Query Evaluation on a Database Given by a Random Graph

Nilesh Dalvi *

University of Washington, Seattle.

Abstract. We consider random graphs, and their extensions to random structures, with edge probabilities of the form $\beta n^{-\alpha}$, where n is the number of vertices, α, β are fixed and $\alpha > 1$ ($\alpha >$ arity $- 1$ for structures of higher arity). We consider conjunctive properties over these random graphs, and investigate the problem of computing their asymptotic conditional probabilities. This provides us a novel approach to dealing with uncertainty in databases, with applications to data privacy and other database problems.

1 Introduction

Let $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ be a vocabulary of relation symbols, and $A(R_i)$ denote the arity of relation R_i. We fix two functions $\alpha, \beta : \mathcal{R} \rightarrow \mathbb{R}^+$ that map relations to positive real numbers, where α satisfies $\alpha(R_i) > A(R_i) - 1$ for all $R_i \in \mathcal{R}$. For any $n > 0$, denote μ_n the probability distribution over the structures with domain $[n]$ given by the following experiment: for each $R_i \in \mathcal{R}$ and $t \in [n]^{A(R_i)}$, choose t to be in R_i with probability $\beta(R_i)n^{-\alpha(R_i)}$. Let $CQ(c, \neq)$ denote the class of boolean conjunctive properties (queries) with constants (c) and inequalities ($\neq$), i.e. formulas of the form: $\exists x_1 \exists x_2 ... (A_1 \wedge A_2 \wedge \cdots \wedge A_k)$ where each A_i is a predicate of the form $R_i(y_1, \cdots, y_j)$, or $y_1 \neq y_2$, where $y_1, \ldots, y_j$ denote variables or constants. $CQ(c)$, $CQ(\neq)$, and CQ denote the classes of conjunctive queries without $\neq$, without constants, and without either $\neq$ or constants respectively.

In this paper, we study the probabilities of conjunctive properties over the class μ_n of random structures. Lynch [17] has shown that if we consider the set of first order properties, of which conjunctive properties are a subset, then for each property q, $\mu_n(q)$ is either $cn^{-d} + o(n^{-d})$ for some $c > 0, d \geq 0$, or is $e^{-\Omega(n^k)}$ for some $k > 0$. Thus, the probability of every first order property is either asymptotically equal to a polynomial in $1/n$ or is exponentially small. Further, the problem of determining which of the two cases hold for a given query is undecidable.

Here we show that when q is a conjunctive query, then we always have $\mu_n(q) = cn^{-d} + o(n^{-d})$. A consequence is that for any two conjunctive queries q, v, the quantity $\mu(q \mid v) = \lim_{n \rightarrow \infty} \mu_n(q \mid v) = \lim_n \mu_n(q \wedge v)/\mu_n(v)$ exists and is a real

* **Acknowledgments** This work was partially supported by the grants NSF SEIII 0513877, NSF 61-2252, and NSF IIS-0428168.

Problem	Combined Complexity (inputs: q, v, v')	Data Complexity (inputs: v, v'; fixed: q)
Compute $\mu(q \mid v)$	#coNP-complete	#coNP-complete
Decide if $\mu(q \mid v) = 0$	Θ_2^p-complete	Θ_2^p-complete
Decide if $\mu(q \mid v) \in (0, 1)$	Σ_2^p-complete	Θ_2^p-complete
Decide if $\mu(q \mid v) = 1$	Π_2^p-complete	Θ_2^p-complete
Decide if $\mu(q \mid v) < p$ for $p \in (0, 1)$	P.coNP-complete	P.coNP-complete
Decide $\mu(q \mid v) < \mu(q \mid v')$	P.coNP-complete	P.coNP-complete
Decide $\mu(q \mid v) < \mu(q \mid vv')$	P.coNP-complete.	P.coNP-complete

Fig. 1. Summary of Main Results.

number in $[0, 1]$. In this paper we investigate the complexity of computing this limit, in various settings, and the main result is below.

Theorem 1. *Fig. 1 shows the complexities of various decision problems and computation problems concerning their asymptotic conditional probabilities. Upper bounds hold for $q, v, v' \in CQ(c, \neq)$; lower bounds hold for both $q, v, v' \in CQ(c)$ and for $q, v, v' \in CQ(\neq)$. For $q, v, v' \in CQ$ all problems are in PTIME.*

1.1 Motivation

Our motivation comes from the following problem in databases: evaluate a property q of an unknown database instance I given some facts v about the instance. The problem appears in a wide range of applications, for instance in *data privacy* [8] where we want to analyze a sensitive query using published facts, in *data integration* [12, 15] where we want to answer queries using views and in *cardinality estimation* [10, 2] where we want to estimate the size of a query using known statistics about the data. In many applications, the standard approach is to use the notion of *certain answers* [12], where the property q is said to be *certain* if it is true on every possible instance I that is consistent with the facts v. However, this approach has two limitations that make it unsuitable for certain applications. First, it does not revel anything about tuples that are not certain answers to the query, whereas in applications like data privacy, we are interested in knowing which tuples are more (or less) likely to be the query answers given the facts in v. The second limitation of the approach is that it cannot incorporate any knowledge about the relative likelihood of possible instances. Applications often have auxiliary information, like statistical knowledge, that makes certain instances more likely than others. In data privacy setting, it is important to take the auxiliary information into account; in cardinality estimation this is often the only kind of information available.

Example 1. (K-Anonymity) Suppose a medical agency wants to publish its data patients data for research purposes, but wants to protect the identity of individual patients. The data is in the table `Patients`(`name`, `age`, `zipcode`, `disease`).

The agency publishes the following view:

$$\texttt{Diseases}(\texttt{age}, \texttt{zipcode}, \texttt{disease}) :- \texttt{Patients}(\texttt{name}, \texttt{age}, \texttt{zipcode}, \texttt{disease})$$

In addition, suppose the following view is publically known:

$$\texttt{Name}(\texttt{name}, \texttt{age}, \texttt{zipcode}) :- \texttt{Patients}(\texttt{name}, \texttt{age}, \texttt{zipcode}, \texttt{disease})$$

The contents of the views are given below. Note that some of the values in the view are partially hidden.

Names=

name	age	zipcode
JOHN	25	98190
MARY	25	98192
MARK	31	98100
FRANK	31	98111
LARRY	32	98100

Diseases=

age	zipcode	disease
25	9819*	FLU
25	9819*	FEVER
31	981**	FLU
**	*****	CANCER

We want to analyze the information given by $v = \{Diseases, Names\}$ about the following query $q(name, disease) :- \texttt{Patients}(name, age, zipcode, disease)$. The `Diseases` table is *2-anonymous* [19], meaning that for every tuple in the `Diseases` table, there are at least two individuals that may have that record. For instance, the tuple *(25, 98192, FLU)* could either refer to *JOHN* or *MARY* or a third person that does not appear in the views. There are no certain answers to the query.

The technique of k-anonymization guarantees that each record in the published data refers to at least k individuals. However, Machanavajjhala et al. [19] have shown that k-anonymization does not guarantee data privacy when the attacker has auxiliary/background knowledge about the data, and they raise the problem of analyzing data privacy in the presence of such information. Examples of background information are: (i) every (`age`, `zipcode`) occurs four times in expectation and (ii) around 80% of all the patients have *FLU*. We need a framework where we can use such statistics on the data and evaluate the likelihood that a particular tuple is the answer to the query given the views.

Bayesian Approach In order to provide such a framework, we consider an alternative approach to the problem using a technique from knowledge representation based on *degrees of beliefs* [3]. Here the uncertainty about the underlying database is expressed as a probability distribution $\mathbf{P}$, called the *prior probability distribution*, or simply the *prior*, which is an assessment of the likelihood of each data instance to occur before observing any facts about the database. Starting from this prior distribution, any subsequent knowledge v about the database is encapsulated by conditioning $\mathbf{P}$ on v. Thus, the probability that the database satisfies a property q is given by the conditional probability $\mathbf{P}(q \mid v)$. Query evaluation thus amounts to computing the conditional probabilities on the prior.

Example 2. Let us revisit Example 1 where we have the relation `Patients`(`name`, `age`, `zipcode`, `disease`), where each attribute takes value from a domain[1] of size

[1] Assume for simplicity that all attributes take values from the same domain.

n. Suppose that the only background knowledge we have is that the expected size of the table is 100. Consider the following probability distribution: there are n^4 tuples that are possible over the domain, pick each of them randomly and independently with probability $100/n^4$. The resulting distribution is a sparse random structure with each tuple having probability $p(n) = 100/n^4$. Further, it is easy to see that the expected size of `Patients` under this distribution is 100.

Now suppose we want to check if $q \equiv \texttt{Patients}(JOHN, 25, 98190, CANCER)$ is true given the views $v = \{\texttt{Names}, \texttt{Patients}\}$. We compute the probability that q is true given v, i.e. the quantity $\mathbf{P}(q \mid v)$. It can be shown that $\mathbf{P}(q \mid v) = 1/101 + O(1/n)$. Intuitively, there are 100 records expected in the database, so there is around 1/100 chance that the facts $\texttt{Names}(JOHN, 25, 98190)$ and $\texttt{Diseases}(**, *****, CANCER)$ talk about the same tuple.

Using a Sparse Random Graph as a Prior Choosing a suitable prior distribution is an important step in this analysis, and the problem is well studied in the area of Knowledge Representation [3]. In our previous work [6], we look at this problem from a database perspective. We consider a framework for specifying prior knowledge about the database that allows statistics like the expected size of a relation, the expected number of distinct values of an attribute and integrity constraints like functional dependencies and inclusion dependencies. Under this framework, we describe how to represent such prior knowledge as a probability distribution using a technique from Knowledge Representation called *entropy maximization*. Applying this technique to Example 2, with the statistics that the expected size of `Patients` relation is 100, we get exactly the probability distribution where each tuple is chosen independently with probability $100/n^4$, which is a sparse random structure.

Other Prior Distributions The probability distribution in Example 2 belongs to a special class of sparse random structures where $p_{R_i(n)} = c_i n^{-Arity(R_i)}$ for each R_i. We studies these distributions in one of our previous works [5]. One of the properties of these distributions is that with high probability, all the tuples in the structure are disjoint. For instance, in Example 2, all the `zipcode` values in the `Patients` tables are distinct with high probability. If we want to incorporate the knowledge that each `zipcode` is expected to have 10 records, we can add this statistics in our framework to obtain a new prior distribution. In this new distribution, the tuples are not independent since each `zipcode` value occurs 10 times in expectation, hence the new distribution will not be a random structure. In [6], we show that statistics like these can be captured using a generalization of random structures. For lack of space, we only describe results in this paper for random structures, but the results also extend to these generalized random structures described in [6].

Domain Size Given a sparse random structure with probability distribution μ_n, and conjunctive properties q and v, we seek the conditional probability $\mu_n(q \mid v)$. In general, we do not know the domain size, n. But the domain is usually large. Hence, we study the behavior of conditional probability for large n by looking at the limit $\lim_{n\to\infty} \mu_n(q \mid v)$, which we denote $\mu(q \mid v)$. Below, we

describe some specific problems related to the computation of $\mu(q \mid v)$ motivated by the data privacy application.

Data Privacy Here, the owner of a database wishes to publish certain facts about a private database, while keeping certain sensitive information hidden. There are two basic problems. The first is *leakage*: does the view v leak information about a sensitive property s ? Various authors [8, 20, 14] have modeled non-leakage by requiring the *a priori* probability of s to be close to the *a posteriori* probability after seeing v, i.e. $\mathbf{P}(s) \approx \mathbf{P}(s \mid v)$. We make this precise by requiring $\mu(s \mid v) = \mu(s)$, which amounts to $\mu(s \mid v) = 0$ since, as we show, $\mu(s) = 0$ for all practical queries. The second problem is *usage*: a legitimate user wants to check a property q over the data, by examining v, and this amounts to checking $\mu(q \mid v) = 1$. In addition to these basic questions, we consider two more complex questions motivated by real application scenarios. In *collusion detection* we know that $\mu(s \mid v) = \mu(s \mid v') = 0$ and have to decide if $\mu(s \mid v, v') = 0$. In *relative security* the data owner has already published some view v, possibly leaking some information about the secret s: the damage cannot be undone, but the data owner wants to publish a second view v' and wants to know if there any additional leakage, i.e. $\mu(s \mid v, v') > \mu(s \mid v)$?

Query Evaluation Vardi [23] has studied the query evaluation problem in databases: given I, q, decide if $I \models q$. In the *combined complexity* both I and q vary, while in the *data complexity*, q is fixed and I varies. The problem we investigate in this paper is related to query evaluation: evaluate q on the observations v, i.e. compute $\mu(q \mid v)$. The database is given by v, with the unknown part filled in by the random graph. Data complexity corresponds here to a fixed q and a variable v.

1.2 Related Work

The study of convergence laws for logical statements on random graphs has been a widely explored areas of model theory. Fagin [9] and Glebskiĭ et al. [11] considered random graphs with $p(n)$ a constant and proved a 0-1 law for statements of first order logic. However, asymptotic limits for *conditional* probabilities do not always exist [9] for this class of graphs, and even the problem of determining if they exist is undecidable [16]. The class of random graphs with edge probabilities of the form $\beta n^{-\alpha}$ with $\alpha > 1$ have also been studied before and there are results on the existence of asymptotic probabilities for statements of first order logic [21, 17, 18]. However, existence of asymptotic conditional probabilities, and the complexity of computing them, has not been studies previously. The applications of sparse random graphs and their generalizations [5, 6] have been discussed before, but again these works did not study the complexity of query evaluation.

2 Computing Asymptotic Probabilities

For a conjunctive query q given by Eq.(1) let $goals(q)$ denote the set of its relational predicates (i.e. of the form $R_i(y_1, \cdots, y_j)$, and called "subgoals"),

$Var(q) = \{x_1, x_2, \ldots\}$ the set of variables, $Const(q)$ the set of constants mentioned in q. Here we show that the asymptotic conditional probabilities always exist for conjunctive queries and are computable. The basic result is:

Theorem 2. *For any conjunctive query $q \in CQ(c, \neq)$, there exists two constants* $\text{coeff}(q)$ *and* $\exp(q)$*, such that*

$$\mu_n(q) = \text{coeff}(q)(1/n)^{\exp(q)} + o((1/n)^{\exp(q)})$$

Corollary 1. *For $q \in CQ(c, \neq)$, the asymptotic probability $\mu(q)$ always exists. It equals* $\text{coeff}(q)$ *if* $\exp(q) = 0$ *and 0 otherwise.*

Corollary 2. *For $q_1, q_2 \in CQ(c, \neq)$, the conditional asymptotic probability, $\mu(q_1 \mid q_2)$, always exists and is as follows:*

$$\mu(q_1 \mid q_2) = \begin{cases} 0 & \exp(q_1 q_2) > \exp(q_2) \\ \frac{\text{coeff}(q_1 q_2)}{\text{coeff}(q_2)} & \exp(q_1 q_2) = \exp(q_2) \end{cases}$$

In the remainder of the section, we show how to compute *coeff*(q) and *exp*(q). For a subgoal $g \in goals(q)$, let $\alpha(g)$ and $\beta(g)$ denote $\alpha(R)$ and $\beta(R)$ where R is the relation to which g refers. Define

$$\begin{aligned} V(q) &= \text{the number of distinct variables in } q \\ \alpha(q) &= \sum\{\alpha(g) \mid g \in goals(q)\} \\ \beta(q) &= \prod\{\beta(g) \mid g \in goals(q)\} \\ D(q) &= \alpha(q) - V(q) \end{aligned}$$

A *substitution* η for a query q is a mapping $\eta : Var(q) \to Var(q) \cup Const(q)$ that does not violate the inequalities in q. We denote $\eta(q)$ the result of applying η to the subgoals of q. For example, if $q \leftarrow R(a, x), R(x, y), R(y, z), x \neq y$ then the substitution $\eta = \{x \to b, y \to y, z \to y\}$ is defined on q and by applying it we obtain the query $q_0 = \eta(q)$, $q_0 \leftarrow R(a, b), R(b, y), R(y, y), b \neq y$. If $\eta(q)$ results in duplicate subgoals, we remove duplicates[2]. We call a query of the form $\eta(q)$ a *unifying query* for q, since it unifies some of the subgoals in q, and denote with $UQ(q)$ the set of all unifying queries for q up to isomorphism. Let P be the partition on $goals(q)$ induced by η, where two subgoals g, g' are in the same equivalence class if $\eta(g) = \eta(g')$. Call η a *most general unifier* if, for any other unifier η' inducing P there exists a substitution θ s.t. $\eta' = \theta \circ \eta$. In this case we call $\eta(q)$ a *most general unifying query* of q, and we define $\text{MGUQ}(q)$ the set of most general unifying queries of q (up to isomorphism). Define:

$$\begin{aligned} E(q) &= \min\{D(q_0) \mid q_0 \in \text{MGUQ}(q)\} \\ \text{MGUQ}^0(q) &= \{q_0 \mid q_0 \in \text{MGUQ}(q), D(q_0) = E(q)\} \\ aut(q) &= |\{\eta \mid \eta(q) \text{ is isomorphic to } q\}| \end{aligned}$$

[2] While this sounds evident, we insist on it because the functions $D(-)$ and $\beta(-)$ return different (and wrong) results if we fail to eliminate duplicates.

We view a subset $q_1 \subseteq goals(q)$ as another conjunctive query, $q_1 = \exists x_1 \exists x_2 \ldots (\bigwedge_{g \in q_1} g)$. Construct a graph whose nodes are $goals(q)$ and edges are pairs of sub-goals that share a variable, and consider all queries $q_1, q_2, \ldots$ given by its connected components. We write $q_i \simeq q_j$ to denote that the queries q_i and q_j are isomorphic, and let k be the number of distinct isomorphism types among queries q_i with $D(q_i) = 0$. Define the following partition on $goals(q)$ into $k+2$ classes:

- $q_N = \bigcup\{q_i \mid D(q_i) < 0\}$.
- $q_P = \bigcup\{q_i \mid D(q_i) > 0\}$.
- $r_1^{t_1}, \cdots, r_k^{t_k}$, where $D(r_i) = 0$, $r_i^{t_i} = \bigcup\{q_j \mid q_j \simeq r_i\}$, $t_i = \mid \{q_j \mid q_j \simeq r_i\} \mid$.

We call q_P the *kernel* of q and we call $r_1^{t_1}, \cdots, r_k^{t_k}$ the *zero subgoals* or q.

Let $\Gamma(x, m) = \sum_{j=1}^{m} x^j e^{-m}/j$ be the Poisson distribution function. Given a query q_s of the form $r_1^{t_1}, \cdots, r_k^{t_k}$, define

$$f(q_S) = \prod_{i=1}^{k} (1 - \Gamma(\frac{\beta(r_i)}{aut(r_i)}, t_i))$$

If q_S is empty, then $f(q_S) = 1$. Given a set Q_s, where each element is a query of the form q_S, define

$$F(Q_s) = \sum_{S \subseteq Q_s} (-1)^{|S|} f(\wedge_{q_s \in S} f(q_s)$$

Now we are ready to describe the *coeff*(q). Group the queries in $\text{MGUQ}^0(q)$ so that all queries that have the same kernel are in one group. Define a pair (q_P, Q_s) for each group where q_P is the kernel, and Q_s is the set consisting of sets of zero sub-goals of all queries in the group. Let $G(q)$ be the set of such pairs and let

$$C(q) = \sum_{(q_P, Q_s) \in G(q)} \frac{\beta(q_P)}{aut(q_P)} F(Q_s)$$

Theorem 3. *For any* $q \in CQ(c, \neq)$, $\exp(q) = max(E(q), 0)$ *and* $\text{coeff}(q) = C(q)$.

Example 3. For a simple illustration, consider the following query q. Assume $\alpha(R) = 4$.

$$\exists x \exists y \exists z \exists u \exists v. R(a, x, y, c), R(z, z, u, c) R(v, a, b, c), y \neq b$$

Here a, b, c are constants. Define:

$$q_1 \equiv \exists y \exists v. R(a, a, y, c), R(v, a, b, c), y \neq b$$
$$q_2 \equiv \exists x \exists y. R(a, x, y, c), R(a, a, b, c), y \neq b$$

Then $\text{MGUQ} = \{q, q_1, q_2\}$, $D(q) = 7$, $D(q_1) = D(q_2) = 6$, hence $exp(q) = 6$, $\text{MGUQ}^0 = \{q_1, q_2\}$, $G(q) = \{(q_1, \emptyset), (q_2, \emptyset)\}$, $aut(q_1) = aut(q_2) = 1$ and $\beta(q) = 2\beta^2(R)$. Thus: $\mu_n(q) = 2\beta^2(R)/n^6 + o(1/n^6)$

3 Complexity Results

We state, explain, and expand here our complexity results that were briefly mentioned in Th. 1.

3.1 Computing *coeff* and *exp*

A direct application of the definitions for $exp(q)$ and $coeff(q)$ leads to an exponential time algorithm. The following gives a tight bound on their complexity:

Theorem 4. $\forall \mathcal{C} \in \{CQ(c), CQ(\neq), CQ(c, \neq)\}$

1. *The problem: given $q \in \mathcal{C}$ and a number k, decide if* $\exp(q) < k$, *is* NP*-complete.*
2. *The problem: given $q \in \mathcal{C}$ compute* $\text{coeff}(q)$, *is* #coNP*-complete.*

The complexity class #coNP [13] is the class of counting problems of the following form

$$f(A) = \#x \ \forall y \ R(x, y, A)$$

where R is some polynomial function. Thus, #coNP counts the number of x that satisfies a certain property where checking the property itself requires an coNP machine.

For pure conjunctive queries, we have:

Theorem 5. *Given a query $q \in CQ$ over some fixed schema, both* $\exp(q)$ *and* $\text{coeff}(q)$ *can be computed in* PTIME.

For $q \in CQ$ one can compute $exp(q)$ in PTIME because here it is always possible to unify completely all subgoals referring to the same relation name, and this unifier has the minimal D. (However, to compute $coeff(q)$ one needs to consider additional unifiers, but it can still be done in polynomial time for a fixed schema). It follows that for conjunctive queries q, v, $\mu(q \mid v)$ can be computed in PTIME and all the problems we described in Sec 1 have PTIME complexity.

Pure conjunctive queries are not very interesting because in practice there is not much we can express without constants. For example, in k-anonymity we need constants to refer to the constants being published and need $\neq$ to state that two published rows correspond to distinct rows in the data. We consider only $CQ(c)$, $CQ(\neq)$, and $CQ(c, \neq)$ in the rest of the paper.

3.2 Conditional Probabilities

We now consider the two decision problems for conditional probabilities that we formulated in Sec 1: deciding $\mu(q \mid v) = 0$ and deciding $\mu(q \mid v) = 1$.

In the following discussion, $\mathcal{C}$ denotes any of $CQ(c)$, $CQ(\neq)$, $CQ(c, \neq)$: all results hold for any of these three classes. Let $S \subseteq [0, 1]$. We define the Asymptotic Conditional Probability problem for S to be:

$$\text{ACP}^S = \{(q, v) \mid q, v \in \mathcal{C}, \mu(q \mid v) \in S\}$$

We only consider the cases when $S = \{0\}$, $(0, 1)$ or $\{1\}$.

Theorem 6. $\mathrm{ACP}^{\{0\}}$ *is* Θ_2^p*-complete.* $\mathrm{ACP}^{(0,1)}$ *is* Σ_2^p*-complete.* $\mathrm{ACP}^{\{1\}}$ *is* Π_2^p*-complete.*

The complexity class Θ_2^p [24], also referred to as $\mathsf{P}^{\mathsf{NP}[O(\log n)]}$, is the class of languages that can be decided by a polynomial time oracle Turing-machine that makes $O(\log n)$ calls to an NP oracle. Thus, $\Theta_2^p \subseteq P^{\mathsf{NP}} = \Delta_2^p \subseteq \Pi_2^p, \Sigma_2^p$.

The $\mathrm{ACP}^{\{1\}}$ property is related to query containment, a well studied problem in finite model theory. For boolean queries containment becomes logical implication, and $v \Rightarrow q$ iff $\forall n.\mu_n(q \mid v) = 1$, while $\mathrm{ACP}^{\{1\}}$ means $\lim_n \mu_n(q \mid v) = 1$. The complexity of query containment for CQ is NP-complete[4]. Similarly ACP^0 is related to non-containment, which, by complementation, is coNP-complete.

Data complexity We study two notions of data complexity. In the first setting, we fix the query and study the complexity as a function of the size of the view. For a query q and set $S \subseteq [0,1]$ we define the following problem:

$$\mathrm{ACP}_q^S = \{v \mid \mu(q \mid v) \in S\}$$

Theorem 7. *Let* q *be any query in* $CQ(c, \neq)$ *and* S *be any of* $\{0\}$*,* $(0,1)$ *and* $\{1\}$*. Then,* ACP_q^S *is in* Θ_2^p*. Further, there exists a query* $q \in CQ(c, \neq)$ *such that* ACP_q^S *is* Θ_2^p*-complete.*

In the second setting, we fix the query as well as a non-boolean view definition and study complexity as a function of the size of the view instance. A non-boolean conjunctive query V is a formula Eq.(1) possibly with free variables. For example the following query V has free variables $\{x, y\}$:

$$\exists z R(x, a, z), S(z, y) \tag{1}$$

Let $\bar{V} = V_1, \ldots, V_m$ be a set of non-boolean views and let $\bar{J} = J_1, \ldots, J_m$ be sets of tuples, with J_i having the same arity as the arity of V_i. Denote $v_i \equiv V_i/J_i$ the boolean conjunctive query stating that all tuples in J_i must be in the result of V_i. For example, if V is given by Eq 1, and $J = \{(a,b),(c,b)\}$, then $V/J \equiv \exists z_1 \exists z_2 R(a, a, z_1)$, $S(z_1, b)$, $R(c, a, z_2)$, $S(z_2, b)$.

For a query q, view definitions $\bar{V}$, and set $S \subseteq [0,1]$ we define the following problem:

$$\mathrm{ACP}_{q,\bar{V}}^S = \{J \mid \mu(q \mid \bar{V}/\bar{J}) \in S\}$$

Theorem 8. *For any* $q, \bar{V} \in CQ(c, \neq)$ *and* $S \in \{\{0\}, (0,1), \{1\}\}$*, the problem* $\mathrm{ACP}_{q,\bar{V}}^S$ *is in* Θ_2^p*. Further, there exists* $q, \bar{V} \in CQ(c, \neq)$ *such that* $\mathrm{ACP}_{q,\bar{V}}^S$ *is* Θ_2^p*-complete.*

Here, too, the problem $\mathrm{ACP}_{q,\bar{V}}^{\{1\}}$ is related to another well studied problem in the literature: the query answering using views problem, under the open world assumption [12]. Indeed, the latter is $\forall n.\mu_n(q \mid \bar{V}/\bar{J}) = 1$, since this means that q is true on all instances I consistent with the observations J, i.e. q is "certain". This problem is known to be in PTIME [1], even for $CQ(c, \neq)$. (One can also check it immediately, since it can be restated as the containment problem $\bar{V}/\bar{J} \subseteq q$, where q is a fixed query.)

$\mu(q \mid v_1)$	$\mu(q \mid v_2)$	$\mu(q \mid v_1v_2)$	v_1	v_2	q
0	0	0	$R(a_1, -)$	$R(a_2, -)$	$R(-, b)$
0	0	(0,1)	$R(a, -)$	$R(-, b)$	$R(a, b)$
0	0	1	$R(a, b, -)$	$R(-, b, c)$	$R(a, b, c)$
0	(0,1)	0	$R(a, b, d)$	$R(a, b, -), R(-, -, c)$	$R(a, -, c)$
0	(0,1)	(0,1)	$R(a, -)$	$R(a, -), R(-, b)$	$R(a, b)$
0	(0,1)	1	$R(a, b, -)$	$R(a, -, -), R(-, b, c)$	$R(a, b, c)$
0	1	0	$R(a, b, d)$	$R(a, b, -), R(-, b, c)$	$R(a, b, c)$
0	1	(0,1)	$R(-, b, d)$	$R(a, b, -), R(-, b, c)$	$R(a, b, c)$
0	1	1	$R(-, b, c)$	$R(a, b, -), R(-, b, c)$	$R(a, b, c)$
(0,1)	(0,1)	0	$R(a, b, -), R(-, -, c), R(a, e, d)$	$R(a, e, -), R(-, -, c), R(a, b, d)$	$R(a, -, c)$
(0,1)	(0,1)	(0,1)	$R(a, -), R(-, b)$	$R(a, -), R(-, b)$	$R(a, b)$
(0,1)	(0,1)	1	$R(a, b, -), R(-, -, c)$	$R(a, -, -), R(-, b, c)$	$R(a, b, c)$
(0,1)	1	0	$R(a, b, -), R(-, -, c), R(a, e, d)$	$R(a, e, -), R(-, e, c), R(a, b, d)$	$R(a, -, c)$
(0,1)	1	(0,1)	$R(a, e, -, -), R(-, -, c, f), R(a, b, g, -)$	$R(a, b, -, -), R(-, b, c, h)$	$R(a, -, c, -)$
(0,1)	1	1	$R(a, b, -), R(-, -, c)$	$R(a, -, c), R(-, b, c)$	$R(a, b, c)$
1	1	0	$R(a, b, -), R(-, b, c), R(a, e, d)$	$R(a, e, -), R(-, e, c), R(a, b, d)$	$R(a, -, c)$
1	1	(0,1)	$R(a, b, -), R(-, b, c), R(-, e, d)$	$R(a, e, -), R(-, e, c), R(-, b, d)$	$R(a, -, c)$
1	1	1	$R(a, b, -), R(-, b, c)$	$R(a, -, c), R(a, b, -)$	$R(a, b, c)$

Fig. 2. Each of the 27 classes $\mathrm{ACP}^S_{S_1,S_2}$ is nonempty, assuming $\alpha(R) = A(R)$.

3.3 Complex Problems

Collusions For $S, S_1, S_2 \in \{0, (0,1), 1\}$, denote $\mathrm{ACP}^S_{S_1,S_2}$ the problem of deciding, for queries (q, v_1, v_2), whether $(q, v_1v_2) \in \mathrm{ACP}^S$ given that $(q, v_1) \in \mathrm{ACP}^{S_1}$ and $(q, v_2) \in \mathrm{ACP}^{S_2}$.

Theorem 9. *The complexity of* $\mathrm{ACP}^S_{S_1,S_2}$ *is same as that of* ACP^S.

The theorem essentially contains 27 statements, for all combinations of S_1, S_2, S. A priori, it is not even clear why all the 27 classes $\mathrm{ACP}^S_{S_1,S_2}$ are nonempty. To see that, Fig. 2 shows for each class $\mathrm{ACP}^S_{S_1,S_2}$ an example of v_1, v_2 and q in that class. For queries, we use the shorthand notation where each "-" stands for a unique existentially quantified variable. There are less than 27 entries due to the symmetry between S_1 and S_2.

Fig. 2 reveals an interesting and counter-intuitive phenomenon, which we refer to as the *non-monotonicity of information disclosure*: publishing more information results in less information disclosure. For example, the entry corresponding to 1, 1, 0 shows that with v_1, v_2, q as given in the figure, we have $\mu(q \mid v_1) = \mu(q \mid v_2) = 1$ but $\mu(q \mid v_1v_2) = 0$. Here the query q is very likely true given either v_1 or v_2 alone but is very likely false given both v_1 and v_2.

Relative Security Finally, we explain the last three entries in Fig. 1 The complexity class P.coNP, also called probabilistic coNP, is the set of languages L for which there is a coNP Turing machine M that uses random bits such that for all strings x: (1) $x \in L \Rightarrow Pr(M \text{ accepts } x) > 1/2$, and (2) $x \notin L \Rightarrow Pr(M \text{ accepts } x) \leq 1/2$.

4 Proofs of Main Results

We include here some proofs and defer the rest to our technical report [7].

Recall the definition of *zero sub-goals* and expression for *exp*(q) and *coeff*(q) from Sec. 2. In all the proofs in this section, we consider only those queries that do not have any zero sub-goals and that they do not have any non-trivial automorphisms, i.e $aut(q) = 1$. We call such queries *simple conjunctive queries*. It is tedious but straightforward to incorporate zero-subgoals and automorphisms in these results, and we omit their discussion here.

For simple conjunctive queries, the expression for *exp* and *coeff* can be simplified to $exp(q) = \min_\eta D(\eta(q))$ and $coeff(q) = \sum_{\eta | D(\eta(q)) = exp(q)} \beta(\eta(q))$

Proposition 1. *Given a conjunctive query q, the complexity of evaluating* coeff(q) *is #*coNP*-complete.*

Proof. *coeff*(q) is simply the size of the set

$$\{(\eta, k) \mid \forall \eta_0 D(\eta(q)) \leq D(\eta_0(q)) \wedge k < \beta(\eta(q))\}$$

Thus, computing *coeff*(q) is in #coNP.

To prove the #coNP-hardness, we will give a reduction from #NSAT. In #NSAT, we are given a *3-CNF* formula ϕ where the set of variables can be partitioned into two sets X and Y, and we need to count the number of assignments of X that can be extended to an assignment of ϕ. #NSAT is known [22] to be #coNP-hard.

Given any *3-CNF* formula ϕ over variables X and Y, we construct two queries q_1 and q_2. Let ϕ have c clauses and let $|X| = k$. The vocabulary consists of a relation R of arity 4 and a relation S of arity 3 with $\beta(R) = \beta(S) = B$, where B is some integer greater than 2^k. We create a unique constant k_x for each variable x a unique constant k_C for each clause C in ϕ, and two extra constants t and f. The query q consists of $q_1 q_2$ where q_1 and q_2 are two queries as described below.

q_1 is constructed as follows. For each clause $C(x, y, z)$ in ϕ, q_1 contains 7 subgoals of the form $R(k_C, v_i, v_j, v_k)$, where v_i, v_j and v_k are such that $C(v_i, v_j, v_k)$ is true. In addition, for each variable $x \in X$ in ϕ, v contains three subgoals $S(k_x, x, x)$, $S(k_x, t, -)$ and $S(k_x, f, -)$.

q_2 is constructed as follows. For each clause $C(x, y, z)$ in ϕ, q_2 contains a subgoal $R(k_C, x', y', z')$, where x', y' and z' are variables. Also, for each variable $x \in X$ in ϕ, v contains a subgoal $S(k_x, -, x')$.

Claim: $\#\phi = (coeff(q_1 q_2) \bmod B^{7c+2k+1}) \,/\, B^{7c+2k}$.

To verify the claim, lets first look at the unifiers of just q_1. Each $S(k_x, x, x)$ can be unified with either $S(k_x, t, -)$ or $S(k_x, f, -)$. There are 2^k such unifiers, one corresponding to each assignment of t or f to variables in X. If η is any such unifier, $\beta(\eta(q_1)) = B^{7c+2k}$ since there are $7c + 2k$ subgoals.. Further, if η corresponds to an assignment that can be extended to an assignment of ϕ, then q_2 can be completely mapped to q_1, i.e. $\eta(q_1)q_2 \equiv \eta(q_1)$.

There are two cases, $\#\phi$ is either 0 or its greater than 0. In the latter case, there is at least one η with the property that $\eta(q_1)q_2 \equiv \eta(q_1)$. Every such η adds the term B^{7c+2k} to the *coeff* resulting in $coeff(q_1 q_2) = \#\phi * B^{7c+2k}$. The claim follows since B is chosen to be greater than 2^k and $\#\phi$ is at most 2^k. In the former case, when $\#\phi = 0$, every minimal unifier of $q_1 q_2$ must contain at least $7c + 2k + 1$ subgoals, so *coeff*$(q_1 q_2)$ is a multiple of $B^{7c+2k+1}$. The claim follows.

Theorem 10. $\text{ACP}^{\{1\}}$ *is* Π_2^p*-complete.*

Proof. We first show that $\text{ACP}^{\{1\}}$ belongs to Π_2^p. $(q, v) \in \text{ACP}^{\{1\}}$ can be restated as

$$\forall\eta_0(\forall\eta_1 D(\eta_0(v)) \leq D(\eta_1(v)) \Rightarrow \exists\eta_2\ \eta_2(qv) = \eta_0(v))$$

Thus, $\text{ACP}^{\{1\}} \in \Pi_2^p$.

For completeness, we give a reduction from the $\forall\exists SAT$ problem defined below, which is known to be Π_2^p-complete.

$\forall\exists SAT = \{(X, Y, \phi(X, Y)) \mid X$,$Y$ sets of variables,$\phi(X, Y)$ a *3CNF* formula, and $\forall X\exists Y\phi(X, Y)\}$

Given $(X, Y, \phi(X, Y))$, we construct two conjunctive queries q and v. The vocabulary consists of R, S of arities 4 and 2 respectively, and there are two constants t and f.

The query v is constructed as follows. Every clause $C(x,\ y,\ z)$ in $\phi(X, Y)$, which is a disjunction of x,y and z or their negations, contributes seven subgoals to v. These are of the form $R(k_C, v_i, v_j, v_k)$, where k_C is a unique constant for each clause and each of v_i,v_j,v_k is either t or f so that the resulting assignment makes the clause true. Every $x \in X$ contributes four subgoal to v given by $S(k_x, t, -)$, $S(k_x, f, -)$, $S(k_x, -, 0)$ and $S(k_x, -, 1)$, where k_x is a unique constant for each x.

The query q is constructed as follows. Corresponding to every clause $C(x, y, z)$ in ϕ, there is a subgoal $R(k_C, x, y, z)$, where k_C is the same constant for the clause as used in the definition of v and x,y,z are variables. For each $x \in X$, there is a subgoal $G(k_x, x, 0)$, where again k_x is the same constant used in v for variable x.

Claim*:* $(q, v) \in \text{ACP}^{\{1\}}$ *iff* $(X, Y, \phi(X, Y)) \in \forall\exists SAT$.

To see this, let us analyze the set $\text{MGUQ}^0(v)$. The subgoals corresponding to R relations cannot be unified with anything else, as all of them contain only constants. The S subgoals corresponding to two different x cannot be unified because of the k_x constants. For same $x \in X$, the four S subgoals can be maximally unified in two possibles ways leading to $S(k_x, t, 0), S(k_x, f, 1)$ or $S(k_x, f, 0), S(k_x, t, 1)$. The choice can be made independently for each x. Thus, the size of $\text{MGUQ}^0(v)$ is $2^{|X|}$. Now, $\mu(q \mid v) = 1$ iff for each of the query v_i in $\text{MGUQ}^0(v)$, q can be mapped to v_i. Each v_i, for each x, contains exactly one of $S(k_x, t, 0)$ and $S(k_x, f, 0)$. The subgoal $S(k_x, x, 0)$ in q has to map to this subgoal. Thus, x will be equated to t or f. After all the S subgoals in q are mapped, each of the X variables will have a truth assignment. As we iterate over v_i, we get all possibles truth assignments for X. Also, after X is given a truth assignment, all the R subgoals of q must map to one of the subgoals of v. This is possible iff the Y variables can be given a truth assignment so that all clauses in ϕ are satisfied. This proves that $(q, v) \in \text{ACP}^{\{1\}}$ iff $(X, Y, \phi(X, Y)) \in \forall\exists SAT$. Thus, $\text{ACP}^{\{1\}}$ is Π_2^p-complete.

Theorem 11. $\mathrm{ACP}^{(0,1]}$ *is* Θ_2^p*-complete.*

Wagner [24] has provided a very useful tool for proving Θ_2^p-hardness of problems, which we state below.

Theorem 12 (Wagner [24]). *Let D be an* NP*-complete set and let A be any arbitrary set. Let χ_D be the characteristic function of D. If there exists a polynomial-time computable function f such that*

$$|\{i \mid x_i \in D\}| \textit{is odd} \Leftrightarrow f(x_1, \ldots, x_{2k}) \in A$$

for all $k \geq 1$ and $x_1, \ldots, x_{2k}$ with $\chi_D(x_1) \geq \cdots \geq \chi_D(x_{2k})$, then A is Θ_2^p-complete[3]*.*

Before we give the proof of Thm. 11, we need few results. Call a *3-CNF* formula with k clauses *almost satisfiable* if there exists an assignment that satisfies at least $k-1$ clauses.

Lemma 1. *There exists a polynomial-time function F such that if ϕ is a 3-CNF formula, $F(\phi)$ is an almost satisfiable 3-CNF formula with the property $\phi \Leftrightarrow F(\phi)$.*

Lemma 2. *There exists PTIME functions g, h s.t. if ϕ_1 and ϕ_2 are almost satisfiable formulas, then $\phi_1 \Rightarrow \phi_2$ iff $\mu(g(\phi_1, \phi_2) \mid h(\phi_1, \phi_2)) > 0$.*

Proof. For each ϕ_i $(i = 1, 2)$, we define two conjunctive queries $q_1(\phi_i)$ and $q_2(\phi_i)$. $q_1(\phi_i)$ is a query over a relation R_i of arity 4. Corresponding to each clause $C(x, y, z)$ in ϕ_i, there are seven subgoals in $q_1(\phi_i)$ of the form $R_i(k_C, v_j, v_k, v_l, 0)$, where k_C is a unique constant for each clause and $(v_j, v_k, v_l) \in \{t, f\}^3$ such that $C(v_j, v_k, v_l)$ is true. We call these the *type-0 subgoals* since all of them end with the constant 0. In addition, $q_1(\phi_i)$ contains eight more subgoals of the form $R_i(x_i, v_j, v_k, v_l, 1)$, where x_i is a variable and $(v_j, v_k, v_l) \in \{t, f\}^3$. We call these the *type-1 subgoals*. $q_2(\phi_i)$ is also a query over R_i. For each clause $C(x, y, z)$ in ϕ_i, $q_1(\phi_i)$ contains a subgoal $R_i(k_C, x, y, z, -)$ where x,y,z are variables. Let S be a new relation and a,b,c fresh constants:

$$g(\phi_1, \phi_2) = q_2(\phi_2)$$
$$h(\phi_1, \phi_2) = q_1(\phi_1) q_2(\phi_1) q_1(\phi_2) S(x_1, x_2, c), S(a, b, c)$$

We will show that g and h satisfy the required property, i.e. $\phi_1 \Rightarrow \phi_2$ iff $\mu(g(\phi_1, \phi_2) \mid h(\phi_1, \phi_2)) > 0$. Let us first analyze the set $\textsc{mguq}^0(h(\phi_1, \phi_2))$. Also, assume without loss of generality that ϕ_1 and ϕ_2 have distinct set of variables. Then, the sub-query $q_1(\phi_1)q_1(\phi_2)$ cannot be further unified. There are two cases: **(i)** ϕ_1 is satisfiable. Then, the sub-query $q_2(\phi_1)$ can be completely mapped to the type-0 subgoals of $q_1(\phi_1)$. Further, $S(x_1, x_2, c)$ can be unified with $S(a, b, c)$. The resulting query is the only one in $\textsc{mguq}^0(h(\phi_1, \phi_2))$. Note that it equates x_2 to b. **(ii)** ϕ_2 is not satisfiable. Then, $q_2(\phi_1)$ cannot be completely mapped to the type-0 subgoals of $q_1(\phi_1)$. But since ϕ_2 is almost satisfiable, all but one subgoal

[3] The class Θ_2^p is referred to as $\mathsf{P}_{\mathsf{bf}}^{\mathsf{NP}}$ in [24]

of $q_2(\phi_1)$ can be mapped to the type-0 subgoals. The remaining subgoal can be unified with a type-1 subgoal, by equating x_1 with the constant for the corresponding clause. $S(x_1, x_2, c)$ can no more be unified with $S(a, b, c)$, since x_1 has been equated with a different constant. One can easily check that the resulting query is the only one in $\text{MGUQ}^0(h(\phi_1, \phi_2))$. Also note that x_2 is still a free variable in this query. In both cases, there is a unique query in $\text{MGUQ}^0(h(\phi_1, \phi_2))$. Call it q_0. $\mu(g(\phi_1, \phi_2) \mid h(\phi_1, \phi_2)) > 0$ holds iff $g(\phi_1, \phi_2) = q_2(\phi_2)$ maps to q_0. If ϕ_1 is not satisfiable, q_0 contains $q_1(\phi_2)$ as a sub-query, otherwise it contains $q_1(\phi_2)$ with x_2 equated to b. If ϕ_2 is satisfiable, it can be mapped to the type-1 subgoal of $q_1(\phi_2)$, and hence can be mapped to q_0. If ϕ_2 is not satisfiable, it can still be mapped to $q_1(\phi_2)$ by using a type-1 subgoal, but then x_2 should be a free variable, i.e., ϕ_1 should also be not satisfiable. Hence, $\mu(g(\phi_1, \phi_2) \mid h(\phi_1, \phi_2)) > 0$ iff $\phi_1 \Rightarrow \phi_2$.

Proof. (**Thm. 11**) First we show that $\text{ACP}^{(0,1]}$ belongs to Θ_2^p. By Cor. 2, $(q, v) \in \text{ACP}^{(0,1]}$ iff $exp(v) = exp(qv)$. The language $\{(q, k) \mid exp(q) \leq k\}$ is in NP since given any (q, k), one only needs to check if there is a substitution η with $D(\eta(q)) \leq k$. Further, $exp(q)$ cannot exceed $D(q)$, which is polynomial in size of q. Thus, $exp(q)$ is determined by a binary search issuing $O(\log n)$ queries to an NP oracle. Since $exp(v) = exp(qv)$ can be checked by explicitly computing $exp(v)$ and $exp(qv)$, we have $\text{ACP}^{(0,1]} \in \Theta_2^p$. For completeness, let $D = \textit{3-SAT}$ be the set of all satisfiable *3-CNF* formulas. We know D is NP-complete. Let $x_1, \ldots, x_{2k}$ be s.t. $\chi_D(x_1) \geq \cdots \geq \chi_D(x_{2k})$. For $i = 1, \ldots, k$, let $Q_i = g(F(x_{2i-1}), F(x_{2i}))$ and $V_i = h(F(x_{2i-1}), F(x_{2i}))$, where F, g, h are functions as defined in Lemmas 1 and 2. Assume that Q_i and V_i use different set of relations for different i. Let $v = V_1 V_2 \ldots V_k$ and $q = Q_1 Q_2 \ldots Q_k$. Then, $\mu(v \mid q) = \prod_{i=1}^{k} \mu(V_i \mid Q_i)$. By Lemma 2, $\mu(V_i \mid Q_i) > 0 \Leftrightarrow \chi_D(x_{2i-1}) = \chi_D(x_{2i})$. Thus, $\mu(v \mid q) > 0 \Leftrightarrow |\{i \mid x_i \in D\}|$is odd. By Thm. 12, ACP^1 is Θ_2^p-complete.

5 Conclusions

We investigate the complexity of a new approach to incompleteness in databases, based on Bayes's notion of a prior probability distribution. In this new framework we study the complexity of several fundamental problems, with applications to information disclosure and query answering using views, and provide tight complexity bounds.

References

1. Serge Abiteboul and Oliver M. Duschka. Complexity of answering queries using materialized views. In *PODS*, pages 254–263, 1998.
2. Brian Babcock and Surajit Chaudhuri. Towards a robust query optimizer: a principled and practical approach. In *SIGMOD*, pages 119–130, 2005.
3. Fahiem Bacchus, Adam J. Grove, Joseph Y. Halpern, and Daphne Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, 87(1-2):75–143, 1996.

4. Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *STOC*, pages 77–90, 1977.
5. Nilesh Dalvi, Gerome Milkau, and Dan Suciu. Asymptotic conditional probabilities for conjunctive queries. In *ICDT*, 2005.
6. Nilesh Dalvi and Dan Suciu. Query answering using probabilistic views. In *VLDB*, pages 805–816, 2005.
7. Nilesh Dalvi and Dan Suciu. Query evaluation on a database given by a random graph, April 2006.
8. Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222, 2003.
9. R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41(1):50–58, 1976.
10. Lise Getoor, Benjamin Taskar, and Daphne Koller. Selectivity estimation using probabilistic models. In *SIGMOD*, pages 461–472, 2001.
11. Y. V. Glebskiĭ, D. I. Kogan, M. I. Liogon'kiĭ, and V. A. Talanov. Range and degree of realizability of formulas in the restricted predicate calculus. *Kibernetika*, 2:17–28, 1969. [Engl. Transl. Cybernetics, vol. 5, 142–154 (1972)].
12. Alon Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, 2001.
13. Lane A. Hemaspaandra and Heribert Vollmer. The satanic notations: Counting classes beyond #p and other definitional adventures. Technical report, Rochester, NY, USA, 1994.
14. Daniel Kifer and J. E. Gehrke. Injecting utility into anonymized datasets. In *SIGMOD*, 2006.
15. Maurizio Lenzerini. Data integration: a theoretical perspective. In *PODS*, pages 233–246, 2002.
16. M. I. Liogon'kiĭ. On the conditional satisfyability ratio of logical formulas. *Mathematical Notes of the Academy of the USSR*, 6:856–861, 1969.
17. James F. Lynch. Probabilities of sentences about very sparse random graphs. *random struct. algorithms*, 3(1):33–54, 1992.
18. James F. Lynch. Infinitary logics and very sparse random graphs. In *Logic in Computer Science*, pages 191–198, 1993.
19. Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, page 24, 2006.
20. Gerome Miklau and Dan Suciu. A formal analysis of information disclosure in data exchange. In *SIGMOD*, 2004.
21. J. Spencer and S. Shelah. Zero-one laws for sparse random graphs. *J. Amer. Math. Soc.*, pages 97–115, 1988.
22. L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.
23. Moshe Y. Vardi. The complexity of relational query languages. In *STOC*, pages 137–146, 1982.
24. K. W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theor. Comput. Sci.*, 51(1-2):53–80, 1987.