Entropy bounds for conjunctive queries with functional dependencies

Tomasz Gogacz
School of Informatics
University of Edinburgh
Edinburgh, UK
tgogacz@inf.ed.ac.uk

Szymon Toruńczyk
Institute of Computer Science
University of Warsaw
Poland
szymtor@mimuw.edu.pl

ABSTRACT

We study the problem of finding the worst-case bound for the size of the result $Q(\mathbb{D})$ of a fixed conjunctive query Q applied to a database $\mathbb{D}$ satisfying given functional dependencies. We provide a precise characterization of this bound in terms of entropy vectors, and in terms of finite groups. In particular, we show that an upper bound provided by Gottlob, Lee, Valiant and Valiant [GLVV12] is tight, answering a question from their paper. Our result generalizes the bound due to Atserias, Grohe and Marx [AGM13], who consider the case without functional dependencies. Our result shows that the problem of computing the worst-case size bound, in the general case, is closely related to difficult problems from information theory.

Categories and Subject Descriptors

H.2.3 [**Information Systems**]: DATABASE MANAGEMENT,Query languages

Keywords

Entropy, Query size, Conjunctive queries, Size bounds, Entropy cone, finite groups

1. INTRODUCTION

Given a natural join query Q we would like to determine a bound $\alpha \in \mathbb{R}$ such that for every database $\mathbb{D}$, the inequality

$$|Q(\mathbb{D})| \ \leq \ c \cdot |\mathbb{D}|^{\alpha} \qquad (1)$$

holds, for some multiplicative factor c depending on Q. Here, $|\mathbb{D}|$ denotes the size of the largest table in the database $\mathbb{D}$, and $|Q(\mathbb{D})|$ denotes the size of the result of the query applied to $\mathbb{D}$. Above, we consider all databases $\mathbb{D}$ over a fixed schema, containing the relation names which appear in Q. In the general problem, which is the main focus of this paper, we may additionally impose some functional dependencies, and require that $\mathbb{D}$ satisfies them.

Obviously, in (1) we can always take $\alpha = |Q|$, the number of relation names appearing in the query Q. Define $\alpha(Q)$ as the infimum of all values α for which there exists a multiplicative factor c so that (1) holds for all databases $\mathbb{D}$. For example, if $Q_1(x, y, z) = R(x, y) \wedge S(y, z)$ then it is not difficult to see that $\alpha(Q_1) = 2$.

Indeed, $|Q_1(\mathbb{D})| \leq |R(\mathbb{D})| \cdot |S(\mathbb{D})| \leq |\mathbb{D}|^2$, and conversely, one can construct a database $\mathbb{D}$ with

$$R(\mathbb{D}) = X \times Y,$$
$$S(\mathbb{D}) = Y \times Z,$$

for some finite sets X, Z of arbitrarily large size N and Y of size 1, and then $|Q_1(\mathbb{D})| = |X \times Y \times Z| = N^2$, whereas $|R(\mathbb{D})| = |S(\mathbb{D})| = N$. Now consider $Q_2(x, y, z) = R(x, y) \wedge S(y, z) \wedge T(z, x)$. The trivial bound gives $\alpha(Q_2) \leq 3$. However, since $Q_2(\mathbb{D}) \subseteq Q_1(\mathbb{D})$ for every $\mathbb{D}$, it follows immediately that $\alpha(Q_2) \leq 2$. With some effort, one can show that in the absence of functional dependencies, $\alpha(Q_2) = 3/2$. This is a consequence of a more general result due to Atserias, Grohe and Marx [AGM13], which we recall now.

Consider the hypergraph whose vertices are the variables appearing in Q, and for each relation name R in Q there is a hyperedge containing those variables which appear in R. A *fractional edge covering* of a hypergraph assigns a positive rational number to each of its hyperedges, so that for every vertex, the numbers assigned to the adjacent hyperedges sum up to at least 1. The total weight of a fractional edge covering is the sum of the numbers assigned to all the hyperedges. Define $\mathrm{AGM}(Q)$ as the least possible total weight of a fractional vertex covering of the hypergraph associated to Q. The AGM bound then states that $\mathrm{AGM}(Q) = \alpha(Q)$, in the absence of functional dependencies. For example, the hypergraph obtained from the query Q_2 is the triangle, and assigning $1/2$ to each edge gives a fractional edge covering with total weight $3/2$, which is the least possible. Hence, $\alpha(Q_2) = \mathrm{AGM}(Q_2) = 3/2$.

Fractional edge coverings of a hypergraph can be computed efficiently using linear programming. So the problem of finding $\alpha(Q)$ is solved, when there are no functional dependencies. However, if we consider only those databases which satisfy a given set of functional dependencies, the problem of computing $\alpha(Q)$ remains unsolved. (To see that the value of $\alpha(Q)$ may change in the presence of functional dependencies, observe that $\alpha(Q_1) = 1$ assuming that y is a key in R and in S.)

In this paper, we make progress towards characterizing the value $\alpha(Q)$ in the presence of functional dependencies. In particular, we show that $\alpha(Q)$ can be characterized in two ways: as an entropy bound $H(Q)$,

and in terms of a number $GC(Q)$ derived from systems of finite groups. The bound $\alpha(Q) \le H(Q)$ was observed in [GLVV12], and it was left as an open problem whether equality holds for all queries Q, in the presence of functional dependencies. We answer this question affirmatively, by providing a matching lower bound based on a construction using finite groups. However, we do not know how to effectively compute the bound $\alpha(Q)$. Moreover, our results demonstrate that this problem is closely connected to notorious problems from information theory.

Finally, we discuss how to treat general conjunctive queries (under the usual semantics and under the bag semantics), and show some preliminary results concerning the computation of the result $Q(\mathbb{D})$.

2. MAIN RESULT AND CONSEQUENCES

The main result of this paper, Theorem 1, expresses $\alpha(Q)$ in terms of entropy. Below we also state a more general result, Theorem 3. We start from recalling the notion of the entropy cone from information theory below; for precise definitions and notation conventions, see Section 3. After that, we formulate the main result of this paper. Next, we show how some known results follow from this result, or its generalization. In Theorem 2 we show that in order to achieve the worst-case size increase $\alpha(Q)$, it is enough to consider databases of a specific, symmetric shape.

Entropy cone.

For a random variable V, let $H(V)$ denote its entropy. All random variables in this paper assume finitely many values. All logarithms are in base 2.

Fix a finite set X. Let $U = (U_x)_{x \in X}$ be a family of random variables indexed by X. For $Y \subseteq X$, let $U[Y]$ denote the joint random variable $(U_y)_{y \in Y}$. The variable $U[Y]$ can be seen as a random variable taking as values tuples indexed by Y. Consider the real-valued vector $\text{ent}(U)$, indexed by subsets Y of X, such that $\text{ent}(U)_Y = H(U[Y])$ for $Y \subseteq X$. Vectors of the form $\text{ent}(U) \in \mathbb{R}^{P(X)}$, where U is a family of random variables indexed by X, are called *entropy vectors* (or *entropic vectors*) with ground set X [ZY98].

The set of all entropy vectors with ground set X forms a subset of $\mathbb{R}^{P(X)}$, denoted Γ_X^*. Its topological closure $\overline{\Gamma_X^*}$ is a convex cone. The sets Γ_X^* and $\overline{\Gamma_X^*}$ are well studied, however, to date, they lack effective descriptions when $|X| \ge 4$. It is known that the closed cone $\overline{\Gamma_X^*}$ is a polyhedron if $|X| \le 3$, and is not a polyhedron if $|X| > 3$ (i.e. it is not described by finitely many linear inequalities). Entropy vectors $v \in \Gamma_X^*$ (and hence, also all $v \in \overline{\Gamma_X^*}$) satisfy the submodularity property,

expressing *Shannon's inequality* for information:

$$v_{Y \cup Z} + v_{Y \cap Z} \le v_Y + v_Z \quad \text{for } Y, Z \subseteq X. \quad (2)$$

Main result.

For a relation name R in a schema Σ, by $\mathbf{V}(R)$ we denote the set of attributes of R. For $X \subseteq \mathbf{V}(R)$ and $x \in \mathbf{V}(R)$, we write $R : X \mapsto x$ to denote the functional dependency (fd) requiring that in R, the values of attributes X determine the value of the attribute x. The value $\alpha(Q)$ is defined as in the introduction, taking into account all databases $\mathbb{D}$ over the schema Σ which satisfy a given set of functional dependencies $\mathcal{F}$. The main result of this paper is the following.

THEOREM 1. *Fix a schema Σ and a set of functional dependencies $\mathcal{F}$. Let Q be a natural join query with variables X. Then $\alpha(Q)$ is equal to the maximal value of v_X, for v ranging over $\overline{\Gamma_X^*}$ and satisfying:*

$$\begin{cases} v_{\mathbf{V}(R)} \le 1 & \text{for } R \in Q, \\ v_{Z \cup \{z\}} = v_Z & \text{for every fd } R : Z \mapsto z \text{ in } \mathcal{F}. \end{cases}$$

We now show how some results known previously can be obtained as consequences of Theorem 1.

Relaxing the condition that $v \in \overline{\Gamma_X^*}$ to the condition that v is submodular gives the following upper bound on $\alpha(Q)$, which is equivalent to a bound in [GLVV12].

COROLLARY 1. *Let $s(Q)$ be the maximal value of v_X, for v ranging over $\mathbb{R}^{P(X)}$ and satisfying:*

$$\begin{cases} v_{Y \cup Z} + v_{Y \cap Z} \le v_Y + v_Z & \text{for } Y, Z \subseteq X, \\ v_{\mathbf{V}(R)} \le 1 & \text{for } R \in Q, \\ v_{Z \cup \{z\}} = v_Z & \text{for every fd } R : Z \mapsto z \text{ in } \mathcal{F}. \end{cases}$$

Then $\alpha(Q) \le s(Q)$.

Note that the bound $s(Q)$ can be computed by linear programming.

We show how the AGM bound can be deduced from Corollary 1. Assume that the set of functional dependencies is empty. It is easy to see that then in the above linear program describing $s(Q)$, the maximum is achieved for a vector v such that $v_Y = \sum_{x \in Y} v_{\{x\}}$ for $Y \subseteq X$. Therefore, we get the following.

COROLLARY 2. *In the absence of functional dependencies, $\alpha(Q) \le s(Q)$, where $s(Q)$ is the maximal value of $\sum_{x \in X} v_x$, for v ranging over $\mathbb{R}^X$ and satisfying*

$$\sum_{x \in \mathbf{V}(R)} v_x \le 1 \quad \text{for } R \in Q. \quad (3)$$

PROOF OF THE AGM BOUND. The linear program (3) corresponds to computing the *fractional vertex packing number*, and is dual to the linear program computing the fractional edge covering number. By strong duality, $s(Q) = \text{AGM}(Q)$, and therefore, $\alpha(Q) \le \text{AGM}(Q)$ by the above corollary. The converse inequality, $\alpha(Q) \ge \text{AGM}(Q)$, is the easier part of the AGM bound, and is shown by constructing a database from a given fractional vertex packing, as follows. Let $v = (p_x/q)_{x \in X}$

be a rational solution to the fractional vertex packing problem with common denominator $q \in \mathbb{N}$. For $Y \subseteq X$, let p_Y denote $\sum_{x \in Y} p_x$. In an optimal solution, we have $p_X/q = s(Q)$ and $\max_{R \in Q}(p_{\mathbf{V}(R)}/q) = 1$.

Choose an arbitrary integer $N > 1$, and for each $x \in X$, a set V_x with N^{p_x} elements. Construct a database $\mathbb{D}$ so that $R(\mathbb{D}) = \prod_{x \in \mathbf{V}(R)} V_x$. It is easy to see that $|R(\mathbb{D})| = N^{p_{\mathbf{V}(R)}}$ and $|Q(\mathbb{D})| = N^{p_X}$. In particular,

$$\frac{\log |Q(\mathbb{D})|}{\log |\mathbb{D}|} = \frac{p_X}{\max_{R \in Q} p_{\mathbf{V}(R)}} = s(Q).$$

Since N can be taken arbitrarily large, this proves $\alpha(Q) \geq s(Q)$. Together with Corollary 2, this shows that in the absence of functional dependencies, $s(Q) = \alpha(Q) = \mathrm{AGM}(Q)$. $\square$

Symmetric databases.

The above proof of the AGM bound shows that in the absence of functional dependencies, the databases $\mathbb{D}$ for which the size-increase $\frac{\log |Q(\mathbb{D})|}{\log |\mathbb{D}|}$ achieves the bound $\alpha(Q)$ are of a very simple, specific form: each table is a full Cartesian product. It follows from [GLVV12] that in the presence of functional dependencies, this is no longer the case: databases of this form, satisfying the given functional dependencies, are arbitrarily far from reaching the value of $\alpha(Q)$.

In this paper, we improve the construction of worst-case databases in the presence of functional dependencies, by constructing databases which are arbitrarily close to achieving the bound $\alpha(Q)$. Interestingly these databases have a very symmetric structure, and their construction uses finite groups.

For a finite group G and a set X, recall that a (left) action of G on X is a mapping $G \times X \to X$ denoted $(g, x) \mapsto g \cdot x$, such that $(g \cdot h) \cdot x = g \cdot (h \cdot x)$ for $g, h \in G$ and $x \in X$. We say that the action is *transitive* if for every $x, y \in X$ there is $g \in G$ such that $g \cdot x = y$. Transitive group actions are very special, and correspond (up to isomorphism) to subgroups of G, where a subgroup H defines the action of G on the coset space G/H of left cosets.

For a fixed group G, by a *G-symmetric* database we mean a database $\mathbb{D}$ together with an action of G on the set of all values appearing in all tables of $\mathbb{D}$, such that the componentwise action of G on each table $R(\mathbb{D})$ is transitive (the componentwise action is given by $(g \cdot r)[x] = g \cdot (r[x])$ for $g \in G$, $r \in R(\mathbb{D})$ and $x \in \mathbf{V}(R)$). A *symmetric* database is a database G which is G-symmetric for some finite group G. For example, in the proof of the AGM bound presented above, the constructed database $\mathbb{D}$ is G-symmetric, for $G = \prod_{x \in \mathbf{V}(Q)} S_N^{p_x}$, where S_N denotes the permutation group on N elements.

The second main result of this paper is the following.

THEOREM 2. *Fix a schema Σ, functional dependencies $\mathcal{F}$, and a natural join query Q. Then, for each $\varepsilon > 0$ there are arbitrarily large symmetric databases $\mathbb{D}$ with*

$$\frac{\log |Q(\mathbb{D})|}{\log |\mathbb{D}|} > \alpha(Q) - \varepsilon.$$

Symmetric databases are essential in our proof of the lower bound given in Theorem 1.

Simple statistics.

We show how Theorem 1 can be deduced from a more general result, which we now state. For a database $\mathbb{D}$ over the schema of Q, its *simple log-statistics* is the vector $\mathrm{sls}(\mathbb{D}) = (\log |R(\mathbb{D})|)_{R \in Q}$.

THEOREM 3. *Fix a schema Σ and a set of functional dependencies $\mathcal{F}$. Let Q be a natural join query with variables X, and let $s = (s_R)_{R \in Q}$ be a vector of non-negative real numbers. Let $\beta(Q, s)$ be the maximal value of v_X, for v ranging over $\overline{\Gamma_X^*}$ and satisfying:*

$$\begin{cases} v_{\mathbf{V}(R)} \leq s_R & \text{for } R \in Q, \\ v_{Z \cup \{z\}} = v_Z & \text{for every fd } R : Z \mapsto z \text{ in } \mathcal{F}. \end{cases}$$

Then,

$$\sup_{\mathbb{D}} \log |Q(\mathbb{D})| = \limsup_{\mathbb{D}} \log |Q(\mathbb{D})| = \beta(Q, s),$$

where $\mathbb{D}$ ranges over all finite databases satisfying the functional dependencies $\mathcal{F}$ and such that $\mathrm{sls}(\mathbb{D}) \leq s$ componentwise.

Theorem 1 is an immediate consequence of Theorem 3, obtained by setting $s_R = \log |\mathbb{D}|$ for each $R \in Q$, and using the fact that $\overline{\Gamma_X^*}$ is a cone. In this paper, we present in detail only the proof of Theorem 1. The proof of Theorem 3 proceeds similarly, and will be presented in the full version of the paper.

We remark that Theorem 3 can be used to derive the following, more precise variant of the AGM bound.

COROLLARY 3 ([AGM13]). *In the absence of functional dependencies, $|Q(\mathbb{D})| \leq \prod_{R \in Q} |R(\mathbb{D})|^{w_R}$, where $(w_R)_{R \in Q}$ is any solution to the fractional edge covering problem.*

Indeed, to deduce this, repeat the reasoning used above when deriving Corollary 2 from Theorem 1, by relaxing the condition $v \in \overline{\Gamma_X^*}$ in Theorem 3 to submodularity of v. We leave the details to the reader.

Geometric inequalities.

As noted elsewhere [NPRR12, Fri04, BKS13], Corollary 3 provides an upper bound on the size of a finite set in multi-dimensional space, in terms of the sizes of its projections. It implies the discrete versions of many inequalities from geometry and analysis, such as Hölder's inequality, Cauchy-Schwartz inequality, Loomis-Whitney inequality, Bollobás-Thomason inequality, Friedgut's inequality. As an illustration, we show how the Loomis-Whitney inequality can be derived from Corollary 3. For $1 \leq j \leq n$, let $\pi_j : \mathbb{R}^n \to \mathbb{R}^{n-1}$ denote the projection along the jth coordinate axis.

THEOREM 4 (LOOMIS-WHITNEY INEQUALITY [LW49]). Continuous variant: *Let $E \subseteq \mathbb{R}^d$ be a measurable set.*

Then, for λ_n denoting the n-dimensional Lebesgue measure,
$$\lambda_d(E) \leq \prod_{j=1}^{d} \lambda_{d-1}(\pi_j(E))^{1/(d-1)}.$$

Discrete variant: *Let $E \subseteq \mathbb{R}^d$ be a finite set. Then*

$$|E| \leq \prod_{j=1}^{d} |\pi_j(E)|^{1/(d-1)}.$$

The discrete variant of Theorem 4 can be seen as a consequence of Corollary 3, as follows. Consider a schema Σ with relations R_i, for $i = 1, \ldots, d$, and attributes x_i, for $i = 1, \ldots, d$, where $\mathbf{V}(R_i) = \{x_j : j \neq i, 1 \leq j \leq d\}$. For a finite set $E \subseteq \mathbb{R}^d$, where E is naturally viewed as a set of functions $r : \{x_1, \ldots, x_d\} \to \mathbb{R}$, define a database $\mathbb{D}$ over Σ with $R_i(\mathbb{D}) = \{r[\mathbf{V}(R_i)] : r \in E\}$, where for $X \subseteq \{x_1, \ldots, x_d\}$, $r[X] : X \to \mathbb{R}$ denotes the restriction of r to X. Clearly, $|R_i(\mathbb{D})| = |\pi_i(E)|$. Let Q be the natural join query consisting of all the R_i's. Then $E \subseteq Q(\mathbb{D})$, and $|Q(\mathbb{D})| \leq \prod_{j=1}^{d} |\pi_j(E)|^{w_i}$ by Corollary 3, where $(w_i)_{i=1}^{d}$ is any fractional edge covering of the hypergraph with vertices $x_1, \ldots, x_d$ and hyperedges which are complements of singletons. In particular, since every vertex belongs to exactly $d-1$ hyperedges, taking $w_i = 1/(d-1)$ yields a fractional edge covering, proving the discrete variant of Theorem 4. The continuous variant can be derived by approximation, in a standard way.

Theorem 3, in principle, can be used to formulate a stronger inequality than the discrete Loomis-Whitney inequality (or other geometric inequalities listed above), for sets $E \subseteq \mathbb{R}^n$ which satisfy given functional dependencies: we say that E satisfies a functional dependency $I \mapsto j$ (where $I \subseteq \{1, \ldots, n\}, j \in \{1, \ldots, n\}$) if for every $v, w \in E$ such that $v_i = w_i$ for all $i \in I$, it is the case that $v_j = w_j$.

COROLLARY 4. *Let $E \subseteq \mathbb{R}^d$ be a finite set, satisfying a given set of functional dependencies $\mathcal{F}$. Then $|E| \leq \beta$, where β is the maximal value of v_X, for $X = \{1, \ldots, d\}$ and v ranging over $\overline{\Gamma_X^*}$ and satisfying:*

$$\begin{cases} v_{X-\{i\}} \leq \log |\pi_j(E)| & \text{for } j \in X, \\ v_{I \cup \{j\}} = v_I & \text{for every fd } I \mapsto j \text{ in } \mathcal{F}. \end{cases}$$

Although the problem of computing the value β is not addressed in this paper, Theorem 2 (or rather, its more precise variant, corresponding to Theorem 3) provides a description of worst-case sets E.

Other results.
A generalization of Theorem 1 to conjunctive queries (with projections) is possible, and we discuss such a result in Section 7. Also, we describe a crude algorithm for evaluating $Q(\mathbb{D})$ in Section 8.

Outline of the paper.
After introducing notation, definitions, and preliminary observations in Section 3, we recall the entropy (upper) bound for $\alpha(Q)$ from [GLVV12] in Section 4. Then, in Section 5 we present several lower bounds for $\alpha(Q)$: we recall the coloring bound from [GLVV12], and later improve it to vector space colorings, and group systems. Finally, in Section 6, we show that the group system bound matches the entropy bound, using a construction from [Lun02]. This proves Theorem 1 and Theorem 2. In Section 7 we show how to generalize Theorem 1 to queries with projections, In Section 8, we describe some basic results concerning the worst-case complexity of computing $Q(\mathbb{D})$, for a fixed query Q and given database $\mathbb{D}$.

3. PRELIMINARIES

To fix notation, we recall some notions concerning databases, and entropy.

Notation.
We assume a fixed *schema* Σ, which specifies a finite set of *attributes* $\mathbf{V}(\Sigma)$, a finite set of *relation names*, and for each relation name R, a finite set $\mathbf{V}(R) \subseteq \mathbf{V}(\Sigma)$ of attributes of R. If X is a set of attributes, then a *row* with attributes X is a function r assigning to each $x \in X$ some value $r[x]$. If r is a row with attributes X and $Y \subseteq X$ then by $r[Y]$ we denote the restriction of r to Y. A *table* with attributes X is a finite set of rows with attributes X. A *database* $\mathbb{D}$ over Σ specifies for each relation name R in Σ a table with attributes $\mathbf{V}(R)$. A *natural join query* is a set Q of relation names in Σ; we denote $\mathbf{V}(Q) = \bigcup_{R \in Q} \mathbf{V}(R)$. Such a query can be applied to a database $\mathbb{D}$, yielding as result the table $Q(\mathbb{D})$ consisting of those rows r with attributes $\mathbf{V}(Q)$ such that $r[\mathbf{V}(R)] \in R(\mathbb{D})$ for every $R \in Q$.

We say that a database $\mathbb{D}$ *satisfies* a *functional dependency* $R : X \mapsto x$ – where X is a set of attributes and x is a single attribute – if for any two rows u, v of $R(\mathbb{D})$, $u[X] = v[X]$ implies $u[x] = v[x]$.

For the rest of this paper, fix a schema Σ and a set of functional dependencies $\mathcal{F}$. Every database $\mathbb{D}$ is assumed to be over this schema, and to satisfy $\mathcal{F}$. Define $\alpha(Q)$ as the smallest value α for which there exists a constant c such that (1) holds for all databases $\mathbb{D}$ over Σ which satisfy the functional dependencies in $\mathcal{F}$. For convenience, we define $|\mathbb{D}|$ to be the maximal size of a relation in $\mathbb{D}$. Since we allow a multiplicative constant in (1), defining $|\mathbb{D}|$ as the sum of the sizes of the relations in $\mathbb{D}$ would give an equivalent definition of $\alpha(Q)$.

Remark 1. Observe that since Q is a natural join query and in the definition of $\alpha(Q)$ we are interested in maximizing $|Q(\mathbb{D})|$ while keeping $|\mathbb{D}|$ bounded, we may assume that if $R : X \mapsto x$ is a functional dependency in $\mathcal{F}$, then also $S : X \mapsto x$ is a functional dependency in $\mathcal{F}$, for every relation name S such that $X \subseteq \mathbf{V}(S)$. Therefore, we may simply write that $\mathcal{F}$ contains the functional dependency $X \mapsto x$ instead of writing $R : X \mapsto x$.

For a database $\mathbb{D}$ (over Σ, satisfying $\mathcal{F}$), denote

$$\alpha(Q, \mathbb{D}) = \frac{\log |Q(\mathbb{D})|}{\log |\mathbb{D}|}. \tag{α}$$

Convention.

Throughout this paper we will define several real-valued parameters of the form $\gamma(Q, x)$, where Q is a query and x is some object. For a fixed query Q, we denote by $\sup_x \gamma(Q, x)$ the supremum, and by $\limsup_x \gamma(Q, x)$ the limit superior over all values x, for which the value $\gamma(Q, x)$ is defined. In particular, $\limsup_x \gamma(Q, x)$ is the smallest value in $\mathbb{R} \cup \{-\infty, +\infty\}$ such that for every real $\varepsilon > 0$, there are only finitely many x's such that $\gamma(Q, x)$ is defined and larger than $\gamma(Q) + \varepsilon$.

Limit superior vs. supremum.

The following simple lemma will simplify several formulations and proofs throughout this paper.

LEMMA 1. *For a natural join query Q,*

$$\alpha(Q) = \limsup_{\mathbb{D}} \alpha(Q, \mathbb{D}) = \sup_{\mathbb{D}} \alpha(Q, \mathbb{D}).$$

PROOF. Let $d = \limsup_{\mathbb{D}} \alpha(Q, \mathbb{D})$. First we show that $d = \alpha(Q)$. By definition, for every $\varepsilon > 0$ there are finitely many databases $\mathbb{D}$ for which

$$\frac{\log |Q(\mathbb{D})|}{\log |\mathbb{D}|} > d + \varepsilon,$$

so $|Q(\mathbb{D})| \le |\mathbb{D}|^{d+\varepsilon}$ for almost all $\mathbb{D}$. By choosing a large enough constant c, we have that

$$|Q(\mathbb{D})| \le c \cdot |\mathbb{D}|^{d+\varepsilon}$$

for all databases $\mathbb{D}$. Hence, $\alpha(Q) \le d + \varepsilon$, for every $\varepsilon > 0$, proving $\alpha(Q) \le d$. The inequality $d \le \alpha(Q)$ is proved similarly.

To show that $\sup_{\mathbb{D}} \alpha(Q, \mathbb{D}) \le \limsup_{\mathbb{D}} \alpha(Q, \mathbb{D})$, we use the following construction. For a database $\mathbb{D}$ and a natural number n, let $\mathbb{D}^n$ be the database defined so that the rows of $R(\mathbb{D}^n)$ are n-tuples of rows of $R(\mathbb{D})$, and for such a row $r = (r_1, \dots, r_n)$, we define $r[x] = (r_1[x], \dots, r_n[x])$ for an attribute $x \in \mathbf{V}(R)$. It is easy to check that $\mathbb{D}^n$ satisfies the same functional dependencies as $\mathbb{D}$, and that $|\mathbb{D}^n| = |\mathbb{D}|^n$ and $|Q(\mathbb{D})| = |Q(\mathbb{D})|^n$. In particular, $\alpha(Q, \mathbb{D}^n) = \alpha(Q, \mathbb{D})$. It follows that if $|\mathbb{D}| > 1$, then by choosing n arbitrarily large, we have arbitrarily large databases $\mathbb{D}^n$ with $\alpha(Q, \mathbb{D}^n) = \alpha(Q, \mathbb{D})$. Therefore, $\limsup_{\mathbb{D}} \alpha(Q, \mathbb{D}) \ge \sup_{\mathbb{D}} \alpha(Q, \mathbb{D})$, the other inequality being obvious. $\square$

Entropy.

In this paper, we only consider random variables taking finitely many values. Formally, a random variable X is a measurable function $X : \Omega \to V$ from a fixed probability space $(\Omega, \mathbb{P})$ of events to a finite set V. In this paper, however, it is not harmful to assume that Ω is a finite probability space, in which case every function $X : \Omega \to V$ is a random variable. By $\mathrm{Im}(X) \subseteq V$ we denote the set of values v such that $\mathbb{P}[X = v] > 0$, where $\mathbb{P}[X = v]$ is a shorthand for $\mathbb{P}[\{\omega \in \Omega : X(\omega) = v\}]$.

For a random variable X taking values in a finite set V, define the *entropy* of X as

$$H(X) = - \sum_{v \in \mathrm{Im}(X)} p_v \log p_v,$$

where $p_v = \mathbb{P}[X = v]$. Clearly, the entropy of X only depends on the distribution of X. Also, the maximal possible entropy of a random variable with values in a finite set V is equal to $\log |V|$, and is attained by the uniform distribution on V, as follows from Jensen's inequality applied to the convex function $-\log(x)$.

According to Shannon's source coding theorem, the value $H(X)$ has the following characterization (up to an additive error of 1): how many questions on average does one need to ask to determine the value of a random variable X, where each question is a question of the form "does X belong to U?" (where $U \subseteq \mathrm{Im}(X)$)? Here we mean the minimum, over all strategies against a given distribution of X, of the average number of questions.

4. UPPER BOUND

We start with presenting an upper bound on $\alpha(Q)$, which we call the *entropy bound*. This bound is essentially from [GLVV12].

The entropy bound.

Fix a natural join query Q. Let U be a random variable U taking as values rows with attributes $\mathbf{V}(\Sigma)$. For a set of attributes $X \subseteq \mathbf{V}(\Sigma)$, define $U[X]$ to be the random variable whose value is the restriction of the value of U to the set of attributes X. In particular, $U[X]$ is a random variable whose values are rows with attributes X, and $\mathrm{Im}(U[\mathbf{V}(R)])$ is a table with attributes $\mathbf{V}(R)$. We say that U satisfies a functional dependency $Y \mapsto x$ if the table $\mathrm{Im}(U)$ satisfies the functional dependency $Y \mapsto x$.

LEMMA 2. *The random variable U satisfies a functional dependency $Y \mapsto x$ if and only if*

$$\mathrm{ent}(U)_Y = \mathrm{ent}(U)_{Y \cup \{x\}}. \tag{4}$$

We remark that in information theory, (4) can be expressed using conditional entropy as $H(U[x] \mid U[Y]) = 0$.

PROOF. By unraveling of definitions. Obviously (4) holds if and only if $U[Y]$ and $U[Y \cup \{x\}]$ have the same distributions. This however means that there exist no elements $r, c \ne d$ such that $\mathbb{P}[U[Y] = r, U[x] = c] > 0$ and $\mathbb{P}[U[Y] = r, U[x] = d] > 0$. This is the definition of functional dependency $Y \mapsto x$ for random variables. $\square$

For a random variable U which satisfies every functional dependency in $\mathcal{F}$ we define

$$H(Q, U) = \frac{H(U[\mathbf{V}(Q)])}{\max_{R \in \Sigma} H(U[\mathbf{V}(R)])}, \tag{H}$$

and let $H(Q) = \sup_U H(Q, U)$.

Observe that a random variable U taking as values rows with attributes $X = \mathbf{V}(Q)$ is the same thing as a tuple $(U_x)_{x \in X}$ of random variables. With this observation, Lemma 2 allows us to characterize $H(Q)$ in terms of the entropy cone $\overline{\Gamma_X^*}$.

PROPOSITION 1. *The value $H(Q)$ is equal to the value described by the optimization problem from Theorem 1.*

PROOF. By definition of the entropy cone $\overline{\Gamma_X^*}$. □

Therefore, to prove Theorem 1, it remains to show that $\alpha(Q) = H(Q)$. Lemma 3 below shows one of the two inequalities, by employing an observation from [GLVV12],

LEMMA 3. *Let Q be a natural join query. Then*

$$H(Q) \geq \alpha(Q).$$

PROOF. For a database $\mathbb{D}$, define a random variable denoted $U_{\mathbb{D}}$ which chooses uniformly at random a row of $Q(\mathbb{D})$. By definition of a natural join query, the values of $U_{\mathbb{D}}[R]$ are rows of $R(\mathbb{D})$. Then $H(Q, U_{\mathbb{D}}) \geq \alpha(Q, \mathbb{D})$ for every database $\mathbb{D}$, since $H(U_{\mathbb{D}}) = \log|Q(\mathbb{D})|$ and $H(U_{\mathbb{D}}[R]) \leq \log|R(\mathbb{D})|$ by the fact that the uniform distribution maximizes entropy. This, together with Lemma 1, proves Lemma 3. □

In Sections 5 and 6 we prove the remaining inequality $H(Q) \leq \alpha(Q)$, thus finishing the proof of Theorem 1.

5. LOWER BOUNDS

The paper [GLVV12] also provides a lower bound for $\alpha(Q)$, using colorings, which we recall below for completeness. We then improve this bound to vector space colorings, and finally, to group systems. In Section 5, fix a natural join query Q over a schema Σ, and a set of functional dependencies $\mathcal{F}$.

5.1 Colorings

A *coloring* of Q is a function f assigning finite sets to $\mathbf{V}(Q)$. We say that f satisfies a functional dependency $X \mapsto x$ if $f(x) \subseteq f(X)$, where $f(X)$ denotes $\bigcup_{y \in X} f(y)$. For a coloring f of Q which satisfies all functional dependencies in $\mathcal{F}$, define

$$C(Q, f) = \frac{|f(\mathbf{V}(Q))|}{\max_{R \in \Sigma} |f(\mathbf{V}(R))|}, \qquad \text{(C)}$$

and let $C(Q) = \sup_f C(Q, f)$.

LEMMA 4 ([GLVV12]). *Let Q be a natural join query. Then $\alpha(Q) \geq C(Q)$.*

PROOF. Consider a coloring f of Q satisfying the functional dependencies $\mathcal{F}$. We construct a database $\mathbb{D}$ with $\alpha(Q, \mathbb{D}) \geq C(Q, f)$. Let $C = f(\mathbf{V}(Q))$ be the set of all colors used by f.

Choose a set N with $|N| > 1$, and consider the table $T = N^C$ with attributes C. Define the database $\mathbb{D}$ so that for each relation name R,

$$R(\mathbb{D}) = \{r[f(\mathbf{V}(R))] : r \in T\}.$$

Then it is not difficult to check that:

- The database $\mathbb{D}$ satisfies the required functional dependencies,

- For every relation name R, $|R(\mathbb{D})| = |N|^k$, where $k = |f(\mathbf{V}(R))|$,

- $|Q(\mathbb{D})| = |N|^{|C|}$.

This yields that $\alpha(Q, \mathbb{D}) = C(Q, f)$, and hence $\alpha(Q) \geq C(Q)$ by Lemma 1. □

It is shown in [GLVV12] that the value $C(Q)$ can be computed by a linear program. In the case without functional dependencies, this program is dual to the program for $\mathrm{AGM}(Q)$, so $C(Q) = \mathrm{AGM}(Q) = \alpha(Q)$.

However, in the presence of functional dependencies, there are queries Q for which $\alpha(Q) > C(Q)$, as shown in the paper [GLVV12], by elaborating an example proposed by Dániel Marx. Interestingly, this construction uses Shamir's secret sharing scheme, which is based on the fact a polynomial of degree k is uniquely determined by any of its k values. There is another secret sharing scheme, Blakley's scheme, which employs vector spaces rather than polynomials, and the fact that any point in a k-dimensional vector space is uniquely determined by a k-tuple of hyperspaces in general position which contain it. More generally, we have the following lemma.

If V is a vector space, W is its subspace and $v \in V$, then $W + v = \{w + v : w \in W\}$ is the unique hyperspace in V that is parallel to W and contains v.

LEMMA 5. *Fix a vector space V, a family $(V_x)_{x \in X}$ of subspaces of V, and a subspace $V_0 \subseteq V$ such that $V_0 \supseteq \bigcap_{x \in X} V_x$. For any given $v \in V$, the hyperspaces $(V_x + v)_{x \in X}$ determine $V_0 + v$, i.e., for every two vectors $v, w \in V$, if $V_x + v = V_x + w$ for all $x \in X$, then $V_0 + v = V_0 + w$.*

PROOF. If $V_1 \subseteq V_0$ then $V_1 + v \subseteq V_0 + v$, and $V_0 + v$ is determined by $V_1 + v$ as follows: $V_0 + v = \{w + w' : w \in V_0, w' \in V_1\}$. In words, $V_0 + v$ is the unique hyperspace parallel to V_0 that contains $V_1 + v$.

Applying this observation to $V_1 = \bigcap_{x \in X} V_x$ yields that $V_0 + v$ is determined by $(\bigcap_{x \in X} V_x) + v = \bigcap_{x \in X} (V_x + v)$. □

The above lemma leads to a "multi-secret" sharing scheme, in which every participant has a publicly known subspace V_x of V, and his secret hyperplane $V_x + v$, where $v \in V$ is fixed and unknown. A set of participants Y can gather and determine the secret of the participant x whenever $V_x \supseteq \bigcap_{y \in Y} V_y$. This secret sharing scheme leads us to construction providing a tighter upper bound, which we describe below.

5.2 Vector space colorings

We consider vector spaces over a fixed finite field $\mathbb{K}$. If V is a vector space, X is a set, and V_x is a subspace of V for $x \in X$, then by $\sum_{x \in X} V_x$ we denote the smallest subspace of V containing every V_x, for $x \in X$. For a subspace W of a vector space V, by $\mathrm{codim}_V W$ we denote $\dim V - \dim W = \dim(V/W)$, where V/W denotes the quotient space, $V/W = \{W + v : v \in V\}$.

A *vector space coloring* of Q is a pair $\mathcal{V} = (V, (V_x)_{x \in \mathbf{V}(Q)})$, where V is a vector space and $(V_x)_{x \in \mathbf{V}(Q)}$ is a family of its subspaces. For such a coloring, define $V_Y = \sum_{x \in Y} V_x$ for $Y \subseteq \mathbf{V}(Q)$. We say that $\mathcal{V}$ *satisfies* a functional dependency $Y \mapsto x$ if $V_x \subseteq V_Y$. For a vector

space coloring $\mathcal{V}$ satisfying all the functional dependencies in $\mathcal{F}$, define

$$\mathrm{VC}_{\mathbb{K}}(Q, \mathcal{V}) = \frac{\dim(V_{\mathbf{V}(Q)})}{\max_{R \in \Sigma} \dim(V_{\mathbf{V}(R)})}, \qquad \text{(VC)}$$

Let $\mathrm{VC}_{\mathbb{K}}(Q) = \sup_{\mathcal{V}} \mathrm{VC}_{\mathbb{K}}(Q, \mathcal{V})$.

PROPOSITION 2. *Let Q be a natural join query. Then*

$$\alpha(Q) \geq \mathrm{VC}_{\mathbb{K}}(Q) \geq C(Q). \qquad (5)$$

The inequality $\mathrm{VC}_{\mathbb{K}}(Q) \geq C(Q)$ is obtained by defining for a coloring f a vector space coloring $\mathcal{V}$ with $V = \mathbb{K}^C$, $V_x = \mathbb{K}^{f(x)} \subseteq \mathbb{K}^C$, where $C = \bigcup_{x \in \mathbf{V}(Q)} f(x)$, and $\mathbb{K}^{f(x)}$ embeds into $\mathbb{K}^C$ in the natural way, by extending a vector with zeros on coordinates in $C - f(x)$. It is easy to see that $\mathcal{V}$ satisfies the same functional dependencies as f, and that $\mathrm{VC}_{\mathbb{K}}(Q, \mathcal{V}) = C(Q, f)$. This proves $\mathrm{VC}_{\mathbb{K}}(Q) \geq C(Q)$. To prove the bound $\alpha(Q) \geq \mathrm{VC}_{\mathbb{K}}(Q)$, we pass to dual vector spaces, as described below.

A *vector space system* over Q is a pair $\mathcal{V} = (V, (V_x)_{x \in \mathbf{V}(Q)})$, where V is a vector space and $(V_x)_{x \in \mathbf{V}(Q)}$ is a family of its subspaces. For such a system, define $V_Y = \bigcap_{y \in X} V_y$ for $Y \subseteq \mathbf{V}(Q)$, and say that $\mathcal{V}$ satisfies a functional dependency $Y \mapsto x$ if $V_x \supseteq V_Y$.

For a vector space system $\mathcal{V}$ which satisfies all the functional dependencies in $\mathcal{F}$, define

$$\mathrm{VC}_{\mathbb{K}}^*(Q, \mathcal{V}) = \frac{\mathrm{codim}_V(V_{\mathbf{V}(Q)})}{\max_{R \in \Sigma} \mathrm{codim}_V(V_{\mathbf{V}(R)})}. \qquad \text{(VC}^*\text{)}$$

Finally, let $\mathrm{VC}_{\mathbb{K}}^*(Q) = \sup_{\mathcal{V}} \mathrm{VC}_{\mathbb{K}}^*(Q, \mathcal{V})$.

LEMMA 6. *There is a bijection between vector space colorings and vector spaces systems, which maps a vector space coloring $\mathcal{V}$ to a vector space system $\mathcal{V}^*$ such that $\mathrm{VC}_{\mathbb{K}}^*(Q, \mathcal{V}^*) = \mathrm{VC}_{\mathbb{K}}(Q, \mathcal{V})$. In particular, $\mathrm{VC}_{\mathbb{K}}(Q) = \mathrm{VC}_{\mathbb{K}}^*(Q)$.*

PROOF. If V is a vector space, let V^* denote its algebraic dual, i.e., the space of all linear functionals from V to $\mathbb{K}$. If L is a subspace of V, then let $L^{\perp} \subseteq V^*$ denote the set of functionals $f \in V^*$ which vanish on L. Then

$$(L_1 \cap L_2)^{\perp} = L_1^{\perp} + L_2^{\perp} \qquad (6)$$

$$L_1 \subseteq L_2 \iff L_1^{\perp} \supseteq L_2^{\perp} \qquad (7)$$

$$\dim(L^{\perp}) = \mathrm{codim}_V(L). \qquad (8)$$

For a vector space coloring $\mathcal{V} = (V, (V_x)_{x \in \mathbf{V}(Q)})$ let $\mathcal{V}^* = (V^*, (V_x^{\perp})_{x \in \mathbf{V}(Q)})$. Using the facts (6),(7),(8), it is easy to check that the mapping $\mathcal{V} \mapsto \mathcal{V}^*$ yields a bijection with the required properties. $\square$

PROOF PROOF OF PROPOSITION 2. We prove the inequality $\alpha(Q) \geq \mathrm{VC}_{\mathbb{K}}^*(Q)$. Let $\mathcal{V} = (V, (V_x)_{x \in \mathbf{V}(Q)})$ be a vector space system. We construct a database $\mathbb{D}$ satisfying

$$\alpha(Q, \mathbb{D}) = \mathrm{VC}_{\mathbb{K}}^*(Q, \mathcal{V}). \qquad (9)$$

For a relation name R and a vector $v \in V$, let r_v be the row such that $r_v[x] = V_x + v \in V/V_x$ for every

attribute $x \in \mathbf{V}(R)$. Define $\mathbb{D}$ by setting

$$R(\mathbb{D}) = \{r_v : v \in V\},$$

for every relation name R. It is easy to verify that:

- The database $\mathbb{D}$ satisfies the functional dependencies. This follows from Lemma 5.

- For each relation name R, consider the mapping $f_R : V \to R(\mathbb{D})$, where $f_R(v) = r_v$. Clearly, the mapping is onto $R(\mathbb{D})$. To compute the size of its image, we analyse the kernel of f_R and observe that $\{w : r_w = r_v\} = V_{\mathbf{V}(R)} + v$ for every $v \in V$. In particular, $|R(\mathbb{D})| = |V|/|V_{\mathbf{V}(R)}| = |\mathbb{K}|^{\mathrm{codim}_V V_{\mathbf{V}(R)}}$.

- Similarly, we verify that $|Q(\mathbb{D})| = |V|/|V_{\mathbf{V}(Q)}| = |\mathbb{K}|^{\mathrm{codim}_V V_{\mathbf{V}(Q)}}$.

Equation (9) follows. $\square$

5.3 Group systems

We relax the notion of a vector space system, by considering finite groups, as follows. Let G be a finite group and $(G_x)_{x \in \mathbf{V}(Q)}$ be a family of its subgroups. For a set of attributes $X \subseteq \mathbf{V}(Q)$, denote by G_X the group $G_X = \bigcap_{x \in X} G_x$, and by G/G_X the space of (left) cosets, $\{g \cdot G_X : g \in G\}$. We call the pair $\mathcal{G} = (G, (G_x)_{x \in X})$ a *group system* for Q, and say that it satisfies a functional dependency $X \mapsto x$ if $G_X \subseteq G_x$. For a group system $\mathcal{G}$ satisfying all the functional dependencies in $\mathcal{F}$, define

$$\mathrm{GC}(Q, \mathcal{G}) = \frac{\log|G/G_{\mathbf{V}(Q)}|}{\max_{R \in \Sigma}(\log|G/G_{\mathbf{V}(R)}|)}, \qquad \text{(GC)}$$

and let $\mathrm{GC}(Q) = \sup_{\mathcal{G}} \mathrm{GC}(Q, \mathcal{G})$.

PROPOSITION 3. *Let Q be a natural join query. Then*

$$\alpha(Q) \geq \mathrm{GC}(Q) \geq \mathrm{VC}_{\mathbb{K}}^*(Q) = \mathrm{VC}_{\mathbb{K}}(Q) \geq C(Q). \quad (10)$$

PROOF. Clearly, $\mathrm{GC}(Q) \geq \mathrm{VC}_{\mathbb{K}}^*(Q)$, since every vector space system is a group system.

The proof of the inequality $\alpha(Q) \geq \mathrm{GC}(Q)$ is analogous to the proof of Proposition 2. From a group system $\mathcal{G}$, we construct a database $\mathbb{D}$. Here, we use cosets $g \cdot G_x$ instead of translates $V_x + v$, and the set of cosets G/G_x instead of quotients V/V_x. By following the same construction of $\mathbb{D}$, we get the following:

- The database $\mathbb{D}$ satisfies the functional dependencies. Here we use a lemma analogous to Lemma 5, which holds for groups as well.

- For each relation name R and vector v, $\{h : h_r = g_r\} = G_{\mathbf{V}(R)} \cdot g$. In particular, $|R(\mathbb{D})| = |G|/|G_{\mathbf{V}(R)}|$;

- $|Q(\mathbb{D})| = |G|/|G_{\mathbf{V}(Q)}|$.

This proves $\alpha(Q, \mathbb{D}) \geq \mathrm{GC}(Q, \mathcal{G})$, and hence $\alpha(Q) \geq \mathrm{GC}(Q)$ by Lemma 1. $\square$

Remark 2. The database $\mathbb{D}$ constructed in the above proof is G-symmetric. Indeed, the values appearing in

the database are cosets the form $g \cdot G_x$ (where $x \in \mathbf{V}(Q)$) and G acts (from the left) on such cosets in the obvious way. Moreover, the action of G on each table $R(\mathbb{D})$ is isomorphic to the action of G on $G/G_{\mathbf{V}(R)}$, in particular, it is transitive.

Also, if $\mathbb{D}$ is G-symmetric and $\mathbb{D}^n$ is defined as in the proof of Lemma 1, then $\mathbb{D}^n$ is G^n-symmetric.

6. TIGHTNESS

In Sections 4 and 5 we have shown that for every natural join query Q,

$$H(Q) \geq \alpha(Q) \geq \mathrm{GC}(Q).$$

In this section, we close the circle by proving the following.

PROPOSITION 4. *Let Q be a natural join query. Then*

$$\mathrm{GC}(Q) \geq H(Q). \tag{11}$$

Together with Proposition 3 and Lemma 3, this proves that $H(Q) = \alpha(Q) = \mathrm{GC}(Q)$. As noted in Proposition 1, this gives Theorem 1. Moreover, from the observation in Remark 2, it follows that the bound $\alpha(Q)$ can be approximated by (arbitrarily large) symmetric databases, proving Theorem 2.

All the necessary ideas to prove the proposition are present in the proof of main theorem from [CY02] in the version presented in the paper [Lun02]. However, we cannot directly apply that theorem, since we need to keep track of the functional dependencies. In the rest of Section 6, we present a self-contained proof of Proposition 4, following the ideas of [Lun02]. For the rest of this section, fix a random variable U, taking values in a finite set of rows with attributes X, and satisfying the functional dependencies $\mathcal{F}$.

We say that a random variable V is *rational* if for every value $v \in \mathrm{Im}(V)$, the probability that V achieves v is a rational number.

LEMMA 7. *For every number $\varepsilon > 0$ there exists a rational random variable V satisfying the same functional dependencies as U, and such that $\|\mathrm{ent}(V) - \mathrm{ent}(U)\| < \varepsilon$ with respect to the euclidean norm on $\mathbb{R}^{P(X)}$.*

PROOF. Let $T = \mathrm{Im}(U)$. Observe that every random variable V with values in T satisfies the same functional dependencies as U. Denote by $\mathcal{D}(T)$ the set of probability distributions on T. Consider the mapping $\mathrm{ent} : \mathcal{D}(T) \to \mathbb{R}^{P(X)}$, which maps a probability distribution $D \in \mathcal{D}(T)$ to the entropy vector $\mathrm{ent}(V) \in \mathbb{R}^{P(X)}$ of any random variable V with distribution D. As is clearly visible from the explicit formula for this mapping, it is continuous. We conclude the lemma by observing that among the set of all probability distributions on T, distributions with rational values form a dense subset. $\square$

To prove Proposition 4, we proceed as follows. For each rational random variable U satisfying the given functional dependencies, we will find a sequence of group

systems $\mathcal{G}_k$ satisfying the functional dependencies $\mathcal{F}$, and such that

$$\lim_{k \to \infty} \mathrm{GC}(Q, \mathcal{G}_k) = H(Q, U). \tag{12}$$

From that, Proposition 4 follows:

$$H(Q) = \sup_U H(Q, U) \overset{Lem.7}{=} \sup_{U \text{ rational}} H(Q, U) =$$

$$\sup_{U \text{ rational}} H(Q, U) \overset{(12)}{\leq} \sup_{\mathcal{G}} \mathrm{GC}(Q, \mathcal{G}) = \mathrm{GC}(Q).$$

From now on, let U be a rational random variable satisfying the functional dependencies $\mathcal{F}$. We will show that there exists a sequence of group systems witnessing (12).

Let $q \in \mathbb{N}$ be a natural number such that for each row $r \in \mathrm{Im}(U)$ the probability $\mathbb{P}[U = r]$ can be represented as a rational number with denominator q. For $k = q, 2q, 3q, \ldots$ let A_k be a matrix whose columns are indexed by attributes, containing exactly $k \cdot \mathbb{P}[U = r]$ copies of the row r, for every row $r \in \mathrm{Im}(U)$. Notice that $k \cdot \mathbb{P}[U = r]$ is always a natural number, and that A_k has exactly k rows in total.

Let G^k denote the group of all permutations of the rows of A_k. For a set of attributes $Y \subseteq X$, let G_Y^k denote the subgroup of G^k which stabilizes the submatrix $A_k[Y]$ of A_k, i.e.,

$$G_Y^k = \{\sigma \in G_Y^k \mid \sigma(r)[y] = r[y] \text{ for each } r \in A_k \text{ and } y \in Y\}.$$

Denote $G_{\{x\}}^k$ by G_x^k. In particular, $G_Y^k = \bigcap_{y \in Y} G_y^k$. The following lemma is immediate.

LEMMA 8. *Suppose that U satisfies the functional dependency $Y \mapsto x$. Then $G_Y^k \subseteq G_x^k$.*

We define $\mathcal{G}_k$ as $(G^k, (G_x^k)_{x \in X})$. By Lemma 8, $\mathcal{G}_k$ is a group system satisfying the functional dependencies $\mathcal{F}$. It remains to prove (12).

Fix a set of attributes $Y \subseteq X$. For a row $r \in \mathrm{Im}(U[Y])$, let p_r denote $\mathbb{P}[U[Y] = r]$. Since r occurs exactly $k \cdot p_r$ times as a row of $A_k[Y]$, it follows that

$$|G_Y^k| = \prod_{r \in \mathrm{Im}(U[Y])} (k \cdot p_r)!.$$

Using Stirling's approximation we get that

$$\lim_{k \to \infty} \frac{1}{k} \log \left(\frac{|G^k|}{|G_Y^k|} \right) = \lim_{k \to \infty} \frac{1}{k} \log \left(\frac{k!}{\prod_{r \in \mathrm{Im}(U[Y])} (k \cdot p_r)!} \right) =$$

$$\lim_{k \to \infty} \frac{1}{k} \left(k \log(k) - \sum_{r \in \mathrm{Im}(U[Y])} (k \cdot p_r) \log (k \cdot p_r) \right) =$$

$$\lim_{k \to \infty} \frac{1}{k} \left(- \sum_{r \in \mathrm{Im}(U[Y])} (k \cdot p_r)(\log (k \cdot p_r - \log k)) \right) =$$

$$\lim_{k \to \infty} \frac{1}{k} \cdot (-k) \cdot \sum_{r \in \mathrm{Im}(U[Y])} p_r \log p_r = \mathrm{ent}(U)_Y.$$

In particular,

$$\text{GC}(Q, \mathcal{G}^k) = \frac{\frac{1}{k}\log(|G^k|/|G_X^k|)}{\max_{R \in Q} \frac{1}{k}\log(|G^k|/|G_{\mathbf{V}R}^k|)} \overset{k \to \infty}{\longrightarrow}$$

$$\overset{k \to \infty}{\longrightarrow} \frac{H(U)}{\max_{R \in Q} H(U[\mathbf{V}(R)])} = H(Q, U).$$

This yields (12) proving Proposition 4, which together with Lemma 3 gives $H(Q) = \text{GC}(Q) = \alpha(Q)$. By Proposition 1, this finishes the proof of Theorem 1. The more general Theorem 3 is proved similarly. Moreover, Theorem 2 follows from Remark 2.

7. GENERAL CONJUNCTIVE QUERIES

The previous sections concern natural join queries: conjunctive queries without existential quantifiers (or projections), in which the variables name coincides with the name of the attribute of the relation in which it appears (in particular, the same variable name cannot occur in the scope of one conjunct, and there are no equalities). In this section, we discuss how to treat arbitrary conjunctive queries.

7.1 Set semantics

Define a *natural conjunctive query* to be a query of the form $Q = \exists Y \ Q'$, where Q' is a natural join query over the schema Σ, and $Y \subseteq \mathbf{V}(\Sigma)$. The set of *free variables* of Q is $\mathbf{V}(Q) = \mathbf{V}(\Sigma) - Y$. For a database $\mathbb{D}$ over the schema Σ, define $Q(\mathbb{D})$ as $T[\mathbf{V}(Q)]$, where $T = Q'(\mathbb{D})$. In other words, $Q(\mathbb{D})$ is the table $Q'(\mathbb{D})$ restricted to the free variables of Q. This is the so-called *set-semantics*, since the result $T[\mathbf{V}(Q)]$ is a set of rows, i.e., each row occurs either 0 or 1 times. The alternative *bag-semantics* is discussed in Section 7.2.

As explained in [GLVV12] for each conjunctive query Q there exists a natural conjunctive join query S, such that $\alpha(Q) = \alpha(S)$. Such S can be constructed in a purely syntactical way from Q by the chase procedure. Because of this, we only consider natural conjunctive queries.

The definition of $\alpha(Q)$ can be lifted without modification to natural conjunctive queries. The generalization of Theorem 1 has the expected form:

THEOREM 5. *Fix a relational schema Σ and a set of functional dependencies $\mathcal{F}$. Let Q be a natural conjunctive query over the schema Σ. Then $\alpha(Q)$ is equal to the maximal value of $v_{\mathbf{V}(Q)}$, for v ranging over $\overline{\Gamma}^*_{\mathbf{V}(\Sigma)}$ and satisfying:*

$$\begin{cases} v_{\mathbf{V}(R)} \leq 1 & \text{for } R \in Q, \\ v_{Z \cup \{z\}} = v_Z & \text{for every fd } Z \mapsto z \text{ in } \mathcal{F}. \end{cases}$$

The proof of the lower bound is exactly the same as the proof of Theorem 1. However the proof of the upper bound has to be modified slightly, as described below.

For a query $Q = \exists Y. \ Q'$, define $H(Q, U)$ and $H(Q)$ as in Section 4, where R in the maximum ranges over Q' rather than Q. We then have the following analogue of Lemma 3, proving the upper bound.

LEMMA 9. *For a natural conjunctive query $Q = \exists Y. \ Q'$, $\alpha(Q) \leq H(Q)$.*

PROOF. For a database $\mathbb{D}$, let $U_{\mathbb{D}}$ be the random variable with values in $Q'(\mathbb{D})$, described as the result r' of the following process: first choose uniformly at random a row $r \in Q(\mathbb{D})$, and then, choose uniformly at random a row $r' \in Q'(\mathbb{D})$ such that $r'[\mathbf{V}(Q)] = r$. The following claim the follows by definition.

CLAIM 1. *The random variable $U_{\mathbb{D}}[\mathbf{V}(Q)]$ is uniformly distributed on $Q(\mathbb{D})$.*

Now we get that:

$$\alpha(Q, \mathbb{D}) \overset{def}{=} \frac{\log|Q(\mathbb{D})|}{\max_{R \in Q'} \log|R(\mathbb{D})|} \overset{Claim \ 1}{=}$$

$$= \frac{H(U_{\mathbb{D}}[\mathbf{V}(Q)])}{\max_{R \in Q'} \log|R(\mathbb{D})|} \leq \frac{H(U_{\mathbb{D}}[\mathbf{V}(Q)])}{\max_{R \in Q'} H(U_{\mathbb{D}}[\mathbf{V}(R)])}$$

$$= H(Q, U_{\mathbb{D}}).$$

By a similar argument as in the proof of Lemma 3, we derive that $\alpha(Q) \leq H(Q)$. $\square$

7.2 Bag semantics

In the bag semantics, tuples may occur in the output with multiplicity other than 1, i.e. if Q is a conjunctive query, then $Q_{bs}(\mathbb{D})$ is a multiset rather than a set, where the multiplicity of a row in the outcome is equal to the number of rows which are projected to it. This is the standard semantics used in SQL. The possibility of having high multiplicity of tuples in the output may affect size of the bound of output. Analogously to the definition in the introduction, we define the size-increase bound $\alpha_{bs}(Q)$ for bag semantics as the smallest value α such that $|Q_{bs}(\mathbb{D})| \leq c \cdot |\mathbb{D}|^\alpha$ for some constant c.

Under bag semantics, the evaluation of a natural join query (without projections) is the same as the evaluation under set semantics. Only projection needs to be handled differently: under bag semantics, tuple repetitions are not removed. This leads us to the following fact:

FACT 1. *Let $Q = \exists Y \ Q'$ be a natural conjunctive query. Then for an arbitrary database $\mathbb{D}$, the size of the output $|Q_{bs}(\mathbb{D})|$ under bag semantics is equal to the size of the output $|Q'(\mathbb{D})|$.*

By the fact above, in order to compute $\alpha_{bs}(Q)$ for the bag semantics we can compute $\alpha_{bs}(Q')$. But Q' is a natural join query, and so, set semantics and bag semantics coincide. By Theorem 1 we get $\alpha(Q') = H(Q')$. Concluding:

COROLLARY 5. *For an arbitrary natural conjunctive query $Q = \exists Y \ Q'$ under bag semantics we have $\alpha_{bs}(Q) = H(Q')$.*

8. EVALUATION

In this section, we give some rudimentary results describing bounding the worst-case complexity of computing the result of a query $Q(\mathbb{D})$, for a given database

$\mathbb{D}$. In the absence of functional dependencies, it is known [NPRR12] that $Q(\mathbb{D})$ can be computed from $\mathbb{D}$ in time proportional to $|\mathbb{D}|^{\alpha(Q)}$. In the bounds presented below, there is an additional factor $|\mathbb{D}|^m$, where m is a parameter depending on the functional dependencies, defined below.

Let $\mathcal{F}$ be a set of functional dependencies over attributes X. A *minimal component* C is an inclusion-minimal nonempty set of attributes $C \subseteq X$ with the property that whenever $Y \mapsto x$ is a functional dependency with $Y \cap C$ nonempty, then $x \in C$. Define $\mathcal{F}[C]$ to be the set of functional dependencies over C which consists of those functional dependencies $Y \mapsto x$ from $\mathcal{F}$, such that $Y \subseteq C$ (and then necessarily $x \in C$ by definition).

We say that a set of attributes $S \subseteq X$ *spans* $\mathcal{F}$, if the smallest subset $\bar{S}$ of X set containing S and closed under functional dependencies (i.e., $Y \mapsto x$ and $Y \subseteq \bar{S}$ implies $x \in \bar{S}$) is equal to X. We say that $\mathcal{F}$ has *width* m if it has a spanning set of size m. We inductively define the *iterative width* of $\mathcal{F}$ to be equal to m, if for every its minimal component C, $\mathcal{F}[C]$ has width m, and after removing from $\mathcal{F}$ the attributes which belong to the minimal components, the resulting set of functional dependencies also has iterative width m, or the set of attributes is empty.

Example 1. If $\mathcal{F}$ is an empty set of functional dependencies over a nonempty set of attributes, then $\mathcal{F}$ has iterative width 1.

Example 2. The set of dependencies $x \mapsto y, y \mapsto z, z \mapsto x$ has width 1, since it is spanned by $\{x\}$. It also has iterative width 1.

Example 3. Let X be a set with three elements and let $\mathcal{F}$ be the set of all dependencies of the form $Y \mapsto x$, with $Y \subseteq X, |Y| = 2$. Then $\mathcal{F}$ has width 2.

Let $\mathbb{D}$ be a database and let C be a minimal component. For a row r over attributes X, denote by r/C the row with attributes X, defined as follows:

$$r/C[x] = \begin{cases} r[x] & \text{for } x \in X - C \\ x & \text{for } x \in X \cap C. \end{cases}$$

Therefore, r/C is obtained by replacing each value of an attribute in C by a placeholder, storing the name of the attribute. Denote by $\mathbb{D}/C$ the database obtained from $\mathbb{D}$ by replacing in each table R, every row r by the row r/C. Clearly, we have the following.

LEMMA 10. *The database $\mathbb{D}/C$ can be computed from $\mathbb{D}$ in linear time.*

The following lemma is immediate, by the fact that C is a minimal component.

LEMMA 11. *The database $\mathbb{D}/C$ satisfies all the functional dependencies of Q.*

Note, however, that the database $\mathbb{D}/C$ usually satisfies more functional dependencies than $\mathbb{D}$, namely, it satisfies all functional dependencies $\emptyset \mapsto x$, for $x \in C$.

LEMMA 12. *Let C be a minimal component, and suppose that $\mathcal{F}[C]$ has width m. Then, for a given database $\mathbb{D}$, the result $Q(\mathbb{D})$ can be computed from $Q(\mathbb{D}/C)$ in time $O(|Q(\mathbb{D}/C)| \cdot |\mathbb{D}|^m)$.*

PROOF. To compute $Q(\mathbb{D})$, proceed as follows.

For each row $s \in Q(\mathbb{D}/C)$, we need to determine whether the placeholders can be replaced by actual values, yielding a row r such that $r[\mathbf{V}(R)] \in R(\mathbb{D})$ for every relation name R.

For an arbitrary attribute $x \in C$, consider the set

$$V_x = \bigcap_{R \in Q} \{r[x] : r \in R(\mathbb{D})\}$$

of all possible values for x. The set V_x has size at most $|\mathbb{D}|$, and can be computed in linear time from $\mathbb{D}$.

Let S be a set spanning C, Let K denote the table consisting of those rows r over S such that $r[x] \in V_x$ for all $x \in S$. The table K has size at most $|\mathbb{D}|^m$.

For a row $s \in Q(\mathbb{D}/C)$ and a row $r \in K$, we say that a row t is *compatible* with s and r if the following conditions hold:

- $t[x] = s[x]$ for $x \in S$,
- $t[x] = r[x]$ for $x \notin C$,
- $t[\mathbf{V}(R)] \in R(\mathbb{D})$ for every relation name R.

CLAIM 2. *If there is a row t compatible with s and r, then it is unique, and, given s, can be computed in constant time (assuming linear time preprocessing independent of s).*

The algorithm for computing $Q(\mathbb{D})$ proceeds by computing, for all $s \in Q(\mathbb{D}/C)$ and all $r \in K$, the row compatible with s and r (if it exists), and adding it to the result. $\square$

By iteratively applying Lemma 12, we obtain the main result of this section.

PROPOSITION 5. *Fix a natural join query Q and functional dependencies $\mathcal{F}$ of iterative width m. Then there is an algorithm which for a given database $\mathbb{D}$ computes $Q(\mathbb{D})$ in time $O(|\mathbb{D}|^{\alpha(Q)+m})$.*

Observe that when the set $\mathcal{F}$ of functional dependencies is empty, Proposition 5 gives the algorithm from [AGM13], whose running time is $O(|\mathbb{D}|^{\alpha(Q)+1})$, since in this case, $\mathcal{F}$ has iterative width 1.

9. CONCLUSION AND FUTURE WORK

We characterized the worst-case size-increase for the evaluation of conjunctive queries, in two ways: in terms of entropy and in terms of finite groups. Our characterization improves a construction and answers a question from [GLVV12]. We also presented a rudimentary result concerning the evaluation of natural join queries.

We see two main directions of a possible future work. One can try to find a method for computing $\alpha(Q)$. This looks hard, and probably will require a deeper understanding of entropy, and the entropy cone in particular. By comparison, we note that in cryptography

and information theory the following, seemingly similar optimization problem, has emerged [BD91] and is considered notoriously difficult. Given an *access structure*, which is an upward-closed family A of subsets of a finite set of participants U, (i.e., $A \subseteq P(U)$ and $V \in A, V \subseteq W \subseteq U$ imply $W \in A$), the aim is to find a secret-sharing scheme in which a set set of participants $V \subseteq U$ can jointly determine the secret s if and only if $V \in A$. The goal is to minimize the ratio of information possessed by the participants to the information stored in the secret. As an optimization problem, this can be expressed as follows. For v ranging over $\overline{\Gamma^*_{A \cup \{s\}}}$, satisfying

$$\begin{cases} v_{X \cup \{s\}} = v_X & \text{if } X \in A, \\ v_{X \cup \{s\}} = v_X + v_s & \text{if } X \notin A, \\ v_u \leq 1 & \text{for } u \in U, \end{cases}$$

maximize the value v_s. The inverse of the optimal value is called the *optimal complexity* or *information rate* of the access scheme A. As noted in [FMBPV12], *"determining the optimal complexity for general access structures has appeared to be an extremely difficult open problem"*.

This demonstrates that optimization problems over the entropy cone can be very difficult. Of course, this will depend very much on the structure of the problem, which may in some cases turn out to be feasible. For instance, the AGM bound demonstrates that in the absence of functional dependencies, it is sufficient to relax the entropy cone to Shannon's information inequalities. Whereas this is no longer true in the presence of functional dependencies, as demonstrated in [GLVV12], it still might be the case that considering only finitely many *non-Shannon* information inequalities is sufficient to compute the optimum.

The paper [NPRR12] manages to find an optimal algorithm for evaluating natural join queries, by proving algorithmic versions of the Loomis-Whitney and Bollobás-Thomason inequalities. Perhaps it is possible to extend this algorithm to the case with functional dependencies, yielding an optimal running time $|\mathbb{D}|^{\alpha(Q)}$, even if the precise value of $\alpha(Q)$ is unknown. However, the worst-case optimal algorithm presented in [NPRR12] uses an optimal solution for fractional edge covering. This suggests that finding a worst-case optimal algorithm for the general case will be impossible without simultaneously computing $\alpha(Q)$.

The fractional edge cover was useful in the analysis of the Hypercube algorithm [BKS13], an algorithm for parallel evaluation of queries. Perhaps some ideas from the current paper can also be applied in the parallel setting.

10. REFERENCES

[AGM13]   Albert Atserias, Martin Grohe, and Dániel Marx. Size bounds and query plans for relational joins. *SIAM J. Comput.*, 42(4):1737–1767, 2013.

[BD91]   ErnestF. Brickell and DanielM. Davenport. On the classification of ideal secret sharing schemes. *Journal of Cryptology*, 4(2):123–134, 1991.

[BKS13]   Paul Beame, Paraschos Koutris, and Dan Suciu. Communication steps for parallel query processing. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pages 273–284, 2013.

[CY02]   Terence H. Chan and Raymond W. Yeung. On a relation between information inequalities and group theory. *IEEE Transactions on Information Theory*, 48(7):1992–1995, 2002.

[FMBPV12]   Oriol Farràs, Jessica Ruth Metcalf-Burton, Carles Padró, and Leonor Vázquez. On the optimization of bipartite secret sharing schemes. *Des. Codes Cryptography*, 63(2):255–271, May 2012.

[Fri04]   Ehud Friedgut. Hypergraphs, entropy, and inequalities. *The American Mathematical Monthly*, 111(9):749–760, 2004.

[GLVV12]   Georg Gottlob, Stephanie Tien Lee, Gregory Valiant, and Paul Valiant. Size and treewidth bounds for conjunctive queries. *J. ACM*, 59(3):16, 2012.

[Lun02]   Desmond S. Lun. A relationship between information inequalities and group theory, 2002.

[LW49]   L. H. Loomis and H. Whitney. An inequality related to the isoperimetric inequality. *Bull. Am. Math. Soc.*, 55:961–962, 1949. MR:0031538. Zbl:0035.38302.

[NPRR12]   Hung Q. Ngo, Ely Porat, Christopher Ré, and Atri Rudra. Worst-case optimal join algorithms: [extended abstract]. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 37–48, 2012.

[ZY98]   Zhen Zhang and Raymond W. Yeung. On characterization of entropy function via information inequalities. *IEEE Transactions on Information Theory*, 44(4):1440–1452, 1998.