Counting Minimal Transversals of ß-Acyclic Hypergraphs

Benjamin Bergougnoux, Florent Capelli, Mamadou Moustapha Kanté

Counting Minimal Transversals of β-Acyclic Hypergraphs☆

Benjamin Bergougnoux[a], Florent Capelli[b], Mamadou Moustapha Kanté[a]

[a] *Université Clermont Auvergne, LIMOS, CNRS, France*
[b] *Birbeck College, University of London, UK*

Abstract

We prove that one can count in polynomial time the number of minimal transversals of β-acyclic hypergraphs. As consequence, we can count in polynomial time the number of minimal dominating sets of strongly chordal graphs, continuing the line of research initiated in [M.M. Kanté and T. Uno, Counting Minimal Dominating Sets, TAMC'17].

Keywords: Counting problem, Minimal transversals, Dominating sets, β-Acyclic Hypergraphs, Strongly chordal graphs

Introduction

A *hypergraph* is a collection of subsets - called *hyperedges* - of a finite ground set, and a *transversal* is a subset of the ground set that intersects every hyperedge. In this short note, we consider the problem of counting the (inclusion-wise) minimal transversals of *β-acyclic hypergraphs*. Counting problems are usually harder than decision problems as for instance computing a (minimal) transversal of a hypergraph can be done in polynomial time while counting the number of (minimal) transversals is #P-complete [1]. As for decision problems, the theory [2] asks for a classification of counting problems into easy ones and those that are hard (or even hard to approximate) (see for instance [3]).

The Hypergraph Dualisation problem - a fifty years open problem - asks for the enumeration of all (inclusion-wise) minimal transversals of a given hypergraph in output-polynomial time and has several applications in several areas such as graph theory, artificial intelligence, datamining, model-checking, network modeling, databases, etc. and is extensively studied (see the survey [4]). The interest in the minimal transversals is twofold: being a transversal is anti-monotone and so we can obtain all the transversals from the minimal ones,

☆The authors are supported by the French Agency for Research under the GraphEN and AGGREG projects.

Email addresses: `benjamin.bergougnoux@uca.fr` (Benjamin Bergougnoux), `florent@dcs.bbk.ac.uk` (Florent Capelli), `mamadou.kante@uca.fr` (Mamadou Moustapha Kanté)

and second there may be sometimes an exponential gap between the number of minimal transversals and the number of transversals (see for instance affine formulas). Because of the broad application of the Hypergraph Dualisation problem and also of the importance of counting in those areas (see for instance the description given in [5] in the case of model checking), the complexity of counting the minimal transversals has been investigated in general as well as in special cases (see the PhD thesis [6] for some references).

A *dominating set* in a graph is a subset of vertices that intersects the closed neighborhood of every vertex. Dominating Set problems are classic and well-studied graph problems, and has applications in many areas such as networks and graph theory [7]. In [8] the authors reduce the Hypergraph Dualisation problem into the enumeration of minimal dominating sets, showing that the two problems are equivalent in the area of enumeration problems (a fact already established in the case of optimisation). The reduction indeed shows that the counting versions are equivalent (under Turing reductions), and such a reduction is of big interest because it allows to study counting and enumeration problems associated with the Hypergraph Dualisation in the perspectives of graph theory, where tools had been developed to tackle combinatorial problems.

Despite the broad application of counting the minimal dominating sets in (hyper)graphs, the problem was not investigated until recently, except in [9] where it is proved that the models of any monadic second-order formula can be counted in polynomial time in graphs of bounded clique-width. Indeed, as far as we know the counting of minimal dominating sets is only considered in [10, 11, 12], and the systematic study of its computational complexity in graph classes is only considered in [12], where the authors proved the #P-completeness in several graph classes and asked whether the following dichotomy conjecture is true. A k-sun is a graph obtained from a cycle of length $2k$ ($k \geq 3$) by adding edges to make the even-indexed vertices pairwise adjacent.

Conjecture 1. *Let $\mathcal{C}$ be a class of chordal graphs. If $\mathcal{C}$ does not contain a k-sun as an induced subgraph, for $k \geq 4$, then one can count in polynomial time the number of minimal dominating sets of any graph in $\mathcal{C}$. Otherwise, the problem is #P-complete.*

We make a first step towards a proof of the first statement of the conjecture and provide a polynomial time algorithm for computing the minimal dominating sets in strongly chordal graphs, which are exactly chordal graphs without k-suns, for $k \geq 3$. In fact, our algorithm counts the minimal transversals of β-acyclic hypergraphs. We can use it to count the number of minimal dominating sets in strongly chordal graphs because the minimal dominating sets of a strongly chordal graph are exactly the minimal transversals of the *closed neighborhood hypergraph*, which is know to be β-acyclic [13].

Besides the polynomial time algorithm, the main contribution of this short note is the modification of the framework considered in [14] in order to count minimal models. The techniques used in [10, 11] are based on structural restrictions and as shown in [14] cannot work for strongly chordal graphs. Instead, Capelli showed in [14] how to construct, from the elimination ordering of a β-acyclic hypergraph associated to a boolean formula, a circuit whose satisfying assignments correspond to the models of the boolean formula. Such circuits are known as *decision Decomposable Negation Normal Form* in knowledge compilation. While the technique allows to count the models of non-monotone formulas, it cannot be used to count the minimal models. Indeed, the branchings of the constructed circuit do not allow to control the minimality. We overcome this difficulty by introducing the notion of *blocked transversals*, which correspond roughly to the minimal transversals of a sub-hypergraph that are transversals of the whole hypergraph. We then show that blocked transversals can be used to control the minimality in the construction of the circuit. However, this control is only possible in the case of monotone Boolean formulas, corresponding to counting the minimal transversals of β-acyclic hypergraphs.

Because of technical definitions, we postpone the details of the algorithm in Section 3.2. The paper is organised as follows. Notations and some technical definitions are given in Section 1, while *blocked transversals* and intermediate lemmas are given in Section 2. The decomposition of β-acyclic hypergraphs proposed in [14] is refined in Section 3.1 to take into account *blocked transversals*. Finally, the algorithm is given in Section 3.2.

1. Definitions and notations

For an integer n, we let $[n]$ denote the set $\{1, 2, \ldots, n\}$. The power set of a set V is denoted by 2^V, and its cardinal is denoted by $\#V$. For two sets A and B, we let $A \setminus B$ denote the set $\{x \in A \mid x \notin B\}$. For a ground set V and subsets $\mathcal{A}_1, \ldots, \mathcal{A}_k$ of 2^V, with $k \geq 2$, we let

$$\bigotimes_{1\leq i\leq k} A_i := \begin{cases} \varnothing & \text{if } \mathcal{A}_i = \varnothing \text{ for some } 1 \leq i \leq k, \\ \{\bigcup_{1\leq i\leq k} T_i \mid T_i \in \mathcal{A}_i\} & \text{otherwise.} \end{cases}$$

If $k = 2$, we write $\mathcal{A}_1 \bigotimes \mathcal{A}_2$ as usual.

1.1. Hypergraphs

A hypergraph $\mathcal{H}$ is a collection of subsets of a finite ground set. The elements of $\mathcal{H}$ are called the *hyperedges* of $\mathcal{H}$ and the *vertex set* of $\mathcal{H}$ is $V(\mathcal{H}) := \bigcup_{e\in\mathcal{H}} e$. Given a set $S \subseteq V(\mathcal{H})$, we denote by $\mathcal{H}(S)$ the set of edges of $\mathcal{H}$ containing a least one vertex in S, that is, $\mathcal{H}(S) := \{e \in \mathcal{H} \mid S \cap e \neq \varnothing\}$; and for $x \in V(\mathcal{H})$, we write $\mathcal{H}(x)$ instead of $\mathcal{H}(\{x\})$. Given $S \subseteq V(\mathcal{H})$, we denote by $\mathcal{H}[S]$ the *hypergraph induced by* S, that is, $\mathcal{H}[S] := \{e \cap S \mid e \in \mathcal{H}\}$. Any subset $\mathcal{H}'$ of $\mathcal{H}$ is called a *sub-hypergraph* of $\mathcal{H}$.

Observe that if there exists $e \in \mathcal{H}$ such that $e \subseteq V(\mathcal{H}) \setminus S$, then $\varnothing \in \mathcal{H}[S]$. We do not enforce hypergraphs to have non-empty edges or to be non-empty. However, a hypergraph with an empty edge may behave counter-intuitively. In the following definitions, we explicitly explain the extremal cases where $\varnothing \in \mathcal{H}$ or $\mathcal{H} = \varnothing$.

Given a hypergraph $\mathcal{H}$, a walk between two distinct edges e_1 and e_k is a sequence $(e_1, x_1, e_2, x_2, \ldots, e_{k-1}, x_{k-1}, e_k)$ such that $x_i \in e_i \cap e_{i+1}$ for all $1 \leq i \leq k-1$. Notice that, $(e_k, x_{k-1}, e_{k-1}, \ldots, x_2, e_2, x_1, e_1)$ is also a walk between e_k and e_1. A maximal set of edges of $\mathcal{H}$ that are pairwise connected by a walk is called a *connected component* of $\mathcal{H}$. It is worth noticing that if $C_1, \ldots, C_k$ are the connected components of $\mathcal{H}$, then $V(C_i) \cap V(C_j) = \varnothing$ for distinct i, j in $\{1, \ldots, k\}$.

A hypergraph $\mathcal{H}$ is said *β-acyclic* if there exists an ordering $x_1, \ldots, x_n$ of $V(\mathcal{H})$ such that for each $1 \leq i \leq n$, the set $\{e \cap \{x_i, \ldots, x_n\} \mid e \in \mathcal{H},\ x_i \in e\}$ is linearly ordered by inclusion. It is well-known that if $\mathcal{H}$ is β-acyclic, then are all the sub-hypergraphs $\mathcal{H}'$ of $\mathcal{H}$ (see for instance [15]). Such an ordering is called a *β-elimination ordering.*

1.2. Transversals

Let $\mathcal{H}$ be a hypergraph. A transversal for $\mathcal{H}$ is a subset $T \subseteq V(\mathcal{H})$ such that for every $e \in \mathcal{H}$, $T \cap e \neq \varnothing$. We denote by $\mathsf{tr}(\mathcal{H})$ the set of transversals of $\mathcal{H}$. Observe that if $\varnothing \in \mathcal{H}$, then $\mathsf{tr}(\mathcal{H}) = \varnothing$ as for every $T \subseteq V(\mathcal{H})$, $\varnothing \cap T = \varnothing$ so T cannot be a transversal of $\mathcal{H}$. Finally, observe that if $\mathcal{H} = \varnothing$, then $\mathsf{tr}(\mathcal{H}) = \{\varnothing\}$.

A transversal T of $\mathcal{H}$ is *minimal* if and only if for every $x \in T$, it holds that $T \setminus \{x\} \notin \mathsf{tr}(\mathcal{H})$. It is easy to see that a transversal T of $\mathcal{H}$ is minimal if and only if for every $x \in T$, there exists $e \in \mathcal{H}$ such that $T \cap e = \{x\}$. A hyperedge e such that $e \cap T = \{x\}$ is said to be *private for x* w.r.t. T. When T is clear from the context, we may refer to such hyperedges as simply private for x. We denote by $\mathsf{mtr}(\mathcal{H})$ the set of minimal transversals of $\mathcal{H}$. Again, observe that if $\mathcal{H} = \varnothing$ then $\mathsf{mtr}(\mathcal{H}) = \{\varnothing\}$.

2. Blocked transversals

Given a hypergraph $\mathcal{H}$ and a vertex $x \in V(\mathcal{H})$, it is easy to check that $\#mtr(\mathcal{H})$ is equal to the sum of $\#mtr(\mathcal{H}[V(\mathcal{H}) \setminus \{x\}])$ and of $\#mtr(\mathcal{H} \setminus \mathcal{H}(x)) - \#(tr(\mathcal{H}) \cap mtr(\mathcal{H} \setminus \mathcal{H}(x)))$. Now, given an ordering of the vertex set of $\mathcal{H}$, we can try to recursively do the same thing in the hypergraph $\mathcal{H} \setminus \mathcal{H}(x)$, which may unfortunately yield an exponential number of computations. The notion of *blocked transversal* introduced in this section is intended to show how to recursively do the computation. We then show in the next section that this recursive definition coupled with the β-elimination ordering yields a polynomial time algorithm in β-acyclic hypergraphs.

Given a hypergraph $\mathcal{H}$, a sub-hypergraph $\mathcal{H}' \subseteq \mathcal{H}$ and $B \subseteq V(\mathcal{H})$, we define the *$B$-blocked transversals* of $\mathcal{H}'$ to be the transversals of $\mathcal{H}'$ that are minimal for $\mathcal{H}' \setminus \mathcal{H}'(B)$. In particular, if y is a vertex of B then y cannot be in a B-blocked transversal of $\mathcal{H}'$. Observe also that if $y \in V(\mathcal{H}) \setminus V(\mathcal{H}')$, then y cannot be in a B-blocked transversal.

In other words, a transversal T of $\mathcal{H}'$ is a B-blocked transversal if and only if for every $x \in T$, there exists $e \in \mathcal{H}' \setminus \mathcal{H}'(B)$ such that $e \cap T = \{x\}$. We call the set $\mathcal{H}'(B)$ the *blocked hyperedges*. Intuitively, $\mathcal{H}'(B)$ are the set of hyperedges that cannot be used as private in a transversal. We denote by $\mathsf{btr}_B(\mathcal{H}')$ the set of B-blocked transversals of $\mathcal{H}'$. In symbols:

$$\mathsf{btr}_B(\mathcal{H}') := \mathsf{tr}(\mathcal{H}') \cap \mathsf{mtr}(\mathcal{H}' \setminus \mathcal{H}'(B)).$$

Observe that by definition, $\mathsf{mtr}(\mathcal{H}) = \mathsf{btr}_\varnothing(\mathcal{H})$. Moreover, if $\mathcal{H}(B) = \mathcal{H} \neq \varnothing$, then $\mathsf{btr}_\mathcal{H}(\mathcal{H}) = \varnothing$ as $\mathsf{mtr}(\varnothing) = \{\varnothing\}$ and $\varnothing \notin \mathsf{tr}(\mathcal{H})$. When $B = \{x\}$, we denote $\mathsf{btr}_B(\mathcal{H})$ by $\mathsf{btr}_x(\mathcal{H})$. Observe that if $\mathcal{H}' \subseteq \mathcal{H}$, then $\mathsf{btr}_B(\mathcal{H}') = \mathsf{btr}_{B \cap V(\mathcal{H}')}(\mathcal{H}')$.

Given $S \subseteq V(\mathcal{H})$, we denote by $\mathsf{tr}(\mathcal{H}, S) := \{T \in \mathsf{tr}(\mathcal{H}) \mid T \subseteq S\}$. We extend this notation to mtr and btr as well. Observe that, for all $S, B \subseteq \mathcal{H}$, we have $\mathsf{btr}_B(\mathcal{H}, S) = \mathsf{btr}_B(\mathcal{H}, S \setminus B)$. Moreover, if $x \notin V(\mathcal{H})$ then $\mathsf{btr}_B(\mathcal{H}, S) = \mathsf{btr}_B(\mathcal{H}, S \setminus \{x\})$.

The end of this section is dedicated to the proof of several crucial lemmas concerning the number of blocked transversals that will be useful in our algorithm. We start by describing the blocked transversals of a hypergraph having more than one connected component:

Lemma 2. *Let $\mathcal{H}$ be a hypergraph, $S, B \subseteq V(\mathcal{H})$ and $C_1, \dots, C_k$ the connected components of $\mathcal{H}[S]$. For each $i \in [k]$, let $\mathcal{H}_i = \{e \in \mathcal{H} \mid e \cap S \in C_i\}$. We have:*

$$\mathsf{btr}_B(\mathcal{H}, S) = \bigotimes_{i=1}^{k} \mathsf{btr}_B(\mathcal{H}_i, S).$$

Proof. Assume that $\varnothing \in \mathcal{H}[S]$, it means that there exists a hyperedge $e \in \mathcal{H}$ such that $e \cap S = \varnothing$. In this case, $\mathsf{btr}_B(\mathcal{H}, S) = \varnothing$. Moreover, there exists $i \in [1, k]$ such that $C_i = \{\varnothing\}$ and $e \in \mathcal{H}_i$. Thus, $\mathsf{btr}_B(\mathcal{H}_i, S) = \varnothing$ and the equality holds in this case.

Assume from now that $\varnothing \notin \mathcal{H}[S]$. Let $T \in \mathsf{btr}_B(\mathcal{H}, S)$. We show that for all $i \leq k$, $T_i = T \cap V(\mathcal{H}_i) \in \mathsf{btr}_B(\mathcal{H}_i, S)$. Let $e \in \mathcal{H}_i$. Since $e \in \mathcal{H}$, we have $e \cap T \neq \varnothing$. As $e \in \mathcal{H}_i$, we have $e \subseteq V(\mathcal{H}_i)$. Thus $e \cap T_i \neq \varnothing$, that is, $T_i \in \mathsf{tr}(\mathcal{H}_i, S)$. Moreover, let $y \in T_i$. By definition of T, there exists $e \in \mathcal{H} \setminus \mathcal{H}(B)$ such that e is private for y *w.r.t.* T. Observe that we have $T_i \subseteq S \cap V(\mathcal{H}_i) = V(C_i)$ since $T \subseteq S$. Thus, $y \in V(C_i)$ and $e \cap S \in C_i$, because C_i is a connected component. We can conclude that $e \in \mathcal{H}_i$ and e is private to y *w.r.t.* T_i and $\mathcal{H}_i \setminus \mathcal{H}(B) = \mathcal{H}_i \setminus \mathcal{H}_i(B)$. Thus T_i is a minimal transversal of $\mathcal{H}_i \setminus \mathcal{H}_i(B)$. That is $T_i \in \mathsf{btr}_B(\mathcal{H}_i, S)$.

Now let $T_1 \in \mathsf{btr}_B(\mathcal{H}_1, S), \dots, T_k \in \mathsf{btr}_B(\mathcal{H}_k, S)$. We show that $T = \bigcup_{i=1}^{k} T_i \in \mathsf{btr}_B(\mathcal{H}, S)$. Let $e \in \mathcal{H}$. As $\mathcal{H} = \bigcup_{i=1}^{k} \mathcal{H}_i$, there exists i such that $e \in \mathcal{H}_i$. Thus $e \cap T_i \neq \varnothing$ and thus $e \cap T \neq \varnothing$, that is, $T \in \mathsf{tr}(\mathcal{H})$. It remains to show that $T \in \mathsf{mtr}(\mathcal{H} \setminus \mathcal{H}(B))$. Let $y \in T$. By definition of T, there exists i such that $y \in T_i$. Thus, there exists $e \in \mathcal{H}_i \setminus \mathcal{H}_i(B)$ that is private for y *w.r.t.* T_i. Moreover, since $\mathcal{H}_i(B) = \mathcal{H}_i \cap \mathcal{H}(B)$, we know that $e \notin \mathcal{H}(B)$. As $C_1, \dots, C_k$ are the connected component of $\mathcal{H}[S]$, we have that, for every $j \neq i$, $V(C_i) \cap V(C_j) = \varnothing$. Moreover, for all $\ell \leq k$, we have $S \cap V(\mathcal{H}_\ell) = V(C_\ell)$. As $e \subseteq V(\mathcal{H}_i)$ and $T_j \subseteq S \cap V(\mathcal{H}_j)$, we have for every $j \neq i$:

$$e \cap T_j \subseteq V(\mathcal{H}_i) \cap S \cap V(\mathcal{H}_j) = V(C_i) \cap V(C_j) = \varnothing.$$

Thus, $e \cap T = e \cap T_i = \{y\}$. In other words, e is private for y *w.r.t.* T and $\mathcal{H} \setminus \mathcal{H}(B)$. That is $T \in \mathsf{btr}_B(\mathcal{H})$. $\square$

Lemma 3. *Let $\mathcal{H}$ be a hypergraph, $S, B \subseteq V(\mathcal{H})$ and $x \in S$. We have*

$$\mathsf{btr}_B(\mathcal{H}, S)(x) \subseteq \{\{x\}\} \bigotimes \mathsf{btr}_B(\mathcal{H}_1, S \setminus \{x\})$$

where $\mathcal{H}_1 := \mathcal{H} \setminus \mathcal{H}(x)$.

Proof. Let $T \in \mathsf{btr}_B(\mathcal{H}, S)(x)$. By definition, $x \in T$ and $T \subseteq S$, thus we only have to show that $T' = T \setminus \{x\} \in \mathsf{btr}_B(\mathcal{H}_1)$. Let $e \in \mathcal{H} \setminus \mathcal{H}(x)$. Since T is a transversal of $\mathcal{H}$, there exists $y \in e \cap T$. Moreover, by definition, $x \notin e$, thus $y \in T'$, that is, T' is a transversal of $\mathcal{H} \setminus \mathcal{H}(x)$. It remains to show that T' is a minimal transversal of $\mathcal{H}_1 \setminus \mathcal{H}_1(B) = \mathcal{H} \setminus (\mathcal{H}(B) \cup \mathcal{H}(x))$. Let $y \in T'$. Since T is a minimal transversal of $\mathcal{H} \setminus \mathcal{H}(B)$, there exists $e \in \mathcal{H} \setminus \mathcal{H}(B)$ such that e is private for y *w.r.t.* T. Since $x \in T$, we have $x \notin e$, otherwise e would not be private for y *w.r.t.* T. Thus $e \in \mathcal{H} \setminus (\mathcal{H}(B) \cup \mathcal{H}(x))$, that is, e is private to y *w.r.t.* T' in $\mathcal{H}_1 \setminus \mathcal{H}_1(B)$. In other words, T' is a minimal transversal of $\mathcal{H}_1 \setminus \mathcal{H}_1(B)$ which concludes the proof. $\square$

Lemma 4. *Let $\mathcal{H}$ be a hypergraph, $S, B \subseteq V(\mathcal{H})$ and $x \in S$. We have*

$$\left(\{\{x\}\} \bigotimes \mathsf{btr}_B(\mathcal{H}_1, S \setminus \{x\})\right) \setminus \mathsf{btr}_B(\mathcal{H}, S)(x) = \{\{x\}\} \bigotimes \mathsf{btr}_{B \cup \{x\}}(\mathcal{H}_2, S \setminus \{x\})$$

where $\mathcal{H}_1 := \mathcal{H} \setminus \mathcal{H}(x)$ and $\mathcal{H}_2 := \mathcal{H} \setminus (\mathcal{H}(B) \cap \mathcal{H}(x))$.

Proof. We prove the lemma by proving first the left-to-right inclusion (Claim 1) and then the right-to-left inclusion (Claim 2). But first, notice that $\mathcal{H}_1 \setminus \mathcal{H}_1(B) = \mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\}) = \mathcal{H} \setminus (\mathcal{H}(B) \cup \mathcal{H}(x))$ since $\mathcal{H}(B) \cup \mathcal{H}(x) = \mathcal{H}(B \cup \{x\})$.

Claim 1. *For every $T \in \left(\{\{x\}\} \bigotimes \mathsf{btr}_B(\mathcal{H}_1, S \setminus \{x\})\right) \setminus \mathsf{btr}_B(\mathcal{H}, S)(x)$, we have $T' = T \setminus \{x\} \in \mathsf{btr}_{B \cup \{x\}}(\mathcal{H}_2, S \setminus \{x\})$.*

Proof. We start by proving that $T' \in \mathsf{tr}(\mathcal{H}_2)$. Assume towards a contradiction that $T' \notin \mathsf{tr}(\mathcal{H}_2)$, *i.e.*, there exists $e \in \mathcal{H}_2$ such that $e \cap T' = \varnothing$. We prove that it implies $T \in \mathsf{btr}_B(\mathcal{H}, S)(x)$. First, observe that $T \in \mathsf{tr}(\mathcal{H})$, since $T' \in \mathsf{tr}(\mathcal{H}_1) = \mathsf{tr}(\mathcal{H} \setminus \mathcal{H}(x))$ and $T = T' \cup \{x\}$. Thus, we have $e \cap T = \{x\}$ and $e \in \mathcal{H}(x)$. As $e \in \mathcal{H}_2 = \mathcal{H} \setminus (\mathcal{H}(B) \cap \mathcal{H}(x))$, we have $e \in \mathcal{H} \setminus \mathcal{H}(B)$ and then e is a private hyperedge for x *w.r.t.* T and $\mathcal{H} \setminus \mathcal{H}(B)$. Furthermore, every vertex in T' has a private hyperedge *w.r.t.* T' and $\mathcal{H}_1 \setminus \mathcal{H}_1(B) = \mathcal{H} \setminus (\mathcal{H}(B) \cup \mathcal{H}(x))$ since $T' \in \mathsf{mtr}(\mathcal{H}_1 \setminus \mathcal{H}_1(B))$. Thus, every vertex in T' has a private hyperedge *w.r.t.* T and $\mathcal{H} \setminus \mathcal{H}(B)$. As $T \in \mathsf{tr}(\mathcal{H})$, we can conclude that $T \in \mathsf{mtr}(\mathcal{H} \setminus \mathcal{H}(B))$. Finally, we have $T \subseteq S$ by assumption. Therefore $T \in \mathsf{btr}_B(\mathcal{H}, S)(x)$ which is a contradiction. Thus, $T' \in \mathsf{tr}(\mathcal{H}_2)$.

We now prove that $T' \in \mathsf{mtr}(\mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\}))$, that is, we prove the minimality of T' in $\mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\})$. Let $y \in T'$. Since $T \in \mathsf{btr}_B(\mathcal{H}_1)$, there exists $f \in \mathcal{H}_1 \setminus \mathcal{H}_1(B)$ such that $f \cap T = \{y\}$. Since $\mathcal{H}_1 \setminus \mathcal{H}_1(B) = \mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\})$, every $y \in T'$ have a private hyperedge in $\mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\})$, that is $T' \in \mathsf{mtr}(\mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\}))$. As $T' \subseteq S \setminus \{x\}$, we can conclude that $T' \in \mathsf{btr}_{B \cup \{x\}}(\mathcal{H}_2, S \setminus \{x\})$. $\square$

Claim 2. *For every $T \in \{x\} \times \mathsf{btr}_{B\cup\{x\}}(\mathcal{H}_2, S \setminus \{x\})$, we have $T \in \{x\} \times \mathsf{btr}_B(\mathcal{H}_1, S \setminus \{x\}) \setminus \mathsf{btr}_B(\mathcal{H}, S)(x)$.*

Proof. We start by proving that $T' = T \setminus \{x\}$ is in $\mathsf{btr}_B(\mathcal{H}_1, S \setminus \{x\})$. First, we show that T' is a transversal of $\mathcal{H}_1$. Let $e \in \mathcal{H}_1$. By definition of $\mathcal{H}_1$, $x \notin e$, thus $e \in \mathcal{H}_2$ as well. Therefore $e \cap T' \neq \varnothing$. We now prove that T' is minimal in $\mathcal{H}_1 \setminus \mathcal{H}_1(B)$. As $T' \in \mathsf{mtr}(\mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\}))$, every vertex in T' has a private hyperedge in $\mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\})$. Moreover, recall that $\mathcal{H}_2 \setminus \mathcal{H}_2(B \cup \{x\}) = \mathcal{H}_1 \setminus \mathcal{H}_1(B)$. Thus, T' is minimal in $\mathcal{H}_1 \setminus \mathcal{H}_1(B)$. As $T' \subseteq S \setminus \{x\}$, we have $T' \in \mathsf{btr}_B(\mathcal{H}_1, S \setminus \{x\})$.

We finish the proof by showing that $T \notin \mathsf{btr}_B(\mathcal{H}, S)(x)$. In order to prove it, we show that there is no private hyperedge for x *w.r.t.* T and $\mathcal{H} \setminus \mathcal{H}(B)$. Indeed, since $T' \in \mathsf{tr}(\mathcal{H}_2)$, every hyperedge in $\mathcal{H}_2$ contains a vertex in T'. By definition of $\mathcal{H}_2$, we have $\mathcal{H} \setminus \mathcal{H}(B) \subseteq \mathcal{H}_2$, thus for every hyperedge e in $\mathcal{H} \setminus \mathcal{H}(B)$, we have $e \cap T \neq \{x\}$, *i.e.*, e is not a private for x. $\square$

By Claim 1 and Claim 2 we can conclude the lemma. $\square$

Finally, we characterise the number of blocked transversals of a hypergraph that do not contain a given vertex.

Lemma 5. *Let $\mathcal{H}$ be a hypergraph, $S, B \subseteq V(\mathcal{H})$ and $x \in S$. We have*

$$\mathsf{btr}_B(\mathcal{H}, S) \setminus \mathsf{btr}_B(\mathcal{H}, S)(x) = \mathsf{btr}_B(\mathcal{H}, S \setminus \{x\})$$

A direct consequence of Lemma 3, Lemma 4 and Lemma 5 is the following equality characterising the number of blocked transversals of $\mathcal{H}$ containing x from the numbers of blocked transversals of two hypergraphs that do not contain x anymore. This will be a crucial step in our dynamic programming scheme.

Theorem 6. *Let $\mathcal{H}$ be a hypergraph, $S, B \subseteq V(\mathcal{H})$ and $x \in S$. We have*

$$\#\mathsf{btr}_B(\mathcal{H}, S) = \#\mathsf{btr}_B(\mathcal{H}, S \setminus \{x\}) + \#\mathsf{btr}_B(\mathcal{H}_1, S \setminus \{x\}) - \#\mathsf{btr}_{B\cup\{x\}}(\mathcal{H}_2, S \setminus \{x\})$$

where $\mathcal{H}_1 := \mathcal{H} \setminus \mathcal{H}(x)$ and $\mathcal{H}_2 := \mathcal{H} \setminus (\mathcal{H}(B) \cap \mathcal{H}(x))$.

3. Counting the minimal transversals of β-acyclic hypergraphs

In this section, we fix $\mathcal{H}$ a β-acyclic hypergraph, $\leq$ a β-elimination ordering of its vertices and we let $\leq_{\mathcal{H}}$ the induced lexicographic ordering on the hyperedges, *i.e.*, $e \leq_{\mathcal{H}} f$ if $\min((e \setminus f) \cup (f \setminus e)) \in e$. We denote by $\mathcal{H}_e^x$ the sub-hypergraph of $\mathcal{H}$ formed by the hyperedges $f \in \mathcal{H}$ such that there exists a walk from f to e going only through hyperedges smaller than e and vertices smaller than x. For a vertex x of $V(\mathcal{H})$, we write $[\leq x]$, $[< x]$ and $[\geq x]$ for, respectively, $\{y \in V(\mathcal{H}) \mid y \leq x\}$, $\{y \in V(\mathcal{H}) \mid y \leq x \wedge y \neq x\}$ and $\{y \in V(\mathcal{H}) \mid x \leq y\}$. Moreover, we write $\mathcal{H}[\leq x]$, $\mathcal{H}[< x]$ and $\mathcal{H}[\geq x]$ instead of, respectively, $\mathcal{H}\big[[\leq x]\big]$, $\mathcal{H}\big[[< x]\big]$ and $\mathcal{H}\big[[\geq x]\big]$.

3.1. Decomposition of β-acyclic hypergraphs

The following two lemmas have been proven in [14, Section III-A].

Lemma 7 (Theorem 3 in [14]). *For every $e \in \mathcal{H}$, $x \in V(\mathcal{H})$, we have $V(\mathcal{H}_e^x) \cap [\geq x] \subseteq e$.*

Lemma 8 (Lemma 2 in [14]). *For every $e, f \in \mathcal{H}$, $x, y \in V(\mathcal{H})$, such that $e \leq_{\mathcal{H}} f$ and $x \leq y$. If $V(\mathcal{H}_e^x) \cap V(\mathcal{H}_f^y) \cap [\leq x] \neq \varnothing$, then $\mathcal{H}_e^x \subseteq \mathcal{H}_f^y$.*

We prove a lemma on the decomposition of $\mathcal{H}_e^x$ graphs that will be used with Lemma 2 to propagate the dynamic programming algorithm:

Lemma 9. *Let $x \in V(\mathcal{H})$, $e \in \mathcal{H}$ and $S \subseteq [\geq x]$. Let*

$$\mathcal{H}' := \begin{cases} \mathcal{H}_e^x & \text{if } S = \varnothing, \\ \mathcal{H}_e^x \setminus \left(\bigcap_{w \in S} \mathcal{H}_e^x(w)\right) & \text{otherwise.} \end{cases}$$

For every connected component C of $\mathcal{H}'[< x]$ different from $\{\varnothing\}$, there exists $y < x$ and $f \leq_{\mathcal{H}} e$ such that $C = \mathcal{H}_f^y[\leq y]$ and $\mathcal{H}_f^y = \{g \in \mathcal{H}' \mid g \cap [< x] \in C\}$.

Proof. Let $y = \max(V(C))$ and $f = \max\{g \in \mathcal{H}' \mid g \cap [< x] \in C\}$. We show that $\mathcal{H}_f^y = \{g \in \mathcal{H}' \mid g \cap [< x] \in C\}$.

First, we observe that $\mathcal{H}_f^y \subseteq \mathcal{H}'$. If $S = \varnothing$, it follows from Lemma 8 because, in this case, $\mathcal{H}' = \mathcal{H}_e^x$. Suppose that $S \neq \varnothing$, by definition of $\mathcal{H}'$, we have $S \nsubseteq f$. Moreover, by Lemma 7, we have that $V(\mathcal{H}_f^y) \cap [\geq y] \subseteq f$. As $S \subseteq [\geq x]$ and $x > y$, we have $S \subseteq [\geq y]$. Thus $S \nsubseteq V(\mathcal{H}_f^y)$ and for all $g \in \mathcal{H}_f^y$, we have $S \nsubseteq g$ since $g \subseteq V(\mathcal{H}_f^y)$. We can conclude that $\mathcal{H}_f^y \subseteq \mathcal{H}'$.

Now, we prove that every $g \in \mathcal{H}_f^y$, we have $g \cap [< x] \in C$. Let $g \in \mathcal{H}_f^y$. By definition of $\mathcal{H}_f^y$ and because $\mathcal{H}_f^y \subseteq \mathcal{H}'$, there exists a path P from f to g going only through vertices smaller than y and hyperedges smaller than f in $\mathcal{H}'$. As $y < x$, we can conclude that $f \cap [< x]$ is connected to $g \cap [< x]$ in $\mathcal{H}'$, *i.e.*, $g \cap [< x] \in C$. In other words, we have $\mathcal{H}_f^y \subseteq \{g \in \mathcal{H}' \mid g \cap [< x] \in C\}$.

It remains to prove the other inclusion. Let $g \in \mathcal{H}'$ with $g \cap [< x] \in C$. Since C is a connected component of $\mathcal{H}'[< x]$, there exists a path P from $f \cap [< x]$ to $g \cap [< x]$. By the maximality of y and f, P goes only through vertices smaller than y and hyperedges smaller than f. We can construct from P a path P' from f to g in $\mathcal{H}'$ going through vertices smaller than y and hyperedges smaller than f. As $\mathcal{H}' \subseteq \mathcal{H}$, we can conclude that $g \in \mathcal{H}_f^y$ and thus, $\mathcal{H}_f^y = \{g \in \mathcal{H}' \mid g \cap [< x] \in C\}$. Finally, observe that $C = \mathcal{H}_f^y[< x] = \mathcal{H}_f^y[\leq y]$ since $y = \max(V(C))$. $\square$

3.2. The algorithm

In this subsection, we describe the dynamic programming algorithm we use to count the number of minimal transversals of a β-acyclic hypergraph. We denote by x_1 the smallest element of $\leq$.

Our goal is to compute $\#\mathsf{btr}_{\varnothing}(\mathcal{H}_e^x, [\leq x])$ and $\#\mathsf{btr}_w(\mathcal{H}_e^x, [\leq x])$ for every $e \in \mathcal{H}$, $x \in V(\mathcal{H})$ and $w \in V(\mathcal{H})$ such that $x < w$. Observe that it is enough for computing the number of minimal transversals of $\mathcal{H}$ as $\#\mathsf{mtr}(\mathcal{H}) = \#\mathsf{btr}_\varnothing(\mathcal{H}_{e_m}^{x_n}, [\leq x_n])$ where e_m is the maximal hyperedge for $\leq_\mathcal{H}$ and x_n is the maximal vertex for $\leq$. Indeed, we have $\mathcal{H}_{e_m}^{x_n} = \mathcal{H}$ and $[\leq x_n] = V(\mathcal{H})$, thus $\mathsf{btr}_\varnothing(\mathcal{H}_{e_m}^{x_n}, [\leq x_n]) = \mathsf{btr}_\varnothing(\mathcal{H}) = \mathsf{mtr}(\mathcal{H})$.

The propagation of the dynamic programming works as follows: we use Theorem 6 to reduce the computation of $\#\mathsf{btr}_B(\mathcal{H}_e^x, [\leq x])$ to the computation of $\#\mathsf{btr}$ for several hypergraphs that do not contain x. We then use Lemma 9 to show that these hypergraphs can be decomposed into disjoint hypergraphs of the form $\mathcal{H}_f^y$ for $f \leq_\mathcal{H} e$ and $y < x$ which allows us to compute $\#\mathsf{btr}_B(\mathcal{H}_e^x, [\leq x])$ from precomputed values of the form $\#\mathsf{btr}_{B'}(\mathcal{H}_f^y, [\leq y])$, where $B' \in \{B, \{x\}\}$.

3.2.1. Base cases.

We observe that for every $e \in \mathcal{H}$, $\mathcal{H}_e^{x_1}[\leq x_1]$ is either equal to $\{x_1\}$ or $\{\varnothing\}$. Thus, for every $e \in \mathcal{H}$ and $w \in V(\mathcal{H})$ such that $w > x_1$, we can compute $\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [\leq x_1])$ and $\#\mathsf{btr}_w(\mathcal{H}_e^x, [\leq x_1])$ in time $O(1)$.

3.2.2. Computing $\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [\leq x])$ by dynamic programming.

We start by explaining how we can compute $\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [\leq x])$ in polynomial time if the values $\#\mathsf{btr}_\varnothing(\mathcal{H}_f^y, [\leq y])$ and $\#\mathsf{btr}_w(\mathcal{H}_f^y, [\leq y])$ have been precomputed for $f \leq_\mathcal{H} e$ and $y < x$, $y < w$.

We start by applying Theorem 6.

$$\begin{aligned}
\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [\leq x]) = {} & \#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [< x]) \\
& + \#\mathsf{btr}_\varnothing(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x]) \\
& - \#\mathsf{btr}_x(\mathcal{H}_e^x, [< x]).
\end{aligned}$$

Now, let $C_1, \ldots, C_k$ be the connected components of $\mathcal{H}_e^x[< x]$. If there exists i such that $C_i = \{\varnothing\}$, then $\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [< x]) = \#\mathsf{btr}_x(\mathcal{H}_e^x, [< x]) = 0$. Otherwise, by applying Lemma 9 with $S = \varnothing$, there exists, for each $1 \leq i \leq k$, $y_i < x$ and $f_i \leq_\mathcal{H} e$ such that $\mathcal{H}_{f_i}^{y_i} = \{g \in \mathcal{H}_e^f \mid g \cap [< x] \in C_i\}$ and $C_i = \mathcal{H}_{f_i}^{y_i}[\leq y_i]$. By Lemma 2,

$$\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [< x]) = \prod_{i=1}^{k} \#\mathsf{btr}_\varnothing(\mathcal{H}_{f_i}^{y_i}, [\leq y_i]),$$

$$\#\mathsf{btr}_x(\mathcal{H}_e^x, [< x]) = \prod_{i=1}^{k} \#\mathsf{btr}_x(\mathcal{H}_{f_i}^{y_i}, [\leq y_i]).$$

We now show how to decompose $\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x])$ into a product of precomputed values. Let $D_1, \ldots, D_l$ be the connected components of $\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x)[< x]$. If there exists i such that $D_i = \{\varnothing\}$, then $\#\mathsf{btr}_x(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x]) = 0$. Otherwise, if we apply Lemma 9 with $S = \{x\}$, then there exists, for each $1 \leq i \leq l$, $y_i < x$ and $f_i \leq_\mathcal{H} e$ such that $\mathcal{H}_{y_i}^{f_i} = \{g \in \mathcal{H}_e^x \setminus \mathcal{H}_e^x(x) \mid g \cap [< x] \in D_i\}$ and $D_i = \mathcal{H}_{y_i}^{f_i}[\leq y_i]$. We can thus conclude by Lemma 2 that

$$\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x]) = \prod_{i=1}^{l} \#\mathsf{btr}_\varnothing(\mathcal{H}_{f_i}^{y_i}, [\leq y_i]).$$

Therefore, if $\#\mathsf{btr}_\varnothing(\mathcal{H}_f^y, [\leq y])$ and $\#\mathsf{btr}_x(\mathcal{H}_f^y, [\leq y])$ have already been computed for every $f <_{\mathcal{H}} e$ and $y \leq x$, we can compute $\#\mathsf{btr}_\varnothing(\mathcal{H}_e^x, [\leq x])$ with at most $3 \times |\mathcal{H}_e^x|$ additional multiplications and 3 additions.

3.2.3. Computing $\#\mathsf{btr}_w(\mathcal{H}_e^x, [\leq x])$ by dynamic programming.
Let $x \leq w$. By Theorem 6, we have:

$$\begin{aligned}\#\mathsf{btr}_w(\mathcal{H}_e^x, [\leq x]) = {} & \#\mathsf{btr}_w(\mathcal{H}_e^x, [< x]) \\ & + \#\mathsf{btr}_w(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x]) \\ & - \#\mathsf{btr}_{\{x,w\}}(\mathcal{H}_e^x \setminus (\mathcal{H}_e^x(x) \cap \mathcal{H}_e^x(w)), [< x]).\end{aligned}$$

We start by explaining how to compute $\#\mathsf{btr}_w(\mathcal{H}_e^x, [< x])$. Let $C_1, \dots, C_k$ be the connected components of $\mathcal{H}_e^x[< x]$. If there exists i such that $C_i = \{\varnothing\}$, then $\#\mathsf{btr}_w(\mathcal{H}_e^x, [< x]) = 0$. Otherwise, by Lemma 9 with $S = \{w\}$, there exists, for each $1 \leq i \leq k$, $f_i \leq_{\mathcal{H}} e$ and $y_i < x$ such that $\mathcal{H}_{f_i}^{y_i} = \{g \in \mathcal{H}_e^x \mid g \cap\ [< x] \in C_i\}$ and $C_i = \mathcal{H}_{f_i}^{y_i}[\leq y_i]$. By Lemma 2, we can conclude that

$$\#\mathsf{btr}_w(\mathcal{H}_e^x, [< x]) = \prod_{i=1}^{k} \#\mathsf{btr}_w(\mathcal{H}_{f_i}^{y_i}, [\leq y_i]).$$

We now explain how to compute $\#\mathsf{btr}_w(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x])$. Let $D_1, \dots, D_l$ be the connected components of $(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x))[< x]$. If there exists i such that $D_i = \{\varnothing\}$, then $\#\mathsf{btr}_w(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x]) = 0$. Otherwise, by applying Lemma 9 with $S = \{x\}$, it follows that for every $1 \leq i \leq l$, there exists $y_i < x$ and $f_i \leq_{\mathcal{H}} e$ such that $\mathcal{H}_{f_i}^{y_i} = \{g \in \mathcal{H}_e^x \setminus \mathcal{H}_e^x(x) \mid g \cap\ [< x] \in D_i\}$ and $D_i = \mathcal{H}_{f_i}^{y_i}[\leq y_i]$. Thus, from Lemma 2,

$$\#\mathsf{btr}_w(\mathcal{H}_e^x \setminus \mathcal{H}_e^x(x), [< x]) = \prod_{i=1}^{l} \#\mathsf{btr}_w(\mathcal{H}_{f_i}^{y_i}, [\leq y_i]).$$

Finally, we explain how to decompose $\#\mathsf{btr}_{\{x,w\}}(\mathcal{H}_e^x \setminus (\mathcal{H}_e^x(x) \cap \mathcal{H}_e^x(w)), [< x])$ into a product of pre-computed values. To ease the notation, we denote $\mathcal{H}_e^x \setminus (\mathcal{H}_e^x(x) \cap \mathcal{H}_e^x(w))$ by $\mathcal{H}'$. Let $K_1, \dots, K_r$ be the connected components of $\mathcal{H}'[< x]$. If there exists i such that $K_i = \{\varnothing\}$, then $\#\mathsf{btr}_{\{x,w\}}(\mathcal{H}', [< x]) = 0$. Otherwise, by Lemma 9 applied with $S = \{x, w\}$, we have that for every i, there exists $y_i < x$ and $f_i \leq_{\mathcal{H}} e$ such that $\mathcal{H}_{f_i}^{y_i} = \{g \in \mathcal{H}' \mid g \cap [< x]\ \in K_i\}$ and $K_i = \mathcal{H}_{f_i}^{y_i}[\leq y_i]$. By Lemma 2, we have:

$$\#\mathsf{btr}_{\{x,w\}}(\mathcal{H}_e^x \setminus (\mathcal{H}_e^x(x) \cap \mathcal{H}_e^x(w)), [< x]) = \prod_{i=1}^{r} \#\mathsf{btr}_{\{x,w\} \cap V(\mathcal{H}_{f_i}^{y_i})}(\mathcal{H}_{f_i}^{y_i}, [\leq y_i]).$$

Claim 3. *For every $i \leq p$, $\{x, w\} \cap V(\mathcal{H}_{f_i}^{y_i}) \neq \{x, w\}$.*

Proof. Assume towards a contradiction that $\{x, w\} \cap V(\mathcal{H}_{f_i}^{y_i}) = \{x, w\}$. Recall that $\mathcal{H}_{f_i}^{y_i}[\leq y_i]$. By Lemma 7, $\{x, w\} \subseteq V(\mathcal{H}_{f_i}^{y_i})$ implies $\{x, w\} \subseteq f$. Thus, we have $f \in \mathcal{H}_e^x(x) \cap \mathcal{H}_e^x(w)$. This is a contradiction, since $f \in \mathcal{H}_{f_i}^{y_i} \subseteq \mathcal{H}_e^x \setminus (\mathcal{H}_e^x(w) \cap \mathcal{H}_e^x(x))$. $\square$

Thus, $\{x, w\} \cap V(\mathcal{H}_{f_i}^{y_i})$ equals either $\{x\}$, or $\{w\}$ or $\varnothing$ by Claim 3. That is, we can compute $\#\mathsf{btr}_{\{x,w\}}(\mathcal{H}_e^x \setminus (\mathcal{H}_e^x(x) \cap \mathcal{H}_e^x(w)), [< x])$ from precomputed terms.

We can conclude that, if $\#\mathsf{btr}_\varnothing(\mathcal{H}_f^y, [\leq y])$ and $\#\mathsf{btr}_w(\mathcal{H}_f^y, [\leq y])$ have already been computed for every $f <_\mathcal{H} e$ and $y \leq w$, we can compute $\#\mathsf{btr}_w(\mathcal{H}_e^x, [\leq x])$ with at most $3 \times |\mathcal{H}_e^x|$ additional multiplications and 3 additions.

It is easy to see that a straightforward greedy algorithm can be used to compute a β-elimination ordering in polynomial time (see [16] for a better algorithm due to Paige and Tarjan). Moreover, the dynamic programming algorithm describes above computes at most $O(n^2|\mathcal{H}|)$ terms and each of them can be computed from the others with a polynomial number of arithmetic operations. Finally, all these terms can be bounded by 2^n since they are all the cardinals of some collection of subsets of the vertices. Thus these arithmetic operations can be done in polynomial time in the size of the input. It follows.

Theorem 10. *Let $\mathcal{H}$ be a β-acyclic hypergraph. We can compute $\#\mathsf{mtr}(\mathcal{H})$ in polynomial time.*

4. Conclusion

We proposed a polynomial time algorithm for counting the minimal transversals of any β-acyclic hypergraph, supporting Conjecture 1, and it seems that the technique can be easily adapted to consider (inclusion-wise) minimal d-dominating sets, which are dominating sets such that each vertex is dominated by at least d vertices [11].

Besides resolving Conjecture 1, there are two immediate questions that deserve to be considered. Firstly, can we count the minimal models of any non-monotone β-acyclic formula in polynomial time? Secondly, can we identify a wide range of graph classes and of counting graph problems for which the techniques of this note apply?

[1] S. P. Vadhan, The complexity of counting in sparse, regular, and planar graphs, SIAM J. Comput. 31 (2) (2001) 398–427 (electronic). `doi:10.1137/S0097539797321602`.
URL `http://dx.doi.org/10.1137/S0097539797321602`

[2] L. G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (2) (1979) 189–201. `doi:10.1016/0304-3975(79)90044-6`.
URL `http://dx.doi.org/10.1016/0304-3975(79)90044-6`

[3] L. A. Goldberg, R. Gysel, J. Lapinskas, Approximately counting locally-optimal structures, J. Comput. Syst. Sci. 82 (6) (2016) 1144–1160. `doi:10.1016/j.jcss.2016.04.001`.
URL `https://doi.org/10.1016/j.jcss.2016.04.001`

[4] T. Eiter, K. Makino, G. Gottlob, Computational aspects of monotone dualization: a brief survey, Discrete Appl. Math. 156 (11) (2008) 2035–2049.

doi:10.1016/j.dam.2007.04.017.
URL http://dx.doi.org/10.1016/j.dam.2007.04.017

[5] A. Durand, M. Hermann, On the counting complexity of propositional circumscription, Inform. Process. Lett. 106 (4) (2008) 164–170. doi:10.1016/j.ipl.2007.11.006.
URL http://dx.doi.org/10.1016/j.ipl.2007.11.006

[6] F. Capelli, Structural restrictions of cnf formulas: application to model counting and knowledge compilation, Ph.D. thesis, Université Paris Diderot (2016).

[7] T. W. Haynes, S. T. Hedetniemi, P. J. Slater, Fundamentals of domination in graphs, Vol. 208 of Monographs and Textbooks in Pure and Applied Mathematics, Marcel Dekker, Inc., New York, 1998.

[8] M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, On the enumeration of minimal dominating sets and related notions, SIAM J. Discrete Math. 28 (4) (2014) 1916–1929. doi:10.1137/120862612.
URL http://dx.doi.org/10.1137/120862612

[9] B. Courcelle, The monadic second-order logic of graphs XVI : Canonical graph decompositions, Logical Methods in Computer Science 2 (2).

[10] M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, T. Uno, On the enumeration and counting of minimal dominating sets in interval and permutation graphs, in: Algorithms and computation, Vol. 8283 of Lecture Notes in Comput. Sci., Springer, Heidelberg, 2013, pp. 339–349. doi:10.1007/978-3-642-45030-3_32.
URL http://dx.doi.org/10.1007/978-3-642-45030-3_32

[11] P. A. Golovach, P. Heggernes, M. M. Kanté, D. Kratsch, S. H. Sæther, Y. Villanger, Output-polynomial enumeration on graphs of bounded (local) linear mim-width, Algorithmica (2017) 1–28doi:10.1007/s00453-017-0289-1.

[12] M. M. Kanté, T. Uno, Counting minimal dominating sets, in: Theory and Applications of Models of Computation - 14th Annual Conference, TAMC 2017, Bern, Switzerland, April 20-22, 2017, Proceedings, 2017, pp. 333–347.

[13] A. Brandstädt, V. B. Le, J. P. Spinrad, Graph classes: a survey, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999. doi:10.1137/1.9780898719796.
URL http://dx.doi.org/10.1137/1.9780898719796

[14] F. Capelli, Understanding the complexity of #sat using knowledge compilation, in: 32th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017, 2017.

[15] T. Eiter, G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, SIAM J. Comput. 24 (6) (1995) 1278–1304. `doi:10.1137/S0097539793250299`.
URL `http://dx.doi.org/10.1137/S0097539793250299`

[16] R. Paige, R. E. Tarjan, Three partition refinement algorithms, SIAM Journal on Computing 16 (6) (1987) 973–989.