur, or a maximal number of duplicates is known. For example, in the case that the database of transactions was created from a relational table, no duplicate transactions can be present. Even if some attributes of the original table are filtered away, the maximal possible number of duplicates might be known. Suppose that the table $R(A_1, \ldots, A_n)$ is transformed as described above, but some, binary valued, attributes $A_1, \ldots, A_k$ are filtered away. In that case, the number of duplicates is at most $2^k$.

## 2.4 Problem Statement

We are now ready to state the main problems studied in this paper: `FREQSAT` under additional constraints.

In [7], the following problem `FREQSAT` was introduced:

**Problem 1** **(`FREQSAT`)**
**Input:** A set of frequency constraints

$$\mathcal{C} = \{\mathrm{freq}\,(I_j) \in [l_j, u_j], j = 1 \ldots m\}$$

**Accept:** iff there exists a database $\mathcal{D}$ over $\bigcup_{j=1}^{m} I_j$ that satisfies $\mathcal{C}$.

In this paper we study extensions of `FREQSAT` wherein more characteristics of the database $\mathcal{D}$ are known. Formally, these characteristics are:

$$\begin{aligned} ltrans(\mathcal{D}) &:= \max\{|J| \mid (tid, J) \in \mathcal{D}\} \\ ntrans(\mathcal{D}) &:= |\mathcal{D}| \\ ndup(\mathcal{D}) &:= \max_{J \subseteq \mathcal{I}} |\{tid \mid (tid, J) \in \mathcal{D}\}| \end{aligned}$$

`FREQSAT`$\{c_1, \ldots, c_k\}$ is the variant of `FREQSAT`where upper bounds on the characteristics $c_1, \ldots, c_k$ are part of the input as well. Under this convention, `FREQSAT`$\{ltrans, ndup\}$ denotes the variant in which besides frequency constraints, also a maximal transaction length and a maximal number of duplicates have been given.

**Problem 2** **(**`FREQSAT`$\{c_1, \ldots, c_k\}$**)**
**Input:** A tuple $(\mathcal{C}, v_1, \ldots, v_k)$, with

$$\mathcal{C} = \{\text{freq}\,(I_j) \in [l_j, u_j], j = 1 \ldots m\}$$

a set of frequency constraints and $v_1, \ldots, v_k$ numbers.
**Accept:** if and only if there exists a database $\mathcal{D}$ over $\bigcup_{j=1}^{m} I_j$ that satisfies $\mathcal{C}$ and for all $l = 1 \ldots k$, $c_l(\mathcal{D}) \leq v_l$.

In the paper we also discuss cases where some of the characteristics are bounded, but not part of the input. These fixed-parameter cases are defined as follows. Let $C_1 = \{c_1, \ldots, c_p\}$, and $C_2 = \{c_{p+1}, \ldots, c_k\}$ be disjoint subsets of $\{ltrans, ntrans, ndup\}$, and let $v_1, \ldots, v_p$ be positive integers. The parameterized variant `FREQSAT`$\{c_1 = v_1, \ldots, c_p = v_p, c_{p+1}, \ldots, c_k\}$ of `FREQSAT` is now defined as follows.

**Problem 3** **(**`FREQSAT`$\{c_1 = v_1, \ldots, c_p = v_p, c_{p+1}, \ldots, c_k\}$**)**
**Input:** A tuple $(\mathcal{C}, v_{p+1}, \ldots, v_k)$, with

$$\mathcal{C} = \{\text{freq}\,(I_j) \in [l_j, u_j], j = 1 \ldots m\}$$

a set of frequency constraints and $v_{p+1}, \ldots, v_k$ numbers.
**Accept:** if and only if there exists a database $\mathcal{D}$ over $\bigcup_{j=1}^{m} I_j$ that satisfies $\mathcal{C}$ and for all $l = 1 \ldots k$, $c_l(\mathcal{D}) \leq v_l$.

The main question in this paper is: what are the computational complexities of the different `FREQSAT`-variants, and what are the relations and differences between them? We will focus mainly on Problem 2.

*Example 2* Suppose that the following set of frequency constraints $\mathcal{C}$ is given:

$$\left\{ \begin{array}{lll} \text{freq}\,(\{a, b\}) \in [3/4, 1], & \text{freq}\,(\{a, c\}) \in [3/4, 1], & \text{freq}\,(\{b, c\}) \in [3/4, 1], \\ \text{freq}\,(\{d, e\}) \in [3/4, 1], & \text{freq}\,(\{d, f\}) \in [1/2, 1], & \text{freq}\,(\{e, f\}) \in [1/2, 1], \\ \text{freq}\,(\{a, b, c, d, e, f\}) = 0 & & \end{array} \right\}.$$

$\mathcal{C}$ is in FREQSAT, because it is satisfiable by the following database:

$\mathcal{D}$ =

| TID | Items | TID | Items |
|---|---|---|---|
| 1 | $a, b, c, d, e$ | 5 | $a, b, c, \quad e, f$ |
| 2 | $a, b, c, d, e$ | 6 | $a, b, \quad d, e, f$ |
| 3 | $a, b, c, d, e$ | 7 | $a, \quad c, d, e, f$ |
| 4 | $a, b, c, d, \quad f$ | 8 | $b, c, d, e, f$ |

The satisfying database $\mathcal{D}$ has $ltrans(\mathcal{D})$ equal to 5, $ntrans(\mathcal{D})$ equal to 8, and $ndup(\mathcal{D})$ equal to 3. Thus, $(\mathcal{C}, 5) \in$ FREQSAT$\{ltrans\}$, and $(\mathcal{C}, 8, 3) \in$ FREQSAT$\{ntrans, ndup\}$.

On the other hand, however, $(\mathcal{C}, 4)$ is not in FREQSAT$\{ntrans\}$, and $(\mathcal{C}, 2)$ is not in FREQSAT$\{ndup\}$. The reason for this is because in every database, the following relations between the frequencies hold [10]:

$$\begin{aligned}
\mathrm{freq}(\{a,b,c\}) &\geq \frac{\mathrm{freq}(\{a,b\}) + \mathrm{freq}(\{a,c\}) + \mathrm{freq}(\{b,c\}) - 1}{2} \qquad (1)\\
&= 5/8\\
\mathrm{freq}(\{d,e,f\}) &\geq \frac{\mathrm{freq}(\{d,e\}) + \mathrm{freq}(\{d,f\}) + \mathrm{freq}(\{e,f\}) - 1}{2} \qquad (2)\\
&= 3/8\\
\mathrm{freq}(\{a,b,c,d,e\}) &\geq \mathrm{freq}(\{a,b,c\}) + \mathrm{freq}(\{d,e\}) - 1 \ = \ 3/8 \qquad (3)
\end{aligned}$$

Suppose that there exists a database that satisfies $\mathcal{C}$ and that has at most 4 transactions. Then, because of equations (1) and (2), and the fact that in the database, every frequency must be of the form $p/q$ with $q \leq 4$, the frequencies of $\{a,b,c\}$ and $\{d,e,f\}$ are respectively at least 3/4 and at least 1/2. Therefore, there must be an overlap of at least 1/4 of the transactions containing $\{a,b,c\}$ and the transactions containing $\{d,e,f\}$, such that the frequency of $\{a,b,c,d,e,f\}$ is at least 1/4. This is however in contradiction with the frequency constraint $\mathrm{freq}(\{a,b,c,d,e,f\}) = 0$ in $\mathcal{C}$. Thus, there cannot exist a database with at most 4 transactions that satisfies $\mathcal{C}$, and hence, $(\mathcal{C}, 4)$ is not in FREQSAT$\{ntrans\}$. This line of reasoning can be extended to show that the smallest database satisfying $\mathcal{C}$ needs to have at least 8 transactions.

For $(\mathcal{C}, 2)$ not in FREQSAT$\{ndup\}$, it suffices to notice that equation (3), together with the fact that every satisfying database has at least 8 transactions, proves that in every satisfying database there must be at least 3 transactions with the same set of items $\{a,b,c,d,e\}$. Indeed, since the set of items is $\{a,b,c,d,e,f\}$, and there are no transactions containing all items ($\mathrm{freq}(\{a,b,c,d,e,f\}) = 0$ is in $\mathcal{C}$), the only way to make (3) true is by having at least 3 out of 8 transactions of the form $(tid, \{a,b,c,d,e\})$.

## 3 FREQSAT Revisited

In this section, we revisit important and well-known properties of FREQSAT that will play an important role in the rest of this paper. These properties include the implementation of FREQSAT as a linear program and the fact that we can simulate constraints on the frequency of arbitrary Boolean expressions over items in FREQSAT, as well as a multiplication lemma that states that we can express that the frequency of an itemset $I$ is a multiple of the frequency of another itemset $J$.

### 3.1 FREQSAT as a Linear Program

**Theorem 1** **([6])** *Let $\mathcal{C} = \{\text{freq}\,(I_j) \in [l_j, u_j] \mid j = 1 \ldots m\}$ be a* FREQSAT*-instance, and let $\mathcal{I} = \bigcup_{j=1}^{m} I_j$. $\mathcal{C}$ is satisfiable if and only if the following linear program has a solution:*

> *Does there exist a $2^{|\mathcal{I}|}$-vector (the entries in the vector are indexed by the subsets $I$ of $\mathcal{I}$)*
>
> $$(X_{\{\}}, X_{\{i_1\}}, X_{\{i_1\}}, \ldots, X_{\{i_1,i_2\}}, \ldots, X_{\mathcal{I}}) \;\; ,$$
>
> *with all entries larger than or equal to $0$, such that the following system $\mathrm{LP}(\mathcal{C})$ of inequalities is satisfied?*

$$\begin{cases} \sum_{I \subseteq \mathcal{I}} X_I = 1 & \\ l_i \leq \sum_{I_i \subseteq I \subseteq \mathcal{I}} X_I \leq u_i & \forall i = 1, \ldots, m \end{cases}$$

Intuitively, the variables $X_I$ in the linear program $\mathrm{LP}(\mathcal{C})$ represent the number of transactions having exactly $I$ as set of items. Notice that the program can have an exponential number of variables in the size of the input. Furthermore, the additional constraints in the previous section can easily be incorporated into the program as follows: an upper bound $k$ on the length of the transactions (i.e., $\mathit{ltrans}(\mathcal{D}) \leq k$), translates to setting all variable $X_I$ with $|I| > k$ to zero (or, equivalently, removing these variables from the linear program.) The bound on the number of transactions, i.e., $\mathit{ntrans}(\mathcal{D}) \leq n$, and the bound on the number of duplicates, i.e., $\mathit{ndup}(\mathcal{D}) \leq d$, can be expressed into a similar linear *integer* program. These reductions, however, only provide very rude upper bounds on the complexity of the variants of FREQSAT. In the subsequent sections we will show more subtle reductions.

From Theorem 1, the following corollary is easy to prove, using standard techniques from linear programming:

**Corollary 1** **([6])** *If there exists a satisfying database for an instance $\mathcal{C}$ of the* FREQSAT*- problem, with $|\mathcal{C}| = m$ constraints, then there exists a database $\mathcal{D}$ such that number of transactions with different sets of items, $|\{J \mid (tid, J) \in \mathcal{D}\}|$, is at most $3m + 1$, and the total number of (non-unique) transactions is at most $2^{p(m)}$. (p is a fixed polynomial, independent of $\mathcal{C}$.)*

### 3.2 Constraints on the Frequency of Arbitrary Boolean Formulas

We now extend the `FREQSAT`-problem from frequency constraints over sets to frequency constraints over arbitrary Boolean formulas.

**Definition 1** Let $\mathcal{I}$ be a set of items. A Boolean formula over $\mathcal{I}$ is recursively defined as follows:

- for all $i \in \mathcal{I}$, $i$ is a Boolean formula over $\mathcal{I}$;
- if $\varphi$ and $\psi$ are Boolean formulas over $\mathcal{I}$, then also $(\neg\varphi)$, $(\varphi \wedge \psi)$ and $(\varphi \vee \psi)$ are Boolean formula's over $\mathcal{I}$.
- Nothing else is a Boolean formula over $\mathcal{I}$.

We recursively define if a transaction $(tid, J)$ satisfies $\varphi$ as follows:

- For all $i \in \mathcal{I}$, $(tid, J)$ satisfies $i$ if and only if $i \in J$;
- $(tid, J)$ satisfies $(\neg\varphi)$ if and only if it does not satisfy $\varphi$;
- $(tid, J)$ satisfies $(\varphi \wedge \psi)$ if it satisfies both $\varphi$ and $\psi$;
- $(tid, J)$ satisfies $(\varphi \vee \psi)$ if it satisfies at least one of $\varphi$ and $\psi$.

Let $\mathcal{D}$ be a database, and $\varphi$ be a formula over $\mathcal{I}$. $\operatorname{freq}(\varphi, \mathcal{D})$ is defined as follows:

$$\operatorname{freq}(\varphi, \mathcal{D}) \ := \ \frac{|\{(tid, J) \in \mathcal{D} \mid (tid, J) \text{ satisfies } \varphi\}|}{|\mathcal{D}|}$$

An *extended frequency constraint over* $\mathcal{I}$ is an expression $\operatorname{freq}(\varphi) \in [l, u]$, with $\varphi$ a Boolean formula over $\mathcal{I}$, and $0 \leq l, u \leq 1$ rational numbers. We say that database $\mathcal{D}$ satisfies $\operatorname{freq}(\varphi) \in [l, u]$, if $\operatorname{freq}(\varphi, \mathcal{D}) \in [l, u]$. We say that $\mathcal{D}$ satisfies a set of extended frequency constraints, if it satisfies every constraint in $\mathcal{C}$.

The *extended* `FREQSAT` *problem* is now defined as the problem of deciding if, given set of extended frequency constraints, there exists a database that satisfies this set.

In [7,12], it is proven that extended `FREQSAT`, i.e., deciding satisfiability of a set of extended frequency constraints, can be simulated in regular `FREQSAT`.

**Definition 2** Let $\mathcal{E}$ be an extended `FREQSAT`-problem, and let $\varphi$ be a Boolean formula over the items in $\mathcal{E}$. The set of entailed frequency for $\varphi$ given $\mathcal{E}$ is defined as the following set:

$$\mathrm{ENT}_{\varphi}(\mathcal{E}) \ := \ \{\operatorname{freq}(\varphi, \mathcal{D}) \ \mid \mathcal{D} \models \mathcal{E}\} \ .$$

**Theorem 2 ([12])** *There exists a polynomial reduction $R$ from extended* `FREQSAT` *to* `FREQSAT`*, such that for any extended* `FREQSAT`*-instance $\mathcal{E}$, $R(\mathcal{E})$ is satisfiable if and only if $\mathcal{E}$ is. Furthermore, $\mathrm{ENT}_{\varphi}(\mathcal{E}) = [l, u]$, if and only if $\mathrm{ENT}_{\{d, i_\varphi\}}(R(\mathcal{E})) = [l/2, u/2]$.*

*There exists a database $\mathcal{D}$ that satisfies $\mathcal{E}$ and the additional constraints $ltrans(\mathcal{D}) = lt$, $ntrans(\mathcal{D}) = nt$, and $ndup(\mathcal{D}) = nd$ if and only if there exists a database $\mathcal{D}'$ that satisfies $R(\mathcal{E})$, with $ltrans(\mathcal{D}') = c + lt$ ($c$ is a parameter that is polynomial in the size of $\mathcal{C}$), $ntrans(\mathcal{D}') = 2 \cdot nt$, and $ndup(\mathcal{D}') = nd$.*

**Proof sketch; full proof can be found in [12]** Let $\mathcal{E} = \{\text{freq}(\varphi_1) \in [l_1, u_1], \ldots, \text{freq}(\varphi_m) \in [l_m, u_m]\}$ be an extended `FREQSAT`-instance. For every subexpression $\sigma$ of the formulas $\varphi_1, \ldots, \varphi_m$ (also for the items), we introduce two new items, $t_\sigma$ and $f_\sigma$. $t_\sigma$ stands for "$\sigma$ is true," and $f_\sigma$ for "$\sigma$ is false." Let $T = (tid, J)$ be a transaction. $V_T$ denotes the truth assignment $V_T$ that assigns true to all items $i$ such that $t_i \in J$, and false to the other items.

$R(\mathcal{E})$ includes the following constraints enforcing that $t_\sigma$ is in a transaction $T$ if and only if the truth assignment $V_T$ makes $\sigma$ true. The main crux in this construction is that in a database that satisfies $R(\mathcal{E})$, only half of the transactions represent valid truth assignments. These transactions will be marked by the fact that they contain the item $d$, and the others contain item $\overline{d}$ (hence, $\overline{d}$ is in fact *not* $d$):

$$\text{freq}(\{d\}) = 0.5,\ \ \text{freq}\left(\{\overline{d}\}\right) = 0.5, \text{freq}\left(\{d, \overline{d}\}\right) = 0\ \ .$$

Furthermore, for every subexpression $\sigma$, $R(\mathcal{E})$ includes the following constraints:

$$\text{freq}(\{t_\sigma\}) = 0.5,\ \ \text{freq}(\{f_\sigma\}) = 0.5, \text{freq}(\{t_\sigma, f_\sigma\}) = 0\ \ \text{, and}$$

$$\begin{array}{lcll}
\text{if } \sigma & = & i: & \text{freq}(\{i, t_i\}) = 0.5, \text{freq}(\{i\}) = 0.5 \\
\text{if } \sigma & = & \sigma_1 \vee \sigma_2: & \text{freq}(\{d, t_\sigma, f_{\sigma_1}, f_{\sigma_2}\}) = 0, \text{freq}(\{d, f_\sigma, t_{\sigma_1}\}) = 0, \\
 & & & \text{freq}(\{d, f_\sigma, t_{\sigma_2}\}) = 0. \\
\text{if } \sigma & = & \sigma_1 \wedge \sigma_2: & \text{freq}(\{d, f_\sigma, t_{\sigma_1}, t_{\sigma_2}\}) = 0, \text{freq}(\{d, t_\sigma, f_{\sigma_1}\}) = 0, \\
 & & & \text{freq}(\{d, t_\sigma, f_{\sigma_2}\}) = 0. \\
\text{if } \sigma & = & \neg\sigma_1: & \text{freq}(\{d, t_\sigma, f_{\sigma_1}\}) = 0, \text{freq}(\{d, f_\sigma, t_{\sigma_1}\}) = 0.
\end{array}$$

In this way, we make sure that every transaction contains either $t_\sigma$, or $f_\sigma$, but not both. We use the transactions containing $\overline{d}$ to compensate the fact that we do not know how many trues and falses we need for $\sigma$. For example, for $a \vee \neg a$, half of the transactions will contain $\{d, t_{a\vee\neg a}\}$, and the other half contains $\{\overline{d}, f_{a\vee\neg a}\}$. Hence, even though only *half* of the transactions contain $t_{a\vee\neg a}$, *all* transactions representing *valid* truth assignments contain $t_{a\vee\neg a}$.

Within the $d$-part of a satisfying database, the trues and falses are consistent with each other. For example, a transaction representing a truth assignment cannot contain $t_{a\vee b}$, $f_a$, and $f_b$ at the same time. The consistency is enforced for every subexpression $\sigma$. Notice that the number of subexpressions of $\sigma$ is bounded by the number of symbols in it, and thus is polynomial.

Hence, for every subexpression $\sigma$, every transaction contains either $t_\sigma$ or $f_\sigma$, but not both. Every transaction $T$ that contains $d$ contains $t_\sigma$ if and only if $V_T(\sigma)$ is true. Furthermore, for all $j = 1 \ldots m$, $R(\mathcal{E})$ contains the constraint

$$\{\text{freq}\left(\{d, t_{\varphi_j}\}\right) \in [l/2, u/2]\}\ \ .$$

That is, we only measure the frequency of the formulas $\varphi$ within the fraction of the database with $d$, that is, the valid truth assignments. Since exactly half of the transactions contain $d$, the bounds of the intervals have to be divided by 2.

We denote the resulting `FREQSAT`-instance by $R(\mathcal{E})$. It is now true that $R(\mathcal{E})$ is satisfiable if and only if $\mathcal{E}$ is. Henceforth, we can reduce extended

FREQSAT to FREQSAT. It is easy to see that if $\mathcal{D}$ satisfies $\mathcal{E}$, we can construct a database $\mathcal{D}'$ that satisfies $R(\mathcal{E})$, by adding the items $d, \overline{d}, t_\sigma, f_\sigma$, etc. and vice versa. Furthermore, to every transaction of $\mathcal{D}$, the same number of items is added (1+the number of sub-formulas $\sigma$). This number depends polynomial on $\mathcal{E}$. The number of duplicates remains the same. □

### 3.3 Multiplication Lemma

In this section, we introduce the *Multiplication Lemma*. This lemma states that it is possible, for a given $n$, to write frequency constraints that express that the frequency of $\varphi$ is exactly $n$ times the frequency of $\psi$. Here we will only sketch the proof of the lemma. For a full proof we refer the reader to [12].

The definition of the construction $\mathcal{M}ULT_n(\varphi, \psi)$ is as follows. Let $\varphi$ be a Boolean formula, and let $m$ be an item, not in $\varphi$. The following frequency constraint, denoted $m = \varphi$, expresses that $m$ is in exact those transactions that satisfy $\varphi$:

$$m \equiv \varphi \;:=\; (\mathrm{freq}\,((m \wedge \neg\varphi) \vee (\neg m \wedge \varphi)) = 0)\ .$$

**Lemma 1** *$\mathcal{D}$ satisfies $m \equiv \varphi$ if and only if*

$$\{(tid, J) \in \mathcal{D} \mid m \in J\} \;=\; \{(tid, J) \in \mathcal{D} \mid (tid, J) \textit{ satisfies } \varphi\}\ .$$

*Proof* $\mathcal{D}$ satisfies $m \equiv \varphi$, if and only if, by definition,

$$\mathrm{freq}\,((m \wedge \neg\varphi) \vee (\neg m \wedge \varphi), \mathcal{D}) = 0\ .$$

This is equivalent with

$$\mathrm{freq}\,((m \wedge \neg\varphi), \mathcal{D}) = 0$$

and

$$\mathrm{freq}\,((\neg m \wedge \varphi), \mathcal{D}) = 0\ .$$

Therefore $\mathcal{D}$ satisfies $m \equiv \varphi$ if and only if there are no transactions in $\mathcal{D}$ that contain $m$ and do not satisfy $\varphi$, because such transactions would satisfy $(m \wedge \neg\varphi))$, and there are no transactions that satisfy $\varphi$ and do not contain $m$, because such transactions would satisfy $(\neg m \wedge \varphi)$. Hence, $m$ is in exactly those transactions that contain $\varphi$.

The main construction in the expression $\mathcal{M}ULT_n(\varphi, \psi)$ is the following expression $\kappa(\varphi_1, \varphi_2)$ ($\kappa$ stands for $\underline{c}$opy) that expresses that $\varphi_1$ and $\varphi_2$ have exactly the same frequency. Notice that this is different from $\varphi_1 \equiv \varphi_2$, because $\kappa$ does not require that the two expressions must be equivalent in the database; only the frequency must be the same.

$$\begin{aligned}\kappa(\varphi_1, \varphi_2) \;:=\; \{ & \mathrm{freq}\,(\varphi_1 \wedge \neg\varphi_2 \wedge r) = 0, \mathrm{freq}\,((\varphi_1 \wedge \neg\varphi_2) \vee r) = 0.5, \\ & \mathrm{freq}\,(\neg\varphi_1 \wedge \varphi_2 \wedge r) = 0, \mathrm{freq}\,((\neg\varphi_1 \wedge \varphi_2) \vee r) = 0.5\ \ \}\end{aligned}$$

**Lemma 2** *Let $\mathcal{C}$ be a set of extended frequency constraints that does not involve item $r$. There exists a database $\mathcal{D}$ that satisfies $\mathcal{C} \cup \kappa(\varphi_1, \varphi_2)$ if and only if there exists a database that satisfies $\mathcal{C}$, and in which* $\mathrm{freq}\,(\varphi_1, \mathcal{D}) = \mathrm{freq}\,(\varphi_2, \mathcal{D})$.

*Proof* If $\mathcal{D}$ satisfies $\kappa(\varphi_1, \varphi_2)$, then, by definition, $\mathrm{freq}\,(\varphi_1 \wedge \neg\varphi_2 \wedge r) = 0$, and $\mathrm{freq}\,(\neg\varphi_1 \wedge \varphi_2 \wedge r) = 0$. As such, $r$ can only be in those transactions that either contain both $\varphi_1$ and $\varphi_2$, or none of them. Hence,

$$\mathrm{freq}\,((\varphi_1 \wedge \neg\varphi_2) \vee r, \mathcal{D}) = \mathrm{freq}\,((\varphi_1 \wedge \neg\varphi_2), \mathcal{D}) + \mathrm{freq}\,(r, \mathcal{D}) \text{ and}$$
$$\mathrm{freq}\,((\neg\varphi_1 \wedge \varphi_2) \vee r, \mathcal{D}) = \mathrm{freq}\,((\neg\varphi_1 \wedge \varphi_2), \mathcal{D}) + \mathrm{freq}\,(r, \mathcal{D}) \quad .$$

From this it follows that:

$$\mathrm{freq}\,((\varphi_1 \wedge \neg\varphi_2), \mathcal{D}) = \mathrm{freq}\,((\neg\varphi_1 \wedge \varphi_2), \mathcal{D}) \quad ,$$

and therefore,

$$\begin{aligned} \mathrm{freq}\,(\varphi_1, \mathcal{D}) &= \mathrm{freq}\,(\varphi_1 \wedge \varphi_2, \mathcal{D}) + \mathrm{freq}\,(\varphi_1 \wedge \neg\varphi_2, \mathcal{D}) \\ &= \mathrm{freq}\,(\varphi_1 \wedge \varphi_2, \mathcal{D}) + \mathrm{freq}\,(\neg\varphi_1 \wedge \varphi_2, \mathcal{D}) \\ &= \mathrm{freq}\,(\varphi_2, \mathcal{D}) \quad . \end{aligned}$$

Hence, any database that satisfies $\mathcal{C}$ and $\kappa(\varphi_1, \varphi_2)$ also satisfies both $\mathcal{C}$ and $\mathrm{freq}\,(\varphi_1, \mathcal{D}) = \mathrm{freq}\,(\varphi_2, \mathcal{D})$.

For the other direction, let $\mathcal{D}$ be a database that satisfies $\mathcal{C}$, and in which the frequency of $\varphi_1$ equals that of $\varphi_2$. We will add to some transactions of this database the item $r$. As such, the resulting database $\mathcal{D}'$ will still satisfy $\mathcal{C}$. Because $\varphi_1$ and $\varphi_2$ have equal frequency,

$$\mathrm{freq}\,(\varphi_1 \wedge \neg\varphi_2, \mathcal{D}) = \mathrm{freq}\,(\neg\varphi_1 \wedge \varphi_2, \mathcal{D}) \quad .$$

Any database that results from adding the item $r$ to half of the transactions that satisfy neither $\varphi_1 \wedge \neg\varphi_2$, nor $\neg\varphi_1 \wedge \varphi_2$ satisfies $\kappa(\varphi_1, \varphi_2)$. □

In many constructions, we use more than one $\kappa$-expression at the same time. It is then understood that for each use of $\kappa$, a new item is substituted for $r$. That is, if we use the set of constraints $\mathcal{C} \cup \kappa(\varphi_1, \varphi_2) \cup \kappa(\varphi_3, \varphi_4)$, we implicitly assume that the item $r$ in $\kappa(\varphi_1, \varphi_2)$ differs from the one used in $\kappa(\varphi_3, \varphi_4)$.

Using the $\kappa$-construction, we can also express that the frequency of one expression is exactly twice the frequency of another expression. The following set of constraints $\delta(\varphi_1, \varphi_2)$ expresses that the frequency of $\varphi_2$ is exactly twice the frequency of $\varphi_1$ ($\delta$ stands for $\underline{\text{d}}$ouble):

$$\delta(\varphi_1, \varphi_2) \;:=\; \kappa(\varphi_1, k_1) \cup \kappa(\varphi_1, k_2) \cup \{\mathrm{freq}\,(k_1 \wedge k_2) = 0, \varphi_2 \equiv (k_1 \vee k_2)\} \quad .$$

**Lemma 3** *There exists a database that satisfies $\mathcal{C}$ and $\delta(\varphi_1, \varphi_2)$ if and only if there exists a database $\mathcal{D}$ that satisfies $\mathcal{C}$ and* $2 \cdot \mathrm{freq}\,(\varphi_1, \mathcal{D}) = \mathrm{freq}\,(\varphi_2, \mathcal{D})$.

*Proof* The lemma follows easily from Lemma 2 and Lemma 1: there exists a database that satisfies $\mathcal{C}$ and $\delta(\varphi_1, \varphi_2)$ if and only if there exists a database $\mathcal{D}$ that satisfies $\mathcal{C}$ and $\delta(\varphi_1, \varphi_2) \setminus \kappa(\varphi_1, k_1)$ and $\text{freq}\,(\varphi_1, \mathcal{D}) = \text{freq}\,(k_1, \mathcal{D})$. Let $f$ be $\text{freq}\,(\varphi_1, \mathcal{D})$. Hence, $\mathcal{D}$ satisfies $\mathcal{C} \cup \{\text{freq}\,(\varphi_1) = f, \text{freq}\,(k_1) = f\}$ and $\delta(\varphi_1, \varphi_2) \setminus \kappa(\varphi_1, k_1)$. By Lemma 2, such a database exists if and only if there exists a database $\mathcal{D}_2$ that satisfies $\mathcal{C} \cup \{\text{freq}\,(\varphi_1) = f, \text{freq}\,(k_1) = f\}$ and $\delta(\varphi_1, \varphi_2) \setminus \kappa(\varphi_1, k_1) \setminus \kappa(\varphi_1, k_2)$ and $\text{freq}\,(\varphi_1, \mathcal{D}) = \text{freq}\,(k_2, \mathcal{D}_2)$. Hence, $\mathcal{D}_2$ satisfies $\mathcal{C}$, $\text{freq}\,(\varphi_1) = \text{freq}\,(k_1) = f$, $\text{freq}\,(\varphi_1) = \text{freq}\,(k_2)$, $\text{freq}\,(k_1 \wedge k_2) = 0$, and $\varphi_2 \equiv (k_1 \vee k_2)$. Because of Lemma 1, this last constraint is equivalent to $\text{freq}\,(\varphi_2, \mathcal{D}_2) = \text{freq}\,(k_1 \vee k_2, \mathcal{D}_2)$. Since $\text{freq}\,(k_1 \wedge k_2) = 0$,

$$\text{freq}\,(\varphi_2,\ \mathcal{D}_2) = \text{freq}\,(k_1, \mathcal{D}_2) + \text{freq}\,(k_2, \mathcal{D}_2) = 2 \cdot \text{freq}\,(\varphi_1, \mathcal{D}_2) \quad ,$$

which proves the lemma. □

Obviously, we can also multiply by $3, 4, \ldots$, by making enough copies of $\varphi_1$ with $\kappa$, and setting $\varphi_2$ equal to $k_1 \vee k_2 \vee \ldots$ This method, however, has a big disadvantage: the formulas to multiply with $n$ would be exponentially large in the size of the representation of $n$. This can easily be solved though, by iterative doubling and adding: let $n$ be a positive integer with binary representation $b_\ell \ldots b_0$. That is, $n = \sum_{j=0}^{\ell} b_j 2^j$. The following set of constraints $\mathcal{M}ULT_n(\varphi_1, \varphi_2)$ expresses that the frequency of $\varphi_2$ is exactly $n$ times the frequency of $\varphi_1$ as follows:

$$\begin{aligned} \mathcal{M}ULT_n(\varphi_1, \varphi_2) \ := \ & \kappa(\varphi_1, b_0) \cup \delta(b_1, b_0) \cup \ldots \cup \delta(b_\ell, b_{\ell-1}) \\ & \cup \{\text{freq}\,(b_j \wedge b_j) = 0 \mid 0 \leq i < j \leq \ell, b_i = b_j = 1\} \\ & \cup \left\{ \varphi_2 = \bigvee \{b_j \mid 0 \leq j \leq \ell, b_j = 1\} \right\} \end{aligned}$$

**Lemma 4 (Multiplication Lemma [12])** *If $\mathcal{D}$ satisfies the set of frequency constraints*

$$\mathcal{M}ULT_{n_1}(\varphi_1^1, \varphi_2^1) \cup \ldots \cup \mathcal{M}ULT_{n_\ell}(\varphi_1^\ell, \varphi_2^\ell) \quad ,$$

*then for all $j = 1 \ldots \ell$,*

$$n_j \cdot \text{freq}\left(\varphi_1^j, \mathcal{D}\right) = \text{freq}\left(\varphi_2^j, \mathcal{D}\right) \quad .$$

**Proof sketch; full proof can be found in [12]** The proof of this lemma is very similar to the proof of Lemma 3. The different $\kappa$- and $\delta$-expressions can be eliminated one by one using a similar technique, using Lemma 2 and Lemma 3 repeatedly. □

In [12,7], it is shown that the multiplication lemma allows for expressing conditional probabilities, and, as such, association rules.

## 4 FREQSAT{ltrans}

In this section we show that knowing an upper bound on the length of the transactions does not affect the complexity of the `FREQSAT`-problem. Moreover, for any subset $C$ of $\{ntrans, ndup\}$, `FREQSAT`$(\{ltrans\} \cup C)$ is equivalent to `FREQSAT`$(C)$. As the original `FREQSAT`-problem does not impose any bound on the length of the transactions whatsoever, this result shows that adding the length of the transactions to the input of the problem does not add complexity to the problem; the frequency constraints are powerful enough to express this constraint. On the other hand, `FREQSAT` easily reduces to `FREQSAT`$\{ltrans\}$ by setting $ltrans$ equal to the number of items.

A straightforward approach to prove the equivalence would be to add constraints $\mathrm{freq}\,(i_1 \dots i_{k+1}) = 0$ to $\mathcal{C}$ for all $i_1, \dots i_{k+1} \in \mathcal{I}$, to enforce that all transactions have maximally length $k$. This reduction, however, can exponentially blow up the set of constraints. Indeed, the number of constraints added is as large as $\binom{|\mathcal{I}|}{k} = \mathcal{O}(|\mathcal{I}|^k)$. In this section we give a more involved reduction that does not have this disadvantage.

**Lemma 5** *Let $\mathcal{J}$ be a finite set of items, $n = |\mathcal{J}|$, $k$ is an integer with $1 \leq k \leq n$.*

*Let $\mathcal{D}$ be a database of transactions that satisfies the following collection $\mathcal{L}_k[\mathcal{J}]$ of frequency expressions:*

$$\forall j \in \mathcal{J} : \mathrm{freq}\,(\{j\}) = \binom{n-1}{k-1} \Big/ \binom{n}{k}$$

$$\forall j_1 \neq j_2 \in \mathcal{J} : \mathrm{freq}\,(\{j_1, j_2\}) = \binom{n-2}{k-2} \Big/ \binom{n}{k}$$

*Then, for all transactions $(tid, J)$ in $\mathcal{D}$, $|J \cap \mathcal{J}| = k$.*

*Proof* Let for all $i = 0 \dots n$,

$$\delta_i \ := \ |\{(tid, J) \in \mathcal{D} \mid |\mathcal{J} \cap J| = i\}| \ .$$

That is, $\delta_i$ is the number of transactions that contain exactly $i$ items of $\mathcal{J}$. It is clear that $\delta \ := \ |\mathcal{D}| \ = \ \sum_{i=0}^{n} \delta_i$. Let

$$S_i \ := \sum_{I \subseteq \mathcal{J}, |I|=i} \mathrm{supp}(I, \mathcal{D})$$

be the sum of the supports of all itemsets of size $i$ that are subset of $\mathcal{J}$. E.g., $S_1 = \sum_{j \in \mathcal{J}} \mathrm{freq}\,(j)$. (Recall that $\mathrm{freq}\,(I, \mathcal{D}) = \mathrm{supp}(I, \mathcal{D})/|\mathcal{D}|$.)

Because $\mathcal{D}$ satisfies $\mathcal{L}_k[\mathcal{J}]$,

$$S_0 = \delta,$$

$$S_1 = \delta n \binom{n-1}{k-1} \Big/ \binom{n}{k},$$

$$S_2 = \delta \binom{n}{2} \binom{n-2}{k-2} \Big/ \binom{n}{k}$$

From these equalities we directly derive the following relations between $S_0$, $S_1$, and $S_2$.

$$k(k-1)S_0 \; = \; (k-1)S_1 \; = \; 2S_2 \tag{4}$$

Another way to compute $S_0$, $S_1$, and $S_2$ is as follows. Every transaction of length $i$ has $i$ subsets of length 1, and $i(i-1)/2$ subsets of length 2. Therefore, we also obtain

$$S_0 \; = \; \sum_{i=0}^{n} \delta_i, \qquad S_1 \; = \; \sum_{i=0}^{n} i\delta_i, \qquad S_2 \; = \; \sum_{i=0}^{n} i(i-1)\delta_i/2$$

These last equalities in combination with (4), lead to

$$0 \; = \; kS_0 - S_1 = \sum_{i=0}^{n}(k-i)\delta_i \tag{5}$$

$$0 \; = \; (k-1)S_1 - 2S_2 = \sum_{i=0}^{n} i(k-i)\delta_i \tag{6}$$

From (5), it follows that

$$\sum_{i=0}^{k-1}(k-i)\delta_i \; = \; \sum_{i=k+1}^{n}(i-k)\delta_i \;\; , \tag{7}$$

and from (6), it follows that

$$\sum_{i=0}^{k-1} i(k-i)\delta_i \; = \; \sum_{i=k+1}^{n} i(i-k)\delta_i \;\; , \tag{8}$$

We now have:

$$\begin{aligned}
(k-1)\sum_{i=0}^{k-1}(k-i)\delta_i &\geq \sum_{i=0}^{k-1} i(k-i)\delta_i \\
&= \sum_{i=k+1}^{n} i(i-k)\delta_i && \text{using (8)} \\
&\geq (k+1)\sum_{i=k+1}^{n}(i-k)\delta_i \\
&= (k+1)\sum_{i=0}^{k-1}(k-i)\delta_i && \text{using (7)}
\end{aligned}$$

Thus,

$$\sum_{i=0}^{k-1}(k-i)\delta_i \; = \; \sum_{i=k+1}^{n}(i-k)\delta_i \; = \; 0 \;\; ,$$

and hence, for all $i \neq k$, $\delta_i = 0$. Therefore, $\delta_k = \delta$, and all transactions have exactly $k$ items in common with $\mathcal{J}$. □

| TID | Items |
|---|---|
| 1 | $a, b, d$ |
| 2 | $b, c, d$ |

$\longrightarrow$

| TID | Items |
|---|---|
| 1 | $a, b, c$ |
| 2 | $a, b, c$ |
| 3 | $a, b, d$ $\delta$ |
| 4 | $a, b, d$ |
| 5 | $a, c, d$ |
| 6 | $a, c, d$ |
| 7 | $b, c, d$ $\delta$ |
| 8 | $b, c, d$ |

**Fig. 1** Embedding of the database $\mathcal{D}$ with 2 transactions of constant length 3 in the database $\bigoplus_2 \mathcal{D}^3$.

The set of constraints $\mathcal{L}_k[\mathcal{J}]$ is satisfied by the following database $\mathcal{D}^k$: for every subset $J$ of $\mathcal{J}$ of length $k$ there is exactly one transaction $(tid, J)$. E.g., for $\mathcal{J} = \{a, b, c\}$, $\mathcal{D}^2$ denotes the database $\{(1, \{a, b\}), (2, \{a, c\}), (3, \{b, c\})\}$. Consider now an arbitrary database $\mathcal{D}$ with $n$ transactions, all of length $k$, over the set of items $\mathcal{J}$. This database can be *embedded* into the database $\bigoplus_n \mathcal{D}^k$ that consists of $n$ copies of $\mathcal{D}^k$; that is, $\bigoplus_n \mathcal{D}^k$ is the database consisting of $n$ copies of every transaction in $\mathcal{D}^k$. E.g., for $\mathcal{J} = \{a, b, c\}$, $\bigoplus_3 \mathcal{D}^2$ denotes the database $\{(1, \{a, b\}), (2, \{a, b\}), (3, \{a, b\}), (4, \{a, c\}), (5, \{a, c\}), (6, \{a, c\}), (7, \{b, c\}), (8, \{b, c\}), (9, \{b, c\})\}$. The $n$ transactions of $\bigoplus_n \mathcal{D}^k$ that form the embedding of $\mathcal{D}$ can be marked by adding a new item $\delta$, not in $\mathcal{J}$. Then, for every itemset $J \subset \mathcal{J}$, $\operatorname{freq}(J, \mathcal{D}) = \binom{n}{k} \operatorname{freq}\left(J \cup \{\delta\}, \bigoplus_n \mathcal{D}^k\right)$. See Fig. 2 for an example of this construction. The next definition and theorem are based on this observation.

**Definition 3** Let $\mathcal{C}$ be the following set of frequency constraints:

$$\mathcal{C} = \{\operatorname{freq}(I_1) \in [l_1, u_1], \ldots, \operatorname{freq}(I_m) \in [l_m, u_m]\} \ .$$

Let $\mathcal{J} = \bigcup_{j=1}^m I_j$, $n = |\mathcal{J}|$, $1 \leq k \leq n$, and let $\delta$ be an item not in $\mathcal{J}$.

$$\lambda_k(\mathcal{C}) \ := \ \left\{\operatorname{freq}(\{\delta\}) = 1 \Big/ \binom{n}{k}\right\} \ \cup \ \mathcal{L}_k[\mathcal{J}] \ \cup$$
$$\bigcup_{j=1}^{m} \left\{\operatorname{freq}(\{\delta\} \cup I_j) \in \left[l_j \Big/ \binom{n}{k}, u_j \Big/ \binom{n}{k}\right]\right\}$$

*Example 3* Consider the following set of frequency constraints

$$\mathcal{C} \ = \ \{\operatorname{freq}(\{a, b, d\}) = 0.5, \operatorname{freq}(\{b, c, d\}) \in [0.4, 0.6]\} \ ,$$

together with the constraint $ltrans = 3$. In Fig. 1 (left), a database has been given that satisfies these constraints. We show now that the set of frequency constraints, $\lambda_3(\mathcal{C})$, without any length constraints, is equivalent. A database that satisfies $\lambda_3(\mathcal{C})$ is also given in Fig. 1 (right).

The set of constraints is over the items $\mathcal{J} = \{a, b, c, d\}$. Hence, $\mathcal{L}_3[\mathcal{J}]$ consists of the constraints:

$$\begin{array}{llll}
\operatorname{freq}(\{a\}) = 3/4, & \operatorname{freq}(\{b\}) = 3/4, & \operatorname{freq}(\{c\}) = 3/4, & \operatorname{freq}(\{d\}) = 3/4 \\
\operatorname{freq}(\{a, b\}) = 1/2, & \operatorname{freq}(\{a, c\}) = 1/2, & \operatorname{freq}(\{a, d\}) = 1/2, & \\
\operatorname{freq}(\{b, c\}) = 1/2, & \operatorname{freq}(\{b, d\}) = 1/2, & \operatorname{freq}(\{c, d\}) = 1/2. &
\end{array}$$

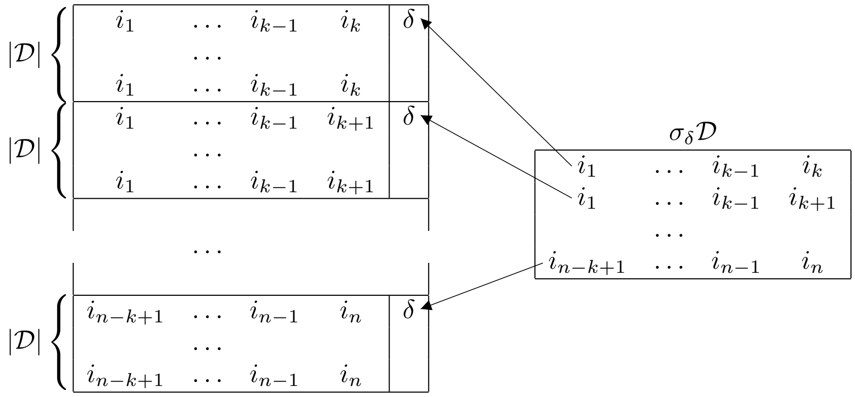

$|\mathcal{D}|$
$i_1 \quad \dots \quad i_{k-1} \quad i_k \quad \delta$
$\dots$
$i_1 \quad \dots \quad i_{k-1} \quad i_k$
$|\mathcal{D}|$
$i_1 \quad \dots \quad i_{k-1} \quad i_{k+1} \quad \delta$
$\dots$
$i_1 \quad \dots \quad i_{k-1} \quad i_{k+1}$
$\dots$
$|\mathcal{D}|$
$i_{n-k+1} \quad \dots \quad i_{n-1} \quad i_n \quad \delta$
$\dots$
$i_{n-k+1} \quad \dots \quad i_{n-1} \quad i_n$
$\sigma_\delta \mathcal{D}$
$i_1 \quad \dots \quad i_{k-1} \quad i_k$
$i_1 \quad \dots \quad i_{k-1} \quad i_{k+1}$
$\dots$
$i_{n-k+1} \quad \dots \quad i_{n-1} \quad i_n$

**Fig. 2** Construction in Theorem 3

These constraints enforce that every transaction contains exactly 3 of $a, b, c, d$. Indeed; these constraints give the exact frequency for all itemsets of size 1 and 2. Thus, these constraints allow us to determine the sum of the frequencies of all itemsets of size 1 ($S_1 = 3$), and of size 2 ($S_2 = 3$). Let now, similarly as in the proof of Lemma 5, $\delta_i$ be the fraction of transactions $(tid, J)$ with $|J \cap \{a, b, c, d\}| = i$. These $\delta_i$'s allow us to give the following alternative characterization of $S_1$ and $S_2$. Hence, we get the following equalities:

$$1 = \delta_0 + \delta_1 + \delta_2 + \delta_3 + \delta_4 \qquad (9)$$

$$S_1 = 3 = \delta_1 + 2 \cdot \delta_2 + 3 \cdot \delta_3 + 4 \cdot \delta_4 \qquad (10)$$

$$S_2 = 3 = \delta_2 + 3 \cdot \delta_3 + 6 \cdot \delta_4 \qquad (11)$$

We now get $6\delta_0 + 3\delta_1 + \delta_2 = 6 \cdot \text{eq}(9) - 3 \cdot \text{eq}(10) + \text{eq}(11) = 0$. As all $\delta_i$ are positive, $\delta_0 = \delta_1 = \delta_2 = 0$. If we combine this with 9 and 10, we easily get that $\delta_4 = 0$, and $\delta_3 = 1$. Hence, all transactions must have exactly 3 of $\{a, b, c, d\}$. The proof of the next theorem comes down to a similar reasoning in a more general setting.

Furthermore, $\lambda_3(\mathcal{C})$ contains the constraints

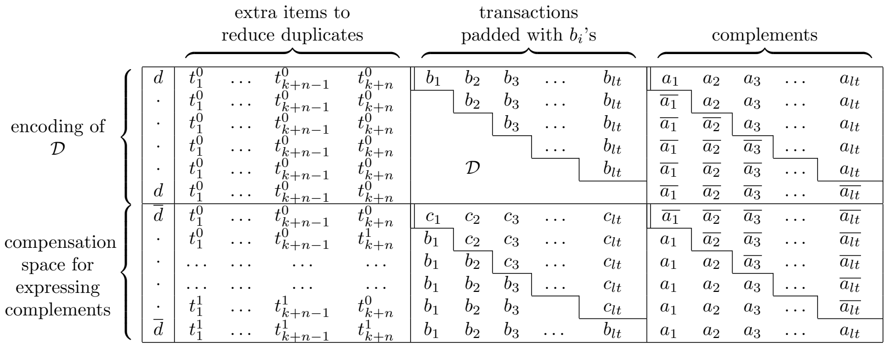
$$\text{freq}(\{\delta\}) = 1/4, \text{freq}(\{a, b, d, \delta\}) = 1/8, \text{ and } \text{freq}(\{b, c, d, \delta\}) \in [1/10, 3/20]$$

As the database satisfying $\mathcal{C}$ and the length constraint given in Fig. 1 (left) contains two transactions, and the transaction length is 3, it can be embedded into $\bigoplus_2 \mathcal{D}^3$. The transactions making up the embedding are marked with the item $\delta$. The database in Fig. 1 (right) is an example of a database satisfying $\lambda_3(\mathcal{C})$.

**Theorem 3** *A set $\mathcal{C}$ of frequency constraints is satisfiable by a database with all transactions of length* equal to $k$ *if and only if $\lambda_k(\mathcal{C})$ is in* `FREQSAT`.

*If $\mathcal{D}$ satisfies $\lambda_k(\mathcal{C})$, then the following database satisfies $\mathcal{C}$:*

$$\sigma_\delta \mathcal{D} \ := \ \{(tid, J) \mid (tid, J \cup \{\delta\}) \in \mathcal{D}\}$$



| | | extra items to reduce duplicates | | | | transactions padded with $b_i$'s | | | | | complements | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| encoding of $\mathcal{D}$ | $d$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^0$ | $b_1$ | $b_2$ | $b_3$ | $\dots$ | $b_{lt}$ | $a_1$ | $a_2$ | $a_3$ | $\dots$ | $a_{lt}$ |
| | $\cdot$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^0$ | | $b_2$ | $b_3$ | $\dots$ | $b_{lt}$ | $\overline{a_1}$ | $a_2$ | $a_3$ | $\dots$ | $a_{lt}$ |
| | $\cdot$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^0$ | | | $b_3$ | $\dots$ | $b_{lt}$ | $\overline{a_1}$ | $\overline{a_2}$ | $a_3$ | $\dots$ | $a_{lt}$ |
| | $\cdot$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^0$ | | | | $\dots$ | $b_{lt}$ | $\overline{a_1}$ | $\overline{a_2}$ | $\overline{a_3}$ | $\dots$ | $a_{lt}$ |
| | $\cdot$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^0$ | | $\mathcal{D}$ | | | $b_{lt}$ | $\overline{a_1}$ | $\overline{a_2}$ | $\overline{a_3}$ | $\dots$ | $a_{lt}$ |
| | $d$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^0$ | | | | | | $\overline{a_1}$ | $\overline{a_2}$ | $\overline{a_3}$ | $\dots$ | $\overline{a_{lt}}$ |
| compensation space for expressing complements | $\overline{d}$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^0$ | $c_1$ | $c_2$ | $c_3$ | $\dots$ | $c_{lt}$ | $\overline{a_1}$ | $\overline{a_2}$ | $\overline{a_3}$ | $\dots$ | $\overline{a_{lt}}$ |
| | $\cdot$ | $t_1^0$ | $\dots$ | $t_{k+n-1}^0$ | $t_{k+n}^1$ | $b_1$ | $c_2$ | $c_3$ | $\dots$ | $c_{lt}$ | $a_1$ | $\overline{a_2}$ | $\overline{a_3}$ | $\dots$ | $\overline{a_{lt}}$ |
| | $\cdot$ | $\dots$ | $\dots$ | $\dots$ | $\dots$ | $b_1$ | $b_2$ | $c_3$ | $\dots$ | $c_{lt}$ | $a_1$ | $a_2$ | $\overline{a_3}$ | $\dots$ | $\overline{a_{lt}}$ |
| | $\cdot$ | $\dots$ | $\dots$ | $\dots$ | $\dots$ | $b_1$ | $b_2$ | $b_3$ | $\dots$ | $c_{lt}$ | $a_1$ | $a_2$ | $a_3$ | $\dots$ | $\overline{a_{lt}}$ |
| | $\cdot$ | $t_1^1$ | $\dots$ | $t_{k+n-1}^1$ | $t_{k+n}^0$ | $b_1$ | $b_2$ | $b_3$ | | $c_{lt}$ | $a_1$ | $a_2$ | $a_3$ | $\dots$ | $\overline{a_{lt}}$ |
| | $\overline{d}$ | $t_1^1$ | $\dots$ | $t_{k+n-1}^1$ | $t_{k+n}^1$ | $b_1$ | $b_2$ | $b_3$ | $\dots$ | $b_{lt}$ | $a_1$ | $a_2$ | $a_3$ | $\dots$ | $a_{lt}$ |

**Fig. 3** Illustration of the construction in the proof of $\texttt{FREQSAT}\{ltrans, ndup\} \leq \texttt{FREQSAT}\{ndup\}$.

*Proof* Let $\mathcal{C}$ be $\{\text{freq}\,(I_1) \in [l_1, u_1], \dots, \text{freq}\,(I_m) \in [l_m, u_m]\}$, and let $\mathcal{J}$ be the set of items $\bigcup_{j=1}^m I_j$.

*If*: Suppose that $\mathcal{D}$ satisfies $\lambda_k(\mathcal{C})$. The number of transactions in $\sigma_\delta\mathcal{D}$ is $|\mathcal{D}|/\binom{n}{k}$, and the number of transactions in $\sigma_\delta\mathcal{D}$ that contain $I_j$ lies between $|\mathcal{D}|l_j/\binom{n}{k}$ and $|\mathcal{D}|u_j/\binom{n}{k}$. Hence, its frequency in $\sigma_\delta\mathcal{D}$ is between $l_j$ and $u_j$. Therefore, $\sigma_\delta\mathcal{D}$ satisfies $\mathcal{C}$. Furthermore, because $\mathcal{D}$ satisfies $\mathcal{L}_k[\mathcal{J}]$, Lemma 5 states that every transaction in $\mathcal{D}$ contains exactly $k$ items from $\mathcal{J}$. Henceforth, all transactions in $\sigma_\delta\mathcal{D}$ have length $k$ ($\delta$ is not in $\mathcal{J}$).

*Only If*: Suppose that $\mathcal{D}$ is a database with all transactions of length $k$ that satisfies $\mathcal{C}$. We construct a database $\mathcal{D}'$ with $|\mathcal{D}|\binom{n}{k}$ transactions that satisfies $\lambda_k(\mathcal{C})$. For every subset $I$ of size $k$ of $\mathcal{J}$, there will be $\text{supp}(I, \mathcal{D})$ transactions $(tid, I \cup \{\delta\})$, and $|\mathcal{D}| - \text{supp}(I, \mathcal{D})$ transactions $(tid, I)$ in $\mathcal{D}'$. Thus, the absolute number of transactions containing $I \cup \{\delta\}$ in $\mathcal{D}'$ is the same as the frequency of $I$ in $\mathcal{D}$, but $|\mathcal{D}'|$ is $\binom{n}{k}$ times larger than $|\mathcal{D}|$. Hence, the frequency of $I_j \cup \{\delta\}$ in $\mathcal{D}'$ equals $\text{freq}\,(I_j, \mathcal{D})\,/\binom{n}{k}$. The frequency of $\delta$ is $|\mathcal{D}|/(|\mathcal{D}|\binom{n}{k}) = 1/\binom{n}{k}$. It also holds that $\mathcal{D}'$ satisfies $\mathcal{L}_k[\mathcal{J}]$; the projection of $\mathcal{D}'$ on $\mathcal{J}$ consists of $|\mathcal{D}|$ copies of $I$, for every subset $I$ of $\mathcal{J}$ of size $k$. □

**Corollary 2** *For all $C \subseteq \{ntrans, ndup\}$,*
$\texttt{FREQSAT}(C \cup \{ltrans\}) \equiv \texttt{FREQSAT}(C)$ .
*($\equiv$ denotes equivalence under logspace reductions.)*

*Proof* Let $\mathcal{C} = \{\text{freq}\,(I_j) \in [l_j, u_j] \mid j = 1 \dots m\}$ be a set of frequency constraints, and let $\mathcal{I} = \bigcup_{j=1}^m I_j$, $|\mathcal{I}| = n$.

$\texttt{FREQSAT}(C \cup \{ltrans\}) \geq \texttt{FREQSAT}(C)$: the number of items is an upper bound for transaction length, and hence, $(\mathcal{C}, v_1, \dots, v_k)$ is a satisfiable instance of the problem $\texttt{FREQSAT}\{c_1, \dots, c_k\}$ if and only if $(\mathcal{C}, n, v_1, \dots, v_k)$ is in $\texttt{FREQSAT}\{ltrans, c_1, \dots, c_k\}$.

$\texttt{FREQSAT}(C \cup \{ltrans\}) \leq \texttt{FREQSAT}(C)$: the proof of this direction is based on Lemma 5. Let $\mathcal{C}$ be a set of frequency constraints. Assume that $\mathcal{C}$ is satisfiable by a database of transactions $\mathcal{D}$, and $\mathcal{D}$ has a maximal transaction size of $lt$. Since Lemma 5 only holds for databases of length *exactly* $lt$, we need to add extra items to compensate for transactions that are too short. When $C$ includes *ndup*, some care is required to avoid that the new items change the number of duplicates.

**Construction in the presence of *ndup*.** Assume that $\mathcal{C}$ is satisfiable by a database of transactions $\mathcal{D}$, and $\mathcal{D}$ has a maximal transaction length $lt$, and a maximal number of duplicates $nd$. Since Lemma 5 only holds for databases of length *exactly lt*, we need to add extra items to compensate for transactions that are too short. We denote the result of the construction $\mathcal{C}'$.

As a first step, we introduce two new items $d$, and $\overline{d}$. The transactions that contain $d$ will be the ones that encode $\mathcal{D}$, the ones with $\overline{d}$ the complement. The following constraints are introduced:

$$\operatorname{freq}(\{d\}) = 0.5, \qquad \operatorname{freq}\left(\{\overline{d}\}\right) = 0.5 \qquad \operatorname{freq}\left(\{d, \overline{d}\}\right) = 0$$

Thus, half of the transactions will embed the satisfying database, while the other half will play an important role in the reduction. The frequency constraints in $\mathcal{C}$ thus become:

$$\{\operatorname{freq}(I_1 \cup \{d\}) \in [l_1/2, u_1/2], \ldots, \operatorname{freq}(I_m \cup \{d\}) \in [l_m/2, u_m/2]\} \ .$$

We add auxiliary items $b_1, \ldots, b_{lt}$ to pad transactions that are too short. However, adding auxiliary items potentially decreases the number of duplicates. Indeed, $I \cup \{b_1\}$ is no longer a duplicate of $I \cup \{b_2\}$. Therefore, we will require that all transactions can be padded in only one way. We do this by requiring that every $b_j$ can only occur together with $b_{j+1}$. Henceforth, a transaction that is $k$ items short, can only be padded by adding $b_{lt-k+1}, \ldots, b_{lt}$. For this purpose, we also introduce the items $a_1, \ldots, a_{lt}, \overline{a_1}, \ldots, \overline{a_{lt}}$ and the following constraints:

$$\begin{array}{lll} \operatorname{freq}(\{a_j\}) = 0.5, & \operatorname{freq}(\{\overline{a_j}\}) = 0.5, & \operatorname{freq}(\{a_j, \overline{a_j}\}) = 0 \\ \operatorname{freq}(\{b_j\}) = 0.5, & \operatorname{freq}(\{a_j, b_j\}) = 0.5, & \operatorname{freq}(\{d, b_j, \overline{a_{j+1}}\}) = 0 \end{array}$$

Thus, the new items $a_j$ occur in exactly those transactions having $b_j$, and $\overline{a_j}$ in exactly those that do not have $b_j$. The constraint $\operatorname{freq}(\{d, b_j, \overline{a_{j+1}}\}) = 0$ thus ensures that within the $d$-part, no $b_j$ is in a transaction without $a_{j+1}$ which are exactly the transactions without $b_{j+1}$ as well. This rather cumbersome construction with the $a_j$'s and $\overline{a_j}$'s is necessary because it allows for expressing the complement of $b_j$ while adding the *exact same* number of items to every transaction.

There is still a problem, however: the transactions that contain $\overline{d}$, can be too short, and there can be too many duplicates in the part of the database having $\overline{d}$. To solve the problem of being too short, we add extra items $c_1, \ldots, c_{lt}$ that can only appear in the $\overline{d}$-part:

$$\operatorname{freq}(\{c_1, \ldots, c_{lt}\}) \in [0, 1], \qquad \text{and for all } i = 1, \ldots lt : \operatorname{freq}(\{c_i, d\}) = 0.$$

By adding these items, we can be sure that in the $\overline{d}$-part, every transaction has the same length $lt$.

We still need to solve the problem of too many duplicates in the $\overline{d}$-part. Let $2^{k-1} < nd \leq 2^k$. Notice that the number of transactions of $\mathcal{D}$ is bounded by $2^{k+n}$, as there are $2^n$ different transactions which can be duplicated at most $2^k$ times. We add $2(k+n)$ new items $t_1^0, \dots, t_{k+n}^0$, $t_1^1, \dots, t_{k+n}^1$. These new items can reduce the number of duplicates in the $\overline{d}$-part to 1; to every transaction we can, e.g., add either $t_1^0$ or $t_1^1$ and $t_2^0$ or $t_2^1$, and $t_3^0$ or $t_3^1$, etc. The number of such possibilities is $2^k$. We however do not want to reduce the duplicates in the $d$-part, and therefore, we add for all $j = 1 \dots k+n$ the following constraint: freq $\left(\{d, t_j^1\}\right) = 0$, thus only allowing $t_j^0$, $j = 1 \dots k+n$ to be added to a transaction in the $d$-part.

$\mathcal{C}$ is satisfiable by a database $\mathcal{D}$ with $ntrans(\mathcal{D}) \leq nt$, $ndup(\mathcal{D}) \leq nd$, and $ltrans(\mathcal{D}) \leq lt$ if and only if $\mathcal{C}'$ is satisfiable by a database of transactions $\mathcal{D}'$ with $ntrans(\mathcal{D}') \leq 2 \cdot nt$, $ndup(\mathcal{D}') \leq nd$, and all transactions of length exactly $1 + 2lt + k + n$. Indeed, suppose $\mathcal{D}$ satisfies $\mathcal{C}$, then items and transactions can be added to $\mathcal{D}$ as depicted in Fig. 3, to get a database that satisfies $\mathcal{C}'$ and the additional constraints $ntrans(\mathcal{D}') \leq 2 \cdot nt$, $ndup(\mathcal{D}') \leq nd$, and all transactions of length exactly $1 + 2lt + k + n$. This addition of items and transactions is as follows. We assume without loss of generality, that the transactions $(tid, J)$ of $\mathcal{D}$ all have $tid \in \{1, \dots, |\mathcal{D}|\}$. Let $(tid, J)$ be a transaction of $\mathcal{D}$. Let $e_1 \dots e_{k+n}$ be the binary encoding of $tid$. $\mathcal{D}'$ will contain for every such transaction $(tid, J)$, two transactions $(tid, J')$ and $(tid + |\mathcal{D}|, \overline{J'})$, with:

$$\begin{aligned}
J &= J \cup \{d, t_1^0, \dots, t_{k+n}^0\} \cup \{b_i \mid i > |J|\} \cup \{a_i \mid i > |J|\} \cup \{\overline{a_i} \mid i \leq |J|\} \\
\overline{J'} &= \{\overline{d}\} \cup \{t_0^i \mid e_i = 0, i = 1 \dots n+k\} \cup \{t_1^i \mid e_i = 1, i = 1 \dots n+k\} \\
&\quad \cup \{c_i \mid i > |J|\} \cup \{\overline{b_i} \mid i \leq |J|\} \cup \{\overline{a_i} \mid i > |J|\} \cup \{a_i \mid i \leq |J|\}
\end{aligned}$$

On the other hand, from every database that satisfies $\mathcal{C}'$, $ntrans(\mathcal{D}') \leq 2 \cdot nt$, $ndup(\mathcal{D}') \leq nd$, and all transactions of length exactly $1 + 2lt + k + n$, we can extract a database that satisfies $\mathcal{C}$ and $ntrans(\mathcal{D}) \leq nt$, $ndup(\mathcal{D}) \leq nd$, and $ltrans(\mathcal{D}) \leq lt$ as follows: let $\mathcal{D}' = \{(tid, J) \in \mathcal{D}' \mid d \in J\}$.

Now, because of Theorem 3, and the proof of this theorem, $\mathcal{C}'$ in turn has a solution $\mathcal{D}'$ with $ntrans(\mathcal{D}') \leq 2 \cdot nt$, $ndup(\mathcal{D}') \leq nd$, and all transactions of length exactly $1 + 2lt + k + n$ if and only if $\lambda_{1+2lt+k+n}(\mathcal{C}')$ has a solution $\mathcal{D}''$ with $ntrans(\mathcal{D}'') = \binom{2n+k+2lt+1}{1+2lt+k+n} nt$, and $ndup(\mathcal{D}'') = nd$. □

### 4.1 Fixed Parameter Variants

A natural question that arises now is what happens if the maximal transaction length is not given as part of the input, but is a fixed parameter instead. This case is handled by the next theorem.

**Theorem 4** *For fixed $k$, the problem* `FREQSAT`$\{ltrans = k\}$ *can be solved in deterministic polynomial time. Furthermore, for all $C \subseteq \{ntrans, ndup\}$,* `FREQSAT`$(C \cup \{ltrans = k\}) \leq$ `FREQSAT`$(C)$ .

*Proof* Let $\mathcal{I}$ be the set of all items in a `FREQSAT`-instance $\mathcal{C}$, and let $|\mathcal{I}| = n$. If the length of the transactions is fixed to $k$, there are maximally $\sum_{i=0}^{k} \binom{n}{k} = \mathcal{O}(n^{k+1})$ different transactions. Therefore, using Theorem 1, we can rewrite the existence of a database that satisfies $\mathcal{C}$, as the following linear program of polynomial size in $n$:

$$\begin{aligned}
x_J &\geq 0 && \forall J \subseteq \mathcal{I},\ \text{with } |J| \leq k \\
\sum_{J \subseteq \mathcal{I},\ \text{with } |J| \leq k} x_J &= 1 && \\
\sum_{J \subseteq I} x_J &\geq l && \text{for all } (\text{freq}\,(I) \in [l,u]) \in \mathcal{C} \\
\sum_{J \subseteq I} x_J &\leq u && \text{for all } (\text{freq}\,(I) \in [l,u]) \in \mathcal{C}
\end{aligned}$$

As linear programming can be performed in polynomial time, and the size of the system is polynomial in $\mathcal{C}$, `FREQSAT` with a constant maximal transaction length can be solved in polynomial time.

The second part of the proof, $\texttt{FREQSAT}(C \cup \{ltrans = k\}) \leq \texttt{FREQSAT}(C)$ follows directly from the proof of Corollary 2, as this direction nowhere requires that *ltrans* is not a fixed parameter. □

Obviously, the direction $\texttt{FREQSAT}(C \cup \{ltrans = k\}) \geq \texttt{FREQSAT}(C)$ does not hold in general; for $C = \{\}$, $\texttt{FREQSAT}(C \cup \{ltrans = k\})$ can be solved in polynomial time, while $\texttt{FREQSAT}(C)$ is **NP**-complete. Furthermore, the case $C = \{ntrans\}$ is handled in Theorem 8. Notice also that the reduction given in Theorem 4 does not imply that $\texttt{FREQSAT}\{ltrans = k\}$ is fixed-parameter tractable, as the size of the linear program that is constructed exponentially depends on $k$. The fixed-parameter complexity of $\texttt{FREQSAT}\{ltrans = k\}$ is hence still open.

## 5 FREQSAT{ntrans}

In the last section we saw that knowing a maximal transaction length does not add expressive power to `FREQSAT`. For the number of transactions *ntrans*, the question whether it adds to the *complexity* is open. In this section we give some indications of $\texttt{FREQSAT}\{ntrans\}$ being more complex. The crux here is that, unlike for `FREQSAT` and $\texttt{FREQSAT}\{ltrans\}$, $\texttt{FREQSAT}\{ntrans\}$ cannot be solved using linear programming. Instead, linear *integer* programming should be used, having far less attractive mathematical properties and complexity than linear programming; linear programming can be solved in deterministic polynomial time, where integer linear programming is complete for **NP**.

We show that `FREQSAT` reduces to $\texttt{FREQSAT}\{ntrans\}$, and that the problem $\texttt{FREQSAT}\{ntrans\}$ is equivalent to the *Intersection Pattern Problem* (`IP`) [17] w.r.t. computational complexity. `IP` is the following problem: *given an $n \times n$ matrix $C$ with integer entries, do there exist sets $S_1, \ldots, S_n$ such that $|S_i \cap S_j| = C[i,j]$?* If such sets exist, $C$ is called an *intersection pattern*. In [17], it is claimed that `IP` is **NP**-complete. However, the inclusion in **NP** has only been proven for the case the entries in the matrix $C$ are bounded by a fixed constant [14]. For the general problem, the inclusion of `IP` in **NP** is still open.

For the entailment, we show that, unlike for FREQSAT, the set $\mathrm{ENT}_I^n(\mathcal{C}) = \{\mathrm{freq}\,(I,\mathcal{D}) \mid \mathcal{D} \models \mathcal{C}, |\mathcal{D}| \leq n\}$, is no longer an interval of the rational numbers. This is of course hardly surprising, since the frequencies in a database with at most $n$ transactions can only be of the form $\frac{p}{q}$, with $0 \leq p \leq n$, and $1 \leq q \leq n$. Therefore, it would be more fair to ask the following question: if $\frac{p_1}{q}, \frac{p_2}{q} \in \mathrm{ENT}_I^n(\mathcal{C})$, is it true that for every $p$ with $p_1 \leq p \leq p_2$, also $\frac{p}{q} \in \mathrm{ENT}_I^n(\mathcal{C})$? We will answer this question negatively. Moreover, given an arbitrary set $R = \{r_1, \ldots, r_k\}$ of rational numbers, we will show that there exists a set of constraints $\mathcal{C}$, an itemset $I$, and a positive integer $n$, all having description size polynomial in the size of $R$, such that $\mathrm{ENT}_I^n(\mathcal{C}) = R$. This shows that the properties of the FREQSAT-problem change fundamentally if we restrict the number of transactions.

Another illustration that we cannot just assume that *ntrans* is a trivial extension without repercussions on complexity, is the fact that for any fixed constant $k$, FREQSAT$\{ltrans = k\}$, is decidable in polynomial time, as we saw in last section, while FREQSAT$\{ltrans = 3, ntrans\}$ is already **NP**-hard!

Finally we show that if the bound on the number of transactions is a fixed constant $c$, the problem is **NP**-complete if $c \geq 2$. We denote the problem: does there exist a database $\mathcal{D}$ with at most 2 transactions that satisfies a given set of constraints, FREQSAT$\{ntrans = 2\}$.

## 5.1 Relation with FREQSAT

**Theorem 5** FREQSAT $\leq$ FREQSAT$\{ntrans\}$

*Proof* Given a FREQSAT-problem $\mathcal{C}$, by Corollary 1, there exists an upper bound $n_{\mathcal{C}}$ (with representation size polynomial in $\mathcal{C}$), such that if $\mathcal{C}$ is satisfiable, then $\mathcal{C}$ is satisfiable by a database of size maximally $n_{\mathcal{C}}$. Hence, $\mathcal{C}$ is in FREQSAT if and only if $(\mathcal{C}, n_{\mathcal{C}})$ is in FREQSAT$\{ntrans\}$. □

## 5.2 INTERSECTION PATTERN

We show that IP is equivalent to FREQSAT$\{ntrans\}$, in the sense that on the one hand, IP is logspace reducible to FREQSAT$\{ntrans\}$, and on the other hand, FREQSAT$\{ntrans\}$ is non-deterministic polynomial many-one reducible to IP. That is, there exists a non-deterministic polynomial-time procedure $R$, such that for every instance $\mathcal{C}$ of FREQSAT$\{ntrans\}$, there is *at least* one execution path of $R$ on input $\mathcal{C}$ that results in a satisfiable instance of IP if and only if $\mathcal{C}$ is satisfiable. Such a reduction shows that if IP is in **NP**, then FREQSAT$\{ntrans\}$ is as well. Indeed; the concatenation of a non-deterministic polynomial time many-one reduction $R$ with a non-deterministic polynomial time decision procedure is again a non-deterministic polynomial time decision procedure.

### *5.2.1 Reduction From* `IP` *to FREQSAT{ntrans}*

Intuitively, the `IP` problem can be seen as a special case of `FREQSAT`$\{ntrans\}$; the set of elements $S_i$ in a realization of an intersection pattern can be simulated by the set of transactions that contain the dedicated element $s_i$. Because the given numbers of elements in the intersections are absolute cardinalities, *ntrans* is needed. The following definition and theorem confirm the correctness of this intuition.

**Definition 4** Let $C$ be an $n \times n$ matrix over the positive integers. $N(C)$ denotes the number $\sum_{1 \le i \le n} C[i,i]$. $\mathcal{C}[C]$ denotes the following instance of the `FREQSAT`$\{ntrans\}$-problem over the set of items $\{e, s_1, \ldots, s_n\}$:

$$\begin{aligned}\mathcal{C}[C] \ := \ & \{\text{freq}\,(e) = 1/N(C)\} \\ & \qquad \cup \{\text{freq}\,(\{s_i, s_j\}) = C[i,j]/N(C) \mid 1 \le i \le j \le n\}\end{aligned}$$

*Example 4* Let $C = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$. The following sets form a realization of $C$: $S_1 = \{1,2\}, S_2 = \{1\}$. The corresponding `FREQSAT`$\{ntrans\}$-problem is $(\mathcal{C}, 3)$ with

$$\mathcal{C} = \left\{ \begin{array}{l} \text{freq}\,(\{e\}) = 1/3, \ \ \text{freq}\,(\{s_1\}) = 2/3, \\ \text{freq}\,(\{s_2\}) = 1/3, \text{freq}\,(\{s_1, s_2\}) = 1/3 \end{array} \right\} .$$

The satisfying database of $\mathcal{C}$ that corresponds with the realization $S_1 = \{1,2\}, S_2 = \{1\}$ is:

$\mathcal{D}$ =

| `TID` | `Items` |
|---|---|
| 1 | $s_1, s_2$ |
| 2 | $s_1$ |
| 3 | $e$ |

**Theorem 6** *$C$ is an intersection pattern if and only if $\mathcal{C}[C]$ is satisfiable by a database with at most $N(C)$ transactions.*

*Proof* If an $n \times n$ matrix $C$ is an intersection pattern, then there exists a realization $S_1, \ldots, S_n$ of $C$ such that $|\bigcup_{i=1}^n S_i| \le \sum_{1 \le i \le n} |S_i| = \sum_{1 \le i \le n} C[i,i] = N(C)$. The constraint $\text{freq}\,(\{e\}) = 1/N(C)$ makes sure that every satisfying database of $\mathcal{C}[C]$ has exactly $N(C)$ transactions. If $\mathcal{D}$ is a satisfying database, then the sets $S_i = \{tid \mid (tid, J) \in \mathcal{D}, s_i \in J\}$, for $i = 1 \ldots n$ form a realization of $C$, and vice versa. □

### *5.2.2 Reduction From FREQSAT{ntrans} to* `IP`

We give a non-deterministic polynomial many-one reduction from the problem `FREQSAT`$\{ntrans\}$ to `IP`. Such a reduction shows that if `IP` is in **NP**, then so is `FREQSAT`$\{ntrans\}$.

Let $(\mathcal{C}, nt)$ be an instance of the `FREQSAT`$\{ntrans\}$ problem. The first step in the reduction is to (non-deterministic, many-one) reduce $\mathcal{C}$ to a set of frequency constraints $\mathcal{C}'$, in which every frequency constraint is of the form $\text{freq}\,(I) = f$, with $|I|$ at most 2. Before going into the technical details, we illustrate this step with an example.

*Example 5* The first step in the reduction is to reduce the cardinalities of the sets in the input to 2. For example; a constraint $\text{freq}(\{a,b,c,d\}) \in [0.1, 0.3]$ in $\mathcal{C}$, must be replaced with a number of constraints that only involve itemsets of cardinality at most 2. This would be easy if we knew the frequencies of the prefixes of $\{a,b,c,d\}$. Indeed; suppose that we know that $\text{freq}(\{a\}) = 0.5$, $\text{freq}(\{a,b\}) = 0.3$, $\text{freq}(\{a,b,c\}) = 0.2$, and $\text{freq}(\{a,b,c,d\}) = 0.1$. Then we could introduce new items $i_{\{a,b\}}$, $i_{\{a,b,c\}}$, and $i_{\{a,b,c,d\}}$. These items replace respectively $\{a,b\}$, $\{a,b,c\}$, and $\{a,b,c,d\}$. We enforce these semantics as follows:

$$\begin{array}{l}\text{freq}\left(\{i_{\{a,b\}}\}\right) = 0.3,\ \text{freq}\left(\{i_{\{a,b\}}, a\}\right) = 0.3,\\ \text{freq}\left(\{i_{\{a,b\}}, b\}\right) = 0.3,\ \text{freq}\left(\{a,b\}\right) = 0.3\end{array}$$

$$\begin{array}{l}\text{freq}\left(\{i_{\{a,b,c\}}\}\right) = 0.2,\ \text{freq}\left(\{i_{\{a,b,c\}}, i_{\{a,b\}}\}\right) = 0.2,\\ \text{freq}\left(\{i_{\{a,b,c\}}, c\}\right) = 0.2,\ \text{freq}\left(\{i_{\{a,b\}}, c\}\right) = 0.2\end{array}$$

$$\begin{array}{l}\text{freq}\left(\{i_{\{a,b,c,d\}}\}\right) = 0.1,\ \text{freq}\left(\{i_{\{a,b,c,d\}}, i_{\{a,b,c\}}\}\right) = 0.1,\\ \text{freq}\left(\{i_{\{a,b,c,d\}}, d\}\right) = 0.1,\ \text{freq}\left(\{i_{\{a,b,c\}}, d\}\right) = 0.1\end{array}$$

In this way, we can replace itemsets of high cardinality by a chain of sets of cardinality at most 2. Of course, in general, we do not know the exact frequencies of the prefixes of the sets that are too long. Therefore, in the non-deterministic polynomial many-one reduction, we start by *guessing* them. If $\mathcal{C}$ has a solution, then there exists a correct guess.

In the second step, we have to encode the FREQSAT$\{ntrans\}$-problem as a matrix $C$. We can at this point assume that $\mathcal{C}$ only contains itemsets of cardinality at most 2, and that the frequencies are given exactly (that is, no intervals). We *guess* the total number of transactions $n$, under the constraint $0 \leq n \leq nt$. In the matrix $C$, every row and column corresponds to one item. The entry $C[i,j]$ that corresponds to the item $i$ and the item $j$ is filled as follows: if there is an expression $\text{freq}(\{i,j\}) = f$ in $\mathcal{C}'$, then $C[i,j] = f \cdot n$. Else, the entry $C[i,j]$ is filled *randomly* by a number between 0 and $n$. If in the end, one of the entries in $C$ is not an integer, we reject, since one of the guesses was wrong. In the other case, an instance for IP has been constructed. The full proof now consists in showing that there exists a series of guesses that leads to an intersection pattern $C$ if and only if the original problem $(\mathcal{C}, nt)$ is in FREQSAT$\{ntrans\}$.

**Definition 5** Let $\mathcal{I}$ be a set of items. We assume an order on $\mathcal{I}$. For $P, I \subseteq \mathcal{I}$, $P \leq I$ denotes that $P$ is a prefix of $I$; that is, the first element in $I \setminus P$ is larger than any element in $P$ w.r.t. the order on $\mathcal{I}$.

Let $I_1, \ldots, I_m$ be subsets of $\mathcal{I}$. Let $\mathcal{P}(I_1, \ldots, I_m)$ denote the set $\{P \mid \exists j : 1 \leq j \leq m : P \leq I_j, P \neq \{\}\}$. $(I_1, \ldots, I_m)$ will be omitted from the notation if it is clear from the context. Assume that for every $P \in \mathcal{P}(I_1, \ldots, I_m)$, a frequency $f_P$ has been given. $Two(\{f_P \mid P \in \mathcal{P}\})$ denotes the following system of frequency constraints over the set of items $\mathcal{I} \cup \{i_P \mid P \in \mathcal{P}\}$ (For every $P \in \mathcal{P}$, $i_P$ denotes a new, distinct item):

$$
\begin{aligned}
Two = & \left\{ \begin{array}{l} \text{freq}\left(\{i_{\{a\}}, a\}\right) = f_{\{a\}}, \\ \text{freq}\left(\{a\}\right) = f_{\{a\}} \end{array} \;\middle|\; \{a\} \in \mathcal{P} \right\} \\
& \bigcup \left\{ \begin{array}{l} \text{freq}\left(\{i_{P\cup\{a\}}, a\}\right) = f_{P\cup\{a\}}, \\ \text{freq}\left(\{i_P, a\}\right) = f_{P\cup\{a\}}, \\ \text{freq}\left(\{i_P, i_{P\cup\{a\}}\}\right) = f_{P\cup\{a\}} \end{array} \;\middle|\; P, P\cup\{a\} \in \mathcal{P}, |P| \geq 1 \right\} \\
& \bigcup \left\{ \text{freq}\left(\{i_P\}\right) = f_P \mid P \in \mathcal{P} \right\}
\end{aligned}
$$

**Lemma 6** *Let $\mathcal{C} = \{\text{freq}(I_1) \in [l_1, u_1], \ldots, \text{freq}(I_m) \in [l_m, u_m]\}$ be a set of frequency constraints. $\mathcal{C}$ is satisfiable by a database with at most ntrans transactions if and only if there exists, for every set $P \in \mathcal{P}(I_1, \ldots, I_m)$, a rational number $0 \leq f_P \leq 1$ such that:*

- $\forall j : 1 \leq j \leq m : f_{I_j} \in [l_j, u_j] \cap \{\frac{a}{b} \mid 0 \leq a \leq b, 1 \leq b \leq ntrans\}$
- $Two(\{f_P \mid P \in \mathcal{P}\})$ *is satisfiable by a database with at most ntrans transactions.*

*Proof* **if:** Let $\mathcal{D}$ be a database with at most *ntrans* transactions that satisfies $\mathcal{C}$. Let $f_P = \text{freq}(P, \mathcal{D})$ for all $P \in \mathcal{P}(I_1, \ldots, I_m)$. We construct a database $\mathcal{D}'$ that satisfies $Two(\{f_P \mid P \in \mathcal{P}\})$ as follows. For every transaction $T = (tid, I) \in \mathcal{D}$, let $T'$ denote the following transaction:

$$(tid, I \cup \{i_P \mid P \subseteq I, P \in \mathcal{P}\}$$

The database $\mathcal{D}' := \{T' \mid T \in \mathcal{D}\}$ satisfies $Two(\{f_P \mid P \in \mathcal{P}\})$: it suffices to note that for every $P \in \mathcal{P}$, $i_P$ is in exactly those transactions that contain $P$, and hence, e.g.,

$$\text{freq}(\{i_P\}, \mathcal{D}') = \text{freq}(P, \mathcal{D}') = \text{freq}(P, \mathcal{D}) = f_P \quad ,$$

and

$$\text{freq}\left(\{i_{P\cup\{a\}}, a\}, \mathcal{D}'\right) = \text{freq}(P \cup \{a\}, \mathcal{D}') = \text{freq}(P \cup \{a\}, \mathcal{D}) = f_{P\cup\{a\}} \quad .$$

In this way, all constraints in $Two$ can easily be shown to hold. Furthermore, as $\mathcal{D}$ satisfies $\mathcal{C}$, $f_{I_j} = \text{freq}(I_j, \mathcal{D})$ must be in $[l_j, u_j]$ for all $1 \leq j \leq m$, and $f_{I_j} \in \{\frac{a}{b} \mid 0 \leq a \leq b, 1 \leq b \leq ntrans\}$ follows from the fact that $\mathcal{D}$ has at most *ntrans* transactions.

**only if:** Let $0 \leq f_P \leq 1$, $P \in \mathcal{P}$ be rational numbers, and let $f_{I_j} \in [l_j, u_j]$ for $1 \leq j \leq m$. We show that any database $\mathcal{D}$ that satisfies $Two$, also satisfies $\mathcal{C}$. To show this claim, it suffices to prove, for any $P \in \mathcal{P}$, that $i_P$ must be in exactly those transactions of $\mathcal{D}$ that contain every element of $P$. We will show this claim by induction on the size of $P$.

- Base case, $P = \{a\} \in \mathcal{P}$. $Two$ contains the following constraints:

  $$\text{freq}\left(\{i_{\{a\}}, a\}\right) = f_{\{a\}}, \text{freq}\left(\{a\}\right) = f_{\{a\}}, \text{freq}\left(\{i_{\{a\}}\}\right) = f_{\{a\}}$$

  Hence, $\text{freq}(\{a\}) = \text{freq}\left(\{i_{\{a\}}\}\right) = \text{freq}\left(\{i_{\{a\}}, a\}\right)$, and thus, any transaction of $\mathcal{D}$ that contains $i_{\{a\}}$ also contains $a$ and vice versa.

- $|P| \geq 2$, $P \in \mathcal{P}$. Let $a$ be the last item of $P$ w.r.t. the order on $\mathcal{I}$. Then, $P \setminus \{a\}$ must be in $\mathcal{P}$ as well. As such, *Two* contains the following constraints:

$$\begin{aligned} &\text{freq}\left(\{i_P, a\}\right) = f_P, &\quad &\text{freq}\left(\{i_{P\setminus\{a\}}, a\}\right) = f_P, \\ &\text{freq}\left(\{i_{P\setminus\{a\}}, i_P\}\right) = f_P, &\quad &\text{freq}\left(\{i_P\}\right) = f_P \end{aligned}$$

  Hence,

$$\text{freq}\left(\{i_P\}\right) = \text{freq}\left(\{i_P, a\}\right) = \text{freq}\left(\{i_{P\setminus\{a\}}, a\}\right) = \text{freq}\left(\{i_{P\setminus\{a\}}, i_P\}\right) \quad ,$$

  and thus, any transaction of $\mathcal{D}$ that contains $i_{P\setminus\{a\}}$ and $a$ at the same time, also contains $i_P$, and vice versa. Furthermore, by induction, a transaction contains $i_{P\setminus\{a\}}$ if and only if it contains all items in $P \setminus \{a\}$. Combining these two facts proves the claim for $P$.

It now follows that $\text{freq}\left(I_j, \mathcal{D}\right) = \text{freq}\left(i_{I_j}, \mathcal{D}\right) = f_{I_j} \in [l_j, u_j]$, since $1 \leq j \leq m$, $I_j \in \mathcal{P}$. □

**Lemma 7** *Let $\mathcal{I} = \{i_1, \ldots, i_n\}$ be a set of items, and let, for all $1 \leq k, l \leq n$, a rational number $0 \leq f_{k,l} \leq 1$ be given, and let ntrans be a positive integer.*

*The system of inequalities*

$$\mathcal{C} \;=\; \{\text{freq}\left(\{i_k, i_l\}\right) = f_{k,l} \mid 1 \leq k, l \leq n\}$$

*is satisfiable by a database with $N$ transactions if and only if*

- *For all $1 \leq k, l \leq n$, $F_{k,l} = N \cdot f_{k,l}$ is an integer, and*
- *the following $(n+1) \times (n+1)$ matrix $C_N(\mathcal{C})$ is an intersection pattern:*
  - *$\forall 1 \leq k, l \leq n : C[k, l] = F_{k,l}$,*
  - *$\forall 1 \leq k \leq n : C[n+1, k] = C[k, n+1] = F_{k,k}$,*
  - *$C[n+1, n+1] = N$.*

*Proof* **if.** Let the sets $S_1, \ldots, S_{n+1}$ be a realization of $C$; i.e., $|S_k \cap S_l| = C[k, l]$ for all $1 \leq k, l \leq n+1$. We assume, without loss of generality, that the elements of the sets $S_k$, $k = 1 \ldots n+1$ are positive integers. Then, the following database has exactly $N$ elements and satisfies $\mathcal{C}$:

$$\mathcal{D} = \{(t, \{i_k \mid 1 \leq k \leq n, t \in S_k\}) \mid t \in S_{n+1}\} \quad .$$

Indeed, since $C[n+1, n+1] = N$, $S_{n+1}$ and hence also $\mathcal{D}$ has $N$ elements. Furthermore, because $C[n+1, k] = C[k, k]$ for all $k = 1 \ldots n$, $S_k \subseteq S_{n+1}$. Therefore,

$$\text{freq}\left(\{i_k, i_l\}, \mathcal{D}\right) = \frac{|\{t \in S_{n+1} \mid t \in S_k, t \in S_l\}| = |S_k \cap S_l|}{|\mathcal{D}|} = \frac{C[k, l]}{N} = f_{k,l} \quad .$$

**only if.** Let $\mathcal{D}$ be a databases that satisfies $\mathcal{C}$, and has exactly $N$ transactions. Then, for all $1 \leq k, l \leq n$:

$$\begin{aligned} N \cdot f_{k,l} &= N \cdot \text{freq}\left(\{i_k, i_l\}, \mathcal{D}\right) \\ &= N \cdot \frac{|\{(tid, J) \in \mathcal{D} \mid i_k, i_l \in J\}|}{|\mathcal{D}|} \\ &= |\{(tid, J) \in \mathcal{D} \mid i_k, i_l \in J\}| \end{aligned}$$

Thus, $N \cdot f_{k,l}$ is an integer. Let $S_{n+1} = \{tid \mid (tid, J) \in \mathcal{D}\}$, and for $k = 1 \ldots n$, $S_k = \{tid \mid (tid, J) \in \mathcal{D}, i_k \in J\}$. $S_1, \ldots, S_{n+1}$ is a realization of $C$. For $1 \leq k, l \leq n$:

$$S_k \cap S_l = \{tid \mid (tid, J) \in \mathcal{D}, i_k, i_l \in J\} \ ,$$

and thus,

$$|S_k \cap S_l| = |\mathcal{D}| \cdot \text{freq}\,(\{i_k, i_l\}, \mathcal{D}) = N \cdot f_{k,l} = C[k,l] \ .$$

The constraints for $S_{n+1}$ are fulfilled as for all $k = 1 \ldots n{+}1$, $S_k \cap S_{n+1} = S_k$. □

**Theorem 7** `FREQSAT`$\{ntrans\}$ *is non-deterministically many-one reducible to* `IP`.

*Proof* Let $\mathcal{C} = \{\text{freq}\,(I_1) \in [l_1, u_1], \ldots, \text{freq}\,(I_m) \in [l_m, u_m]\}$ be a set of frequency constraints, and let $ntrans$ be a positive integer. By Lemma 6, $\mathcal{C}$ is satisfiable by a database with at most $ntrans$ transactions, if and only if there exists, for every $P \in \mathcal{P}$, a rational number $f_P \in \{\frac{a}{b} \mid 0 \leq a \leq b, 1 \leq b \leq ntrans\}$ such that $Two(\{f_P \mid P \in \mathcal{P}\})$ is satisfiable by a database with at most $ntrans$ transactions, and for $j = 1 \ldots m$, $f_{I_j} \in [l_j, u_j]$. Let $\mathcal{J}$ be the set of all items that occur in $Two(\{f_P \mid P \in \mathcal{P}\})$. It is clear that $Two(\{f_P \mid P \in \mathcal{P}\})$ is satisfiable by a database with at most $ntrans$ transactions if and only if there exist numbers $f_{\{i_1,i_2\}} \in \{\frac{a}{b} \mid 0 \leq a \leq b, 1 \leq b \leq ntrans\}$ for all $i_1, i_2 \in \mathcal{J}$, such that these numbers are consistent with the system of inequalities $Two$; i.e., for all $(\text{freq}\,(\{i_1, i_2\}) = f_{1,2})$ in $Two$, $f_{\{i_1,i_2\}} = f_{1,2}$, and the system of inequalities

$$\mathcal{C}' \ = \ \{\text{freq}\,(\{i_1, i_2\}) = f_{\{i_1,i_2\}} \mid i_1, i_2 \in \mathcal{J}\} \tag{12}$$

is satisfiable. Indeed; since $Two \subseteq \mathcal{C}'$ ($Two$ only contains constraints over sets of at most 2 elements), any database that satisfies $\mathcal{C}'$, satisfies $Two$ as well, and for the other direction, if $\mathcal{D}$ satisfies $Two$, then we can choose the numbers as follows: for all $i_1, i_2 \in \mathcal{J}$, $f_{\{i_1,i_2\}} = \text{freq}\,(\{i_1, i_2\}, \mathcal{D})$. Then, $\mathcal{D}$ satisfies $\mathcal{C}'$ as well.

$\mathcal{C}'$ is satisfiable by a database with at most $ntrans$ transactions, if and only if there exists an integer $N \leq ntrans$, such that there exists a database with exactly $N$ transactions that satisfies $\mathcal{C}'$, which, by Lemma 7, is equivalent with: for all $i_1, i_2 \in \mathcal{J}$, $N \cdot f_{\{i_1,i_2\}}$ is a positive integer and $C_N(\mathcal{C}')$ is an intersection pattern.

Hence, `FREQSAT`$\{ntrans\}$ is non-deterministically polynomial many-one reducible to `IP`; $\mathcal{C}$ is satisfiable by a database with at most $ntrans$ transactions if and only if there exists a positive integer $N \leq ntrans$, and a choice of rational numbers $f_{\{i_1,i_2\}} \in \{\frac{a}{b} \mid 0 \leq a \leq b, 1 \leq b \leq ntrans, b|N\}$, for all $i_1, i_2 \in \mathcal{J}$, with $f_{\{I_j\}} \in [l_j, u_j]$, for all $j = 1 \ldots m$, such that $C_N(\mathcal{C}')$, as constructed in the proof, is an intersection pattern. The reduction $R$ maps the system $\mathcal{C}$ non-deterministically to the intersection pattern associated with one of these choices. $\mathcal{C}$ is satisfiable, if and only if one of the branches of $R(\mathcal{C})$ is an intersection pattern. □

**Corollary 3** `IP` *is in* **NP** *if and only if* `FREQSAT`$\{ntrans\}$ *is in* **NP**.

*Proof* The if-direction follows from Theorem 6, and the only-if direction follows directly from Theorem 7. □

## 5.3 Entailment

In this section we show that, in contrast to FREQSAT, where the entailed interval on the frequency of an itemset is always an interval, in FREQSAT$\{ntrans\}$, the entailed set can be *any* finite set of rational numbers. Moreover, given a set of rational numbers $R$, there exists a system of constraints of polynomial size in the description of $R$, such that the entail set of a target set given the system of constraints is exactly $R$, thus effectively showing that the entail sets do not have any connectedness or compactness properties that can be exploited in algorithms.

To make the construction less involved, we will be using *extended* FREQSAT-expressions; that is, we allow expressions involving the conjunction, disjunction, and negation of items. This does not change the problem, because from Theorem 2, we know that we can extend FREQSAT$\{ntrans\}$ to arbitrary Boolean formulas without adding extra complexity or fundamentally changing the entail sets[1].

We first illustrate the principle on a small example. Then the example will be generalized.

*Example 6* Assume that the maximal number of transactions is set to $nt$. Consider the following set of expressions over the items $a, b, c$:

$$\begin{array}{ll}
\mathrm{freq}\,(\{i\}) = 1/nt & \mathrm{freq}\,(\{a,b\}) = 0 \\
\mathrm{freq}\,(\{a,c\}) = 0 & \mathrm{freq}\,(\{b,c\}) = 0 \\
\mathrm{freq}\,(a \vee c) = k/nt & \mathrm{freq}\,(b \vee c) = k/nt
\end{array}$$

The first constraint makes sure that there are exactly $nt$ transactions. The next three constraints enforce that the transactions with $a$, the ones with $b$, and the ones with $c$ are disjoint. Let $A$ be the set of transactions with $a$, $B$ the ones with $b$, and $C$ the ones with $c$. The last two constraints express that $|A \cup C| = |B \cup C| = k/nt$. Let's now consider the set $\mathrm{ENT}^{nt}_{a \vee b}(\mathcal{C})$. Suppose that $C$ contains $l$ items, $0 \leq l \leq k$. Then, both $A$ and $B$ contain $k-l$ transactions, and hence, $|A \cup B| = 2(k-l)$. Therefore, $\mathrm{ENT}^{nt}_{a \vee b}(\mathcal{C}) = \{\frac{2 \cdot l}{nt} \mid l = 0 \ldots k\}$. Thus, $0/nt, 2/nt \in \mathrm{ENT}^{nt}_{a \vee b}(\mathcal{C})$, but $1/nt$ is not in $\mathrm{ENT}^{nt}_{a \vee b}(\mathcal{C})$.

**Construction in general.** We now show that we can express every arbitrary set. Let $R = \{r_1, \ldots, r_k\}$ be a set of positive rational numbers between 0 and 1. First, we equalize the denominators, that is, let $R = \{\frac{p_1}{q}, \ldots, \frac{p_k}{q}\}$. In the construction we use new items, $n_i$ and $d_i$ for $i = 1 \ldots, k$, and the item $j$. The bound on the number of transactions is $q$, and the set of constraints is the following:

$$\left\{\begin{array}{lcl}
\mathrm{freq}\,(\{j\}) & = & 1/q, \\
\mathrm{freq}\,(d_1 \vee \ldots \vee d_k) & = & 1/q \\
\mathrm{freq}\,(d_i \wedge d_j) & = & 0 \quad 1 \leq i < j \leq k
\end{array}\right\} \bigcup_{i=1}^{k} \mathcal{M}_{p_i}(d_i, n_i)$$

[1] It must be remarked, though, that in the reduction from extended FREQSAT to FREQSAT, a factor 2 has to be taken into account; that is, if the entail set for $\varphi$ given an extended FREQSAT problem is $R$, the entail set for the corresponding itemset $\{d, t_\varphi\}$ given the reduction to FREQSAT will be $\{r/2 \mid r \in R\}$. This complication, however, is immaterial for our claim that the entail set in FREQSAT$\{ntrans\}$ does not possess any connectedness or compactness property.

The first constraint makes sure that the number of transactions is exactly $q$. The second and third line ensure that exactly one $d_i$, $i = 1 \dots k$ has frequency $1/q$; the others have frequency 0. Let $d_l$ be the non-zero one. Then, the Multiplication Lemma 4 is used to express that for all $i = 1 \dots k$, $\mathrm{freq}\,(n_i)$ is $p_i$ times the frequency of $d_i$. Hence, the frequency of $n_l$ is $p_l/q$, the frequencies of the other $n_i$'s are zero. Therefore, the frequency of $n_1 \vee \dots \vee n_k$ is $p_l/q$. Because $l$ was chosen arbitrary, it holds that $\mathrm{ENT}^q_{n_1 \vee \dots \vee n_k}(\mathcal{E}) = R$.

*Example 7* Consider the set $R = \{1/2, 1/3, 1/4\}$. First we equalize the denominators: $R = \{6/12, 4/12, 3/12\}$. We set the upper bound on the number of transactions to 12 and make sure that there are exactly 12 transactions by adding the constraint $\mathrm{freq}\,(\{j\}) = 1/12$.

New items $d_1, d_2, d_3$ are introduced. We add the following constraints to ensure that for exactly one $i = 1, 2, 3$, $\mathrm{freq}\,(d_i) = 1/12$ and the other $\mathrm{freq}\,(d_i) = 0$.

$$\begin{array}{ll} \mathrm{freq}\,(d_1 \vee d_2 \vee d_3) = 1/12, & \mathrm{freq}\,(d_1 \wedge d_2) = 0 \\ \mathrm{freq}\,(d_1 \wedge d_3) = 0, & \mathrm{freq}\,(d_2 \wedge d_3) = 0 \end{array}$$

Next, the items $n_1, n_2, n_3$ are introduced that have a frequency of respectively $3 \cdot \mathrm{freq}\,(d_1)$, $4 \cdot \mathrm{freq}\,(d_2)$, and $6 \cdot \mathrm{freq}\,(d_3)$:

$$\mathcal{M}_3(d_1, n_1), \mathcal{M}_4(d_2, n_2), \mathcal{M}_6(d_3, n_3)$$

Hence, exactly one of $\mathrm{freq}\,(d_j)$ is 1/12, the other are 0. Therefore, either $\mathrm{freq}\,(n_1) = 3/12, \mathrm{freq}\,(n_2) = 0, \mathrm{freq}\,(n_3) = 0$, or $\mathrm{freq}\,(n_1) = 0, \mathrm{freq}\,(n_2) = 4/12, \mathrm{freq}\,(n_3) = 0$, or $\mathrm{freq}\,(n_1) = 0, \mathrm{freq}\,(n_2) = 0, \mathrm{freq}\,(n_3) = 6/12$.

Finally, the set of frequencies for $n_1 \vee n_2 \vee n_3$ entailed by this set of constraints equals $\{3/12, 4/12, 6/12\}$.

## 5.4 Fixed Parameter Variants

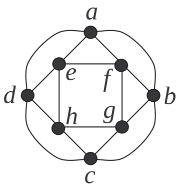
We now study some cases of FREQSAT($C \cup \{ntrans\}$), where some of the parameters are fixed.

### *5.4.1* FREQSAT$\{ltrans = 3, ntrans\}$ *is* **NP**-*Hard*

Another illustration of the complexity of giving the number of transactions as part of the input, is the fact that FREQSAT$\{ltrans = k\}$ can be solved in polynomial time, while adding the number of transactions to the input makes the problem **NP**-hard.

The **NP**-hardness will be shown by reducing the following *triangle partition problem* to FREQSAT$\{ltrans = 3, ntrans\}$: given a graph $G$, with $|V| = 3k$, can $G$ be divided into disjoint triangles; that is, is it possible to partition the vertices into $V_1, \dots, V_k$, such that for all $i = 1 \dots k$, $|V_i| = 3$, and for all $v, w \in V_i$, there is an edge between $v$ and $w$. This problem is known to be **NP**-complete [17].

**Theorem 8** FREQSAT$\{ltrans = 3, ntrans\}$ *is* **NP**-*Hard.*



| TID | Items | TID | Items |
|---|---|---|---|
| 1 | $a, e, f$ | 5 | $b, c, g$ |
| 2 | $a, b, f$ | 6 | $c, g, h$ |
| 3 | $a, d, e$ | 7 | $c, d, h$ |
| 4 | $b, f, g$ | 8 | $d, e, h$ |

**Fig. 4** A graph $G$ that cannot be triangulated, with a database satisfying $\mathcal{C}_G$

*Proof* Let $G(V, E)$ be a graph, $|V| = 3k$. Consider now the following set of frequency constraints $\mathcal{C}_G$ over the set of items $V$:

$$\begin{aligned} \text{freq}\,(\{v\}) &= 3/|V| \qquad && \forall v \in V \\ \text{freq}\,(\{v, w\}) &= 0 && \forall (v, w) \notin E \end{aligned}$$

$(\mathcal{C}, ntrans = k)$ is in $\texttt{FREQSAT}\{ltrans = 3, ntrans\}$ if and only if $G$ can be triangulated in disjoint triangles. Indeed; $V_1, \ldots, V_k$ is a triangulation if and only if $\{(1, V_1), \ldots, (k, V_k)\}$ satisfies $\mathcal{C}$. □

*Example 8* Consider the graph given in Fig. 4. This graph $G$ clearly cannot be triangulated, as the number of vertices (8) is not a multiple of 3. Thus, the system of frequency constraints $\mathcal{C}_G$ cannot be satisfied by a database of transactions with at most 8/3 transactions of length at most 3. To illustrate the impact of the bound on the number of transactions, a database that does satisfy $\mathcal{C}_G$, and the bound on the transaction length, but without a bound on the number of transactions is given in the figure as well.

### *5.4.2* $\texttt{FREQSAT}\{ntrans{=}2\}$ *is* $\mathbf{NP}$*-complete*

For *ntrans* given as part of the input, we do not know the exact implications for the complexity, although some evidence has been given that it substantially alters the problem. For a fixed number of transactions, though, we do have the exact complexity. We will show that if the fixed bound on the number of transactions is at least 2, the problem is $\mathbf{NP}$-complete. Hence, $\texttt{FREQSAT}\{ntrans = k\}$ is in $\mathbf{NP}$-complete and is thus clearly not fixed-parameter tractable (assuming $\mathbf{P} \neq \mathbf{NP}$.)

**Theorem 9** $\texttt{FREQSAT}\{ntrans{=}2\}$ *is* $\mathbf{NP}$*-complete.*

*Proof* Let $\mathcal{C}$ be an instance of the $\texttt{FREQSAT}\{ntrans = c\}$-problem over the set of items $\mathcal{I}$, with $c \geq 2$. $\texttt{FREQSAT}\{ntrans = c\}$ is clearly in $\mathbf{NP}$, because a satisfying database has size at most $c \cdot |\mathcal{I}|$, and can thus be used as a certificate for membership.

For the hardness, we reduce 3-colorability to $\texttt{FREQSAT}\{ntrans = 2\}$. Let $G(V, E)$ be a graph. We define $\mathcal{COL}(G)$ over the following set of items: $\{R_v, G_v, B_v, \overline{R}_v, \overline{B}_v, \overline{G}_v \mid v \in V\} \cup \{c\}$. $R_v$ $(G_v, B_g)$ stands for "vertex $v$ is red (green, blue)", and $\overline{R}_v$ $(\overline{G}_v, \overline{B}_g)$ stands for "vertex $v$ is NOT red (green, blue).” The item $c$ will mark one of the two transactions as the one containing the coloring of $G$ as follows:

$$\begin{array}{lll}
\multicolumn{3}{l}{\forall v \in V :} \\
\quad \text{freq}\left(\{R_v\}\right) \;=\; 0.5 & \text{freq}\left(\{G_v\}\right) \;=\; 0.5 & \text{freq}\left(\{B_v\}\right) \;=\; 0.5 \\
\quad \text{freq}\left(\{\overline{R}_v\}\right) \;=\; 0.5 & \text{freq}\left(\{\overline{G}_v\}\right) \;=\; 0.5 & \text{freq}\left(\{\overline{B}_v\}\right) \;=\; 0.5 \\
\quad \text{freq}\left(\{R_v, \overline{R}_v\}\right) \;=\; 0 & \text{freq}\left(\{G_v, \overline{G}_v\}\right) \;=\; 0 & \text{freq}\left(\{B_v, \overline{B}_v\}\right) \;=\; 0 \\
\quad \text{freq}\left(\{c, R_v, G_v\}\right) \;=\; 0 & \text{freq}\left(\{c, R_v, B_v\}\right) \;=\; 0 & \text{freq}\left(\{c, G_v, B_v\}\right) \;=\; 0 \\
\multicolumn{3}{l}{\forall (v, w) \in E :} \\
\quad \text{freq}\left(\{c, R_v, R_w\}\right) \;=\; 0 & \text{freq}\left(\{c, G_v, G_w\}\right) \;=\; 0 & \text{freq}\left(\{c, B_v, B_w\}\right) \;=\; 0 \\
\text{freq}\left(\{c\}\right) \;=\; 1/2 & &
\end{array}$$

The first 9 constraints ensure that both transactions contain for every vertex exactly one of $R_v, \overline{R}_v$ ($B_v, \overline{B}_v$, $G_v, \overline{G}_v$). The next three constraints make sure that within the transaction with $c$, every vertex has “at most one color”. The next three constraints ensure that for every edge $(v, w)$, $v$ and $w$ “do not have the same color” in the transaction with $c$.

It is now easy to see that the graph $G$ is 3-colorable if and only if $\mathcal{COL}(G)$ is satisfiable by a database with 2 transaction. □

The problem FREQSAT$\{ntrans = 1\}$ is in **P**. This can be seen as follows: every constraint in $\mathcal{C}$ has one of the following three forms:

$$\text{freq}\,(I) = 0, \qquad \text{freq}\,(I) = 1, \qquad \text{freq}\,(I) \in [0, 1].$$

Every other constraint can straightforwardly be transformed to one of these three forms. Constraints of the third form can be omitted, as they are always fulfilled. So, let

$$\mathcal{C} \;=\; \{\text{freq}\,(I_j) = 0, j = 1 \ldots k\} \cup \{\text{freq}\,(I_j) = 1, j = k+1 \ldots m\} \;\; .$$

$\mathcal{C}$ is satisfiable if and only if for all $l = k+1 \ldots m$ it holds that $I_l \setminus (\bigcup_{j=1}^{k} I_j)$ is non-empty. In that case is the database $\{(1, \bigcup_{j=1}^{k} I_j)\}$ a satisfying database. Also the other direction holds: if $\mathcal{C}$ is not satisfiable, then $\{(1, \bigcup_{j=1}^{k} I_j)\}$ is not a satisfying database. As we only need to check one database, the problem FREQSAT$\{ntrans = 1\}$ is clearly in **P**.

## 6 FREQSAT{ndup}

In this section we study FREQSAT$\{ndup\}$. First we show that we can always reduce a FREQSAT$\{ndup\}$-instance $(\mathcal{C}, nd)$ to an instance $(\mathcal{C}', 1)$. Hence, we show that the following problem: *given* $\mathcal{C}$*, decide whether* $(\mathcal{C}, 1)$ *is in* FREQSAT$\{ndup\}$, is equivalent to FREQSAT$\{ndup\}$. We denote this problem FREQSAT$\{ndup = 1\}$.

We furthermore show that FREQSAT$\{ntrans\}$ reduces to FREQSAT$\{ndup\}$, and that FREQSAT$\{ndup = 1\}$ is **PP**-hard. Hence, knowing the number of duplicates does add complexity to the FREQSAT-problem (assuming **NP** $\neq$ **PP**).

## 6.1 Fixed Parameter Variant

For *ndup*, we start with the fixed parameter variant, because we will use the results here to simplify the proofs for the general case. The following theorem states that fixing *ndup* to 1 does not change the complexity of the problem.

**Theorem 10** *Let $C \subseteq \{ltrans, ntrans\}$.*

$$\texttt{FREQSAT}(C \cup \{ndup\}) \equiv \texttt{FREQSAT}(C \cup \{ndup = 1\}) \ .$$

*Proof* Let $\mathcal{C}$ be a set of frequency constraints, and let $nd$ be a positive integer. Let the binary representation of $nd$ be $B_l \dots B_0$. We introduce $l+1$ new items, $b_0, \dots, b_l$. We use the $b_j$'s to eliminate duplicates. That is, $nd+1$ transactions with set of items $I$, will be replaced by transactions with set of items: $I$, $I \cup \{b_0\}$, $I \cup \{b_1\}$, $I \cup \{b_0, b_1\}$, $\dots$, $I \cup \{b_j \mid B_j = 1\}$. Let $I \cup B$ be an itemset, with $B \subseteq \{b_0, \dots, b_l\}$, and $I \cap \{b_0, \dots, b_l\} = \emptyset$. $\nu(I \cup B)$ is defined as the *number associated with* $I$; that is:

$$\nu(I \cup B) = \sum_{b_j \in B} 2^j \ .$$

We have to make sure that the numbers of the transactions are never higher than $nd$. This can be done as follows: for all $\ell$ such that $B_\ell = 0$, add the constraint $\text{freq}\left(\{b_j \mid B_j = 1, j > \ell\} \cup \{b_\ell\}\right) = 0$. For example, for $5 = 101_b$, the constraint $\text{freq}\left(\{b_2, b_1\}\right) = 0$ would be added, disallowing for bit 2 and bit 1 to be 1 at the same time, because bit 1 and 2 being 1 together, would result in at least 6. Let $\mathcal{B}_{nd}$ be the set of these constraints.

$$\Delta_{nd}(\mathcal{C}) \ := \ \mathcal{C} \cup \mathcal{B}_{nd-1} \ .$$

The constraints in $\mathcal{B}_{nd-1}$ allow to reduce the number of duplicates from $nd$ to 1, because every set $I$ can be extended with a $B \subseteq \{b_0, \dots, b_l\}$, with $\nu(B)$ ranging from 0 to $nd-1$. It is now true that $(\mathcal{C}, nt, nd)$ is in $\texttt{FREQSAT}\{ntrans, ndup\}$, if and only if $(\Delta_{nd}(\mathcal{C}), nt)$ is a satisfiable instance of the $\texttt{FREQSAT}\{ntrans, ndup = 1\}$-problem, and $(\mathcal{C}, nd)$ is in $\texttt{FREQSAT}\{ndup\}$, if and only if $\Delta_{nd}(\mathcal{C})$ is in $\texttt{FREQSAT}\{ndup = 1\}$. For the other sets $C$ that include *ltrans*, it suffices to notice that $\texttt{FREQSAT}(C \cup \{ndup\})$ is always equivalent to $\texttt{FREQSAT}(C \setminus \{ltrans\} \cup \{ndup\})$. □

*Example 9* The binary representation of 10 is 1010. Hence, $\mathcal{B}_{10}$ is the following set of constraints:

$$\{\text{freq}\left(\{b_3, b_2\}\right) = 0, \text{freq}\left(\{b_3, b_1, b_0\}\right) = 0\} \ .$$

Every database that satisfies these constraints can have transactions $(tid, J)$ with $J \cap \{b_0, b_1, b_2, b_3\}$ equal to:

$$\begin{array}{l} \{\}, \{b_0\}, \{b_1\}, \{b_0, b_1\}, \{b_2\}, \{b_0, b_2\}, \{b_1, b_2\} \\ \{b_0, b_1, b_2\}, \{b_3\}, \{b_0, b_3\}, \{b_1, b_3\} \end{array}$$

These transactions have respectively as associated numbers 0, …, 10. The constraints in $\mathcal{B}_{10}$ disallow transactions that contain

$$\{b_3, b_1, b_0\}, \{b_3, b_2\}, \{b_3, b_2, b_0\}, \{b_3, b_2, b_1\}, \{b_3, b_2, b_1, b_0\} \ .$$

These transactions have respectively as associated numbers 11, …, 15.

Therefore, adding the items $b_0, \ldots, b_3$, and $\mathcal{B}_{10}$ makes it possible to reduce the number of duplicates with a factor 11.

## 6.2 General Case

**Theorem 11**
$\texttt{FREQSAT}\{ndup\} \equiv \texttt{FREQSAT}\{ntrans, ndup\}$
$\texttt{FREQSAT}\{ntrans\} \leq \texttt{FREQSAT}\{ndup\}$

*Proof* $\texttt{FREQSAT}\{ndup\} \leq \texttt{FREQSAT}\{ntrans, ndup\}$: With $n$ items and $nd$ duplicates, one can have maximally $nd \cdot 2^n$ transactions. Hence, $(\mathcal{C}, nd) \in \texttt{FREQSAT}\{ndup\}$ if and only if $(\mathcal{C}, nd \cdot 2^n, nd) \in \texttt{FREQSAT}\{ntrans, ndup\}$.

For the other direction, $\texttt{FREQSAT}\{ndup\} \geq \texttt{FREQSAT}\{ntrans, ndup\}$, we will use Theorem 10, and show the equivalent statement $\texttt{FREQSAT}\{ndup = 1\} \geq \texttt{FREQSAT}\{ntrans, ndup = 1\}$. Let $\mathcal{C} = \{\mathrm{freq}\,(I_j) \in [l_j, u_j], j = 1 \ldots m\}$, $\mathcal{I} = \bigcup_{j=1^m} I_j$. Let $b_l \ldots b_0$ be the binary representation of the bound on the number of transactions, $nt$.
$(\mathcal{C}, nt) \in \texttt{FREQSAT}\{ntrans, ndup = 1\}$ if and only if

$$\begin{aligned}
&\{\mathrm{freq}\,(\{d\} \cup I_j) \in [l_j/2, u_j/2], j = 1 \ldots m\} \\
&\cup \{\mathrm{freq}\,(\{d\}) = 0.5, \mathrm{freq}\left(\overline{d}\right) = 0.5, \mathrm{freq}\left(d, \overline{d}\right) = 0\} \\
&\cup \mathcal{B}_{nt-1} \cup \{\mathrm{freq}\,(\{b_j, d\}) = 0, j = 1 \ldots l\} \\
&\cup \{\mathrm{freq}\left(\{i, \overline{d}\}\right) = 0 \mid i \in \mathcal{I}\}
\end{aligned}$$

is in $\texttt{FREQSAT}\{ndup = 1\}$. In this reduction, the simulating database is split into two equally sized parts. The actual database consists of the transactions containing $d$. In the other part, every transaction contains $\overline{d}$ and some items of $\{b_0, \ldots, b_l\}$. Since $\mathcal{B}_{nt-1}$ holds, and the number of duplicates is 1, the $\overline{d}$-part has maximally $nt$ transactions. Because both parts have equal size, the actual database, that is embedded as the $d$-part, contains maximally $nt$ transactions as well.

$\texttt{FREQSAT}\{ntrans\} \leq \texttt{FREQSAT}\{ndup\}$:
$(\mathcal{C}, nt)$ is a satisfiable instance of $\texttt{FREQSAT}\{ntrans\}$ if and only if $(\mathcal{C}, nt, nt)$ is in $\texttt{FREQSAT}\{ntrans, ndup\}$. Indeed; any database with at most $nt$ transactions has at most $nt$ duplicates, and, obviously, any database with at most $nt$ transactions and $nt$ duplicates has at most $nt$ transactions. Therefore, $\texttt{FREQSAT}\{ntrans\}$ reduces to $\texttt{FREQSAT}\{ntrans, ndup\}$, which is equivalent to $\texttt{FREQSAT}\{ndup\}$, as shown in the first part of this proof. □

### *6.2.1* `FREQSAT`$\{ndup = 1\}$ *is in* **PSPACE**

We will now show an upper bound on the complexity of `FREQSAT`$\{ndup = 1\}$. Because of Theorems 10 and 11, this upper bound is an upper bound on the complexity of all problems studied in this paper.

**Theorem 12** `FREQSAT`$\{ndup = 1\}$ *is in* **PSPACE**.

*Proof* Let $\mathcal{C} = \{\text{freq}\,(I_j) \in [l_j, u_j], j = 1 \dots m\}$, and let $\mathcal{I} = \bigcup_{j=1}^{m} I_j$. Every database $\mathcal{D}$ that satisfies $\mathcal{C}$, and with $ndup(\mathcal{D}) \leq 1$, has at most $2^{|\mathcal{I}|}$ transactions.

We show a non-deterministic procedure to decide the satisfiability of $\mathcal{C}$ that uses at most polynomial space in the length of $\mathcal{C}$. In this way we show that `FREQSAT`$\{ndup = 1\}$ is in **NPSPACE**, and thus by Savitch's Theorem [26, p. 149-150], also in **PSPACE**.

We "guess" a database $\mathcal{D}$, transaction by transaction. We avoid generating the same transaction twice, by requiring that every new transaction comes lexicographically strictly after the previous one. During database generation, we maintain $m$ counters for $I_1, \dots, I_m$, and 1 counter for $|\mathcal{D}|$. For every new transaction $(tid, J)$, we increment the counter $|\mathcal{D}|$, and we do the checks $I_j \subseteq J$. For all $j$ such that $I_j \subseteq J$, the counter for $I_j$ is incremented. After at most $2^{|\mathcal{I}|}$ guesses, we stop the database generation. We then check whether $Counter(I_j)/Counter(|\mathcal{D}|)$ is within the interval $[l_j, u_j]$. If this is the case for all $j = 1 \dots m$, we accept, otherwise, we reject. □

### *6.2.2 FREQSAT*$\{ndup\}$ *is PP-hard*

In this section we show that the complexity of `FREQSAT`$\{ndup\}$ is provably harder than the complexity of `FREQSAT` (assuming $\mathbf{NP} \neq \mathbf{PP}$.)

We say that a language $L$ is in **PP** if there exists a non-deterministic polynomially bounded Turing machine $N$ such that, for all inputs $x$, $x \in L$ if and only if *more than half of the computations of* $N$ *on input* $x$ *end up accepting.* We say that $N$ decides $L$ "by majority". It is known that **NP** is included in **PP**. It is also widely believed that this inclusion is strict, for a number of reasons. First, **PP** is closed under complement, whereas **NP** is believed to be not. Second, *Toda's theorem* states that the polynomial hierarchy **PH** is a subset of $\mathbf{P}^{\mathbf{PP}}$. Hence, $\mathbf{PP} = \mathbf{NP}$ would cause the polynomial hierarchy **PH** to collapse to $\mathbf{P}^{\mathbf{NP}}$. $\mathbf{P}^{\mathbf{PP}}$ is included in **PSPACE**. The `MAJSAT`-problem, asking if more than half of the truth assignments for a given formula $\phi$ are accepting, is **PP**-complete.

**Theorem 13** `FREQSAT`$\{ndup\}$ *is* **PP**-*hard.*

*Proof* By Theorem 2, we know that we can use arbitrary Boolean formulas instead of itemsets, without loss of generality.

We reduce `MAJSAT` to `FREQSAT`$\{ndup = 1, ntrans\}$. This reduction proves the theorem, as, by Theorems 10 and 11, `FREQSAT`$\{ndup = 1, ntrans\}$ is equivalent to `FREQSAT`$\{ndup\}$. Let $\varphi$ be the given formula with variables $x_1, \dots, x_n$. We construct a set of constraints $\mathcal{C}$, such that $(\mathcal{C}, 2^n)$ is in `FREQSAT`$\{ndup =$

$1, ntrans\}$ if and only if more than half of the truth assignments of $\varphi$ are accepting.

We introduce items $x$ and $\overline{x}$ for every variable in $\varphi$. These items will express respectively "$x$ is true", and "$x$ is false". For every variable $x$, we add the following constraints:

$$\mathrm{freq}\left(\{x\}\right) = 0.5, \qquad \mathrm{freq}\left(\{\overline{x}\}\right) = 0.5, \quad \mathrm{freq}\left(\{x, \overline{x}\}\right) = 0$$

Because the number of transactions is set to $2^n$, and no duplicates are allowed, for every truth assignment $A$ for $\varphi$, there will be exactly one transaction $(tid, J)$ with $x \in J$ iff $A(x) = 1$, and $\overline{x} \in J$ iff $A(x) = 0$.

The requirement that $\varphi$ is true in more than half of the truth assignments can thus now be stated as follows:

$$\mathrm{freq}\left(\varphi\right) \in [(2^{n-1} + 1)/2^n, 1] \ .$$

□

Notice that it is unlikely that this **PP** lower bound is also an upper bound on the complexity of `FREQSAT`$\{ndup\}$. Intuitively, it is not very likely that `FREQSAT`$\{ndup\}$ is equivalent to co-`FREQSAT`$\{ndup\}$; the former asks if there exists one database satisfying certain constraints, while the latter asks if for every database it holds that certain constraints are violated. It is unlikely that the one problem can be reduced to the other. On the other hand, however, **PP** is closed under complement.

## 7 Related Work and Applications

In this section we discuss related work and applications in the area of probabilistic logics, privacy preserving data mining, condensed representations, and pruning in frequent set mining.

**Probabilistic Logics.** The `FREQSAT`- problem is very much related to probabilistic logic [18] and reasoning about uncertainty and belief [27], studied in the field of artificial intelligence. E.g., as was proven in [7], the complexity of the `pSAT`-problem introduced by *Nilsson* [25], and extensions to intervals, conditional constraints, etc. [20,19,16,22,21] are closely related to the `FREQSAT`-problem. The main difference between the work we present in this paper, and the literature on probabilistic logics is in the extra constraints we put on the database of transactions. These constraints, that are quite natural in the context of itemset mining, would correspond to less natural constraints on the underlying probability distributions of the probabilistic logics.

**Privacy Preserving Data Mining.** Data Mining can be a serious threat to the privacy. Therefore, methods are developed to adapt databases in such a way that still meaningful data mining results can be produced from it, but the privacy of the individual data are not compromised [2]. It is, however, conceivable that the mining is done by a trusted party. In that case, there is no risk of disclosure based on the original data. Even though, the results of the mining themselves can disclose more of the original data than is desirable. The process of trying to reconstruct parts of the original database from data mining results is called *inverse data mining* [24]. The FREQSAT-problem, its various variants and the entailment problems can be situated in this context. The results of a frequent set mining operation can be represented as an instance of FREQSAT. Inverse data mining would then amount to deriving the frequencies of other itemsets, not in the result set. In this context, the high complexities of the problems studied in this paper are bad news: suppose that we want to publish some itemsets with their frequencies, but first we want to assess how much these frequencies disclose of the original dataset. This problem can be stated as one of the variants of FREQSAT. The high complexity of the FREQSAT-problems in this paper, however, shows that there is little hope that it is effectively possible to assess the degree of disclosure. On the bright side, the high complexity means also that it is potentially very hard to break the privacy. However, the situation is different from that of, for example, public key encryption. In inverse mining, partial information can be derived with incomplete methods, whereas, in general, in public key encryption, the code cannot be *partially* broken. Hence, in inverse mining, the more computing power one has, the more one can derive. Therefore, unless one has superior computing power over potentially malicious parties, the results of mining cannot be guaranteed to be safe.

In [28], the following problem of approximate inverse frequent itemset mining is studied. Given some itemsets with their absolute support, does there exist a database such that these support constraints are *approximately* satisfied, in the sense that a difference proportional to the number of constraints given is allowed. This problem is shown to be **NP**-complete. Also an approximate algorithm to determine information leakage is given.

In [29,13], heuristic methods for generating a database (approximately) satisfying given frequency constraints are given. The idea behind this database generation is to, instead of publishing a confidential database, generate a new database with the same frequency information that can be published for analysis purposes. The feasibility of these approaches depends highly on the assumption that many of the items are (conditionally) independent.

**Condensed Representations.** Another application is making condensed representations [23] of frequent itemsets. For an overview of condensed representations for the itemset domain, see [11]. In such condensed representations typically only non-redundant information is stored. Entailment of frequencies as in the FREQSAT-problem allows for derivation of frequencies. The stronger the deduction mechanism, the more redundancy in the set of frequencies can be found. The complexity results in this paper indicate that complete deduction in the most general context is infeasible, and hence, incomplete, yet tractable methods are more appropriate. In [9], for a special case of FREQSAT, entailment can be decided in polynomial time. This special case is then used to make the Non-Derivable Itemsets representation. In this representation, all itemsets are removed if their frequency can be derived perfectly from the other frequencies in the set.

**Frequent Itemset Mining Algorithms.** A third application is improving the pruning of frequent itemset mining algorithms. All frequent set mining algorithms use the monotonicity rule to prune substantial parts of the

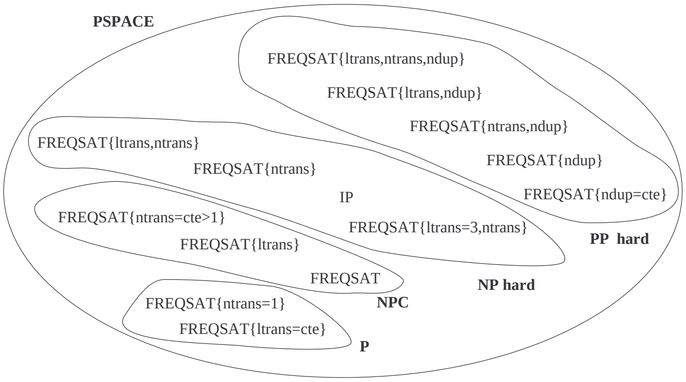


**Fig. 5** Visualization of the different complexity results. "cte" indicates a fixed parameter (constant expression)

search space. This monotonicity rule can be seen as a very simple example of deduction. Based on partial frequency information of some itemsets, bounds on the frequencies of yet to be counted sets are derived. If these bounds establish that a certain set must be certainly frequent or certainly infrequent, the counting of it can be omitted in some cases. In the context of FREQSAT, frequency constraints can be used to model the frequency information gathered in previous scans over the database of transactions. The deduction can then be used to identify sets that are certainly frequent/infrequent. In [3,4, 8,9], in some form, deduction rules are used in order to improve pruning and speed up frequent set mining algorithms. In [15] it is studied how the pruning in candidate-based algorithms influences their performance. Improving pruning with FREQSAT might result in a higher success rate for these algorithms. Other complexity results in frequent set mining include [5] and [30], settling complexity issues in the context of mining maximally frequent itemsets.

## 8 Summary and Conclusion

The complexity of different variants of the the FREQSAT-problem, where extra characteristics of the underlying database of transactions are known was studied. Figure 5 illustrates the relations between the different variants.

The main open questions remain the complexity of FREQSAT$\{ntrans\}$ and of FREQSAT$\{ndup\}$. For FREQSAT$\{ntrans\}$, we showed that it is **NP**-complete if IP is in **NP**. We also illustrated that FREQSAT$\{ntrans\}$ has different properties than FREQSAT by showing that the set $\mathrm{ENT}_I^{nt}(\mathcal{C})$ can be any set of rational numbers, whereas in FREQSAT, this set is always an interval of the rational numbers.

FREQSAT$\{ndup\}$ is the most complex of the different variants of FREQSAT. Its complexity is between **PP** and **PSPACE**. The exact complexity is unknown. Assuming that $\mathbf{NP} \neq \mathbf{PP}$, FREQSAT$\{ndup\}$ is provably harder than FREQSAT.

Finally, for the different characteristics, also the complexity when they are fixed are studied. For *ltrans*, the switch from input-parameter to fixed parameter results in a reduction in complexity from **P** to **NP**. For *ntrans*, this switch results in certainty about the membership in **NP**, while for *ndup*, the switch does not change anything at all. Notice that for the fixed parameter setting, not all combinations were studied.

We consider as further work: the study of the exact complexities for all cases, and the study of the missing combinations for the fixed parameter setting. It would also be very interesting to see if parameters can be found for the `FREQSAT`-problem for which the problem is fixed-parameter tractable.

## Acknowledgement

Part of this work was done while the author was still working at the University of Antwerp as a Post-doctoral Fellow of the Research Foundation - Flanders (FWO). The work has been partially supported by the EU contract IQ FP6-516169.

## References

1. R. Agrawal, T. Imilienski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, pages 207–216, Washington, D.C., 1993.
2. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, pages 439–450, 2000.
3. Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, and L. Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2):66–75, 2000.
4. R. J. Bayardo. Efficiently mining long patterns from databases. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, pages 85–93, Seattle, Washington, 1998.
5. Endre Boros, Vladimir Gurvich, Leonid Khachiyan, and Kazuhisa Makino. On the complexity of generating maximal frequent and minimal infrequent sets. In *STACS*, pages 133–141, 2002.
6. T. Calders. *Axiomatization and Deduction Rules for the Frequency of Itemsets.* PhD thesis, University of Antwerp, Belgium, 2003.
7. T. Calders. Computational complexity of itemset frequency satisfiability. In *Proc. PODS Int. Conf. Principles of Database Systems*, pages 143–154, 2004.
8. T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proc. PKDD Int. Conf. Principles of Data Mining and Knowledge Discovery*, pages 74–85. Springer, 2002.
9. T. Calders and B. Goethals. Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1):171–206, 2007.
10. T. Calders and J. Paredaens. Axiomatization of frequent itemsets. *Theoretical Computer Science*, 290(1):669–693, 2003.
11. T. Calders, C. Rigotti, and J-F. Boulicaut. A survey on condensed representations for frequent sets. In J-F. Boulicaut, L. de Raedt, and H. Mannila, editors, *Constraint-based mining and inductive databases*, volume 3848 of *LNCS*. Springer, 2005.
12. Toon Calders. Complexity of and axiomatization for the freqsat problem. Technical Report 06-03, University of Antwerp, 2006.
13. Xia Chen and Maria E. Orlowska. A further study on inverse frequent set mining. In *Proc. ADMA Int. Conf. Advanced Data Mining and Applications*, pages 753–760, 2005.

14. V. Chvátal. Recognizing intersection patterns. *Annals of Discrete Mathematics - Combinatorics 79*, 8(I):249–251, 1980.
15. Nele Dexters, Paul W. Purdom, and Dirk Van Gucht. A probability analysis for candidate-based frequent itemset algorithms. In *Proceedings of the 2006 ACM Symposium on Applied Computing, DM track*, volume 1 of 2, pages 541 – 545, 2006.
16. A. M. Frisch and P. Haddawy. Anytime deduction for probabilistic logic. *Artificial Intelligence*, 69(1,2):93–112, 1994.
17. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, New York, 1979.
18. T. Hailperin. *Sentential Probability Logic.* Lehigh University Press, 1996.
19. P. Hansen and B. Jaumard. Probabilistic satisfiability. Les Cahiers du GERAD G-96-31, GERAD, 1996.
20. P. Hansen, B. Jaumard, G.-B. D. Nguets, and M. P. de Aragäo. Models and algorithms for probabilistic and bayesian logic. In *Proc. IJCAI Int. Joint Conf. Artificial Intelligence*, pages 1862–1868, Montreal, Canada, 1995.
21. T. Lukasiewicz. Probabilistic logic programming with conditional constraints. INFSYS Research Report 1843-00-01, Institut für Informationssysteme, Abteilung Wissenbasierte Systeme, 2000.
22. T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Transactions on Computational Logic*, 2(3):289–339, 2001.
23. H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations. In *Proc. KDD Int. Conf. Knowledge Discovery in Databases*, 1996.
24. Taneli Mielikäinen. On inverse frequent set mining. In *2nd Workshop on Privacy Preserving Data Mining (PPDM)*, 2003.
25. N. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–87, 1986.
26. C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
27. J. B. Paris. *The Uncertain Reasoner's Companion.* Tracts in Theoretical Computer Science 39. Cambridge University Press, 1994.
28. Y. Wang and X. Wu. Approximate inverse frequent itemset mining: Privacy, complexity, and approximation. In *Proc. IEEE Int. Conf. on Data Mining*, 2005.
29. Xintao Wu, Ying Wu, Yongge Wang, and Yingjiu Li. Privacy aware market basket data set generation: A feasible approach for inverse frequent set mining. In *Proc. SIAM Int. Conf. on Data Mining*, 2005.
30. Guizhen Yang. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 344–353, New York, NY, USA, 2004. ACM Press.