Reductions for Frequency-Based Data Mining Problems

Stefan Neumann*
University of Vienna
Vienna, Austria
Email: stefan.neumann@univie.ac.at

Pauli Miettinen
Max Planck Institute for Informatics
Saarland Informatics Campus, Germany
Email: pauli.miettinen@mpi-inf.mpg.de

***Abstract*—Computational complexity of various maximal pattern mining problems, including maximal frequent items, maximal frequent subgraphs in labelled graphs, and maximal subsequences with no repetitions, is studied, and the complexities are found to be equivalent under novel constrained reductions. The results extend those of Kimelfeld and Kolaitis [ACM TODS, 2014].**

I. Introduction

The computational complexity of central data mining problems is surprisingly little studied. This is especially true for the *frequency-based problems*, that is, for problems where the goal is to enumerate all sufficiently frequent patterns (that admit other possible constraints). Problems such as frequent itemset, subgraph, or subsequence mining all belong to this family of problems. Often the only computational complexity argument for these problems is that the output can be exponentially large w.r.t. the input, and hence any algorithm might need exponential time to enumerate the results.

This view is too limited for two reasons. First, there are more fine-grained models of complexity than just the running time as a function of the input. For enumeration problems we can use the framework of Johnson et al. [1]: instead of studying the total running time w.r.t. the input size, we consider it as a function of the total size of input *and output*, or study the time it takes to create a *new* pattern when a set of patterns is already known (see Section III for details). This allows us to argue about the time complexity of enumeration problems with potentially exponential output sizes. Another approach is the counting complexity framework of Valiant [2] (see Section III).

The second reason why we argue that the "output is exponential" is a too limited view for the computational complexity is that we also want to explore the relationships between the problems, that is, questions like "can we solve problem X efficiently if we can solve problem Y efficiently?" The main tool for answering such questions are *reductions* between problems. In this work, we introduce a new type of reduction between frequency-based problems called *maximality-preserving reduction* (see Section IV). Our reduction maps the maximal patterns of one problem to the maximal patterns of the other problem, thus allowing us to study questions like "can we find the maximal frequent subgraphs on labelled graphs using maximal frequent itemset mining algorithms?" Surprisingly, the answer to this question is positive, although it requires that we consider specially constrained maximal frequent pattern mining problems; we call the general class of such problems *feasible frequency-based problems* (see Section V).

Our Contributions: We study a number of maximal pattern mining problems, including maximal subgraph mining in labelled graphs, maximal frequent itemset mining, and maximal subsequence mining with no repetitions (see Section II for definitions). Figure 1 summarizes our results: the arrows show which problem can be reduced to which other problem either using non-constraining reductions (black and red lines), or with constraints on the feasible solutions (dashed lines). The figure shows that all problems can be reduced to each other (potentially with constraints). Given that the constrained reductions are transitive (Lemma 5), we obtain our main result:

Theorem 1 (Informal)**.** *Maximal subgraph mining in labelled graphs, maximal frequent itemset mining, and maximal subsequence mining with no repetitions are equally hard problems when we are allowed to constrain the pattern space.*

In some sense, our results unify all existing hardness results for frequency-based problems by putting them into a general framework using maximality-preserving reductions. These reductions preserve all interesting theoretical aspects like NP- or #P-hardness, but are still restricted enough to maintain the special properties of the transactions.

Due to space constraints, some proofs and empirical evaluation are postponed to the complete technical report [3].

II. Preliminaries

We define frequency-based problems, enumeration problems, and counting complexity. We further present the problems we consider in the paper.

Frequency-based Problems: A *frequency-based* problem $\mathcal{P}$ consists of: (1) A set of labels $\mathcal{L}$; for example, $\mathcal{L} = \{1, \dots, n\}$. (2) A set *trans*$(\mathcal{P})$ consisting of possible transactions over the labels $\mathcal{L}$. (3) A set *patterns*$(\mathcal{P}) \subseteq$ *trans*$(\mathcal{P})$ of possible patterns over the labels $\mathcal{L}$. (4) A partial order $\sqsubseteq$ over *trans*$(\mathcal{P})$. A similar definition was given by Gunopulos et al. [4].

*S.N. gratefully acknowledges the financial support from the Doctoral Programme "Vienna Graduate School on Computational Optimization" which is funded by the Austrian Science Fund (FWF, project no. W1260-N35).

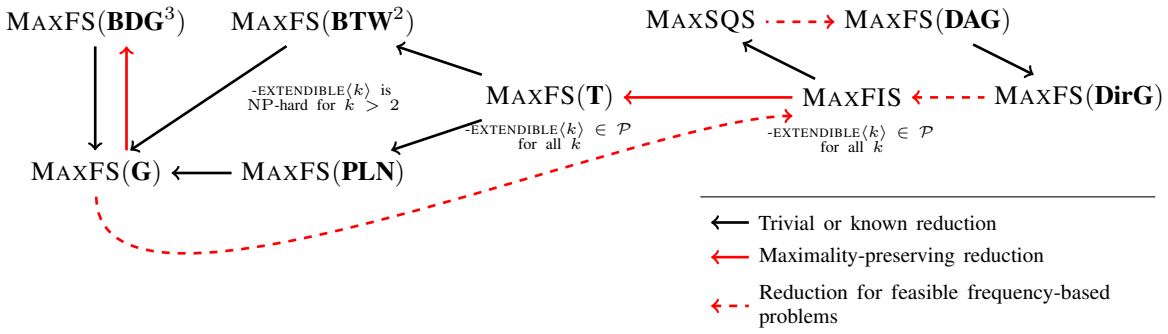

Fig. 1. The hierarchy of maximal frequency-based problems with this paper's results. Arrows point from the "easier" to the "harder" problem. See Section II for the abbreviated problem names. Maximality-preserving reductions are defined in Section IV; feasible frequency-based problems are defined in Section V.

Given a frequency-based problem $\mathcal{P}$, a *database* $D_\mathcal{P}$ is a finite multiset of elements from *trans*$(\mathcal{P})$. For a database $D_\mathcal{P}$ and a *support threshold* τ, a pattern $p \in$ *patterns*$(\mathcal{P})$ is called τ*-frequent* if $\text{supp}(p, D_\mathcal{P}) := |\{t \in D_\mathcal{P} : p \sqsubseteq t\}| \geq \tau$. In other words, a pattern p is frequent if it appears in at least τ transactions of the database. When τ is clear from the context, we will call p only *frequent*. A pattern $p \in$ *patterns*$(\mathcal{P})$ is a *maximal frequent* pattern if p is frequent and all patterns $q \in$ *patterns*$(\mathcal{P})$ with $p \sqsubset q$ are not frequent. Given a database $D_\mathcal{P}$, we denote the set of all maximal frequent patterns by $\text{Max}(D_\mathcal{P}, \tau)$, i.e., $\text{Max}(D_\mathcal{P}, \tau) = \{p \in$ *patterns*$(\mathcal{P}) : p$ is a maximal τ-frequent pattern in $D_\mathcal{P}\}$.

When the parameter τ is not part of the input but fixed to some integer, we write $\mathcal{P}^\tau$ to denote the resulting problem.

Enumeration Problems: An *enumeration relation* $\mathcal{R}$ is a set of strings $\mathcal{R} = \{(x, y)\} \subset \{0,1\}^* \times \{0,1\}^*$ s.t. the set $\mathcal{R}(x) := \{y \in \{0,1\}^* : (x, y) \in \mathcal{R}\}$ is finite for every x. A string $y \in \mathcal{R}(x)$ is called a *witness* for x. We call $\mathcal{R}$ an NP-*relation* if (1) there exists a polynomial p such that $|y| \leq p(|x|)$ for all $(x, y) \in \mathcal{R}$, and (2) there exists a polynomial-time algorithm deciding if $(x, y) \in \mathcal{R}$ for any given pair (x, y).

Following [5], we define the following problems for an enumeration relation $\mathcal{R}$:

- $\mathcal{R}$-ENUMERATE: The input is a string x. The task is to output the set $\mathcal{R}(x)$ without repetitions.
- $\mathcal{R}$-EXTEND: The input is a string x and a set $Y \subseteq \mathcal{R}(x)$. The task is to compute a string y such that $y \in \mathcal{R}(x) \setminus Y$ or to output that no such element exists.
- $\mathcal{R}$-EXTENDIBLE: The input is a string x and a set $Y \subseteq \mathcal{R}(x)$. The task is to decide whether $\mathcal{R}(x) \setminus Y \neq \emptyset$.
- $\mathcal{R}$-EXTENDIBLE$\langle k \rangle$: The input is a string x and a set $Y \subseteq \mathcal{R}(x)$ with the restriction that $|Y| < k$. The task is to decide whether $\mathcal{R}(x) \setminus Y \neq \emptyset$.

$\mathcal{R}$-EXTEND is the decision version of $\mathcal{R}$-EXTEND. Repeatedly running an algorithm for $\mathcal{R}$-EXTEND solves $\mathcal{R}$-ENUMERATE. An algorithm solving $\mathcal{R}$-EXTEND solves $\mathcal{R}$-EXTENDIBLE.

Enumeration Complexity: Johnson et al. [1] introduced different notions for the complexity of enumeration problems. Let $\mathcal{R}$ be an enumeration relation. An algorithm solving $\mathcal{R}$-ENUMERATE is called an *enumeration algorithm*.

For enumeration problems it might be the case that the output $\mathcal{R}(x)$ is exponentially larger than the input x. Due to this, measuring the running time of an enumeration algorithm only as a function of $|x|$ can be too restrictive; instead, one can include the size of $\mathcal{R}(x)$ in the complexity analysis. Then the running time of an algorithm is measured as function of $|x| + |\mathcal{R}(x)|$. This gives rise to the following definitions.

Let $\mathcal{A}$ be an enumeration algorithm. $\mathcal{A}$ runs in *total polynomial time* if its running time is polynomial in $|x|+|\mathcal{R}(x)|$. $\mathcal{A}$ has *polynomial delay* if the time spent between outputting two consecutive witnesses of $\mathcal{R}(x)$ is always polynomial in $|x|$. $\mathcal{A}$ runs in *incremental polynomial time* if on input x and after outputting a set $Y \subseteq \mathcal{R}(x)$ it takes time polynomial in $|x| + |Y|$ to produce the next witness from $\mathcal{R}(x) \setminus Y$.

Note that $\mathcal{R}$-ENUMERATE is in incremental polynomial time iff $\mathcal{R}$-EXTEND is in polynomial time. Further, a polynomial total time algorithm can be used to decide if $\mathcal{R}(x) \neq \emptyset$.

Relationship to Frequency-Based Problems: We note that frequency-based problems are special cases of enumeration problems. Let $\mathcal{P}$ be a frequency-based problem. We define the enumeration relation $\mathcal{R}$ corresponding to $\mathcal{P}$ by setting $\mathcal{R} = \{(x, y) : x = (D_\mathcal{P}, \tau), y \in \text{Max}(D_\mathcal{P}, \tau)\}$, i.e., $\mathcal{R}$ consists of all possible databases $D_\mathcal{P}$, support thresholds τ and all maximal frequent patterns y for the tuples $(D_\mathcal{P}, \tau)$.

Observe that $\mathcal{R}(x) = \mathcal{R}(D_\mathcal{P}, \tau) = \text{Max}(D_\mathcal{P}, \tau)$ and, hence, the problem $\mathcal{R}$-ENUMERATE is exactly the same problem as outputting all maximal frequent patterns in $\text{Max}(D_\mathcal{P}, \tau)$. The problem $\mathcal{R}$-EXTEND is to output a maximal frequent pattern in $\text{Max}(D_\mathcal{P}, \tau) \setminus Y$ for a given set of maximal patterns Y. The problems $\mathcal{R}$-EXTENDIBLE and $\mathcal{R}$-EXTENDIBLE$\langle k \rangle$ are the corresponding decision versions of the problems.

As $\mathcal{R}$ and $\mathcal{P}$ yield the same enumeration problems, we write $\mathcal{P}$-ENUMERATE, $\mathcal{P}$-EXTENDIBLE, $\mathcal{P}$-EXTEND, $\mathcal{P}$-EXTENDIBLE$\langle k \rangle$. We write $\mathcal{P}$ to denote $\mathcal{P}$-ENUMERATE.

Counting Complexity: For an enumeration relation $\mathcal{R}$, the function $\#\mathcal{R} : \{0,1\}^* \to \mathbb{N}$ returns the number of witnesses for a given string, i.e., $\#\mathcal{R}(x) = |\mathcal{R}(x)|$ for $x \in \{0,1\}^*$. The complexity class #P (pronounced "sharp P") contains all functions $\#\mathcal{R}$ for which $\mathcal{R}$ is an NP-relation; it was introduced by Valiant [2]. A function $F\colon \{0,1\}^* \to \mathbb{N}$ is #P-*hard* if there

exists a Turing reduction from every function in #P to F.

For two NP-relations $\mathcal{R}, \mathcal{Q} : \{0,1\}^* \to \mathbb{N}$, a *parsimonious reduction from* $\#\mathcal{R}$ *to* $\#\mathcal{Q}$ is a polynomial-time computable function $f\colon \{0,1\}^* \to \{0,1\}^*$ such that $\#\mathcal{R}(x) = \#\mathcal{Q}(f(x))$ for all $x \in \{0,1\}^*$. Note that a parsimonious reduction from a #P-hard problem $\mathcal{R}$ to a problem $\mathcal{Q}$ implies that $\mathcal{Q}$ is #P-hard.

Observe that an algorithm solving $\mathcal{R}$-ENUMERATE can solve $\#\mathcal{R}$ by counting the number of witnesses in its output.

Problems Considered in This Paper: All problems considered in this paper are frequency-based problems where the goal is to enumerate all maximal patterns of the data. For brevity, we only define $\mathcal{L}$, *trans*$(\cdot)$, *patterns*$(\cdot)$, and $\sqsubseteq$ for each problem (see, e.g., [6] for more thorough definitions).

The *maximal frequent itemset mining* problem, denoted MAXFIS, is as follows: We have n labels $\mathcal{L} = \{1, \ldots, n\}$; *trans*(MAXFIS) and *patterns*(MAXFIS) are given by $2^{\mathcal{L}}$; $\sqsubseteq$ is the standard subset relationship $\subseteq$.

The *maximal frequent subsequence mining* problem, denoted MAXSQS: $\mathcal{L} = \{1, \ldots, n\}$ is the set of labels. A *sequence* $S = \langle S_1, \ldots, S_m \rangle$ of length m consists of events S_i with $S_i \in \mathcal{L}$; we require that *each label appears at most once per sequence*. We set *trans*(MAXSQS) and *patterns*(MAXSQS) to the sets consisting of all sequences of arbitrary lengths. For two sequences $S = \langle S_1, \ldots S_r \rangle$ and $T = \langle T_1, \ldots, T_k \rangle$, we have $T \sqsubseteq S$ if $k \leq r$ and there exist indices $1 \leq i_1 \leq \cdots \leq i_k \leq r$ s.t. $T_j = S_{i_j}$ for each $j = 1, \ldots, k$.

Let $\mathcal{G}$ be a class of *vertex-labelled* graphs, which contain each label at most once. The *maximal frequent subgraph mining* problem, MAXFS$(\mathcal{G})$, is as follows: We have n labels $\mathcal{L} = \{1, \ldots, n\}$; *trans*(MAXFS$(\mathcal{G})$) and *patterns*(MAXFS$(\mathcal{G})$) are given by all labelled graphs in $\mathcal{G}$ with labels from $\mathcal{L}$; $\sqsubseteq$ is the standard subgraph relationship for labelled graphs (i.e., we consider arbitrary subgraphs, not necessarily induced subgraphs).

In this paper we consider the following graph classes, all of which are labelled and connected: undirected trees (**T**); undirected graphs of bounded degree $\leq b$ (**BDG**^{b}); undirected graphs of bounded treewidth $\leq w$ (**BTW**^{w}); undirected planar graphs (**PLN**); general undirected graphs (**G**); directed acyclic graphs (**DAG**); and directed graphs (**DirG**).

We only consider labelled graphs *in which each label appears at most once*. In this setting, the subgraph isomorphism problem can be solved in polynomial-time. This a necessary since Kimelfeld and Kolaitis [7, Prop. 3.4] showed that for certain *unlabelled* graph classes $\mathcal{G}$, MAXFS$(\mathcal{G})$ is not an NP-relation.

III. Related Work

Counting Complexity: The study of counting problems was initiated when Valiant [2] introduced #P. Provan and Ball [8] showed #P-hardness for many graph problems. Later, more #P-hardness results were obtained [9], [10].

Johnson et al. [1] introduced polynomial total time, polynomial delay, and incremental polynomial time to obtain a better understanding of the complexity of enumeration problems.

Computational Complexity of Data Mining Problems: Gunopulos et al. [4] introduced a class of problems similar to frequency-based problems. For these problems they proved #P-hardness for mining frequent patterns, and provided an algorithm to mine maximal frequent sets.

Yang [11] proved that the following problems are #P-complete: MAXFIS, MAXFS(**T**), MAXFS(**G**), MAXSQS.

Boros et al. [12] showed that MAXFIS-EXTENDIBLE and MAXFIS-EXTEND are NP-complete.

Kimelfeld and Kolaitis [7] considered mining maximal frequent subgraphs from certain graph classes; their results distinguish the computational complexities of several graph mining problems. They proved the following results, which are also depicted in Figure 1: (1) For fixed k, the problem MAXFS(**T**)-EXTENDIBLE$\langle k \rangle$ can be solved in polynomial time. (2) For fixed τ, the problem MAXFS$^\tau(\mathcal{G})$-ENUMERATE can be solved in polynomial time for any class of graphs $\mathcal{G}$ from Section II. (3) The following problems are NP-complete: (a) MAXFS$(\mathcal{G})$-EXTENDIBLE for $\mathcal{G} \in \{\mathbf{G}, \mathbf{PLN}, \mathbf{BDG}^b, \mathbf{BTW}^w\}$ with $w \geq 1$ and $b \geq 3$; (b) MAXFS$(\mathcal{G})$-EXTENDIBLE$\langle k \rangle$ for $\mathcal{G} \in \{\mathbf{G}, \mathbf{PLN}, \mathbf{BDG}^b, \mathbf{BTW}^w\}$ with $w > 1$, $b > 2$, $k > 2$.

In [5], Kimelfeld and Kolaitis give computational hardness results for subgraph mining problems in which the set *patterns*() is a subset of *trans*().

Mining Maximal Frequent Patterns: Many algorithms were proposed to mine maximal frequent patterns from different types of data such as itemsets [13]–[15], subsequences [16], trees [17], [18], and general graphs [19]. However, the main focus of those papers was not to investigate the computational complexity of these problems. See (for example) the book by Aggarwal [6] for more references to algorithms for computing maximal frequent patterns.

Constraint-based Pattern Mining: Many algorithms were proposed to mine frequent patterns with constraints [20]–[26]. We refer to Han et al. [27] for details. Greco et al. [28] explored mining taxonomies of process models; this can be viewed as constraint-based pattern mining. The work on constraint programming for itemset mining by Raedt et al. [29], [30] can be used to mine frequency-based problems with constraints.

IV. Maximality-Preserving Reductions

We introduce maximality-preserving reductions and state some of their properties. We further prove reductions between the problems MAXFIS, MAXSQS, and MAXFS$(\mathcal{G})$ for $\mathcal{G} \in \{\mathbf{T}, \mathbf{BDG}^3, \mathbf{G}\}$. Combining our reductions with the statements from Section III, we obtain the following theorem.

Theorem 2. *Fix natural numbers k, τ.*

1) MAXFIS-EXTENDIBLE$\langle k \rangle$ *is in polynomial time.*
2) MAXFIS$^\tau$-ENUMERATE *is in polynomial time.*
3) MAXFS(**G**) *and* MAXFS(**BDG**^{3}) *have the same enumeration and counting complexities. More concretely, let* $\mathcal{P} \in \{\text{MAXFS}(\mathbf{G}), \text{MAXFS}(\mathbf{BDG}^3)\}$. *Then:*
   - $\mathcal{P}$-ENUMERATE *is* #P*-hard.*
   - $\mathcal{P}$-EXTENDIBLE *is* NP*-hard.*
   - *For $k > 2$,* $\mathcal{P}$-EXTENDIBLE$\langle k \rangle$ *is* NP*-hard.*
   - $\mathcal{P}^\tau$-ENUMERATE *is solvable in polynomial time.*

A. Definition and Properties

We define maximality-preserving reductions to make explicit which properties are required by reductions in order to be useful for understanding the complexity of frequency-based problems.

Definition 1. Let $\mathcal{P}, \mathcal{Q}$ be two frequency-based problems. Let $(D_\mathcal{P}, \tau)$ be an instance for $\mathcal{P}$. A *maximality-preserving reduction* from $\mathcal{P}$ to $\mathcal{Q}$ defines an instance $(D_\mathcal{Q}, \tau)$ using a polynomial-time computable injective function f: *trans*$(\mathcal{P}) \to$ *trans*$(\mathcal{Q})$ with the following properties:

1) $f($*patterns*$(\mathcal{P})) \subseteq$ *patterns*$(\mathcal{Q})$.
2) For $p, p' \in$ *trans*$(\mathcal{P})$, $p \sqsubseteq_\mathcal{P} p'$ iff $f(p) \sqsubseteq_\mathcal{Q} f(p')$.
3) The inverse f^{-1}: *trans*$(\mathcal{Q}) \to$ *trans*$(\mathcal{P})$ of f can be computed in polynomial time.
4) $p \in \textsc{Max}(D_\mathcal{P}, \tau)$ iff $f(p) \in \textsc{Max}(D_\mathcal{Q}, \tau)$, where $D_\mathcal{Q} = f(D_\mathcal{P}) = \{f(t) : t \in D_\mathcal{P}\}$. Additionally, for all $q \in \textsc{Max}(D_\mathcal{Q}, \tau)$, $f^{-1}(q)$ exists.

The properties can be interpreted as follows: Property 1 asserts that f maintains validity of patterns; this condition is necessary when *patterns*$(\mathcal{Q}) \subsetneq$ *trans*$(\mathcal{Q})$. Property 2 asserts that f maintains subset properties. Property 3 is necessary to recover patterns in $\mathcal{P}$ from those found in $\mathcal{Q}$. Property 4 requires that the maximal frequent patterns in $D_\mathcal{P}$ are the same as those in $D_\mathcal{Q}$ under the mapping f; here, the database $D_\mathcal{Q}$ is given by applying the function f to each transaction in $D_\mathcal{P}$.

Properties: Property 4 implies that there exists a bijective relationship between the maximal frequent patterns in $D_\mathcal{P}$ and in $D_\mathcal{Q}$. Hence, $|\textsc{Max}(D_\mathcal{P}, \tau)| = |\textsc{Max}(D_\mathcal{Q}, \tau)|$. This shows that maximality-preserving reductions are parsimonious reductions and that they preserve #P-hardness.

In fact, maximality-preserving reductions are slightly stronger than parsimonious reductions. They do not only preserve the *number* of maximal frequent patterns, but they enable us to *recover* the maximal frequent patterns in $D_\mathcal{P}$ from those in $D_\mathcal{Q}$: By injectivity of f and due to Property 4, $\textsc{Max}(D_\mathcal{P}, \tau)$ can be reconstruct in polynomial time from $\textsc{Max}(D_\mathcal{Q}, \tau)$. Hence, maximality-preserving reductions preserve properties of extendibility problems as discussed in Section II.

Further, by Property 2, the support of a pattern p in $D_\mathcal{P}$ is a lower bound on the support of $f(p)$ in $D_\mathcal{Q}$ (since for each transaction $t \in D_\mathcal{P}$ with $p \sqsubseteq t$, $f(p) \sqsubseteq f(t)$).

However, although the number of transactions and *maximal* frequent patterns in both databases remains the same, the number of *frequent* patterns in $D_\mathcal{Q}$ might be exponentially larger than the number of frequent patterns in $D_\mathcal{P}$.

B. Reductions

We present maximality-preserving reductions, some of which are similar to ones presented in, e.g., [5], [11]. We only prove Property 4 of maximality-preserving reductions. The proofs of Properties 1–3 follow from the definitions of f.

Lemma 3. *There exist maximality-preserving reductions between the following problems: (1) From* MaxFIS *to* MaxFS(**T**). *(2) From* MaxFIS *to* MaxSQS. *(3) From* MaxFS(**G**) *to* MaxFS(**BDG**^{3}).

Note that (3) is the tightest result we could hope for, since graphs with degree bounded by 2 are simply cycles or line graphs. In this short version of the paper, we only prove point (3); see the technical report [3] for the remaining proofs.

Proof. Construction of f. Let $G = (V, E)$ be a graph with unbounded degree of the vertices over labels $\mathcal{L} = \{1, \dots, n\}$. Denote the label of a vertex $v \in V$ by *label*(v). We construct a graph $G' = (V', E')$ with bounded degree 3 over the set of labels $\mathcal{L}' = \{1, \dots, n\}^2$.

Intuitively, the construction of f is picked such that every original vertex $v \in V$ is split into a line graph consisting of n vertices v_i, where each v_i has an additional non-line-graph-edge in G' iff vertices v and i share an edge in G.

Formally, for each vertex $v \in V$, we insert vertices $v_1, \dots, v_n$ into V' with edges (v_i, v_{i+1}) for $i = 1, \dots, n-1$. Each vertex v_i is labeled by $(\textit{label}(v), i)$. For each edge $(u, v) \in E$, we insert an edge $(u_{\textit{label}(v)}, v_{\textit{label}(u)})$ into G'.

Observe that the resulting graph $G' = f(G)$ indeed has bounded degree 3: Consider any vertex $v_i \in V'$. The vertex has at most 2 neighbors from the line graph $(v_1, \dots, v_n)$. The only additional edge it could have is to vertex $i_{\textit{label}(v)}$.

Maximality-preserving. Let $p \in \textsc{Max}(D_{\textsc{MaxFS}(\mathbf{G})}, \tau)$. We need to show that $f(p) \in \textsc{Max}(D_{\textsc{MaxFS}(\mathbf{BDG}^3)}, \tau)$. By construction of f, $\operatorname{supp}(f(p), D_{\textsc{MaxFS}(\mathbf{BDG}^3)}) = \operatorname{supp}(p, D_{\textsc{MaxFS}(\mathbf{G})})$; hence, $f(p)$ is frequent in $D_{\textsc{MaxFS}(\mathbf{BDG}^3)}$. It remains to show that $f(p)$ is maximal. For contradiction, suppose there is a maximal frequent pattern q with $f(p) \sqsubset q$ in $D_{\textsc{MaxFS}(\mathbf{BDG}^3)}$. Then q contains an edge (u_i, v_j) with $i = \textit{label}(v)$, $j = \textit{label}(u)$, which is not contained in $f(p)$.

Case 1: $u_i \in f(p)$ and $v_j \in f(p)$. Consider the graph $q' = f(p) \cup (u_i, v_j)$. Then $f^{-1}(q')$ exists and must be frequent in $D_{\textsc{MaxFS}(\mathbf{G})}$ by Property 2. Contradiction.

Case 2: Assume w.l.o.g. that $u_i \in f(p)$ and $v_j \notin f(p)$. Since q is maximal and by construction of f and $D_{\textsc{MaxFS}(\mathbf{BDG}^3)}$, q must contain the line graph L with vertices $v_1, \dots, v_n$. Consider the graph $q' = f(p) \cup (u_i, v_j) \cup L$. By construction of f and $D_{\textsc{MaxFS}(\mathbf{BDG}^3)}$, q' has a preimage $p' = f^{-1}(q')$ which is frequent and satisfies $p \sqsubset p'$. Contradiction.

Case 3: $u_i \notin f(p)$ and $v_j \notin f(p)$. Since q is connected and $f(p) \sqsubset q$, we only need to consider the first two cases.

The second part of Property 4 is implied by the previous case distinctions. Proving that $f(p) \in \textsc{Max}(D_{\textsc{MaxFS}(\mathbf{BDG}^3)}, \tau)$ implies $p \in \textsc{Max}(D_{\textsc{MaxFS}(\mathbf{G})}, \tau)$ is similar. □

V. Constraining the Set of Patterns

We generalize frequency-based problems by allowing to constrain the set of patterns using a feasibility function. We introduce maximality-preserving reductions for this problem class and prove that all problems discussed in this paper exhibit the same hardness after introducing the feasibility function.

A. Feasible Frequency-Based Problems

A *feasible frequency-based problem* (FFBP) $\mathcal{P}$ is a frequency-based problem with an additional polynomial-time computable operation ϕ: *patterns*$(\mathcal{P}) \to \{0, 1\}$ which can be described using constant space. The operation ϕ is part of the input;

this is the reason for restricting the description length of the function to constant size. We call ϕ the *feasibility function*.

Given a feasible frequency-based problem $\mathcal{P}$, $p \in$ *patterns*$(\mathcal{P})$ is a *feasible frequent pattern* (FFP) if p is frequent and $\phi(p) = 1$. The goal is to find all maximal FFPs; we denote the set of all FFPs by $\textsc{Max}(D_\mathcal{P}, \tau, \phi_\mathcal{P})$. We define MAXFFIS, MAXFSQS, and MAXFFS$(\mathcal{G})$ for a graph class $\mathcal{G}$ as before for maximal frequency-based problems.

FFBPs are generalizations of frequency-based problems since setting $\phi_\mathcal{P} \equiv 1$ gives the underlying frequency-based problem.

The main result of this section is given in the following theorem, which follows from the reductions presented later in this section and the results from Section III.

Theorem 4. *The* FFBP*-version of all problems discussed in this paper have the same enumeration and counting complexities. More concretely, for any* FFBP*-problem $\mathcal{P}$ discussed in this paper:*

- $\mathcal{P}$-ENUMERATE *is* #P*-hard.*
- $\mathcal{P}$-EXTENDIBLE *is* NP*-hard.*
- *For $k > 2$, the problem $\mathcal{P}$-*EXTENDIBLE$\langle k \rangle$ *is* NP*-hard.*
- *For fixed τ, the problem $\mathcal{P}^\tau$-*ENUMERATE *is solvable in polynomial time.*

Theorem 4 shows that the hierarchy given in Figure 1 for frequency-based problems collapses when a feasibility function is introduced. Since many practical algorithms (like the Apriori algorithm) for finding maximal frequent patterns allow to add such a feasibility function, our reductions give a theoretical justification why many such algorithms can be extended to a broader range of problems.

B. Maximality-Preserving Reductions for FFPPs

Definition 2. Let $\mathcal{P}, \mathcal{Q}$ be two FFBPs. Let $(D_\mathcal{P}, \tau, \phi_\mathcal{P})$ be an instance for $\mathcal{P}$. A *maximality-preserving reduction* from $\mathcal{P}$ to $\mathcal{Q}$ defines an instance $(D_\mathcal{Q}, \tau, \phi_\mathcal{Q})$ using a polynomial-time computable injective function $f\colon$ *trans*$(\mathcal{P}) \to$ *trans*$(\mathcal{Q})$ with the following properties:

1) $f($*patterns*$(\mathcal{P})) \subseteq$ *patterns*$(\mathcal{Q})$.
2) For $p, p' \in$ *trans*$(\mathcal{P})$, $p \sqsubseteq_\mathcal{P} p'$ iff $f(p) \sqsubseteq_\mathcal{Q} f(p')$.
3) The inverse $f^{-1}\colon$ *trans*$(\mathcal{Q}) \to$ *trans*$(\mathcal{P})$ of f can be computed in polynomial time.
4) $p \in \textsc{Max}(D_\mathcal{P}, \tau, \phi_\mathcal{P})$ iff $f(p) \in \textsc{Max}(D_\mathcal{Q}, \tau, \phi_\mathcal{Q})$, where $D_\mathcal{Q} = f(D_\mathcal{P}) = \{f(t) : t \in D_\mathcal{P}\}$. Additionally, for all $q \in \textsc{Max}(D_\mathcal{Q}, \tau, \phi_\mathcal{Q})$, $f^{-1}(q)$ exists.

Compared to Definition 1 we only changed Property 4 to assert that the maximal patterns are feasible. In general, the constructed function $\phi_\mathcal{Q}$ will depend on $\phi_\mathcal{P}$, f and f^{-1}.

Properties: We show that maximality-preserving reductions for FFBPs are transitive, which is the crucial property to argue that multiple reductions can be used in a row. We also show that if for two frequency-based problems $\mathcal{P}$ and $\mathcal{Q}$ there exists a maximality-preserving reduction from $\mathcal{P}$ to $\mathcal{Q}$, then there exists a reduction between their FFBP-versions.

Lemma 5. *(1) Let $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ be* FFBP *s. Assume there exist maximality-preserving reductions from $\mathcal{P}$ to $\mathcal{Q}$ via a function g and $\phi_\mathcal{Q}$, and from $\mathcal{Q}$ to $\mathcal{R}$ via a function h and $\phi_\mathcal{R}$. Then there exists a maximality-preserving reduction from $\mathcal{P}$ to $\mathcal{R}$.*

(2) Let $\mathcal{P}$ and $\mathcal{Q}$ be two frequency-based problems, and let $\mathcal{P}'$ and $\mathcal{Q}'$ be the FFBP*-versions of those problems. Suppose there exists a maximality-preserving reduction from $\mathcal{P}$ to $\mathcal{Q}$ via a mapping g. Then there exists a maximality-preserving reduction from $\mathcal{P}'$ to $\mathcal{Q}'$.*

Proof of (1). Let $(D_\mathcal{P}, \phi_\mathcal{P})$ be an instance for $\mathcal{P}$. We construct an instance (D^*, ϕ_*) for $\mathcal{R}$: Set $f\colon$ *trans*$(\mathcal{P}) \to$ *trans*$(\mathcal{R})$ to $f(p) = h(g(p))$ for $p \in$ *trans*$(\mathcal{P})$. For $r \in$ *patterns*$(\mathcal{R})$, we set $\phi_*(r) = 1$ iff the following four conditions are satisfied: (1) $h^{-1}(r)$ and $f^{-1}(r)$ exist; (2) $\phi_\mathcal{R}(r) = 1$; (3) $\phi_\mathcal{Q}(h^{-1}(r)) = 1$; and (4) $\phi_\mathcal{P}(f^{-1}(r)) = 1$.

We check the properties from Definition 2. Property 1 and Property 2 are satisfied since f is the composition g and h. Property 3 holds since $f^{-1} = g^{-1} \circ h^{-1}$ and both g^{-1} and h^{-1} can be computed in polynomial time.

The rest of the proof is devoted to proving Property 4.

Let $p \in \textsc{Max}(D_\mathcal{P}, \tau, \phi_\mathcal{P})$. Then p is feasible w.r.t. $\phi_\mathcal{P}$. By the reduction from $\mathcal{P}$ to $\mathcal{Q}$, $g(p) \in \textsc{Max}(D_\mathcal{Q}, \tau, \phi_\mathcal{Q})$, where $D_\mathcal{Q} = g(D_\mathcal{P})$. Note that $g(p)$ is feasible w.r.t. $\phi_\mathcal{Q}$. Using the reduction from $\mathcal{Q}$ to $\mathcal{R}$, we obtain $r := h(g(p)) \in \textsc{Max}(D_\mathcal{R}, \tau, \phi_\mathcal{R})$, where $D_\mathcal{R} = h(D_\mathcal{Q})$; additionally, r is feasible w.r.t. $\phi_\mathcal{R}$. Now observe that $r = f(p)$ and that r is feasible w.r.t. the operation ϕ_* defined above. Note that r is frequent in D^* since for each transaction $t \in D_\mathcal{P}$ with $p \sqsubseteq_\mathcal{P} t$, $r = f(p) \sqsubseteq_\mathcal{R} f(t)$ by Property 2 of f. To prove that $r \in \textsc{Max}(D^*, \tau, \phi_*)$, it remains to show that r is maximal. Suppose not. Then there exists a pattern $r' \in \textsc{Max}(D^*, \tau, \phi_*)$ such that $r \sqsubset_\mathcal{R} r'$. Since r' is feasible, let $p' = f^{-1}(r')$. By Property 2 of f, we have that $p \sqsubset_\mathcal{P} p'$ and that p' is frequent since $p' \sqsubset_\mathcal{P} t$ for $t \in D_\mathcal{P}$ iff $f(p') = r' \sqsubset_\mathcal{R} f(t)$. This contradicts the maximality of p. Hence, $r \in \textsc{Max}(D^*, \tau, \phi_*)$.

Let $r \in \textsc{Max}(D^*, \tau, \phi_*)$. Since r is feasible w.r.t. ϕ_*, there exists $p = f^{-1}(r) \in$ *patterns*$(\mathcal{P})$ that is feasible w.r.t. $\phi_\mathcal{P}$. By Property 2, p is frequent in $D_\mathcal{P}$. It remains to show that p is maximal. We argue by contradiction. Suppose there exists a frequent pattern p' with $p \sqsubset p'$. Then $f(p') \in \textsc{Max}(D_*, \tau, \phi_*)$ by the previous paragraph, and $r \sqsubset f(p')$ by Property 2 of f. This contradicts the maximality of r. Hence, $p \in \textsc{Max}(D_\mathcal{P}, \tau, \phi_\mathcal{P})$. □

Proof of (2). Construction of f. Set $f \equiv g$. Set $\phi_{\mathcal{Q}'}(q) = 1$ iff $f^{-1}(q)$ exists and $\phi_{\mathcal{P}'}(f^{-1}(q)) = 1$.

Maximality-preserving. Properties 1–3 for f are satisfied since they are satisfied for g. We prove Property 4 for f.

Let $p \in \textsc{Max}(D_\mathcal{P}, \tau, \phi_\mathcal{P})$. We show that $f(p) \in \textsc{Max}(D_\mathcal{Q}, \tau, \phi_\mathcal{Q})$. Since $f^{-1}(f(p)) = p$ is feasible w.r.t. $\phi_\mathcal{P}$, $f(p)$ is feasible w.r.t. $\phi_\mathcal{Q}$. By Property 2 of f, $f(p)$ is frequent in $D_\mathcal{Q}$. It remains to show that $f(p)$ is maximal. Suppose not. Then there is a pattern $q \in \textsc{Max}(D_\mathcal{Q}, \tau, \phi_\mathcal{Q})$ s.t. $f(p) \sqsubset q$. Since q is feasible, there exists a feasible pattern $p' = f^{-1}(q) \in$ *patterns*$(\mathcal{P})$. By Property 2, we have $p \sqsubset p'$. Additionally, the pattern p' is frequent in $D_\mathcal{P}$: for each transaction $t \in D_\mathcal{Q}$ with $q \sqsubset_\mathcal{Q} t$, $p' \sqsubset_\mathcal{P} f^{-1}(t)$ (by Property 2 of f and definition of $D_\mathcal{Q}$). Contradiction.

Let $q \in \text{MAX}(D_\mathcal{Q}, \tau, \phi_\mathcal{Q})$. Since q is feasible, $p = f^{-1}(q)$ exists and is feasible w.r.t. $\phi_\mathcal{P}$. We show that $p \in \text{MAX}(D_\mathcal{P}, \tau, \phi_\mathcal{P})$. By Property 2 of f, p is frequent in $D_\mathcal{P}$. We prove the maximality of p by contradiction. Suppose there exists a pattern $p' \in \text{MAX}(D_\mathcal{P}, \tau, \phi_\mathcal{P})$ with $p \sqsubset p'$. Then by the previous paragraph the pattern $f(p')$ is a feasible frequent pattern in $D_\mathcal{Q}$ with $q = f(p) \sqsubset f(p')$. Contradiction. □

C. Reductions

Lemma 6. *There exist maximality-preserving reductions between the following problems: (1) From* MAXFFS(**G**) *to* MAXFFIS. *(2) From* MAXFFS(**DirG**) *to* MAXFFIS. *(3) From* MAXFSQS *to* MAXFFS(**DAG**).

Due to space constraints, we only prove point (1) of the lemma; the technical report [3] contains the remaining proofs.

Proof of (1). Let $(D_{\text{MAXFFS}(\mathbf{G})}, \tau, \phi_{\text{MAXFFS}(\mathbf{G})})$ be an instance for MAXFFS(**G**) with graphs with labels from $\{1, \ldots, n\}$.

Construction of f. For MAXFFIS we use the labels $\mathcal{L} = \{1, \ldots, n\}^2$. Let $G = (V, E)$ be a graph from $D_{\text{MAXFFS}(\mathbf{G})}$. We construct an itemset $I(G) := f(G)$ by mapping the graph onto the labels of its edges, i.e., we construct an itemset $I(G) = \{(\mathit{label}(u), \mathit{label}(v)) : (u, v) \in E\}$.

For $I \in \mathit{patterns}(\text{MAXFFIS})$, we set $\phi_{\text{MAXFFIS}}(I) = 1$ iff (1) $f^{-1}(I)$ exists and $\phi_{\text{MAXFFS}(\mathbf{G})}(f^{-1}(I)) = 1$, and (2) for each pair of tuples $(a, b), (c, d) \in I$ there exists a sequence $(a, b) = (e_1, e'_1), \ldots, (e_k, e'_k) = (c, d)$ of tuples $(e_i, e'_i) \in I$ with the following property: For each pair of consecutive tuples (e_i, e'_i) and (e_{i+1}, e'_{i+1}), there exists some $\ell \in \{1, \ldots, n\}$ with $\ell \in \{e_i, e'_i\}$ and $\ell \in \{e_{i+1}, e'_{i+1}\}$. Intuitively, condition (2) of ϕ_{MAXFFIS} asserts that the graph corresponding I is connected.

Maximality-preserving. Note that any feasible frequent itemset in D_{MAXFFIS} corresponds to a frequent *connected* graph in $D_{\text{MAXFFS}(\mathbf{G})}$ due to the choice of ϕ_{MAXFFIS}. Observe that there exists a bijection between connected subgraphs G and feasible itemsets $I(G) \subseteq \mathcal{L}'$. Further observe that for two frequent subgraphs G and H, $G \subseteq H$ iff $f(G) \subseteq f(H)$. It follows that a graph G and an itemset I must have the same supports in $D_{\text{MAXFFS}(\mathbf{G})}$ and D_{MAXFFIS}, respectively. The maximality then follows from the subset-property we observed. □

VI. CONCLUSIONS

We showed that when considering a generalized version of frequency-based problems, FFBP, the computational hardness of many frequency-based problems collapses. Hence, our reductions provide a unifying framework for the existing computational hardness results of fundamental data mining problems. Additionally, our reductions give a formal explanation why algorithms similar to the Apriori algorithm can be used for such a wide range of problems by only slightly adjusting the candidate generation.

REFERENCES

[1] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, "On generating all maximal independent sets," *Inf. Proc. Lett.*, vol. 27, no. 3, pp. 119–123, 1988.

[2] L. G. Valiant, "The complexity of computing the permanent," *Theor. Comput. Sci.*, vol. 8, pp. 189–201, 1979.

[3] S. Neumann and P. Miettinen, "Reductions for frequency-based data mining problems," *CoRR*, vol. abs/1709.00900, 2017. [Online]. Available: http://arxiv.org/abs/1709.00900

[4] D. Gunopulos, R. Khardon, H. Mannila, S. Saluja, H. Toivonen, and R. S. Sharm, "Discovering all most specific sentences," *ACM Trans. Database Syst.*, vol. 28, no. 2, pp. 140–174, 2003.

[5] B. Kimelfeld and P. G. Kolaitis, "The complexity of mining maximal frequent subgraphs," *ACM Trans. Database Syst.*, vol. 39, no. 4, pp. 32:1–32:33, 2014.

[6] C. C. Aggarwal, *Data Mining - The Textbook*. Springer, 2015.

[7] B. Kimelfeld and P. G. Kolaitis, "The complexity of mining maximal frequent subgraphs," in *PODS*, 2013, pp. 13–24.

[8] J. S. Provan and M. O. Ball, "The complexity of counting cuts and of computing the probability that a graph is connected," *SIAM J. Comput.*, vol. 12, no. 4, pp. 777–788, 1983.

[9] H. B. Hunt, III, M. V. Marathe, V. Radhakrishnan, and R. E. Stearns, "The complexity of planar counting problems," *SIAM J. Comput.*, vol. 27, no. 4, pp. 1142–1167, 1998.

[10] S. P. Vadhan, "The complexity of counting in sparse, regular, and planar graphs," *SIAM J. Comput.*, vol. 31, no. 2, pp. 398–427, 2001.

[11] G. Yang, "The complexity of mining maximal frequent itemsets and maximal frequent patterns," in *KDD*, 2004, pp. 344–353.

[12] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino, "On maximal frequent and minimal infrequent sets in binary matrices," *Ann. Math. Artif. Intell.*, vol. 39, no. 3, pp. 211–221, 2003.

[13] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Min. Knowl. Discov.*, vol. 8, no. 1, pp. 53–87, 2004.

[14] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, "MAFIA: A maximal frequent itemset algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1490–1504, 2005.

[15] R. J. Bayardo, Jr., "Efficiently mining long patterns from databases," in *SIGMOD*, 1998, pp. 85–93.

[16] R. Agrawal and R. Srikant, "Mining sequential patterns," in *ICDE*, 1995, pp. 3–14.

[17] M. J. Zaki, "Efficiently mining frequent trees in a forest: Algorithms and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 8, pp. 1021–1035, 2005.

[18] Y. Xiao, J. Yao, Z. Li, and M. H. Dunham, "Efficient data mining for maximal frequent subtrees," in *ICDM*, 2003, pp. 379–386.

[19] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *ICDM*, 2001, pp. 313–320.

[20] R. T. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang, "Exploratory mining and pruning optimizations of constrained association rules," in *SIGMOD*, 1998, pp. 13–24.

[21] G. Grahne, L. V. S. Lakshmanan, and X. Wang, "Efficient mining of constrained correlated sets," in *ICDE*, 2000, pp. 512–521.

[22] F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, "Exante: Anticipated data reduction in constrained pattern mining," in *PKDD*, 2003, pp. 59–70.

[23] F. Bonchi and C. Lucchese, "On closed constrained frequent pattern mining," in *ICDM*, 2004, pp. 35–42.

[24] M. N. Garofalakis, R. Rastogi, and K. Shim, "SPIRIT: sequential pattern mining with regular expression constraints," in *VLDB*, 1999, pp. 223–234.

[25] J. Pei, J. Han, and W. Wang, "Mining sequential patterns with constraints in large databases," in *CIKM*, 2002, pp. 18–25.

[26] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, and R. Trasarti, "A constraint-based querying system for exploratory pattern discovery," *Inf. Syst.*, vol. 34, no. 1, pp. 3–27, 2009.

[27] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Min. Knowl. Discov.*, vol. 15, no. 1, pp. 55–86, 2007.

[28] G. Greco, A. Guzzo, and L. Pontieri, "Mining taxonomies of process models," *Data Knowl. Eng.*, vol. 67, no. 1, pp. 74–102, 2008.

[29] L. D. Raedt, T. Guns, and S. Nijssen, "Constraint programming for itemset mining," in *KDD*, 2008, pp. 204–212.

[30] T. Guns, S. Nijssen, and L. D. Raedt, "k-pattern set mining under constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 402–418, 2013.