Boolean Tensor Decomposition for Conjunctive Queries with Negation

Mahmoud Abo Khamis
RelationalAI, Berkeley, USA

Hung Q. Ngo
RelationalAI, Berkeley, USA

Dan Olteanu
Department of Computer Science, University of Oxford, UK

Dan Suciu
Department of Computer Science and Engineering, University of Washington, USA

Abstract

We propose an approach for answering conjunctive queries with negation, where the negated relations have bounded degree. Its data complexity matches that of the InsideOut and PANDA algorithms for the positive subquery of the input query and is expressed in terms of the fractional hypertree width and the submodular width respectively. Its query complexity depends on the structure of the conjunction of negated relations; in general it is exponential in the number of join variables occurring in negated relations yet it becomes polynomial for several classes of queries.

This approach relies on several contributions. We show how to rewrite queries with negation on bounded-degree relations into equivalent conjunctive queries with not-all-equal (NAE) predicates, which are a multi-dimensional analog of disequality ($\neq$). We then generalize the known color-coding technique to conjunctions of NAE predicates and explain it via a Boolean tensor decomposition of conjunctions of NAE predicates. This decomposition can be achieved via a probabilistic construction that can be derandomized efficiently.

2012 ACM Subject Classification Theory of computation → Database query processing and optimization (theory); Information systems → Database query processing

Keywords and phrases color-coding, combined complexity, negation, query evaluation

Digital Object Identifier 10.4230/LIPIcs.ICDT.2019.21

Related Version An extended online version of this work includes the missing proofs [1], see `https://arxiv.org/abs/1712.07445`.

Funding This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 682588. This project is also supported in part by NSF grants AITF-1535565 and III-1614738.

Acknowledgements The authors would like to thank the anonymous reviewers for their suggestions that helped improve the readability of this paper.

1 Introduction

This paper considers the problem of answering conjunctive queries with negation of the form

$$Q(\boldsymbol{X}_F) \leftarrow \mathsf{body} \wedge \bigwedge_{S \in \overline{\mathcal{E}}} \neg R_S(\boldsymbol{X}_S), \tag{1}$$

where body is the body of an arbitrary conjunctive query, $\boldsymbol{X}_F = (X_i)_{i \in F}$ denotes a tuple of variables (or attributes) indexed by a set F of positive integers, and $\overline{\mathcal{E}}$ is the set of hyperedges

of a multi-hypergraph[1] $\overline{\mathcal{H}} = (\overline{\mathcal{V}}, \overline{\mathcal{E}})$. Every hyperedge $S \in \overline{\mathcal{E}}$ corresponds to a bounded-degree relation R_S on attributes $\boldsymbol{X}_S$. For instance, the equality (=) relation is a bounded-degree (binary) relation, because every element in the active domain has degree one; the edge relation E of a graph with bounded maximum degree is also a bounded-degree relation. Section 2 formalizes this notion of bounded degree.

We exemplify using three Boolean queries[2] over a directed graph $G = ([n], E)$ with n nodes and $N = |E|$ edges: the k-walk query[3], the k-path query, and the induced (or chordless) k-path query. They have the same body and encode graph problems of increasing complexity:

$$W \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1}).$$

$$P \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1}) \wedge \bigwedge_{\substack{i,j \in [k+1] \\ i+1<j}} X_i \neq X_j. \tag{2}$$

$$I \leftarrow E(X_1, X_2) \wedge E(X_2, X_3) \wedge \cdots \wedge E(X_k, X_{k+1}) \wedge \bigwedge_{\substack{i,j \in [k+1] \\ i+1<j}} (\neg E(X_i, X_j) \wedge X_i \neq X_j)). \tag{3}$$

The hypergraph $\overline{\mathcal{H}}$ for the k-walk query W is empty since it has no negated relations. This query can be answered in $O(kN \log N)$ time using for instance the Yannakakis dynamic programming algorithm [34]. The k-path query P has the hypergraph $\overline{\mathcal{H}} = ([k+1], \{(i,j) \mid i, j \in [k+1], i+1 < j\})$. It can be answered in $O(k^k N \log N)$-time [30] and even better in $2^{O(k)} N \log N$-time using the color-coding technique [8]. The induced k-path query I has the hypergraph $\overline{\mathcal{H}}$ similar to that of P, but every edge (i,j) has now multiplicity two due to the negated edge relation and also the disequality. This query is W[2]-hard [10]. However, if the graph G has a maximal degree that is bounded by some constant d, then the query can be answered in $O(f(k,d) \cdot N \log N)$-time for some function f that depends exponentially on k and d [30]. Our results imply the above complexities for the three queries.

1.1 Main Contribution

In this paper we propose an approach to answering conjunctive queries with negation on bounded-degree relations of arbitrary arities. Our approach is the first to exploit the bounded degree of the negated relations. The best known algorithms for positive queries such as InsideOut [2] and PANDA [3] can also answer queries with negation, albeit with much higher complexity since already one negation can increase their worst-case runtime. For example, the Boolean path queries with a disequality between the two end points takes linear time with our approach, but quadratic time with existing approaches [2, 3]. The data complexity of our approach matches that of InsideOut and PANDA for the positive subquery $Q(\boldsymbol{X}_F) \leftarrow \mathsf{body}$. To lower its query complexity, we use a range of techniques including color-coding, probabilistic construction of Boolean tensor decompositions, and derandomization of this construction.

▶ **Theorem 1.1.** *Any query Q of the form (1), where for each $S \in \overline{\mathcal{E}}$ the relation R_S has bounded degree and $f(\cdot)$ is a function of Q, can be answered over a database of size N in time $O(f(Q) \cdot \log N \cdot (N^{\mathsf{fhtw}_F(\mathsf{body})} + |\mathsf{output}|))$ using a reduction to* InsideOut *and $O(f(Q) \cdot (\mathsf{poly}(\log N) \cdot N^{\mathsf{subw}_F(\mathsf{body})} + \log N \cdot |\mathsf{output}|))$ using a reduction to* PANDA.

[1] In a multi-hypergraph, each hyperedge S can occur multiple times. All hypergraphs in this paper are multi-hypergraphs.
[2] We denote Boolean queries $Q(\boldsymbol{X}_F)$ where $F = \emptyset$ by Q instead of $Q()$. We also use $[n] = \{1, \ldots, n\}$.
[3] Unlike a path, in a walk some vertices may repeat.

The complexities of InsideOut [2] and PANDA [3] depend on the fractional hypertree width fhtw [17] and respectively the submodular width subw [23]. The widths fhtw_F and subw_F are fhtw and respectively subw computed on the subset of hypertree decompositions of the positive subquery of Q for which the set F of free variables form a connected subtree. The dependency of the function f on the structure of Q, and in particular on the hypergraph $\overline{\mathcal{H}}$ of the negated relations in Q, is an important result of this paper.

Theorem 1.1 draws on three contributions:

1. A rewriting of queries of the form (1) into equivalent conjunctive queries with not-all-equal predicates, which are a multi-dimensional analog of disequality $\neq$ (Proposition 4.3);
2. A generalization of color-coding [8] from cliques of disequalities to arbitrary conjunctions of not-all-equal predicates; and
3. An alternative view of color coding via Boolean tensor decomposition of conjunctions of not-all-equal predicates (Lemma 5.1). This decomposition admits a probabilistic construction that can be derandomized efficiently (Corollary 5.7).

Contribution 1 (Section 4) gives a rewrite of the query Q into an equivalent disjunction of queries Q_i of the form (cf. Proposition 4.3)

$$Q_i(\boldsymbol{X}_F) \leftarrow \mathsf{body}_i \wedge \bigwedge_{S \in \mathcal{E}_i} \mathsf{NAE}(\boldsymbol{Z}_S).$$

For each query Q_i, body_i may be different from body in Q, since fresh variables $\boldsymbol{Z}_S$ and unary predicates may be introduced. Its fractional hypertree and submodular widths remain however at most that of body. We thus rewrite the conjunction of the negated relations into a much simpler conjunction of NAE predicates without increasing the data complexity of Q. The number of such queries Q_i depends exponentially on the arities and the degrees of the negated relations, which is the reason why we need the constant bound on these degrees.

Contribution 2 (Section 5) is based on the observation that a conjunction of NAE predicates can be answered by an adaptation of the color-coding technique [8], which has been used so far for checking cliques of disequalities. The crux of this technique is to randomly color each value in the active domain with one color from a set whose size is much smaller than the size of the active domain, and to use these colors instead of the values themselves to check the disequalities. We generalize this idea to conjunctions of NAE predicates and show that such conjunctions can be expressed equivalently as disjunctions of simple queries over the different possible colorings of the variables in these queries.

Contribution 3 (Section 5) explains color coding by providing an alternative view of it: Color coding is a (Boolean) tensor decomposition of the (Boolean) tensor defined by the conjunction $\bigwedge_S \mathsf{NAE}(\boldsymbol{Z}_S)$. As a tensor, $\bigwedge_S \mathsf{NAE}(\boldsymbol{Z}_S)$ is a multivariate function over variables in the set $U = \bigcup_S \boldsymbol{Z}_S$. The tensor decomposition rewrites it into a disjunction of conjunctions of *univariate* functions over individual variables Z_i (Lemma 5.1). That is,

$$\bigwedge_S \mathsf{NAE}(\boldsymbol{Z}_S) \equiv \bigvee_{j \in [r]} \bigwedge_{i \in U} f_i^{(j)}(Z_i),$$

where r is the (Boolean tensor) rank of the tensor decomposition, and for each $j \in [r]$, the inner conjunction $\bigwedge_{i \in U} f_i^{(j)}(Z_i)$ can be thought of as a rank-1 tensor of inexpensive Boolean univariate functions $f_i^{(j)}(\cdot)$ $(\forall i \in U)$. The key advantages of this tensor decomposition are that (i) the addition of univariate conjuncts to body_i does not increase its (fractional hypertree and submodular) width and (ii) the dependency of the rank r on the database size N is only a $\log N$ factor. Lemma 5.1 shows that the rank r depends on two quantities:

$r = P(\mathcal{G}, c) \cdot |\mathcal{F}|$. The first is the chromatic polynomial of the hypergraph of $\bigwedge_S \mathsf{NAE}(\boldsymbol{Z}_S)$ using c colors. The second is the size of a family of hash functions that represent proper c-colorings of homomorphic images of the input database. The number c of needed colors is at most the number $|U|$ of variables in $\bigwedge_S \mathsf{NAE}(\boldsymbol{Z}_S)$. We show it to be the maximum chromatic number of a hypergraph defined by any homomorphic image of the database.

We give a probabilistic construction of the tensor decomposition that generalizes the construction used by the color-coding technique. It selects a color distribution dependent on the query structure, which allows the rank of $\bigwedge_S \mathsf{NAE}(\boldsymbol{Z}_S)$ to take a wide range of query complexity asymptotics, from polynomial to exponential in the query size. This is more refined than the previously known bound [8], which amounts to a tensor rank that is exponential in the query size. We further derandomize this construction by adapting ideas from derandomization for k-restrictions [6] (with k being related to the Boolean tensor rank).

Section 6 shows how to use the Boolean tensor decomposition in conjunction with InsideOut [2] and PANDA [3] to evaluate queries of the form (1) with the complexity given by Theorem 1.1. The query complexity captured by the function f is given by the number of NAE predicates and the rank of the tensor decomposition of their conjunction.

2 Preliminaries

In this paper we consider arbitrary conjunctive queries with negated relations of the form (1). We make use of the following naming convention. Capital letters with subscripts such as X_i or A_j denote variables. For any set S of positive integers, $\boldsymbol{X}_S = (X_i)_{i \in S}$ denote a tuple of variables indexed by S. Given a relation R over variables $\boldsymbol{X}_S$ and $J \subseteq S$, $\pi_J R$ denotes the projection of R onto variables $\boldsymbol{X}_J$, i.e., we write $\pi_J R$ instead of $\pi_{\boldsymbol{X}_J} R$. If X_i is a variable, then the corresponding lower-case x_i denotes a value from the active domain $\mathsf{Dom}(X_i)$ of X_i. Bold-face $\boldsymbol{x}_S = (x_i)_{i \in S}$ denotes a tuple of values in $\prod_{i \in S} \mathsf{Dom}(X_i)$.

We associate a hypergraph $\mathcal{H}(R)$ with a relation $R(\boldsymbol{X}_S)$ as follows. The vertex set is $\{(i, v) \mid i \in S, v \in \mathsf{Dom}(X_i)\}$. Each tuple $\boldsymbol{x}_S = (x_i)_{i \in S} \in R$ corresponds to a (hyper)edge $\{(i, x_i) | i \in S\}$. $\mathcal{H}(R)$ is a $|S|$-uniform hypergraph (all hyperedges have size $|S|$).

2.1 Hypergraph coloring and bounded-degree relations

Hypergraph coloring. Let $\mathcal{G} = (U, \mathcal{A})$ denote a multi-hypergraph and k be a positive integer. A *proper c-coloring* of $\mathcal{G}$ is a mapping $h : U \to [c]$ such that for every edge $S \in \mathcal{A}$, there exists $u, v \in S$ with $u \neq v$ such that $h(u) \neq h(v)$. The *chromatic polynomial* $P(\mathcal{G}, c)$ of $\mathcal{G}$ is the number of proper c-colorings of $\mathcal{G}$ [18]. A vertex (edge) coloring of $\mathcal{G}$ is an assignment of colors to the vertices (edges) of $\mathcal{G}$ so that no two adjacent vertices (incident edges) have the same color. The *chromatic number* $\chi(\mathcal{G})$ and the *chromatic index* $\chi'(\mathcal{G})$ are the smallest numbers of colors needed for a vertex coloring and respectively an edge coloring of $\mathcal{G}$. Coloring a (hyper)graph is equivalent to coloring it without singleton edges.

Bounded-degree relation. The *maximum degree* of a vertex in a hypergraph $\mathcal{G} = (U, \mathcal{A})$ is denoted by $\Delta(\mathcal{G})$: $\Delta(\mathcal{G}) = \max_{v \in U} |\{S \in \mathcal{A} \mid v \in S\}|$. For a relation $R_S(\boldsymbol{X}_S)$, its maximum degree $\Delta(\mathcal{H}(R_S))$ is the maximum number of tuples in R_S with the same value for a variable $X \in \boldsymbol{X}_S$: $\Delta(\mathcal{H}(R_S)) = \max_{\substack{i \in S \\ v \in \mathsf{Dom}(X_i)}} |\{\boldsymbol{x}_S \in R_S \mid x_i = v\}|$. We will use a slightly different notion of *degree* of a relation denoted by $\deg(R_S)$, which also accounts for the arity $|S|$ of the relation R_S. Proposition 2.3 connects the two notions.

▶ **Definition 2.1** (Matching). *A k-ary relation $M(\boldsymbol{X}_S)$ is called a (k-dimensional)* matching *if for every two tuples $\boldsymbol{x}_S, \boldsymbol{x}'_S \in M$, either $\boldsymbol{x}_S = \boldsymbol{x}'_S$, i.e., $\boldsymbol{x}_S$ and $\boldsymbol{x}'_S$ are the same tuple, or it holds that $x_i \neq x'_i, \forall i \in S$.*

▶ **Definition 2.2** (Degree). *The* degree *of a relation $R_S(\boldsymbol{X}_S)$, denoted by $\deg(R_S)$, is the smallest integer d for which R_S can be written as the disjoint union of d matchings. The degree $\deg(R_S)$ is* bounded *if there is a constant d_S such that $\deg(R_S) \leq d_S$.*

It is easy to see that $\deg(R_S) = \chi'(\mathcal{H}(R_S))$. If R_S is a binary relation, then $\mathcal{H}(R_S)$ is a bipartite graph and $\deg(R_S) = \chi'(\mathcal{H}(R_S)) = \Delta(\mathcal{H}(R_S))$. This follows from König's line coloring theorem [20], which states that the chromatic index of a bipartite graph is equal to its maximum degree. When the arity k is higher than two, to the best of our knowledge there does not exist such a nice characterization of the chromatic index of R_S in terms of the maximum degree of individual vertices in the graph, although there has been some work on bounding the chromatic index of (linear) uniform hypergraphs [5, 22, 13, 29, 29]. In our setting, we are willing to live with sub-optimal decomposition of a bounded-degree relation into matchings as long as it can be done in linear time.

▶ **Proposition 2.3.** *Let $R_S(\boldsymbol{X}_S)$ denote a k-ary relation of size N and $\ell = \Delta(\mathcal{H}(R_S))$. Then:*

- $\ell \leq \deg(R_S) \leq k(\ell - 1) + 1$;
- *We can compute in $O(N)$-time disjoint k-ary matchings $M_1, \ldots, M_{k\ell-k+1}$ such that $R_S = \bigcup_{j=1}^{k(\ell-1)+1} M_j$.*

Proof. The fact that $\ell \leq \deg(R_S)$ is obvious. To show that $\deg(R_S) \leq k(\ell - 1) + 1$, note that any edge in $\mathcal{H}(R_S)$ is adjacent to at most $k(\ell - 1)$ other edges of $\mathcal{H}(R_S)$, hence greedy coloring can color the edges of $\mathcal{H}(R_S)$ in time $O(N)$ using $k(\ell - 1) + 1$ colors. ◀

The two notions of degree of a relation are thus equivalent up to a constant factor given by the arity of the relation.

2.2 FAQ, width parameters, and corresponding algorithms

▶ **Definition 2.4** (The FAQ problem [2]). *The input to* FAQ *is a set of functions and the output is a function which is a series of aggregations (e.g. sums) over the product of input functions. In particular, the input to* FAQ *consists of the following:*

- *A multi-hypergraph $\mathcal{H} = (\mathcal{V} = [n], \mathcal{E})$.*
- *Each vertex $i \in \mathcal{V} = [n]$ corresponds to a variable X_i over a discrete domain $\mathsf{Dom}(X_i)$.*
- *For each hyperedge $S \in \mathcal{E}$, there is a corresponding input function (also called a* factor*) $\psi_S : \prod_{i \in S} \mathsf{Dom}(X_i) \to \mathbf{D}$ for some fixed $\mathbf{D}$.*
- *A number $f \in [n]$. Let $F := [f]$. Variables in $\boldsymbol{X}_F$ are called* free variables*, while variables in $\boldsymbol{X}_{[n]-F}$ are called* bound variables*.*
- *For each $i \in [n] - F$, there is a commutative semiring $(\mathbf{D}, \bigoplus^{(i)}, \bigotimes)$. (All the semirings share the same $\mathbf{D}$ and $\bigotimes$ but can potentially have different $\bigoplus^{(i)}$).*[4]

The FAQ *problem is to compute the following function $\varphi(\boldsymbol{x}_F) : \prod_{i \in F} \mathsf{Dom}(X_i) \to \mathbf{D}$*

$$\varphi(\boldsymbol{x}_F) := \bigoplus_{x_{f+1}}^{(f+1)} \cdots \bigoplus_{x_n}^{(n)} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_s). \tag{4}$$

[4] More generally, instead of $(\mathbf{D}, \oplus^{(i)}, \otimes)$ being a semiring, we also allow some $\oplus^{(i)}$ to be identical to $\otimes$.

Consider the conjunctive query $Q(\boldsymbol{X}_F) \leftarrow \bigwedge_{S\in\mathcal{E}} R_S(\boldsymbol{X}_S)$ where $\boldsymbol{X}_F$ is the set of free variables. The FAQ framework models each input relation R_S as a Boolean function $\psi_S(\boldsymbol{x}_S)$, called a "factor", in which $\psi_S(\boldsymbol{x}_S) = \mathsf{true}$ iff $\boldsymbol{x}_S \in R_S$. Then, computing the output $Q(\boldsymbol{X}_F)$ is equivalent to computing the Boolean function $\varphi(\boldsymbol{x}_F)$ defined as $\varphi(\boldsymbol{x}_F) = \bigvee_{x_{f+1}} \cdots \bigvee_{x_n} \bigwedge_{S\in\mathcal{E}} \psi_S(\boldsymbol{x}_S)$. Instead of Boolean functions, this expression can be defined in SumProd form over functions on a commutative semiring $(\boldsymbol{D}, \oplus, \otimes)$:

$$\varphi(\boldsymbol{x}_F) = \bigoplus_{x_{f+1}} \cdots \bigoplus_{x_n} \bigotimes_{S\in\mathcal{E}} \psi_S(\boldsymbol{x}_S). \tag{5}$$

The semiring $(\{\mathsf{true}, \mathsf{false}\}, \vee, \wedge)$ was used for Q above.

We next define tree decompositions and the fhtw and subw parameters. We refer the reader to the recent survey by Gottlob et al. [15] for more details and a historical context. In what follows, the hypergraph $\mathcal{H}$ should be thought of as the hypergraph of the input FAQ query, although the notions of tree decomposition and width parameters are defined independently of queries.

A *tree decomposition* of a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is a pair (T, χ), where T is a tree and $\chi : V(T) \to 2^{\mathcal{V}}$ maps each node t of the tree to a subset $\chi(t)$ of vertices such that:

(a) Every hyperedge $S \in \mathcal{E}$ is a subset of some $\chi(t)$, $t \in V(T)$ (i.e. every edge is covered by some bag);

(b) For every vertex $v \in \mathcal{V}$, the set $\{t \mid v \in \chi(t)\}$ is a non-empty (connected) sub-tree of T. This is called the *running intersection property*.

The sets $\chi(t)$ are often called the *bags* of the tree decomposition. Let $\mathsf{TD}(\mathcal{H})$ denote the set of all tree decompositions of $\mathcal{H}$. When $\mathcal{H}$ is clear from context, we use TD for brevity.

▶ **Definition 2.5** (F-connex tree decomposition [9, 32]). *Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ and a set $F \subseteq \mathcal{V}$, a tree decomposition (T, χ) of $\mathcal{H}$ is* F-connex *if there is a subset $V' \subseteq V(T)$ that forms a connected subtree of T and satisfies $\bigcup_{t\in V'} \chi(t) = F$. We use TD_F to denote the set of all F-connex tree decompositions of $\mathcal{H}$. (Note that when $F = \emptyset$, $\mathsf{TD}_F = \mathsf{TD}$.)*

To define width parameters, we use the polymatroid characterization from [3]. A function $f : 2^{\mathcal{V}} \to \mathbb{R}_+$ is called a (non-negative) *set function* on $\mathcal{V}$. A set function f on $\mathcal{V}$ is *modular* if $f(S) = \sum_{v\in S} f(\{v\})$ for all $S \subseteq \mathcal{V}$, it is *monotone* if $f(X) \le f(Y)$ whenever $X \subseteq Y \subseteq \mathcal{V}$, and it is *submodular* if $f(X \cup Y) + f(X \cap Y) \le f(X) + f(Y)$ for all $X, Y \subseteq \mathcal{V}$. A monotone, submodular set function $h : 2^{\mathcal{V}} \to \mathbb{R}_+$ with $h(\emptyset) = 0$ is called a *polymatroid*. Let Γ_n denote the set of all polymatroids on $\mathcal{V} = [n]$.

Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, define the set of *edge dominated* set functions, denoted by $\mathsf{ED}_{\mathcal{H}}$ or ED when $\mathcal{H}$ is clear from the context, as follows:

$$\mathsf{ED} := \{h \mid h : 2^{\mathcal{V}} \to \mathbb{R}_+, h(S) \le 1, \forall S \in \mathcal{E}\}. \tag{6}$$

We can now define the submodular width and fractional hypertree width of a given hypergraph $\mathcal{H}$ (or of a given FAQ query with hypergraph $\mathcal{H}$): [5]

$$\mathsf{fhtw}(\mathcal{H}) := \min_{(T,\chi)\in\mathsf{TD}} \max_{h\in\mathsf{ED}\cap\Gamma_n} \max_{t\in V(T)} h(\chi(t)), \quad \mathsf{fhtw}_F(\mathcal{H}) := \min_{(T,\chi)\in\mathsf{TD}_F} \max_{h\in\mathsf{ED}\cap\Gamma_n} \max_{t\in V(T)} h(\chi(t)),$$

$$\mathsf{subw}(\mathcal{H}) := \max_{h\in\mathsf{ED}\cap\Gamma_n} \min_{(T,\chi)\in\mathsf{TD}} \max_{t\in V(T)} h(\chi(t)), \quad \mathsf{subw}_F(\mathcal{H}) := \max_{h\in\mathsf{ED}\cap\Gamma_n} \min_{(T,\chi)\in\mathsf{TD}_F} \max_{t\in V(T)} h(\chi(t)).$$

[5] Although this definition for fhtw differs from the original one [17, 15], the two definitions have been shown to be equivalent [3].

It is known that $\mathsf{subw}(\mathcal{H}) \leq \mathsf{fhtw}(\mathcal{H})$, and there are classes of hypergraphs with bounded subw and unbounded fhtw [23]. Furthermore, fhtw is strictly less than other width notions such as (generalized) hypertree width and tree width.

▶ **Theorem 2.6** ([2]). *Given an* FAQ φ *over a single semiring with hypergraph* $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ *and free variables* $F \subseteq \mathcal{V}$ *over a database of size* N*, the* InsideOut *algorithm can answer* φ *in time* $O(|\mathcal{E}| \cdot |\mathcal{V}|^2 \cdot \log N \cdot (N^{\mathsf{fhtw}_F(\mathcal{H})} + |\mathsf{output}|))$.

▶ **Theorem 2.7** ([3]). *Given an* FAQ φ *over the Boolean semiring with hypergraph* $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ *and free variables* $F \subseteq \mathcal{V}$ *over a database of size* N*, the* PANDA *algorithm can answer* φ *in time* $O(|\mathcal{V}| \cdot 2^{2^{|\mathcal{V}|}} \cdot (\mathsf{poly}(\log N) \cdot N^{\mathsf{subw}_F(\mathcal{H})} + \log N \cdot |\mathsf{output}|))$.

3 Example

We illustrate our approach using the following Boolean query[6]:

$$C \leftarrow R(X,Y), S(Y,Z), \neg T(X,Z) \tag{7}$$

where all input relations have sizes upper bounded by N and thus the *active* domain of any variable X has size at most N. The query C can be answered trivially in time $O(N^2)$ by joining R and S first, and then, for each triple (x,y,z) in the join, by verifying whether $(x,z) \notin T$ with a (hash) lookup. Suppose we know that the degree of relation T is less than two. Can we do better than $O(N^2)$ in that case? The answer is YES.

Rewriting to not-all-equal predicates

By viewing T as a bipartite graph of maximum degree two, it is easy to see that T can be written as a disjoint union of two relations $M_1(X,Z)$ and $M_2(X,Z)$ that represent *matchings* in the following sense: for any $i \in [2]$, if $(x,z) \in M_i$ and $(x',z') \in M_i$, then either $(x,z) = (x',z')$ or $x \neq x'$ and $z \neq z'$. Let $\mathsf{Dom}(Z)$ denote the active domain of the variable Z. Define, for each $i \in [2]$, a singleton relation $W_i(Z) \leftarrow \mathsf{Dom}(Z) \wedge \neg(\pi_Z M_i)(Z)$. Clearly, $|W_i| \leq N$ and given M_i, W_i can be computed in $O(N)$ preprocessing time. For each $i \in [2]$, create a new variable X_i with domain $\mathsf{Dom}(X_i) = \mathsf{Dom}(X)$. Then,

$$\neg M_i(X,Z) \equiv W_i(Z) \vee \exists X_i \big[M_i(X_i,Z) \wedge \mathsf{NAE}(X,X_i) \big]. \tag{8}$$

The predicate NAE stands for not-all-equal: It is the negation of the conjunction of pairwise equality on its variables. For arity two as in the rewriting of $\neg M_i(X,Z)$, $\mathsf{NAE}(X,X_i)$ stands for the disequality $X \neq X_i$.

From $T = M_1 \vee M_2$ and (8), we can rewrite the original query C from (7) into a disjunction of Boolean conjunctive queries without negated relations but with one or two extra existential variables that are involved in *disequalities* ($\neq$): $C \equiv \bigvee_{i \in [4]} C_i$, where

$$\begin{aligned}
C_1 &\leftarrow R(X,Y) \wedge S(Y,Z) \wedge W_1(Z) \wedge W_2(Z).\\
C_2 &\leftarrow R(X,Y) \wedge S(Y,Z) \wedge W_1(Z) \wedge M_2(X_2,Z) \wedge X \neq X_2.\\
C_3 &\leftarrow R(X,Y) \wedge S(Y,Z) \wedge W_2(Z) \wedge M_1(X_1,Z) \wedge X \neq X_1.\\
C_4 &\leftarrow R(X,Y) \wedge S(Y,Z) \wedge M_1(X_1,Z) \wedge M_2(X_2,Z) \wedge X \neq X_1 \wedge X \neq X_2.
\end{aligned}$$

[6] If R, S, and T would record direct train connections between cities, then this query would ask whether there exists a pair of cities with no direct train connection but with connections via another city.

It takes linear time to compute the matching decomposition of T into M_1 and M_2 since: (1) the relation T is a bipartite graph with degree at most two, and it is thus a union of even cycles and paths; and (2) we can traverse the cycles and paths and add alternative edges to M_1 and M_2. In general, when the maximum degree is higher and when T is not a binary predicate, Proposition 2.3 shows how to decompose a relation into high-dimensional matchings efficiently. The number of queries C_i depends exponentially on the arities and degrees of the negated relations.

Boolean tensor decomposition

The acyclic query C_1 can be answered in $O(N \log N)$ time using for instance InsideOut [2]; this algorithm first sorts the input relations in time $O(N \log N)$. The query C_2 can be answered as follows. Let $\forall i \in [\log N], f_i : \mathsf{Dom}(X) \to \{0,1\}$ denote the function such that $f_i(X)$ is the ith bit of X in its binary representation. Then, by noticing that

$$X \neq X_2 \equiv \bigvee_{b \in \{0,1\}} \bigvee_{i \in [\log N]} f_i(X) = b \wedge f_i(X_2) \neq b \tag{9}$$

we can break up the query C_2 into the disjunction of $2 \log N$ acyclic queries of the form

$$C_2^{b,i} \leftarrow R(X,Y) \wedge S(Y,Z) \wedge W_1(Z) \wedge M_2(X_2,Z) \wedge f_i(X) = b \wedge f_i(X_2) \neq b. \tag{10}$$

For a fixed b, both $f_i(X) = b$ and $f_i(X_2) \neq b$ are singleton relations on X and X_2, respectively. Then, C_2 can be answered in time $O(N \log^2 N)$. The same applies to C_3. We can use the same trick to answer C_4 in time $O(N \log^3 N)$. However, we can do better than that by observing that when viewed as a Boolean tensor in (9), the disequality tensor has the Boolean rank bounded by $O(\log N)$. In order to answer C_4 in time $O(N \log^2 N)$, we will show that the three-dimensional tensor $(X \neq X_1) \wedge (X \neq X_2)$ has the Boolean rank bounded by $O(\log N)$ as well. To this end, we extend the color-coding technique. We can further shave off a $\log N$ factor in the complexities of C_2, C_3, and C_4, as explained in Section 6.

Construction of the Boolean tensor decomposition

We next explain how to compute a tensor decomposition for the conjunction of disequalities in C_4. We show that there exists a family $\mathcal{F}$ of functions $f : \mathsf{Dom}(X) \to \{0,1\}$ satisfying the following conditions:

- **(i)** $|\mathcal{F}| = O(\log |\mathsf{Dom}(X)|) = O(\log N)$,
- **(ii)** For every triple $(x, x_1, x_2) \in \mathsf{Dom}(X)^3$ for which $x \neq x_1 \wedge x \neq x_2$, there is a function $f \in \mathcal{F}$ such that $f(x) \neq f(x_1) \wedge f(x) \neq f(x_2)$, and
- **(iii)** $\mathcal{F}$ can be constructed in time $O(N \log N)$.

We think of each function f as a "coloring" that assigns a "color" in $\{0,1\}$ to each element of $\mathsf{Dom}(X)$. Assuming (i) to (iii) hold, it follows that

$$X \neq X_1 \wedge X \neq X_2 \equiv \bigvee_{(c,c_1,c_2)} \bigvee_{f \in \mathcal{F}} f(X) = c \wedge f(X_1) = c_1 \wedge f(X_2) = c_2, \tag{11}$$

where (c, c_1, c_2) ranges over all triples in $\{0,1\}^3$ such that $c \neq c_1$ and $c \neq c_2$. Given this Boolean tensor decomposition, we can solve C_4 in time $O(N \log^2 N)$.

We prove (i) to (iii) using a combinatorial object called the *disjunct matrices*. These matrices are the central subject of combinatorial group testing [24, 12].

▶ **Definition 3.1** (k-disjunct matrix). *A $t \times N$ binary matrix $\boldsymbol{A} = (a_{ij})$ is called a* k-disjunct matrix *if for every column $j \in [N]$ and every set $S \subseteq [N]$ such that $|S| \leq k$ and $j \notin S$, there exists a row $i \in [t]$ for which $a_{ij} = 1$ and $a_{ij'} = 0$ for all $j' \in S$.*

It is known that for every integer $k < \sqrt{N}$, there exists a k-disjunct matrix (or equivalently a combinatorial group testing [24]) with $t = O\left(k^2 \log N\right)$ rows that can be constructed in time $O(k^2 N \log N)$ [31]. (If $k \geq \sqrt{N}$, we can just use the identity matrix.) In particular, for $N = |\mathsf{Dom}(X)|$ and $k = 2$, a 2-disjunct matrix $\boldsymbol{A} = (a_{ij})$ of size $O(\log N) \times N$ can be constructed in time $O(N \log N)$. From the matrix we define the function family $\mathcal{F}$ by associating a function f_i to each row i of the matrix, and every member $x \in \mathsf{Dom}(X)$ to a distinct column j_x of the matrix. Define $f_i(x) = a_{i,j_x}$ and (i)–(iii) straightforwardly follow.

4 Untangling bounded-degree relations

In this section we introduce a rewriting of queries of the form (1) into queries with so-called *not-all-equal* predicates, under the assumption that the relation $R_S(\boldsymbol{X}_S)$ for every hyperedge $S \in \overline{\mathcal{E}}$ has bounded degree $\deg(R_S)$.

▶ **Definition 4.1** (Not-all-equal). *Let $k \geq 2$ be an integer, and S be a set of k integers. The relation $\mathsf{NAE}_k(\boldsymbol{X}_S)$, or $\mathsf{NAE}(\boldsymbol{X}_S)$ for simplicity, holds true iff not all variables in $\boldsymbol{X}_S$ are equal:* $\mathsf{NAE}(\boldsymbol{X}_S) = \neg \bigwedge_{\{i,j\} \in \binom{S}{2}} X_i = X_j$.

The disequality ($\neq$) relation is exactly NAE_2. The negation of a matching is connected to NAE predicates as follows.

▶ **Proposition 4.2.** *Let $M(\boldsymbol{X}_S)$ be a k-ary matching, where $k = |S| \geq 2$. For any $i, j \in S$, define the unary relation $W_i(X_i) \leftarrow \mathsf{Dom}(X_i) \wedge \neg(\pi_i M)(X_i)$ and the binary relation $M_{ij} = \pi_{i,j} M$. For any $\ell \in S$, it holds that*

$$\neg M(\boldsymbol{X}_S) \equiv \left(\bigvee_{i \in S \setminus \{\ell\}} W_i(X_i) \right) \vee \exists \boldsymbol{Y}_{S \setminus \{\ell\}} \left[\mathsf{NAE}(X_\ell, \boldsymbol{Y}_{S \setminus \{\ell\}}) \wedge \bigwedge_{j \in S \setminus \{\ell\}} M_{\ell j}(Y_j, X_j) \right]. \quad (12)$$

Proof. The intuition for this rewriting is as follows. A value $x_i \in \mathsf{Dom}(X_i)$ occurs in at most one tuple in the matching M. Therefore, any value in a tuple determines the rest of the tuple. The rewriting in (12) first turns every tuple in M into a tuple whose values are all the same, i.e., all-equal values. The negation of M consists of tuples with at least two different values, i.e., not-all-equal values.

We next prove that the rewriting is correct.

In one direction, consider a tuple $\boldsymbol{x}_S \notin M$, i.e., $\neg M(\boldsymbol{x}_S)$ holds, and suppose $x_i \notin W_i$ for all $i \in S \setminus \{\ell\}$. This means, for every $i \in S \setminus \{\ell\}$, there is a unique tuple $\boldsymbol{t}^{(i)} = (t_j^{(i)})_{j \in S} \in M$ such that $x_i = t_i^{(i)}$. Define $y_j = t_\ell^{(j)}$ for all $j \in S \setminus \{\ell\}$. The tuple $\boldsymbol{y}_{S \setminus \{\ell\}}$ satisfies $(y_j, x_j) = (t_\ell^{(j)}, t_j^{(j)}) \in M_{\ell j}$, for all $j \in S \setminus \{\ell\}$. Moreover, one can verify that $\mathsf{NAE}(x_\ell, \boldsymbol{y}_{S \setminus \{\ell\}})$ holds. In particular, if $y_j = x_\ell$ for all $j \in S \setminus \{\ell\}$, then all tuples $\boldsymbol{t}^{(j)} \in M$ are the same tuple (since M is a matching) and that tuple is $\boldsymbol{x}_S$. Hence $\boldsymbol{x}_S \in M$ which is a contradiction.

Conversely, suppose there exists a tuple $(\boldsymbol{x}_S, \boldsymbol{y}_{S \setminus \{\ell\}})$ satisfying the right hand side of (12). If $x_i \in W_i$ for any $i \in S \setminus \{\ell\}$, then $\boldsymbol{x}_S \notin M$, i.e., $\boldsymbol{x}_S$ satisfies the left hand side of (12). Now, suppose $x_i \notin W_i$ for all $i \in S \setminus \{\ell\}$. Suppose to the contrary that $\boldsymbol{x}_S \in M$. Then, for all $j \in S \setminus \{\ell\}$ we have $y_j = x_\ell$ since $M_{\ell j}(y_j, x_j)$ must hold. This means that $\mathsf{NAE}(x_\ell, \boldsymbol{y}_{S \setminus \{\ell\}}) = \neg \wedge_{j \in S \setminus \{\ell\}} x_\ell = y_j$ does not hold. This contradicts our hypothesis. ◀

We use the connection to NAE predicates to decompose a query containing a conjunction of negated bounded-degree relations into a disjunction of positive terms, as given next by Proposition 4.3. We call this rewriting *untangling*.

Let fhtw_F and subw_F denote the fractional hypertree width and respectively the submodular width of the conjunctive query $Q(\boldsymbol{X}_F) \leftarrow \mathsf{body}$ (These notions are defined in Section 2.2).

▶ **Proposition 4.3.** *Let Q be the query defined in Eq. (1): $Q(\boldsymbol{X}_F) \leftarrow \mathit{body} \wedge \bigwedge_{S\in\overline{\mathcal{E}}} \neg R_S(\boldsymbol{X}_S)$. We can compute in linear time a collection of B hypergraphs $\mathcal{H}_i = (\mathcal{V}_i, \mathcal{E}_i)$ such that*

$$Q(\boldsymbol{X}_F) \equiv \bigvee_{i\in[B]} Q_i(\boldsymbol{X}_F), \qquad \text{where } \forall i \in [B] : Q_i(\boldsymbol{X}_F) \leftarrow \mathit{body}_i \wedge \bigwedge_{S\in\mathcal{E}_i} \mathsf{NAE}(\boldsymbol{Z}_S), \tag{13}$$

and body_i is the body of a conjunctive query satisfying

$$\mathsf{fhtw}_F(\mathit{body}_i) \le \mathsf{fhtw}_F(\mathit{body}), \qquad \text{and} \qquad \mathsf{subw}_F(\mathit{body}_i) \le \mathsf{subw}_F(\mathit{body}).$$

Furthermore, the number B of queries is bounded by $B \le \prod_{S\in\overline{\mathcal{E}}} (|S|)^{|S|(\deg(R_S)-1)+1}$.

Proof. From Proposition 2.3, each relation $R_S(\boldsymbol{X}_S)$ can be written as a disjoint union of $D_S \le |S|(\deg(R_S)-1)+1$ matchings M_S^ℓ, $\ell \in [D_S]$. These matchings can be computed in linear time. Hence, the second half of the body of query Q can be rewritten equivalently as

$$\bigwedge_{S\in\overline{\mathcal{E}}} \neg R_S(\boldsymbol{X}_S) \equiv \bigwedge_{S\in\overline{\mathcal{E}}} \neg \bigvee_{\ell\in[D_S]} M_S^\ell(\boldsymbol{X}_S) \equiv \bigwedge_{\substack{S\in\overline{\mathcal{E}}\\ \ell\in[D_S]}} \neg M_S^\ell(\boldsymbol{X}_S).$$

To simplify notation, let $\overline{\mathcal{E}}_1$ denote the multiset of edges obtained from $\overline{\mathcal{E}}$ by duplicating the edge $S \in \overline{\mathcal{E}}$ exactly D_S times. Furthermore, for the ℓ-th copy of S, associate the matching M_S^ℓ with the copy of S in $\overline{\mathcal{E}}_1$; use M_S to denote the matching corresponding to that copy. Then, we can write Q equivalently $Q(\boldsymbol{X}_F) \leftarrow \mathsf{body} \wedge \bigwedge_{S\in\overline{\mathcal{E}}_1} \neg M_S(\boldsymbol{X}_S)$.

For each $S \in \overline{\mathcal{E}}_1$, fix an arbitrary integer $\ell_S \in S$. From Proposition 4.2, the negation of M_S can be written as

$$\neg M_S(\boldsymbol{X}_S) \equiv \Big(\bigvee_{i\in S\setminus\{\ell_S\}} W_i^S(X_i)\Big) \vee \exists \boldsymbol{Y}^S_{S\setminus\{\ell_S\}} \Big[\bigwedge_{j\in S\setminus\{\ell_S\}} (\pi_{\ell_S,j} M_S)(Y_j^S, X_j) \wedge \mathsf{NAE}(X_{\ell_S}, \boldsymbol{Y}^S_{S\setminus\{\ell_S\}})\Big],$$

where W_i^S is a unary relation on variable X_i, and $\boldsymbol{Y}^S_{S\setminus\{\ell_S\}} = (Y_i^S)_{i\in S\setminus\{\ell_S\}}$ is a tuple of fresh variables, only associated with (the copy of) S. In particular, if S and S' are two distinct items in the multiset $\overline{\mathcal{E}}_1$, then Y_i^S and $Y_i^{S'}$ are two distinct variables.

Each negated term $\neg M_S(\boldsymbol{X}_S)$ is thus expressed as a disjunction of $|S|$ positive terms. We can then express the conjunction of $|\overline{\mathcal{E}}_1|$ negated terms as the disjunction of $\prod_{S\in\overline{\mathcal{E}}_1} |S|$ conjunctions. For this, define a collection of tuples $\mathcal{T} = \prod_{S\in\overline{\mathcal{E}}_1} S$. In particular, every member $\boldsymbol{T} \in \mathcal{T}$ is a tuple $\boldsymbol{T} = (t_S)_{S\in\overline{\mathcal{E}}_1}$ where $t_S \in S$. The second half of the body of query Q can be rewritten equivalently as

$$\begin{aligned}
&\bigwedge_{S\in\overline{\mathcal{E}}} \neg R_S(\boldsymbol{X}_S) \equiv \bigwedge_{S\in\overline{\mathcal{E}}_1} \neg M_S(\boldsymbol{X}_S)\\
&\equiv \bigwedge_{S\in\overline{\mathcal{E}}_1} \left(\bigvee_{i\in S\setminus\{\ell_S\}} W_i^S(X_i) \vee \exists \boldsymbol{Y}^S_{S\setminus\{\ell_S\}} \Big[\bigwedge_{j\in S\setminus\{\ell_S\}} (\pi_{\ell_S,j} M_S)(Y_j^S, X_j) \wedge \mathsf{NAE}\big(X_{\ell_S}, \boldsymbol{Y}^S_{S\setminus\{\ell_S\}}\big)\Big] \right)\\
&\equiv \bigvee_{\boldsymbol{T}\in\mathcal{T}} \bigwedge_{\substack{S\in\overline{\mathcal{E}}_1\\ t_S\neq\ell_S}} W_{t_S}^S(X_{t_S}) \wedge \bigwedge_{\substack{S\in\overline{\mathcal{E}}_1\\ t_S=\ell_S}} \exists \boldsymbol{Y}^S_{S\setminus\{\ell_S\}} \Big[\bigwedge_{j\in S\setminus\{\ell_S\}} (\pi_{\ell_S,j} M_S)(Y_j^S, X_j) \wedge \mathsf{NAE}\big(X_{\ell_S}, \boldsymbol{Y}^S_{S\setminus\{\ell_S\}}\big)\Big]
\end{aligned}$$

The original query Q is equivalent to the disjunction

$$Q(\boldsymbol{X}_F) \equiv \bigvee_{\boldsymbol{T}\in\mathcal{T}} Q_{\boldsymbol{T}}(\boldsymbol{X}_F)$$

of up to $\prod_{S\in\overline{\mathcal{E}}_1}|S|$ queries $Q_{\boldsymbol{T}}$ defined by

$$\underbrace{\mathsf{body} \wedge \bigwedge_{\substack{S\in\overline{\mathcal{E}}_1\\ t_S\neq\ell_S}} W^S_{t_S}(X_{t_S}) \wedge \bigwedge_{\substack{S\in\overline{\mathcal{E}}_1\\ t_S=\ell_S\\ j\in S\setminus\{\ell_S\}}} (\pi_{\ell_S,j}M_S)(Y^S_j, X_j)}_{\mathsf{body}_i} \wedge \bigwedge_{\substack{S\in\overline{\mathcal{E}}_1\\ t_S=\ell_S}} \mathsf{NAE}\big(X_{\ell_S}, \boldsymbol{Y}^S_{S\setminus\{\ell_S\}}\big) \tag{14}$$

In the above definition of $Q_{\boldsymbol{T}}$, let us denote all but the last conjunction of NAE predicates by body_i. It holds that $\mathsf{fhtw}_F(\mathsf{body}_i) \le \mathsf{fhtw}_F(\mathsf{body})$, and $\mathsf{subw}_F(\mathsf{body}_i) \le \mathsf{subw}_F(\mathsf{body})$ [1]. We now turn to the conjunction of NAE predicates in (14). Since each $S \in \overline{\mathcal{E}}$ is repeated at most $|S|(\deg(R_S)-1)+1$ times in $\overline{\mathcal{E}}_1$, it follows that the number $\prod_{S\in\overline{\mathcal{E}}_1}|S|$ of conjunctive queries $Q_{\boldsymbol{T}}$ is at most $\prod_{S\in\overline{\mathcal{E}}}|S|^{|S|(\deg(R_S)-1)+1}$. ◀

5 Boolean tensor decomposition

Thanks to the untangling result in Proposition 4.3, we only need to concentrate on answering queries of the form (13). To deal with the conjunction of NAE predicates, this section describes the construction of a Boolean tensor decomposition of a conjunction $\bigwedge_{S\in\mathcal{A}}\mathsf{NAE}(\boldsymbol{X}_S)$ of NAE predicates. The multi-hypergraph of this conjunction has the query variables as vertices and the NAE predicates as hyperedges.

▶ **Lemma 5.1.** *Let $\mathcal{G}=(U,\mathcal{A})$ be the multi-hypergraph of a conjunction $\bigwedge_{S\in\mathcal{A}}\mathsf{NAE}(\boldsymbol{X}_S)$, N an upper bound on the domain sizes for variables $(X_i)_{i\in U}$, and c a positive integer. Suppose there exists a family $\mathcal{F}$ of functions $f:[N]\to[c]$ satisfying the following property*

$$\text{for any proper } N\text{-coloring } h: U\to[N] \text{ of } \mathcal{G} \text{ there exists a function } f\in\mathcal{F} \text{ such that } f\circ h \text{ is a proper } c\text{-coloring of } \mathcal{G}. \tag{15}$$

Then, the following holds:

$$\bigwedge_{S\in\mathcal{A}}\mathsf{NAE}(\boldsymbol{X}_S) \equiv \bigvee_{g}\bigvee_{f\in\mathcal{F}}\bigwedge_{i\in U} f(X_i)=g(i), \tag{16}$$

where g ranges over all proper c-colorings of $\mathcal{G}$. In particular, the Boolean tensor rank of the left-hand side of (16) is bounded by $r=P(\mathcal{G},c)\cdot|\mathcal{F}|$.

Proof. Let $\boldsymbol{x}_U$ denote any tuple satisfying the LHS of (16). Define $h:U\to[N]$ by setting $h(i)=x_i$. Then h is a proper N-coloring of $\mathcal{G}$, which means there exists $f\in\mathcal{F}$ such that $g=f\circ h$ is a proper c-coloring of $\mathcal{G}$. Then the conjunct on the RHS corresponding to this particular pair (g,f) is satisfied.

Conversely, let $\boldsymbol{x}_U$ denote any tuple satisfying the RHS of (16). Then, there is a pair (g,f) whose corresponding conjunct on the RHS of (16) is satisfied, i.e., $f(x_i)=g(i)$ for all $i\in U$. Recall that g is a proper c-coloring of $\mathcal{G}$. If there exists $S\in\mathcal{A}$ such that $\mathsf{NAE}(\boldsymbol{x}_S)$ does *not* hold, then $x_i=x_j$ for all $i,j\in S$, implying $g(i)=f(x_i)=f(x_j)=g(j)$ for all $i,j\in S$, contradicting the fact that g is a proper coloring.

For the Boolean tensor rank statement, note that (16) is a Boolean tensor decomposition of the formula $\bigwedge_{S\in\mathcal{A}}\mathsf{NAE}(\boldsymbol{X}_S)$, because $f(X_i)=g(i)$ is a unary predicate on variable X_i. This predicate is of size bounded by N. ◀

To explain how Lemma 5.1 can be applied, we exemplify two techniques, showing the intimate connections of our Boolean tensor decomposition problem to combinatorial group testing and perfect hashing.

▶ **Example 5.2** (Connection to group testing)**.** Consider the case when the graph $\mathcal{G}$ is a *k-star*, i.e., a tree with a center vertex and k leaf vertices. Let $\boldsymbol{A}$ be a $O(k^2 \log N) \times N$ binary k-disjunct matrix, which can be constructed in time $O(kN \log N)$ (This is due to known results on k-restriction and error codes, recalled in the extended paper [1]). We can assume $k < \sqrt{N}$ to avoid triviality. Consider a family $\mathcal{F}$ of functions $f : [N] \to \{0, 1\}$ constructed as follows: there is a function f for every row i of $\boldsymbol{A}$, where $f(j) = a_{ij}$, for all $j \in [N]$. The family $\mathcal{F}$ has size $O(k^2 \log N)$. We show that $\mathcal{F}$ satisfies condition (15). Let $h : U \to [N]$ denote any coloring of the star. Let $j \in [N]$ be the color h assigns to the center, and S be the set of colors assigned to the leaf nodes. Clearly $j \notin S$. Hence, there is a function $f \in \mathcal{F}$ for which $f(j) = 1$ and $f(j') = 0$ for all $j' \in S$, implying $f \circ h$ is a proper 2-coloring of $\mathcal{G}$.

A consequence of our observation is that for a k-star $\mathcal{G}$ the conjunction $\bigwedge_{S \in \mathcal{A}} \mathsf{NAE}(\boldsymbol{X}_S)$ has Boolean rank bounded by $O(k^2 \log N)$.

▶ **Example 5.3** (Connection to perfect hashing)**.** Consider now the case when the graph $\mathcal{G}$ is a k-clique. Let $\mathcal{F}$ denote any (N, c, k)-perfect hash family, i.e., a family of hash functions from $[N] \to [c]$ such that for every subset $S \subseteq [N]$ of size k, there is a function f in the family for which its image is also of size k. It is easy to see that this hash family satisfies (15). From [6], it is known that we can construct in polytime an (N, k^2, k)-perfect hash family of size $O(k^4 \log N)$. However, it is not clear what the runtime exponent of their construction is. What we need for our application is that the construction should run in linear data complexity and in polynomial query complexity. We use a result from [31] to exhibit such a construction in Theorem 5.6; furthermore, our hash family has size only $O(k^2 \log N)$.

We next construct the smallest family $\mathcal{F}$ satisfying Lemma 5.1. We first bound the size of $\mathcal{F}$ using the probabilistic method [7] and then specify how to derandomize the probabilistic construction of $\mathcal{F}$ to obtain a deterministic algorithm. For this, we need some terminology.

Every coloring $h : U \to [N]$ of $\mathcal{G} = (U, \mathcal{A})$ induces a homomorphic image $h(\mathcal{G}) = (h(U), h(\mathcal{A}))$, which is the graph on vertex set $h(U)$ and edge set $h(\mathcal{A})$ defined by

$$h(U) = \{h(v) \mid v \in U\} \subseteq [N], \qquad h(\mathcal{A}) = \big\{h(S) = \{h(v) \mid v \in S\} \mid S \in \mathcal{A}\big\} \subseteq 2^{[N]}.$$

Here, we overload notation to allow h range over sets and graphs. Let $\mathsf{col}(\mathcal{G}, N)$ denote the set of proper N-colorings h of $\mathcal{G}$. Each such proper N-coloring is a homomorphic image of $\mathcal{G}$. Define c as the maximum chromatic number over all homomorphic images of $\mathcal{G}$: $c = \max_{h \in \mathsf{col}(\mathcal{G}, N)} \chi(h(\mathcal{G}))$. For a given $h \in \mathsf{col}(\mathcal{G}, N)$, let $g : h(U) \to [c]$ be a proper c-coloring of $h(\mathcal{G})$. The *multiplicity* of a color $i \in [c]$ is the number of vertices colored i by g. The *signature* of g is the vector $\boldsymbol{\mu}(g) = (\mu_i)_{i \in [c]}$, where μ_i is the multiplicity of color i. Let $T_c(h)$ denote the collection of all signatures of proper c-colorings of $h(\mathcal{G})$. For a given signature $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_c) \in T_c(h)$, let $n(\boldsymbol{\mu}, h)$ denote the number of proper c-colorings of $h(\mathcal{G})$ whose signature is $\boldsymbol{\mu}$.

▶ **Example 5.4.** Suppose $\mathcal{G}$ is the k-clique and $c = k$. Then, every proper k-coloring of $h(\mathcal{G})$ has signature $\boldsymbol{\mu} = \mathbf{1}_k = (1, 1, \ldots, 1)$: $T_c(h)$ has only one member, but $n(\mathbf{1}, h) = k!$. If $\mathcal{G}$ is the k-star then $c = 2$ and for any $h \in \mathsf{col}(\mathcal{G}, N)$, $h(\mathcal{G})$ is an ℓ-star for some $\ell \in [k]$. Then, $T_2(h)$ has two signatures: $\boldsymbol{\mu} = (\ell, 1)$ and $\boldsymbol{\mu}' = (1, \ell)$; furthermore, $n(\boldsymbol{\mu}, h) = n(\boldsymbol{\mu}', h) = 1$.

▶ **Definition 5.5** (Strongly Explicit Construction)**.** *A family $\mathcal{F}$ of functions $f : [N] \to [c]$ is said to be* strongly explicit *if there is an algorithm that, given an index to a function f in $\mathcal{F}$ and a number $j \in [N]$, returns $f(j)$ in $\mathsf{poly}(\log |\mathcal{F}|, \log N)$-time.*

The next theorem gives two upper bounds on the size of a family of hash functions satisfying (15) that we use to define the rank of our Boolean tensor decomposition: The first bound is for such families in general, whereas the second is for strongly explicit families that we can use effectively.

▶ **Theorem 5.6.** *Let $\mathcal{G} = (U, \mathcal{A})$ be a multi-hypergraph, $c = \max_{h \in \mathsf{col}(\mathcal{G},N)} \chi(h(\mathcal{G}))$, and $\boldsymbol{p} = (p_1, \ldots, p_c) \in \mathbb{R}^c_+$ be a fixed non-negative real vector such that $\|\boldsymbol{p}\|_1 = 1$. Define*

$$\theta(\boldsymbol{p}) = \min_{h \in \mathsf{col}(\mathcal{G},N)} \sum_{\boldsymbol{\mu} \in T_c(h)} n(\boldsymbol{\mu}, h) \prod_{i=1}^{c} p_i^{\mu_i} \tag{17}$$

Then, the following hold:

(a) *There exists a family $\mathcal{F}$ of functions $f : [N] \to [c]$ satisfying (15) such that*

$$|\mathcal{F}| \le \left\lceil \frac{\ln P(\mathcal{G}, N)}{\theta(\boldsymbol{p})} \right\rceil \le \frac{|U| \log N}{\theta(\boldsymbol{p})}. \tag{18}$$

(b) *There is a strongly explicit family $\mathcal{F}'$ of functions $f : [N] \to [c]$ satisfying (15) such that*

$$|\mathcal{F}'| = O\left(\frac{|U|^3 \cdot \log |U| \cdot \log N}{\theta(\boldsymbol{p})} \right). \tag{19}$$

The next corollary follows immediately from Lemma 5.1 and Theorem 5.6.

▶ **Corollary 5.7.** *Let $\mathcal{G} = (U, \mathcal{A})$ be a multi-hypergraph, $c = \max_{h \in \mathsf{col}(\mathcal{H},N)} \chi(h(\mathcal{G}))$, and*

$$\theta^* = \max_{\boldsymbol{p}: \|\boldsymbol{p}\|_1 = 1, \boldsymbol{p} \ge \boldsymbol{0}} \theta(\boldsymbol{p}), \tag{20}$$

where $\theta(\boldsymbol{p})$ is defined in (17). The following hold:

(a) *The Boolean rank of the function $\bigwedge_{F \in \mathcal{A}} \mathsf{NAE}(\boldsymbol{X}_F)$ is upper bounded by $\frac{P(\mathcal{G},c) \cdot \ln P(\mathcal{G},N)}{\theta^*}$.*

(b) *Given $\boldsymbol{p}$, there is a strongly explicit Boolean tensor decomposition of $\bigwedge_{F \in \mathcal{A}} \mathsf{NAE}(\boldsymbol{X}_F)$ whose rank is upper bounded by $P(\mathcal{G}, c) \cdot \frac{|U|^3 \cdot \log |U| \cdot \log N}{\theta(\boldsymbol{p})}$.*

To apply the above result, we need to specify $\boldsymbol{p}$ to maximize $\theta(\boldsymbol{p})$. We do not know how to compute the optimizer $\boldsymbol{p}^*$ in closed form. We next discuss several observations that allow us to bound θ^* from below or compute it exactly. In the following, for any tuple $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_\ell)$ of positive integers, let $K_{\boldsymbol{\mu}}$ denote the complete ℓ-partite graph defined as follows. For every $i \in [\ell]$ there is an independent set I_i of size μ_i. All independent sets are disjoint. The vertex set is $\bigcup_{i \in [\ell]} I_i$ and the vertices not belonging to the same independent set are connected. Without loss of generality, we assume $\mu_1 \ge \cdots \ge \mu_\ell$ when specifying the graph $K_{\boldsymbol{\mu}}$. For example, $K_{\boldsymbol{1}_k}$ is the k-clique, and $K_{(k,1)}$ is the k-star.

▶ **Proposition 5.8.** *The following hold:*

(a) *Given a multi-hypergraph $\mathcal{G} = (U, \mathcal{A})$ with $|U| = k$ and $c = \max_{h \in \mathsf{col}(\mathcal{G},N)} \chi(h(\mathcal{G}))$, it holds that $\theta^* \ge \frac{1}{c^c} \ge \frac{1}{k^k}$.*

(b) *Suppose $\mathcal{G} = K_{\boldsymbol{\mu}}$ for some positive integer tuple $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_\ell)$, where $\mu_1 \ge \cdots \ge \mu_\ell \ge 1$. Let S_ℓ denote the set of all permutations of $[\ell]$, and $\mathsf{FP}(\boldsymbol{\mu})$ denote the number of permutations $\pi \in S_\ell$ for which $\mu_i = \mu_{\pi(i)}, \forall i \in [\ell]$. Then,*

$$\theta^* = \max_{\boldsymbol{p}} \sum_{\pi \in S_\ell} \prod_{i=1}^{\ell} p_i^{\mu_{\pi(i)}} \ge \sum_{\pi \in S_\ell} \prod_{i=1}^{\ell} \left(\frac{\mu_i}{\|\boldsymbol{\mu}\|_1} \right)^{\mu_{\pi(i)}} \ge \mathit{FP}(\boldsymbol{\mu}) \prod_{i=1}^{\ell} \left(\frac{\mu_i}{\|\boldsymbol{\mu}\|_1} \right)^{\mu_i}. \tag{21}$$

▶ **Corollary 5.9.** *Let $\ell \in [k]$ be an integer. Let $\boldsymbol{\mu} = (k-\ell, \mathbf{1}_\ell)$. Then, when $\mathcal{G} = K_{\boldsymbol{\mu}}$ we have*

$$\theta^* \geq \frac{\ell!}{k^\ell}\left(\frac{k-\ell}{k}\right)^{k-\ell} \geq \frac{\ell!}{e^\ell}\frac{1}{k^\ell},$$

where $e = 2.7..$ is the base of the natural log. In particular, $\mathcal{G}$ is a $(k-1)$-star when $\ell = 1$ and the bound is $\theta^* \geq \frac{1}{ek}$. When $\ell = k$, then $\mathcal{G}$ is a k-clique and the bound is $\theta^* = k!/k^k$.

For any constant $\ell \in [k]$, the bound for θ^* is $\Omega(1/k^\ell)$; in particular, the lower bound for θ^* ranges anywhere between $\Omega(1/k)$, $\Omega(1/k^2)$, up to $\Omega(k!/k^k)$. There is a spectrum of these bounds, leading to a spectrum of Boolean tensor ranks for our decomposition.

▶ **Example 5.10.** From (18) and the above corollary, it follows that when $\mathcal{G}$ is a k-star, the corresponding Boolean rank is bounded by $O(k^2 \log N)$, matching the group testing connection from Example 5.2. The reason is twofold. We need two colors to color a k-star and the chromatic polynomial of a k-star using two colors is two. The size of the family $\mathcal{F}$ of hash functions is upper bounded by $\frac{|U|\log N}{\theta^*}$ where θ^* is at least $\frac{1}{ek}$ and $|U| = k+1$. Then, $|\mathcal{F}| \leq e \cdot k \cdot (k+1)\log N = O(k^2 \log N)$. This matches the tailor-made construction from Example 5.2. However, our strongly explicit construction in Theorem 5.6(b) yields a slightly larger Boolean tensor decomposition of rank $O(k^4 \log k \log N)$.

When applying part (b) of Proposition 5.8 to the problem of detecting k-paths in a graph, i.e., the query P in the introduction, we obtain the Boolean rank $O(\frac{k^{k+3}}{k!} \cdot \log k \cdot \log N)$. This is because (1) we would need two colors and the chromatic polynomial for the k-path hypergraph using two colors is two, and (2) the size of the family of strongly explicit functions is $O(\frac{(k+1)^3 \log(k+1) \log N}{\theta^*})$ with $\theta^* = k!/k^k$.

6 How to use the tensor decomposition

Sections 4 and 5 introduced two rewriting steps. The first step transforms a conjunctive query with negation of the form (1) into a disjunction of conjunctive queries with NAE predicates of the form (13). The second step transforms a conjunction of NAE predicates into a disjunction of conjunctions of one-variable-conditions of the form (16). The first step exploited the bounded degrees of the negated relations to bound from above the number of disjuncts and independently of the database size. The second step uses a generalization of the color-coding technique to further rewrite a conjunction of NAE predicates into a Boolean tensor decomposition whose rank depends on the structure of the multi-hypergraph of the conjunction. Both rewriting steps preserve the equivalence of the queries.

In this section, we show that the query obtained after the two rewriting steps can be evaluated efficiently. This query has the form $Q(\boldsymbol{X}_F) \leftarrow \bigvee_{j\in[B]} Q_j(\boldsymbol{X}_F)$ where $\forall j \in [B]$:

$$Q_j(\boldsymbol{X}_F) \leftarrow \bigvee_{g\in \mathsf{col}(\mathcal{G}_j, c_j)} \bigvee_{f\in\mathcal{F}_j} \underbrace{\left[\bigwedge_{S\in\mathcal{E}_j} R_S(\boldsymbol{X}_S) \wedge \bigwedge_{i\in U_j} f(X_i) = g(i)\right]}_{Q_j^{(g,f)}(\boldsymbol{X}_F)} \tag{22}$$

In particular, we will show that the data complexity of any conjunctive query with negation of the form (1) is the same as for its positive subquery $Q(\boldsymbol{X}_F) \leftarrow \mathsf{body}$.

The subsequent development in this section uses the InsideOut algorithm and the FAQ framework (see Section 2.2 and [2]). For each $j \in [B]$, we distinguish two multi-hypergraphs for the query $Q_j(\boldsymbol{X}_F)$: $\mathcal{H}_j = (V_j, \mathcal{E}_j)$ and associated relations $(R_S)_{S\in\mathcal{E}_j}$ for $\bigwedge_{S\in\mathcal{E}_j} R_S(\boldsymbol{X}_S)$;

and $\mathcal{G}_j = (U_j, \mathcal{A}_j)$ for $\bigwedge_{i \in U_j} f(X_i) = g(i)$, where $U_j \subseteq V_j$. For the rest of this section, we will fix some $j \in [B]$ and drop the subscript j for brevity. In particular, we will use $\mathcal{H} = (V, \mathcal{E}), \mathcal{G} = (U, \mathcal{A}), \mathcal{F}$ to denote $\mathcal{H}_j = (V_j, \mathcal{E}_j), \mathcal{G}_j = (U_j, \mathcal{A}_j), \mathcal{F}_j$ respectively.

A better semiring for shaving off a $\log N$ factor

Let $r = P(\mathcal{G}, c) \cdot |\mathcal{F}|$ denote the Boolean tensor rank in the decomposition (16). If we were only interested in bounding the rank, we can use the bound on $|\mathcal{F}|$ from Part (a) of Theorem 5.6. However, for the purpose of using the Boolean tensor decomposition in an algorithm, we have to be able to explicitly and efficiently construct the family $\mathcal{F}$ of functions. We thus need to use the bound on $|\mathcal{F}|$ from Part (b) of Theorem 5.6. To facilitate the explanations below, define $w = |\mathcal{F}|/\log N$ so that the Boolean rank is decomposed into $r = P(\mathcal{G}, c) \cdot w \cdot \log N$; that is, $w = \frac{|U|^3 \cdot \log |U|}{\theta(\boldsymbol{p})}$ from Part (b) of Theorem 5.6.

By Theorem 2.6, we can answer query (22) by running r instantiations of InsideOut, each of which computes $Q_j^{(g,f)}$ for some fixed pair (g, f), and then take the disjunction of $Q_j^{(g,f)}$ over g and f. The runtime is

$$O(P(\mathcal{G}, c) \cdot w \cdot (|\mathcal{E}| + |U|) \cdot |V|^2 \cdot (\log N)^2 \cdot (N^{\mathsf{fhtw}_F(\mathcal{H})} + |\mathsf{output}|)). \tag{23}$$

The atoms $f(X_i) = g(i)$ are singleton factors, i.e., factors on one variable, and thus do not increase the fractional hypertree width or the submodular width of the query.

These r instantiations of InsideOut are run on sum-product instances over the Boolean semiring. We can however reformulate the problem as sum-product over a different semiring, which helps reduce the runtime. The new semiring $(\boldsymbol{D}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ is defined as follows. The domain $\boldsymbol{D}$ is set to $\boldsymbol{D} = \{0,1\}^r$, the collection of all r-bit vectors. The "addition" and "multiplication" operators $\oplus$ and $\otimes$ are bit-wise max and min (essentially, bit-wise $\vee$ and $\wedge$). The additive identity is $\mathbf{0} = \mathbf{0}_r$, the r-bit all-0 vector. The multiplicative identity is $\mathbf{1} = \mathbf{1}_r$, the r-bit all-1 vector. To each input relation R_S, we associate a function $\psi_S(\boldsymbol{X}_S) : \prod_{i \in S} \mathsf{Dom}(X_i) \to \boldsymbol{D}$, where $\psi_S(\boldsymbol{x}_S) = \mathbf{1}$ if $\boldsymbol{x}_S \in R_S$ and $\mathbf{0}$ otherwise. Also, define $|U|$ extra singleton factors $\overline{\psi}_i : \mathsf{Dom}(X_i) \to \boldsymbol{D}$ $(\forall i \in U)$, where

$$\overline{\psi}_i(x_i) = (b_{g,f})_{g \in \mathsf{col}(\mathcal{G},c), f \in \mathcal{F}}, \quad \text{where} \quad b_{g,f} = \begin{cases} 1 & \text{if } f(x_i) = g(i) \\ 0 & \text{if } f(x_i) \neq g(i). \end{cases} \tag{24}$$

▶ **Proposition 6.1.** *The query* (22) *is equivalent to the following* SumProd *expression*

$$\varphi(\boldsymbol{x}_F) = \bigoplus_{x_{|F|+1}} \cdots \bigoplus_{x_{|V|}} \bigotimes_{S \in \mathcal{E}} \psi_S(\boldsymbol{x}_S) \otimes \bigotimes_{i \in U} \overline{\psi}_i(x_i). \tag{25}$$

The runtime of InsideOut *for the expression* $\varphi(\boldsymbol{x}_F)$ *is*

$$O(P(\mathcal{G}, c) \cdot w \cdot (|\mathcal{E}| + |U|) \cdot |V|^2 \cdot \log N \cdot (N^{\mathsf{fhtw}_F(\mathcal{H})} + |\mathsf{output}|)). \tag{26}$$

Proof. For any $\boldsymbol{x}_F$, we have $Q(\boldsymbol{x}_F) = \mathsf{true}$ iff $\varphi(\boldsymbol{x}_F) \neq \mathbf{0}$. This is because for each $\boldsymbol{x}_F$, the value $\varphi(\boldsymbol{x}_F) \in \mathbf{D}$ is an r-bit vector where each bit represents the answer to $Q_j^{(g,f)}(\boldsymbol{X}_F)$ for some pair (g, f) (There are exactly $r = P(\mathcal{G}, c) \cdot |\mathcal{F}|$ such pairs).

The runtime of InsideOut follows from the observation that each operation $\oplus$ or $\otimes$ can be done in $O(r/\log N)$-time, because those are bit-wise $\vee$ or $\wedge$ and the r-bit vector can be stored in $O(r/\log N)$ words in memory. Bit-wise $\vee$ and $\wedge$ of two words are done in $O(1)$-time. ◀

We can further lower the data complexity of our approach using PANDA (See Section 2.2 and [3]). By Theorem 2.7, the complexity from (26) becomes:

$$O(P(\mathcal{G}, c) \cdot w \cdot |V| \cdot 2^{2^{|V|}} \cdot (\mathsf{poly}(\log N) \cdot N^{\mathsf{subw}_F(\mathcal{H})} + \log N \cdot |\mathsf{output}|)). \tag{27}$$

The data complexity for conjunctive queries with negation

We are now ready to prove Theorem 1.1. From Proposition 4.3, we untangle Q into a disjunction of B different queries Q_j for $j \in [B]$ of the form (13) where $B \leq \prod_{S \in \overline{\mathcal{E}}} (|S|)^{|S|(d_S-1)+1} = O(1)$ in data complexity. From (16), each of these queries is equivalent to query (22). For a fixed $g \in \mathsf{col}(\mathcal{G}, c)$ and $f \in \mathcal{F}$, the inner conjunction $Q_j^{(g,f)}(\boldsymbol{X}_F)$ in (22) has at most the widths fhtw_F and subw_F of body in the original query (1). Query (22) can be solved in time (26) using InsideOut or (27) using PANDA.

7 Related Work

Color-coding. The color-coding technique [8] underlies existing approaches to answering queries with disequalities [27, 9, 21], the homomorphic embedding problem [14], and motif finding and counting in computational biology [4]. This technique has been originally proposed for checking cliques of inequalities. It is typically used in conjunction with a dynamic programming algorithm, whose analysis involves combinatorial arguments that make it difficult to apply and generalize to problems beyond the path query from Eq. (2). For example, it is unclear how to use color coding to recover the Plehn and Voigt result [30] for the induced path query from Eq. (3). In this paper, we generalize the technique to arbitrary conjunctions of NAE predicates and from graph coloring to hypergraph coloring.

Queries with disequalities. Our work also generalizes prior work on answering queries with disequalities, which are a special case of queries with negated relations of bounded degree.

Papadimitriou and Yannakakis [27] showed that any acyclic join query Q with an arbitrary set of disequalities on k variables can be evaluated in time $2^{O(k \log k)} \cdot |D| \cdot |Q(D)| \cdot \log^2 |D|$ over any database D. This builds on, yet uses more colors than the color-coding technique.

Bagan et al [9] extended this result to free-connex acyclic queries; they also shaved off a $\log |D|$ factor by using a RAM model of computation differently from ours, where sorting can be done in linear time.

Koutris et al [21] introduced a practical algorithm for conjunctive queries with disequalities: Given a select-project-join (SPJ) plan for the conjunctive query without disequalities, the disequalities can be solved uniformly using an extended projection operator. The reliance on SPJ plans is a limitation, since it is known that such plans are suboptimal for join processing [25] and are inadequate to achieve the fhtw and subw complexity bounds. Our approach uses the InsideOut [2] and PANDA [3] query evaluation algorithms and inherits their low data complexity, thus achieving both bounds as stated in Theorem 1.1.

Differently from prior work and in line with our work, Koutris et al [21] also investigated query structures for which the combined complexity becomes polynomial: This is the case for queries whose augmented hypergraphs have bounded treewidth (an augmented hypergraph is the hypergraph of the skeleton conjunctive query extended with one hyperedge per disequality). Koutris et al [21] further proposed an alternative query answering approach that uses the probabilistic construction of the original color-coding technique coupled with any query evaluation algorithm. When restricted to queries with disequalities, our query complexity analysis is more refined than [21] as the number of colors used in our generalization of color-coding is sensitive to the query structure.

Tensor decomposition. Our Boolean tensor decomposition for conjunctions of NAE predicates draws on the general framework of tensor decomposition used in signal processing and machine learning [19, 33]. It is a special case of sum-product decomposition and a powerful

tool. Typical dynamic programming algorithms solve subproblems by *combining* relations and *eliminating* variables [34, 26, 2]. The sum-product decomposition is the dual approach that *decomposes* a formula and *introduces* new variables. The PANDA algorithm [3] achieves a generalization of the submodular width by rewriting a conjunction as a sum-product over tree decompositions. By combining PANDA with our Boolean tensor decomposition, we can answer queries with negation in time defined by the submodular width.

While close in spirit to k-restrictions [6], our approach to derandomization of the construction of the Boolean tensor decomposition is different since we would like to execute it in time defined by the fhtw-bound for computing body. Our derandomization uses a code-concatenation technique where the outer-code is a linear error-correcting code on the Gilbert-Varshamov boundary [31] that can be constructed in linear time. As a byproduct, the code enables an efficient construction of an (N, k^2, k)-perfect hash family of size $O(k^2 \log N)$. To the best of our knowledge, the prior constructions yield families of size $O(k^4 \log N)$ [6].

Data sparsity. We connect two notions of sparsity in this work. One is the bounded degree of the input relations that are negated in the query. There are notions of sparsity beyond bounded degree, cf. [28] for an excellent and comprehensive course on sparsity. The most refined sparsity notion is that of *nowhere denseness* [16], which characterizes the *input monotone* graph classes on which FO model checking is fixed-parameter tractable. We leave as future work the generalization of our work to queries with negated nowhere-dense relations.

The second notion of sparsity used in this work is given by the Boolean tensor rank of the Boolean tensor decomposition of the conjunction of NAE predicates. We note that the relation represented by such a conjunction is not necessarily nowhere dense.

8 Concluding remarks

In this paper, we studied the complexity of answering conjunctive queries with negation on relations of bounded degree. We give an approach that matches the data complexity of the best known query evaluation algorithms InsideOut [2] and PANDA [3].

An intriguing venue of future research is to further lower the query complexity of our approach. Proposition 5.8 presented lower bounds on θ^* that are dependent on the structure of the multi-hypergraph $\mathcal{G}$ of the input query. It is an intriguing open problem to give a lower bound on θ^* that is dependent on some known parameter of $\mathcal{G}$. The extended version of this paper [1] discusses two further ideas on how to reduce the query complexity:

- Cast coloring as a join of "coloring predicates" and apply the InsideOut algorithm on the resulting query with the coloring predicates taken into account; and
- Exploit symmetry to answer the k-path query in time $2^{O(k)} N \log N$ [8] instead of $O(k^k N \log N)$.

Our approach extends immediately to unions of conjunctive queries with negated relations and of degree bounds on the positive relations. In the latter case, we can achieve a runtime depending on the *degree-aware* version of the submodular width [3].

We finally note that our Boolean tensor decomposition technique cannot be generalized to more powerful semirings such as the sum-product semiring over the reals due to an intrinsic computational difficulty: The counting version of the (induced) k-path query from Section 1 is #W[1]-hard [11, 14].

References

1 Mahmoud Abo Khamis, Hung Q. Ngo, Dan Olteanu, and Dan Suciu. Boolean Tensor Decomposition for Conjunctive Queries with Negation. *CoRR*, abs/1712.07445, 2017. `arXiv:1712.07445`.

2 Mahmoud Abo Khamis, Hung Q. Ngo, and Atri Rudra. FAQ: Questions Asked Frequently. In *PODS*, pages 13–28, 2016.

3 Mahmoud Abo Khamis, Hung Q. Ngo, and Dan Suciu. What Do Shannon-type Inequalities, Submodular Width, and Disjunctive Datalog Have to Do with One Another? In *PODS*, pages 429–444, 2017.

4 Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Trans. Information Theory*, 38(2):509–516, 1992.

5 Noga Alon and Jeong Han Kim. On the degree, size, and chromatic index of a uniform hypergraph. *J. Combin. Theory Ser. A*, 77(1):165–170, 1997.

6 Noga Alon, Dana Moshkovitz, and Shmuel Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2(2):153–177, 2006.

7 Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley, 1992.

8 Noga Alon, Raphael Yuster, and Uri Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995.

9 Guillaume Bagan, Arnaud Durand, and Etienne Grandjean. On Acyclic Conjunctive Queries and Constant Delay Enumeration. In *CSL*, pages 208–222, 2007.

10 Yijia Chen and Jörg Flum. On Parameterized Path and Chordless Path Problems. In *CCC*, pages 250–263, 2007.

11 Yijia Chen, Marc Thurley, and Mark Weyer. Understanding the Complexity of Induced Subgraph Isomorphisms. In *ICALP*, pages 587–596, 2008.

12 Ding-Zhu Du and Frank K. Hwang. *Combinatorial group testing and its applications*, volume 12 of *Series on Applied Mathematics*. World Scientific Publishing Co. Inc., second edition, 2000.

13 V. Faber. Linear Hypergraph Edge Coloring. *ArXiv e-prints*, 2016. `arXiv:1603.04938`.

14 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

15 Georg Gottlob, Gianluigi Greco, Nicola Leone, and Francesco Scarcello. Hypertree Decompositions: Questions and Answers. In *PODS*, pages 57–74, 2016.

16 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding First-Order Properties of Nowhere Dense Graphs. *J. ACM*, 64(3):17:1–17:32, 2017.

17 Martin Grohe and Dániel Marx. Constraint Solving via Fractional Edge Covers. *ACM Trans. Alg.*, 11(1):4, 2014.

18 Tommy R. Jensen and Bjarne Toft. *Graph coloring problems*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1995. A Wiley-Interscience Publication.

19 Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Rev.*, 51(3):455–500, 2009.

20 D. König. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Math. Ann.*, 77:453–465, 1916.

21 Paraschos Koutris, Tova Milo, Sudeepa Roy, and Dan Suciu. Answering Conjunctive Queries with Inequalities. *Theory Comput. Syst.*, 61(1):2–30, 2017.

22 Valentas Kurauskas and Katarzyna Rybarczyk. On the chromatic index of random uniform hypergraphs. *SIAM J. Discrete Math.*, 29(1):541–558, 2015.

23 Dániel Marx. Tractable Hypergraph Properties for Constraint Satisfaction and Conjunctive Queries. *J. ACM*, 60(6):42:1–42:51, November 2013. `doi:10.1145/2535926`.

24 Hung Q. Ngo and Ding-Zhu Du. A Survey on Combinatorial Group Testing Algorithms with Applications to DNA Library Screening. In *DIMACS*, volume 55, pages 171–182. Amer. Math. Soc., 2000.

25 Hung Q. Ngo, Christopher Ré, and Atri Rudra. Skew Strikes Back: New Developments in the Theory of Join Algorithms. In *SIGMOD Rec.*, pages 5–16, 2013.

26 Dan Olteanu and Jakub Závodnỳ. Size Bounds for Factorised Representations of Query Results. *TODS*, 40(1):2:1–2:44, 2015.

27 Christos H. Papadimitriou and Mihalis Yannakakis. On the Complexity of Database Queries. *J. Comput. Syst. Sci.*, 58(3):407–427, 1999.

28 Michał Pilipczuk and Sebastian Siebertz. Sparsity. Technical report, University of Warsaw, December 2017. URL: `https://www.mimuw.edu.pl/~mp248287/sparsity/`.

29 Nicholas Pippenger and Joel Spencer. Asymptotic behavior of the chromatic index for hypergraphs. *J. Combin. Theory Ser. A*, 51(1):24–42, 1989.

30 Jürgen Plehn and Bernd Voigt. Finding Minimally Weighted Subgraphs. In *Graph-Theoretic Concepts in Computer Science*, pages 18–29, 1990.

31 Ely Porat and Amir Rothschild. Explicit Nonadaptive Combinatorial Group Testing Schemes. *IEEE Trans. Information Theory*, 57(12):7982–7989, 2011.

32 Luc Segoufin. Enumerating with Constant Delay the Answers to a Query. In *ICDT*, pages 10–20, 2013.

33 Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor Decomposition for Signal Processing and Machine Learning. *Trans. Sig. Proc.*, 65(13):3551–3582, 2017.

34 Mihalis Yannakakis. Algorithms for Acyclic Database Schemes. In *VLDB*, pages 82–94, 1981.