Bounds and Algorithms for Joins via Fractional Edge Covers

Martin Grohe

Humboldt Universität zu Berlin
grohe@informatik.hu-berlin.de

1 Introduction

Among the operations of relational algebra, the join operation tends to be the most costly. There is a wealth of research in the database literature devoted to efficient join processing. Short of fully computing the result of a join or a sequence of joins (we use the term *join query* in the following), in many applications it is also important to get good bounds on the size of the result.

A relatively new idea exploits the structure of the join query to obtain nontrivial bounds on the size of the result and to design algorithms computing the result in time linear in the estimated size of the result. These bounds are based on a combinatorial parameter known as *fractional edge cover number* of the query. The purpose of this paper is to explain this idea and give a survey of the results based on it.

Consider the natural-join query

$$Q = R_1 \bowtie \ldots \bowtie R_m.$$

Given a database instance D of schema $\{R_1, \ldots, R_m\}$, we want to bound the size of the query answer $Q(D)$ in terms of the sizes $N_i := |R_i(D)|$ of the input relations. As a start, suppose that there is a relation R_i that contains all attributes appearing in the query. Then, trivially, the size of the query answer is bounded by the size of the relation: $|Q(D)| \leq N_i$. Suppose next that, instead of one relation that contains all attributes, we have relations $R_{i_1}, \ldots, R_{i_k}$ that together contain all attributes. Then $|Q(D)| \leq \prod_{j=1}^{k} N_{i_j}$. We call $R_{i_1}, \ldots, R_{i_k}$ an *edge cover* of Q. Now we can try to find the edge cover that gives us the best bound. We can express this as an integer linear program in the variables $x_1, \ldots, x_m$, where $x_i = 1$ expresses that R_i is in the edge cover. Suppose that the attributes appearing in Q are $A_1, \ldots, A_n$.

$$\text{minimise} \quad \sum_i x_i \log N_i, \tag{1}$$

$$\text{where} \quad \sum_{\substack{i \text{ such that } A_j \\ \text{attribute of } R_i}} x_i \geq 1 \qquad \text{for } j = 1, \ldots, n \tag{2}$$

$$x_i \in \{0, 1\} \qquad \text{for } i = 1, \ldots, m. \tag{3}$$

Then for every solution $\boldsymbol{x} = (x_1, \ldots, x_m) \in \{0,1\}^m$ of this integer linear program, we have

$$|Q(D)| \leq \prod_{i=1}^{m} N_i^{x_i} = 2^{\sum_i x_i \log N_i}. \tag{4}$$

We call the value $\rho(Q, D) = \sum_i x_i \log N_i$ of an optimal solution of the integer linear program the *edge cover number* of Q in D.

So — we have found a complicated way to state a trivial observation. It is hard to imagine, though, how we can obtain nontrivial bounds on the size of the query answer if we just know the query and the size of the input relations. Surprisingly, there are such bounds. Let us look at the LP relaxation of the integer linear program (1)–(3), where we replace the integrality constraints $x_i \in \{0, 1\}$ by the inequalities

$$0 \leq x_i \qquad \text{for } i = 1, \ldots, m. \tag{5}$$

(There is no need to add inequalities $x_i \leq 1$, because in an optimal solution it never makes sense to let $x_i > 1$.) We call a rational solution $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathbb{Q}^m$ to this linear program a *fractional edge cover* of Q. We call the value $\rho^*(Q, D) := \sum_i x_i \log N_i$ of an optimal solution $\boldsymbol{x}$ to the linear program the *fractional edge cover number* of Q in D. It turns out that the bound (4) remains valid for fractional edge covers and that it is actually tight.

Theorem 1 (Grohe and Marx [6], Atserias, Grohe, and Marx [1]). *Let Q be a join query. Then for every database instance D,*

$$|Q(D)| \leq 2^{\rho^*(Q,D)}.$$

Furthermore, there are arbitrarily large database instances D such that $|Q(D)| = 2^{\rho^*(Q,D)}$.

It is neither obvious why this theorem should hold nor why it is an improvement over the trivial bound (4) for the (integral) edge cover number. I will try to answer both questions with the following examples. In a way, these examples form the core of the whole paper.

1.1 Examples

We give two examples. The first illustrates the main idea of the upper bound of Theorem 1. The second shows that the fractional edge cover number of Q in D may be substantially smaller than the edge cover number.

Example 1. Let us consider the query

$$Q(A, B, C) = R(A, B) \bowtie S(B, C) \bowtie T(C, A)$$

with attributes A, B, C and relation schemas $R = R(A, B)$, $S = S(B, C)$, and $T = T(C, A)$. Let D be a database instance of this schema, and let $N_R := |R(D)|$, $N_S := |S(D)|$, and $N_T := |T(D)|$. We want to give an upper bound on the size $N_Q := |Q(D)|$ of the query answer.

The linear program associated with Q and D looks as follows:

$$\begin{aligned} \text{minimise } & x_R \log N_R + x_S \log N_S + x_T \log N_T, \\ \text{where } & x_R + x_T \geq 1 \\ & x_R + x_S \geq 1 \\ & x_S + x_T \geq 1 \\ & x_R, x_S, x_T \geq 0. \end{aligned}$$

Observe that $x_R = x_S = x_T = 1/2$ is a feasible solution to this linear program. It is an optimal solution if $N_R = N_S = N_T$. We shall prove that

$$N_Q \leq 2^{(1/2)\log N_R + (1/2)\log N_S + (1/2)\log N_T} = \sqrt{N_R \cdot N_S \cdot N_T}. \tag{6}$$

It is worth thinking about how to prove this bound for a minute. The special case where D is an undirected graph and $R = S = T$ is the edge relation may be most intuitive. In this special case, Q asks for all triangles in the graph, and (6) says that there are at most $M^{3/2}$ triangles, where $M := N_R = N_S = N_T$ is the number of edges of the graph. (This bound on the number of triangles appeared in [2]; the slightly better bound $(2M)^{3/2}/6$ can be found in [7]). Even in this special case, I see no obvious direct proof for the bound.

In our proof, we take an information theoretic approach. We ask how many bits we need on average to describe a tuple chosen from $Q(D)$ uniformly at random. To be clear what is meant here, let us describe this as a two-player game: suppose that player (P) wants to inform player (M) about the outcome of an experiment where a tuple $(a, b, c) \in Q(D)$ was drawn uniformly at random. Both players know the query Q and the database D and thus the query answer $Q(D)$ in advance, but only (P) knows the outcome (a, b, c) of the experiment. The players may agree

on a coding system that allows (P) to transmit (a, b, c) using as few bits as possible on average. For example, they may use a Huffman code. The quantity "average number of bits" we look for is essentially the *entropy* $H(X_Q)$ of a random variable X_Q that, for all $(a, b, c) \in Q(D)$, takes value (a, b, c) with probability $1/N_Q$.[1] As the distribution is uniform, the best the two players can do is number the tuples in $Q(D)$ in advance and then have (P) send the number corresponding to (a, b, c) in binary. This essentially shows that $H(X_Q) = \log N_Q$.

We now give a different protocol that yields an estimate of $H(X_Q)$ in terms of $H(X_R)$, $H(X_S)$, and $H(X_T)$, where X_R is the random variable that picks an element $(a, b) \in R(D)$ uniformly at random and X_S, X_T are defined similarly. The same argument that showed $H(X_Q) = \log N_Q$ shows that $H(X_R) = \log N_R$ and $H(X_S) = \log N_S$ and $H(X_T) = \log N_T$.

Here is the protocol. (P) transmits the tuple $(a, b, c) \in Q(D)$ in three steps. In the first step, he transmits a using an optimal coding system for the projection of X_q on the first component. The distribution of the projected random variable is known as the marginal distribution; note that it is not necessarily uniform, because some elements a may be contained in more tuples $(a, b, c) \in Q(D)$ than others. In the second step, (P) transmits b, taking into account that (M) already knows a. He uses an optimal coding system for the random variable that picks a b such that (a, b) can be extended to a tuple $(a, b, c) \in Q(D)$ with a distribution that takes the number of such extensions into account. In the third step, (P) transmits c, taking into account that (M) already knows a, b, and using an optimal coding system for the random variable that picks a c such that $(a, b, c) \in Q(D)$. More formally, we write X_Q as a triple (X_A, X_B, X_C) of random variables describing the first, second, and third component of the tuple. As indicated above, the random variables X_A, X_B, X_C are not uniformly distributed. And of course they are not independent. The protocol is based on the fact that

$$H(X_Q) = H(X_A) + H(X_B \mid X_A) + H(X_C \mid X_A, X_B).$$

Here the *conditional entropy* $H(X_B \mid X_A)$ of "X_B given X_A" is essentially the average, taken over all a, of the average number of bits transmitted with an optimal coding system for b given a. The conditional entropy $H(X_C \mid X_A, X_B)$ of "X_C given X_A and X_B" has a similar meaning.

Based on the fact that the uniform distribution on a domain always has the highest entropy (because there are no clever coding systems that exploit imbalances in the distribution), we make a few crucial observations:

(i) $H(X_A) + H(X_A \mid X_B) = H(X_A, X_B) \leq H(X_R)$,
because transmitting (a, b) such that there is a c with $(a, b, c) \in Q(D)$ requires fewer bits than transmitting an arbitrary $(a, b) \in R(D)$ chosen uniformly at random;

(ii) $H(X_B \mid X_A) + H(X_C \mid X_A, X_B) \leq H(X_B) + H(X_C \mid X_B) = H(X_B, X_C) \leq H(X_S)$,
where for the first inequality we note that dropping information can only increase the entropy and for the second inequality we argue as in (i).

(iii) $H(X_A) + H(X_C \mid X_A, X_B) \leq H(X_A) + H(X_C \mid X_A) = H(X_A, X_C) \leq H(X_T)$.

Putting things together, we see that

$$\begin{aligned}
2 \log N_Q &= 2H(X_Q) \\
&= 2\big(H(X_A) + H(X_B \mid X_A) + H(X_C \mid X_A, X_B)\big) \\
&= \big(H(X_A) + H(X_B \mid X_A)\big) + \big(H(X_B \mid X_A) + H(X_C \mid X_A, X_B)\big) \\
&\quad + \big(H(X_A) + H(X_C \mid X_A, X_B)\big) \\
&\leq H(X_R) + H(X_S) + H(X_T) \\
&= \log N_R + \log N_S + \log N_T.
\end{aligned}$$

This implies (6).

[1] To be precise, we have $H(X_Q) \leq$ expected number of transmitted bits of an optimal coding system $< H(X_Q) + 1$. This is Shannon's famous Source Coding Theorem [9].

A formal treatment of the arguments given in Example 1, including definitions of entropy and conditional entropy, can be found in Section 2.

Example 2 ([6]). Let $m \in \mathbb{N}^+$ be even, and let $n := \binom{m}{m/2}$. For every $m/2$-element subset $s \subseteq [m] := \{1, \dots, m\}$, let $A(s)$ be an attribute, and for every $i \in [m]$, let R_i be a relation schema with attributes $A(s)$ for all s that contain i. Let $Q := R_1 \bowtie \dots \bowtie R_m$, and let D be a database instance with $|R_i(D)| = N$ for all $i \in [m]$. Then

$$\rho^*(Q, D) \leq 2 \log N,$$

because $\boldsymbol{x} = (x_1, \dots, x_m)$ with $x_i := 2/m$ is a solution for the linear program (1), (2), and (5).

On the other hand,

$$\rho(Q, D) \geq (m/2 + 1) \log N.$$

To see this, let $\boldsymbol{x} = (x_1, \dots, x_m) \in \{0, 1\}^m$ be a solution to the integer linear program (1)–(3). Then at most $m/2 - 1$ of the x_is are 0, because otherwise there is a set $s \subseteq [m]$ such that $|s| = m/2$ and $x_i = 0$ for all $i \in s$, and then equation (2) is violated for the index j of the attribute $A(s)$. Thus at least $(m/2 + 1)$ of the x_is are 1, and we have $\sum_i x_i \log N \geq (m/2 + 1) \log N$.

1.2 Algorithms

It was shown in [6] that there is an algorithm computing the result of a join query Q in a database D of size N in time

$$O\big(N + M \cdot 2^{\rho^*(Q,D)}\big), \tag{7}$$

where $M := \max_R |R(D)|$ is the maximum size of a relation of D. Here we are mainly concerned with data complexity and ignore a small polynomial factor in terms of the query size. It was observed in [1] that for every join query Q there is a *join-project plan* (i.e., a relational-algebra expression equivalent to the query that uses only joins and projections) that can be executed in time (7). Furthermore, it was shown that there are queries Q such that every *join plan* for Q has an execution time that is worse by a factor $O(N^{\log |Q|})$.

Very recently, Ngo et al. [7] found an algorithm for answering join queries that avoids the factor M in the running time (7) and is thus worst-case optimal.

Theorem 2 (Ngo, Porat, Ré, and Rudra [7]). *There is an algorithm for answering a join query Q in a database D of size N in time*

$$O\big(N + 2^{\rho^*(Q,D)}\big). \tag{8}$$

Interestingly, Ngo et al. [7] also showed that the running time (8) cannot be achieved by executing a join-project plan for the query.

1.3 Further Results

Gottlob, Lee, and Valiant [5, 4] extended Theorem 1 from join queries to conjunctive queries. They obtained similar bounds in a setting that involves key dependencies. These were extended by Valiant and Valiant [10, 4] to a a setting with arbitrary functional dependencies.

In a completely different direction, Atserias et al. [1] also considered an average case scenario (all results described so far were worst-case results). In the average case model, the size of the query answer is governed by a different combinatorial parameter of the query, the *maximum density*. Contrasting the worst-case results, it was shown that for every query there is a join plan whose execution is almost always optimal (in a precise probabilistic sense).

1.4 The Rest of this Paper

In Section 2, we give a proof of Theorem 1. In Section 3 we discuss extensions to conjunctive queries. Finally, in Section 4, we sketch the simple algorithm for answering join queries with running time (7) and discuss query plans.

1.5 Notation

We denote by $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, $\mathbb{N}$, $\mathbb{N}^+$ the reals, rationals, integers, nonnegative integers, and positive integers, respectively. For every $n \in \mathbb{N}$ we let $[n] := \{1, \dots, n\}$.

2 Bounds for Join Queries

2.1 Entropy and Shearer's Lemma

Random variables are mappings defined on some probability space. We only consider finite probability spaces. For each element $a \in \mathrm{rg}(X)$ of the range of a random variable X we have a probability $\Pr(X = a)$; this defines a probability distribution on the range. We allow arbitrary ranges for random variables (and not just real numbers). In our applications, the ranges will be sets of tuples of a database instance. If we have random variables X, Y with ranges A, B, respectively, then we may form a new random variable (X, Y) with range $A \times B$ by letting $\Pr((X, Y) = (a, b)) := \Pr(X = a, Y = b)$ (the comma in probabilities means conjunction). Conversely, if we have a random variable Z with range $A \times B$, then we may decompose it into two random variables X, Y with ranges A, B, respectively, such that $Z = (X, Y)$. We have $\Pr(X = a) = \sum_{b \in B} \Pr(Z = (a, b))$ and $\Pr(Y = b) = \sum_{a \in A} \Pr(Z = (a, b))$.

In the following, let X, Y be random variables with ranges A, B, respectively. The *entropy* of X is

$$H(X) := \sum_{a \in A} \Pr(X = a) \log \frac{1}{\Pr(X = a)}.$$

In Section 1.1, we interpreted $H(X)$ as the expected number of bits needed to encode a randomly chosen value of X with an optimal coding system. A more immediate interpretation that also gives a good intuition (at least qualitatively) is to think of $H(X)$ as a measure for the *uncertainty* of X. If there is an $a \in A$ such that $\Pr(X = a) = 1$ then there is no uncertainty, and we have $H(X) = 0$. On the other hand, if X is uniformly distributed, i.e., $\Pr(X = a) = 1/|A|$ for all $a \in A$, then we have $H(X) = \log |A|$. It is easy to see that this is the maximum entropy that a random variable X with range A may have, that is,

$$H(X) \leq \log |A| \tag{9}$$

for all X with $\mathrm{rg}(X) = A$.

The *joint entropy* $H(X, Y)$ of X and Y is the entropy of (X, Y), i.e.,

$$H(X, Y) = \sum_{a \in A, b \in B} \Pr(X = a, Y = b) \log \frac{1}{\Pr(X = a, Y = b)}.$$

For $b \in B$ with $\Pr(Y = b) \neq 0$, the *conditional probability* of $X = a$ given $Y = b$ is defined as $\Pr(X = a \mid Y = b) := \frac{Pr(X=a, Y=b)}{\Pr(Y=b)}$, and the *conditional entropy* of X given $Y = b$ is

$$H(X \mid Y = b) := \sum_{a \in A} \Pr(X = a \mid Y = b) \log \frac{1}{\Pr(X = a \mid Y = b)}.$$

Finally, the *conditional entropy* of X given Y is

$$\begin{aligned} H(X \mid Y) &:= \sum_{b \in B} \Pr(Y = b) \cdot H(X \mid Y = b) \\ &= \sum_{b \in B} \Pr(Y = b) \cdot \sum_{a \in A} \Pr(X = a \mid Y = b) \log \frac{1}{\Pr(X = a \mid Y = b)}. \end{aligned}$$

A straightforward calculation shows that

$$H(X, Y) = H(X) + H(Y|X). \tag{10}$$

Indeed,

$$\begin{aligned}
H(X,Y) &= \sum_{a\in A, b\in B} \Pr(X=a, Y=b) \log \frac{1}{\Pr(X=a, Y=b)} \\
&= \sum_{a\in A} \Pr(X=a) \sum_{b\in B} \Pr(Y=b \mid X=a) \left(\log \frac{1}{\Pr(X=a)} + \log \frac{1}{\Pr(Y=b \mid X=a)} \right) \\
&= \sum_{a\in A} \Pr(X=a) \log \frac{1}{\Pr(X=a)} \sum_{b\in B} \Pr(Y=b \mid X=a) \\
&\quad + \sum_{a\in A} \Pr(X=a) \sum_{b\in B} \Pr(Y=b \mid X=a) \log \frac{1}{\Pr(Y=b \mid X=a)} \\
&= H(X) + H(Y \mid X),
\end{aligned}$$

where the last equality holds because $\sum_{b\in B} \Pr(Y=b \mid X=a) = 1$.

It is slightly more difficult to prove that

$$H(X \mid Y) \le H(X). \tag{11}$$

Intuitively, this is clear because the uncertainty about X can only decrease with the additional information $Y=b$. A formal proof uses Jensen's inequality.

The definitions and equations (10) and (11) can easily be generalised to more than two random variables. In particular, for random variables $X_1, \ldots, X_n$,

$$H(X_1, \ldots, X_n) = H(X_1) + H(X_2 \mid X_1) + H(X_3 \mid X_1, X_2) + \ldots + H(X_n \mid X_1, \ldots, X_{n-1}), \tag{12}$$

and for all $J \subseteq [n]$

$$H\big(X \;\big|\; X_1, \ldots, X_n)\big) \le H\big(X \;\big|\; (X_j : j \in J)\big). \tag{13}$$

The following lemma first appeared in [3]. Our formulation and proof of the lemma are from [8].

Lemma 1 (Shearer's Lemma). *Let I be a finite set, and for each $i \in I$, let X_i be a random variable. For each $J \subseteq I$, let $X_J := (X_j : j \in J)$. Let $\mathcal{J} \subseteq 2^I$ be a multiset of subsets of I such that each $i \in I$ appears in at least q members of $\mathcal{J}$. Then*

$$H(X_I) \le \frac{1}{q} \sum_{J\in\mathcal{J}} H(X_J).$$

Proof. Let $<$ be an arbitrary linear order on I. By (12), for every $J \subseteq I$ we have

$$H(X_J) = \sum_{j\in J} H\big(X_j \;\big|\; (X_i : i \in J \text{ with } i < j)\big).$$

Thus

$$\begin{aligned}
\sum_{J\in\mathcal{J}} H(X_J) &= \sum_{J\in\mathcal{J}} \sum_{j\in J} H\big(X_j \;\big|\; (X_i : i \in J \text{ with } i < j)\big) \\
&\ge \sum_{J\in\mathcal{J}} \sum_{j\in J} H\big(X_j \;\big|\; (X_i : i \in I \text{ with } i < j)\big) && \text{by (13)} \\
&\ge q \cdot \sum_{j\in I} H\big(X_j \;\big|\; (X_i : i \in I \text{ with } i < j)\big) && \text{because every } j \text{ appears in at least } q \text{ sets } J \in \mathcal{J} \\
&= q \cdot H(X_I).
\end{aligned}$$

□

2.2 Proof of the Upper Bound

Consider a join query

$$Q = R_1 \bowtie \ldots \bowtie R_m.$$

Suppose that the attributes of Q are $A_1, \ldots, A_n$. For each $i \in [m]$, let J_i be the set of all $j \in [n]$ such that A_j is an attribute of R_i. Let D be a database instance of schema $\{R_1, \ldots, R_m\}$. For all $i \in [m]$, let $N_i := |R_i(D)|$. Let $L(Q, N_1, \ldots, N_m)$ be the linear program

$$\text{minimise} \quad \sum_{i \in [m]} x_i \log N_i, \tag{14}$$

$$\text{where} \quad \sum_{i \in [m] \text{ with } j \in J_i} x_i \geq 1 \qquad \text{for all } j \in [n] \tag{15}$$

$$x_i \geq 0 \qquad \text{for all } i \in [m]. \tag{16}$$

(This is precisely the linear program from the introduction, which we repeat for the reader's convenience.) Let $\boldsymbol{x} = (x_1, \ldots, x_m) \in \mathbb{Q}^m$ be a rational solution to $L(Q, N_1, \ldots, N_m)$. We shall prove that

$$|Q(D)| \leq 2^{\sum_{i \in [m]} x_i \log N_i}. \tag{17}$$

This will imply the upper bound of Theorem 1.

Let $p_1, \ldots, p_m \in \mathbb{N}$ such that $x_i = p_i/q$ for all i. Let $\mathcal{J}$ be a collection of subsets of $[n]$ that contains p_i copies of J_i, for each $i \in [m]$. Then every $j \in [n]$ occurs in at least q sets in $\mathcal{J}$, because

$$\sum_{i \in [m] \text{ with } j \in J_i} p_i = q \cdot \sum_{i \in [m] \text{ with } j \in J_i} x_i \geq q$$

by (15).

Without loss of generality we assume that $Q(D) \neq \emptyset$; otherwise (17) is trivial. Let $X = (X_1, \ldots, X_n)$ be uniformly distributed over $Q(D)$. Then

$$\begin{aligned} \log |Q(D)| &= H(X) \\ &\leq \frac{1}{q} \cdot \sum_{i=1}^{m} p_i \cdot H(X_j \mid j \in J_i) \qquad \text{by Shearer's Lemma} \\ &\leq \sum_{i=1}^{m} x_i \log N_i. \end{aligned}$$

This implies (17). □

Remark 1. The proof of the upper bound of Theorem 1 through Shearer's Lemma is inherently nonconstructive. As a by-product of their algorithm for answering join queries (see Theorem 2), Ngo et al. [7] gave a constructive (but far more complicated) proof of the upper bound.

2.3 LP Duality and Proof of the Lower Bound

Let $Q, R_1, \ldots, R_m, A_1, \ldots, A_n$, and $J_1, \ldots, J_m$ be as in the previous subsection, and let $N_1, \ldots, N_m \in \mathbb{N}^+$ be arbitrary. The *dual* of the linear program $L(Q, N_1, \ldots, N_m)$ is the following linear program $D(Q, N_1, \ldots, N_m)$ in the variables $y_1, \ldots, y_n$.

$$\text{maximise} \quad \sum_{j=1}^{n} y_j, \tag{18}$$

$$\text{where} \quad \sum_{j \in J_i} y_j \leq \log N_i \qquad \text{for all } i \in [m] \tag{19}$$

$$y_j \geq 0 \qquad \text{for all } j \in [n]. \tag{20}$$

By linear programming duality, for all solutions $(x_1, \ldots, x_m)$ to $L(Q, N_1, \ldots, N_m)$ and $(y_1, \ldots, y_n)$ to $D(Q, N_1, \ldots, N_m)$ we have

$$\sum_{i=1}^{m} x_i \log N_i \geq \sum_{j=1}^{n} y_j,$$

with equality if both solutions are optimal.

Now suppose that all the N_i are powers of 2, say, $N_i = 2^{L_i}$ for some $L_i \in \mathbb{N}$. Then all coefficients of $D(Q, N_1, \ldots, N_m)$ are integers, and hence there exists an optimal rational solution. Let $(y_1, \ldots, y_n) \in \mathbb{Q}^n$ be such an optimal solution. Let $p_1, \ldots, p_n, q \in \mathbb{N}$ such that $y_j = p_j/q$. Observe that $(p_1, \ldots, p_n)$ is an optimal solution to the linear program $D(Q, N_1^q, \ldots, N_m^q)$. We shall construct a database instance D with $|R_i(D)| = N_i^q$ and

$$|Q(D)| = 2^{\sum_{j=1}^{n} p_j} = 2^{\rho^*(Q,D)}. \tag{21}$$

This will imply the lower bound of Theorem 1.

To define the instance D, for every $i \in [m]$, we first define a relation $R_i'(D)$ to be the set of all tuples t such that for all $j \in J_i$ the projection $\pi_{A_j}(t)$ is in $[2^{p_j}]$. (So $R_i'(D)$ is the cartesian product of the sets $[2^{p_j}]$, for $j \in J_i$, if we forget about the names of the attributes.) Then

$$|R_i'(D)| = \prod_{j \in J_i} 2^{p_j} = 2^{\sum_{j \in J_i} p_j} \leq 2^{q \log N_i} = N_i^q.$$

We choose $R_i(D) \supseteq R_i'(D)$ with $|R_i(D)| = N_i^q$ arbitrarily. Then $Q(D)$ contains all tuples t such that for all $j \in [n]$ the projection $\pi_{A_j}(t)$ is in $[2^{p_j}]$. Hence

$$|Q(D)| \geq \prod_{j=1}^{n} 2^{p_j} = 2^{\sum_{j=1}^{n} p_j} = 2^{\rho^*(Q,D)}.$$

Actually, we must have equality here because we already know that $|Q(D)| \leq 2^{\rho^*(Q,D)}$. □

Remark 2. It would be nicer if for all $N_1, \ldots, N_q \in \mathbb{N}^+$ we could construct a database instance D with $R_i(D) = N_i$ and $|Q(D)| = 2^{\rho^*(Q,D)}$. We cannot always do that. However, it was proved in [1] that we can always construct an instance D with $R_i(D) = N_i$ and $|Q(D)| \geq 2^{\rho^*(Q,D)-n}$. In general, this is best possible.

2.4 Bounds Depending on the Query Only

The bounds of Theorem 1 depend on the sizes of the individual relations in the database. Obviously, any reasonable estimate on the size of the query answer should depend on the size of the database, but maybe we only have an estimate of the size of the whole database instead of the sizes of the individual relations. In this situation, we can use the database size as an upper bound on the size of all relations.

Observe that if $N_i = N$ for all $i \in [m]$, then an optimal solution $\boldsymbol{x}$ to the linear program $L(Q, N, \ldots, N)$ no longer depends on N (only its value does). We let $L(Q)$ be the linear program obtained from $L(Q, N, \ldots, N)$ by replacing the cost function (14) by $\sum_{i \in [m]} x_i$ and let $\rho^*(Q)$ be the optimal value of this linear program. Then $\rho^*(Q) = \rho^*(Q, D)/\log N$ for all database instances D with $|R_i(D)| = N$ for all $i \in [m]$. Observe that in the dual $D(Q)$ of $L(Q)$ we replace inequalities (19) by $\sum_{j \in J_i} y_j \leq 1$.

Defining the size $||D||$ of a database instance D as the sum $\sum_{i \in [m]} |R_i(D)|$ of the sizes of all relations, we obtain the following corollary to Theorem 1.

Corollary 1. *Let Q be a join query. Then for every database instance D*

$$|Q(D)| \leq ||D||^{\rho^*(Q)}.$$

Furthermore, for every $N \in \mathbb{N}$ there is a database instance D of size $||D|| \geq N$ such that $|Q(D)| \geq (||D||/m)^{\rho^*(Q)}$.

3 Conjunctive Queries

In this section, we extend the bounds of Theorem 1 to conjunctive queries.

3.1 Projections of Join Queries

We start by considering conjunctive queries of the following special form:

$$P(B_1, \ldots, B_k) = \pi_{B_1,\ldots,B_k} Q(A_1, \ldots, A_n), \tag{22}$$

where $Q(A_1, \ldots, A_n)$ is a join query and $B_1, \ldots, B_k \in \{A_1, \ldots, A_n\}$. Here $\pi_{B_1,\ldots,B_k}$ is a projection operator. We allow k to be 0, in which case the query is Boolean.

Consider a conjunctive query P of the form (22). Without loss of generality we assume that $B_j = A_j$ for all $j \in [k]$. Suppose that $Q = R_1 \bowtie \ldots \bowtie R_m$ as before, and let J_i be the set of indices of the attributes of R_i. Let $N_1, \ldots, N_m \in \mathbb{N}^+$. We try to bound the size $P(D)$ of the query answer in a database instance D with $|R_i(D)| = N_i$. We modify our linear programs $L(Q, N_1, \ldots, N_m)$ and $D(Q, N_1, \ldots, N_m)$, essentially ignoring the attributes that are "projected out". The primal linear program $L(P, N_1, \ldots, N_m)$ is defined as follows.

$$\text{minimise} \quad \sum_{i\in[m]} x_i \log N_i, \tag{23}$$

$$\text{where} \quad \sum_{i\in[m] \text{ with } j\in J_i} x_i \geq 1 \qquad \text{for all } j \in [k] \tag{24}$$

$$x_i \geq 0 \qquad \text{for all } i \in [m]. \tag{25}$$

This linear program has the following dual $D(P, N_1, \ldots, N_m)$.

$$\text{maximise} \quad \sum_{j=1}^{k} y_j, \tag{26}$$

$$\text{where} \quad \sum_{j\in J_i\cap[k]} y_j \leq \log N_i \qquad \text{for all } i \in [m] \tag{27}$$

$$y_j \geq 0 \qquad \text{for all } j \in [k]. \tag{28}$$

For a database instance D with $|R_i(D)| = N_i$, let $\rho^*(P, D)$ be the value of the optimal solution to the linear programs. As an easy consequence of Theorem 1, we obtain the following bounds for projections of join queries.

Corollary 2. *Let P be a conjunctive query of the form (22). Then for every database instance D,*

$$|P(D)| \leq 2^{\rho^*(P,D)}.$$

Furthermore, there are arbitrarily large database instances D such that $|P(D)| = 2^{\rho^*(P,D)}$.

Proof. Let P, Q, $R_1, \ldots, R_m$, $A_1, \ldots, A_n$, $B_j = A_j$ for $j \in [k]$, and $J_1, \ldots, J_m$ be as above.

For every $i \in [m]$, let R'_i be a relation schema with attributes A_j for $j \in J_i \cap [k]$, and let $Q' := R'_1 \bowtie \ldots \bowtie R'_m$. For every database instance D of schema $\{R_1, \ldots, R_m\}$, let D' be the instance of schema $\{R'_1, \ldots, R'_m\}$ with $R'_i(D') := \pi_{A_1,\ldots,A_k} R_i(D)$. Then $P(D) \subseteq Q'(D')$. Observe that for all $N_1, \ldots, N_m \in \mathbb{N}^+$ we have $L(Q', N_1, \ldots, N_m) = L(P, N_1, \ldots, N_m)$.

For the upper bound, let D be a database instance with $|R_i(D)| =: N_i$. Then by Theorem 1, we have

$$|P(D)| \leq |Q'(D')| \leq 2^{\rho^*(Q',D')}.$$

Here $\rho^*(Q', D')$ is the optimal value of the linear program $L(Q', N'_1, \ldots, N'_m)$, where $N'_i := |R'_i(D')| \leq N_i$. As the linear programs $L(Q', N'_1, \ldots, N'_m)$ and $L(Q', N_1, \ldots, N_m) = L(P, N_1, \ldots, N_m)$ only differ in their cost functions (23), we have $\rho^*(Q', D') \leq \rho^*(P, D)$.

For the lower bound, we observe that for every database instance D' of schema $\{R'_1, \ldots, R'_m\}$ we can construct an instance D of schema $\{R_1, \ldots, R_m\}$ such that $|P(D)| = |Q'(D')|$ and $|R_i(D)| = |R'_i(D')|$ for all $i \in [m]$. We simply choose a default value, say 1, and extend all tuples $t \in R'_i(D')$ by letting $t(A) := 1$ for all attributes of R_i not in $\{A_1, \ldots, A_k\}$. Then the lower bound follows from the lower bound of Theorem 1. □

Remark 3. As we did in Section 2.4, Gottlob, Lee, and Valiant [5] state their bounds in terms of the query only. For a conjunctive query P as above, they look at the dual linear program $D(P)$:

$$\begin{aligned} \text{maximise} \quad & \sum_{j=1}^{k} y_j, & \\ \text{where} \quad & \sum_{j \in J_i \cap [k]} y_j \leq 1 & \text{for all } i \in [m] \\ & y_j \geq 0 & \text{for all } j \in [k]. \end{aligned}$$

They interpret rational solutions $\boldsymbol{y} = (y_1, \ldots, y_k)$ of $D(P)$ as colourings of the query in the following sense.

A *valid colouring* C of P assigns to each $j \in [k]$ (or to the attribute A_j) a finite set $C(j)$ of colours in such a way that $C(j) \neq \emptyset$ for at least one $j \in [k]$. The *value* of a colouring C is

$$v(C) := \frac{\left|\bigcup_{j \in [k]} C(j)\right|}{\max_{i \in [m]} \left|\bigcup_{j \in J_i \cap [k]} C(j)\right|}.$$

The *colouring number* of P is defined to be the maximum of the values of all its valid colourings. We will see that this maximum always exists and is equal to the value of the linear program $D(P)$.

Let C be a valid colouring of P. Without loss of generality, we may assume that $C(j) \cap C(j') = \emptyset$ for all $j \neq j' \in [k]$, because if a colour appears in $C(j) \cap C(j')$ then dropping this colour from one of the sets gives a colouring with the same or a better value. Let $q := \max_{i \in [m]} \left|\bigcup_{j \in J_i \cap [k]} C(j)\right|$. For $j \in [k]$, let $p_j := |C(j)|$ and $y_j := p_j/q$. Then $(y_1, \ldots, y_k)$ is a solution of $D(Q)$ of value $\sum_{j=1}^{k} y_j = v(C)$.

Conversely, let $(y_1, \ldots, y_k) \in \mathbb{Q}^k$ be a rational solution of $D(Q)$ such that $\sum_{j \in J_i \cap [k]} = 1$ for some $i \in [m]$. Clearly, an optimal solution has this property. Suppose that $y_j = p_j/q$ for all $j \in [k]$. We define a valid colouring by letting $C(j)$ be a set of p_j fresh colours (so that $C(1), \ldots, C(k)$ are mutually disjoint). Then $v(C) = \sum_{j=1}^{k} y_j$.

As $D(Q)$ has integer coefficients, there is an optimal rational solution, and it yields a colouring of optimal value.

Th purpose of viewing solutions of the dual linear program as colourings in this way is that it yields a natural extension to the setting with functional dependencies.

3.2 Arbitrary Conjunctive Queries

We view a general conjunctive query as an expression

$$C(\bar{X}) \leftarrow R_1(\bar{X}_1), \ldots, R_m(\bar{X}_m), \tag{29}$$

where $R_1, \ldots, R_m$ are (not necessarily distinct) relation names of arities $k_1, \ldots, k_m$, respectively, and $\bar{X}, \bar{X}_i$ are tuples of not necessarily distinct variables of lengths $k, k_1, \ldots, k_m$, respectively, such that all variables appearing in $\bar{X}$ also appear in some $\bar{X}_i$. Without loss of generality we assume that the set of all variables appearing in the query C is $\{X_1, \ldots, X_n\}$, where $X_1, \ldots, X_n$ are pairwise distinct, and that $\bar{X} = (X_1, \ldots, X_k)$. (We can do this because it does not affect the size of the query answer if we repeat variables in the head $C(\bar{X})$.) For each $i \in [m]$, let $J_i \subseteq [n]$ be the set of indices of the variables appearing in $\bar{X}_i$.

For a database instance D of schema $\{R_1, \dots, R_m\}$, the query answer $C(D)$ is the set of all k-tuples $(a_1, \dots, a_k)$ that can be extended to an n-tuple $\bar{a} = (a_1, \dots, a_n)$ such that for each $i \in [m]$ the projection of $\bar{a}$ to the indices of the variables in $\bar{X}_i$ (in the right order) is in $R_i(D)$.

Note that the variables play the role the attributes played so far. Projections of join queries, as considered in the previous subsection, correspond to the case that the relations $R_1, \dots, R_m$ are pairwise distinct and that for each $i \in [m]$ the variables appearing in $\bar{X}_i$ are pairwise distinct.

For all $N_1, \dots, N_m \in \mathbb{N}^+$, we define the following linear program $L(C, N_1, \dots, N_m)$.

$$\text{minimise} \quad \sum_{i \in [m]} x_i \log N_i, \tag{30}$$

$$\text{where} \quad \sum_{i \in [m] \text{ with } j \in J_i} x_i \geq 1 \qquad \text{for all } j \in [k] \tag{31}$$

$$x_i \geq 0 \qquad \text{for all } i \in [m]. \tag{32}$$

Note that if the relations $R_1, \dots, R_m$ are pairwise distinct and for each $i \in [m]$ the variables appearing in $\bar{X}_i$ are pairwise distinct, then this is precisely the linear program (23)–(25) defined in the previous subsection.

We let $\rho^*(C, N_1, \dots, N_m)$ be the value of an optimal solution of $L(C, N_1, \dots, N_m)$. The following lemma is an easy consequence of Corollary 2.

Lemma 2. *Let C be a conjunctive query of the form (29) such that $R_1, \dots, R_m$ are pairwise distinct. Then for all $N_1, \dots, N_m \in \mathbb{N}^+$ and all database instances D of schema $\{R_1, \dots, R_m\}$ with $|R_i(D)| \leq N_i$,*

$$|C(D)| \leq 2^{\rho^*(C, N_1, \dots, N_m)}.$$

Furthermore, there are arbitrarily large $N_1, \dots, N_m \in \mathbb{N}^+$ and databases instances D with $|R_i(D)| = N_i$ such that $|C(D)| = 2^{\rho^*(C, N_1, \dots, N_m)}$.

Proof. For every $i \in [m]$, let $\ell_i := |J_i|$. Then obviously we have $\ell_i \leq k_i$, where equality holds if the variables in $\bar{X}_i$ are distinct. For each $i \in [m]$, we let $R_i^{\neq}$ be a fresh ℓ_i-ary relation symbol. We let $\bar{X}_i^{\neq}$ be the ℓ_i-tuple of variables that contains the same variables as $\bar{X}_i$, but without repetitions, in the order of their first appearance in $\bar{X}_i$. We let

$$C^{\neq}(\bar{X}) \leftarrow R_1^{\neq}(\bar{X}_1^{\neq}), \dots, R_m^{\neq}(\bar{X}_m^{\neq}),$$

Then $C^{\neq}(\bar{X})$ is a projection of a join query of the type considered in the previous subsection.

For every database instance D of schema $\{R_1, \dots, R_m\}$ we define an instance $D^{\neq}$ of schema $\{R_1^{\neq}, \dots, R_m^{\neq}\}$ by letting $R_i^{\neq}(D)$ be the set of all $\bar{X}_i^{\neq}$-tuples obtained from $\bar{X}_i$-tuples in $R_i(D)$. To make this precise, suppose that $\bar{X}_i = (X(i,1), \dots, X(i,k_i))$, where of course each $X(i,j)$ is an element of $\{X_1, \dots, X_n\}$. Let $J(1), \dots, J(\ell_i)$ be the partition of $[k_i]$ such that $X(i,j) = X(i,j')$ for all $p \in [\ell_i]$ and $j, j' \in J(p)$, and $X(i,j) \neq X(i,j')$ for all $p \neq p' \in [\ell_i]$ and $j \in J(p), j' \in J(p')$. For each $p \in [\ell_i]$, let $j(p)$ be the minimum of $J(p)$. Without loss of generality we assume that $j(1) < j(2) < \ldots < j(\ell_i)$. Then $\bar{X}_i^{\neq} = (X(i, j(1)), \dots, X(i, j(\ell_i)))$. Now we let $R_i^{\neq}(D)$ be the set of all ℓ_i-tuples $(a^{\neq}(1), \dots, a^{\neq}(\ell_i))$ such that there is a k_i-tuple $(a(1), \dots, a(k_i)) \in R_i(D)$ with $a(j) = a^{\neq}(p)$ for all $p \in [\ell_i]$, $j \in J(p)$. It is immediate from the definitions that $C(D) = C^{\neq}(D^{\neq})$ and $|R_i(D)| \geq |R_i^{\neq}(D)|$.

Now the upper bound follows from Corollary 2 and the observation that

$$L(C, N_1, \dots, N_m) = L(C^{\neq}, N_1, \dots, N_m).$$

To prove the lower bound, we start with an instance D' of schema $\{R_1^{\neq}, \dots, R_m^{\neq}\}$ with sufficiently large $N_i := |R_i^{\neq}(D')|$ such that

$$|C^{\neq}(D')| = 2^{\rho^*(C^{\neq}, D')} = 2^{\rho^*(C^{\neq}, N_1, \dots, N_m)} = 2^{\rho^*(C, N_1, \dots, N_m)}.$$

It is easy to construct an instance D of schema $\{R_1, \dots, R_m\}$ such that $D' = D^{\neq}$ and $|R_i(D)| = |R_i^{\neq}(D')| = N_i$ for all $i \in [m]$. Then we have $|C(D)| = |C^{\neq}(D')| = 2^{\rho^*(C, N_1, \dots, N_m)}$. ◻

For general conjunctive queries, we obtain a slightly weaker lower bound that takes into account the multiplicities with which the relations appear in the query: the inequalities $|R_i(D)| \leq N_i$ are replaced by (33).

Theorem 3 (Gottlob, Lee, Valiant [5]). *Let C be a conjunctive query of the form (29). Then for all $N_1, \ldots, N_m \in \mathbb{N}^+$ and all database instances D of schema $\{R_1, \ldots, R_m\}$ with $|R_i(D)| \leq N_i$,*

$$|C(D)| \leq 2^{\rho^*(C, N_1, \ldots, N_m)}.$$

Furthermore, there are arbitrarily large $N_1, \ldots, N_m \in \mathbb{N}^+$ and databases instances D with

$$|R_i(D)| \leq \sum_{\substack{j \in [m] \\ \text{with } R_j = R_i}} N_j \tag{33}$$

for all $i \in [m]$ such that $|C(D)| \geq 2^{\rho^*(C, N_1, \ldots, N_m)}$.

Proof. For every $i \in [m]$, let R'_i be a fresh k_i-ary relation name, and let

$$C'(\bar{X}) \leftarrow R'_1(\bar{X}_i), \ldots, R'_m(\bar{X}_m).$$

Observe that for all $N_1, \ldots, N_m \in \mathbb{N}^+$ we have $L(C, N_1, \ldots, N_m) = L(C', N_1, \ldots, N_m)$.

For every database instance D of schema $\{R_1, \ldots, R_m\}$, let D' be the instance of schema $\{R'_1, \ldots, R'_m\}$ with $R'_i(D') := R_i(D)$. Then $C(D) = C'(D')$. As all relations appearing in C' are distinct, the upper bound follows directly from Lemma 2.

To prove the lower bound, we choose a database instance D' of schema $\{R'_1, \ldots, R'_m\}$ for sufficiently large $N_i := |R'_i(D')|$ such that

$$|C'(D')| = 2^{\rho^*(C', N_1, \ldots, N_m)} = 2^{\rho^*(C, N_1, \ldots, N_m)}.$$

Such a D' exists by Lemma 2. We define an instance $D_\cup$ of schema $\{R_1, \ldots, R_m\}$ by letting

$$R_i(D_\cup) := \bigcup_{\substack{j \in [m] \\ \text{with } R_j = R_i}} R'_j(D').$$

Obviously, $D_\cup$ satisfies (33), and we have $C(D_\cup) \supseteq C'(D')$ and hence $|C(D_\cup)| \geq |C'(D')| \geq 2^{\rho^*(C, N_1, \ldots, N_m)}$. □

As for join queries (see Corollary 1), it is easy to formulate a version of the theorem where the bounds only depend on the size of the database instance and the query and not on the sizes of the individual relations. This is how Gottlob et al. [5] originally phrased their theorem.

4 Query Plans

A *query plan* for a query Q is an expression φ in the relational algebra, using (binary) join operators, projection operators, and possibly other relational algebra operators, such that $Q(D) = \varphi(D)$ for every database instance D. A query plan is a *join plan* if the only operator it uses is the join operator, and it is a *join-project plan* if it only uses joins and projections. A *subplan* of a query plan is defined in the natural way; all nodes of the parse tree of the plan correspond to subplans. *Executing* a query plan φ in a database instance means computing $\psi(D)$ for all subplans ψ, until finally $\varphi(D)$ is computed. As all relational algebra operations can be implemented with a running time linear in the size of the input(s) plus the output, the time it takes to execute a query plan is linear in the size of the maximal intermediate result.

Obviously, for every join query there is a join plan. However, it may happen that the intermediate results of the executions of all possible join plans for a query are substantially larger than the final result.

Example 3. Recall the query $Q := R \bowtie S \bowtie T$ of Example 1. Every join plan for Q contains either $R \bowtie S$ or $R \bowtie T$ or $S \bowtie T$ as a subplan, and in a database instance D with $R(D) = S(D) = T(D) = N$ it may happen that $|(R \bowtie S)(D)|, |(R \bowtie T)(D)|, |(S \bowtie T)(D)| = \Theta(N^2)$, whereas we have seen in Example 1 that $|Q(D)| \leq N^{3/2}$.

The query of Example 2 is underlying the following theorem.

Theorem 4 ([1]). *There are arbitrarily large join queries Q and databases D such that $\rho^*(Q) \leq 2$ and every join plan for Q has a subplan ψ with $|\psi(D)| \geq ||D||^{\Omega(\log |Q|)}$.*

Thus join plans may lead to relatively inefficient algorithms for answering join queries. Join-project plans, on the other hand, are almost optimal.

Theorem 5 ([1]). *For every join query Q there is a join-project plan φ such that for every subplan ψ of φ and a every database instance D,*

$$|\psi(D)| \leq 2^{\rho^*(Q,D)} \cdot M,$$

where M is the maximum size of the projection of a relation in D to a single attribute.

Proof. Let $Q = R_1 \bowtie \ldots \bowtie R_m$. As always, suppose that the attributes of Q are $A_1, \ldots, A_n$, and let J_i be the set of indices of the attributes of R_i. The idea is to evaluate Q by iteratively computing the projections $\pi_{A_1,\ldots,A_j} Q(D)$, for $j = 1, \ldots, n$.

We let

$$\varphi_1 := \Big(\cdots \big(\big(\pi_{A_1}(R_1) \bowtie \varphi_{A_1}(R_2) \big) \bowtie \pi_{A_1}(R_3) \big) \bowtie \ldots \bowtie \pi_{A_1}(R_m) \Big),$$

and for $j = 1, \ldots, n-1$,

$$\varphi_{j+1} := \Big(\cdots \big(\big(\varphi_j \bowtie \pi_{A_1,\ldots,A_{j+1}}(R_1) \big) \bowtie \pi_{A_1,\ldots,A_{j+1}}(R_2) \big) \bowtie \ldots \bowtie \pi_{A_1,\ldots,A_{j+1}}(R_m) \Big).$$

Either by a direct argument or by an application of the upper bound of Corollary 2, it is easy to see that the plan $\varphi := \varphi_n$ has the desired properties. The crucial observation is that

$$\big| \big(\varphi_j \bowtie \pi_{A_1,\ldots,A_{j+1}}(R_1) \big)(D) \big| \leq \big| \varphi_j(D) \times \pi_{A_{j+1}} R_1(D) \big| \leq 2^{\rho^*(Q,D)} \cdot M,$$

because $\pi_{A_{j+1}} R_1(D) \leq M$ and for the conjunctive query $P_j := \pi_{A_1,\ldots,A_j} R_1 \bowtie \ldots \bowtie R_m$, for which φ_j is a query plan, we have $\rho^*(P_j, D) \leq \rho^*(Q, D)$ and thus $|\varphi_j(D)| = |P_j(D)| \leq 2^{\rho^*(P_j,D)} \leq 2^{\rho^*(Q,D)}$ by Corollary 2. □

The execution of the join-project plan of Theorem 5 leads to an algorithm answering Q in time $O(2^{\rho^*(Q,D)} \cdot M)$ (ignoring a small polynomial factor depending on Q). We remind the reader of the algorithm of Theorem 2, which avoids the factor M in the running time. Ngo et al. [7] showed that a running time of $O(2^{\rho^*(Q,D)})$ cannot be achieved with the execution of a join-project plan.

References

1. A. Atserias, M. Grohe, and D. Marx. Size bounds and query plans for relational joins. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 739–748, 2008.
2. N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on Computing*, 14:210–223, 1985.
3. F. Chung, P. Frank, R. Graham, and J. Shearer. Some intersection theorems for ordered sets and graphs. *Journal of Combinatorial Theory, Series A*, 43:23–37, 1986.
4. G. Gottlob, S.T. Lee, G. Valiant, and P. Valiant. Size and treewidth bounds for conjunctive queries. *Journal of the ACM*. To appear.

5. G. Gottlob, S.T. Lee, and G.J. Valiant. Size and treewidth bounds for conjunctive queries. In *Proceedings of the 28th ACM Symposium on Principles of Database Systems*, pages 45–54, 2009.
6. M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 289–298, 2006.
7. H.Q. Ngo, E. Porat, C. Ré, and A. Rudra. Worst-case optimal join algorithms. In *Proceedings of the 31st ACM Symposium on Principles of Database Systems*, 2012.
8. J. Radhakrishnan. Entropy and counting. In J.C. Misra, editor, *Computational Mathematics, Modelling and Algorithms*. Narosa Pub House, 2003.
9. C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, 1948.
10. G. Valiant and P. Valiant. Size bounds for conjunctive queries with general functional dependencies. *Arxiv preprint arXiv:0909.2030*, 2009.