Alcom-FT Technical Report Series
ALCOMFT-TR-03-108

Computational Complexity of Auditing Finite Attributes in Statistical Databases

Peter Jonsson *,

Department of Computer and Information Science, Linköpings Universitet, SE-581 83 Linköping, Sweden, tel: +46 13 282415, fax: +46 13 282499

Andrei Krokhin [b]

[b] *Department of Computer Science, University of Warwick,Coventry, CV4 7AL, UK*

---

* Corresponding author.
Email addresses: `Peter.Jonsson@ida.liu.se` (Peter Jonsson), `Andrei.Krokhin@dcs.warwick.ac.uk` (Andrei Krokhin).

Preprint submitted to Elsevier Science *28 November 2003*

Abstract: We study the computational complexity of auditing finite attributes in databases allowing statistical queries. Given a database that supports statistical queries, the auditing problem is to check whether an attribute can be completely determined or not from a given set of statistical information. Some restricted cases of this problem have been investigated earlier, e.g. the complexity of statistical sum queries is known by the work of Kleinberg et al. (J. CSS 66 (2003) 244-253). We characterise *all* classes of statistical queries such that the auditing problem is polynomial-time solvable. We also prove that the problem is **coNP**-complete in all other cases under a plausible conjecture on the complexity of constraint satisfaction problems. This result is obtained by exploiting connections between auditing and constraint satisfaction, and using certain algebraic techniques. We also study a generalisation of the auditing problem where one asks if a set of statistical information imply that an attribute is restricted to K or less different values. We characterise all classes of polynomial-time solvable problems also in this case.

Key words: Database, statistical query, auditing, constraint satisfaction problem, computational complexity

1 Introduction

A *statistical database* is a collection of data about which queries concerning general properties of certain subsets of the data may be answered without revealing 'secret' detailed information about the data. A well-known example is databases allowing *statistical sum queries.* For instance, we may have a database with attributes $(\mathsf{name}, \mathsf{age}, \mathsf{salary})$ supporting queries of the form 'give me the sum of salaries of all individuals whose age satisifes a certain condition'. If we assume that the projection $(\mathsf{name}, \mathsf{age})$ is publicly available, what measures suffice to protect the confidentiality of the salary information? This suggests an obvious security problem: how to prevent or make difficult the extraction of data about particular individuals from the answers to statisticial queries. This is the *statistical database security* problem [1] and many different approaches have been proposed for dealing with this problem. Examples include perturbation of the database itself [24], perturbation of query answers [2] and query restriction [11]. An introduction to security issues in connection with statistical databases can be found in [7]. Yet another approach is to *audit* the statistical queries in order to determine when enough information has been given out so that compromise becomes possible [8,20] and we focus on this approach in this article. Kleinberg et al. [21] have studied the complexity of this problem for statistical sum queries. Formally speaking, they studied the following problem:

INSTANCE: A set $\{x_1, \ldots, x_n\}$ of variables taking their values from the set $D = \{0, 1, \ldots, p\}$, a family of subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$ of $\{1, \ldots, n\}$, and m integers $b_1, \ldots, b_m$.
QUESTION: Is there an $i \leq n$ such that in all 0-1-...-p solutions of the system of equations $\sum_{i \in S_j} x_i = b_j$, $j = 1, \ldots, m$, the variable x_i has the same value.

For Boolean domains (where $D = \{0, 1\}$), they showed that this problem is **coNP**-complete. This article is a direct continuation of their work so we restrict our attention to auditing finite attributes only. This does not imply that the entire database has to contain finite attributes only; the attributes used for specifying query sets (also known as 'characteristic attributes') may very well be continuous. In the example above, name and age are typically characteristic attributes if one wants to audit salary information.

Our main result is the identification of *all* classes of statistical queries having a tractable (i.e. in **PTIME**) auditing problem. The algorithm for the tractable cases can also identify the values of the compromised data efficiently. Our results also imply that, under a widely believed conjecture, the problem is **coNP**-complete in all other cases – this conjecture is known to be true if we only consider attributes with at most three values. To exemplify the use of our results we study the auditing problem for a number of statistical queries (such as MAX, MEAN and MEDIAN), and we completely classify the problem of auditing Boolean attributes.

We also study an extension of the auditing problem where the question is whether the possible values of an attribute can be narrowed down to a set of size at most K (for some $K > 0$) or not. Obviously, the usual auditing problem corresponds to the case when $K = 1$. Solutions to the ordinary auditing problem can only be used for deciding whether a database is compromised or not. The extended problem can be used for 'early warnings', i.e. it can be used for checking whether data is near to be compromised so that adequate countermeasures can be taken. We characterise all tractable cases of the extended auditing problem, too. We complement this result with the following observation: For any domain size $d \geq 3$, and any K with $1 \leq K < d - 1$, there exist a finite set Γ of relations over $\{1, \ldots, d\}$ such that K-auditing is a polynomial-time solvable problem, but $(K + 1)$-auditing is **coNP**-complete.

The results were obtained by (1) exploiting a connection between the auditing problem and the constraint satisfaction problem (CSP); and (2) using powerful algebraic techniques for studying constraint satisfaction problems. In the basic version of CSP, we are given a set of variables taking their values from a finite domain and a set of constraints (e.g. relations) restricting the values different variables can simultaneously assume – the question is whether the variables can be assigned values that are consistent with all constraints. Clearly, CSP has many connections with databases. For instance, the conjunctive-query evaluation problem [22] is to find the predicate (or decide whether it is non-empty) on variables $y_1, \ldots, y_m$ given by a formula of the form $(\exists x_1) \ldots (\exists x_n) : \mathcal{C}$ where $\mathcal{C} = \varrho_1(s_1) \wedge \ldots \wedge \varrho_q(s_q)$, $x_1, \ldots, x_n, y_1, \ldots, y_m$ are the variables, and $\varrho_1, \ldots, \varrho_q$ are the predicates used in $\mathcal{C}$. It is easy to see that this problem has a close relationship to CSP.

Constraints are typically specified by relations, so CSP can be parameterised by restricting the set of allowed relations which can be used as constraints. The problem of determining the complexity of CSP for all possible parameter sets has attracted much attention (see, e.g., [4,9,12]). For the Boolean (i.e., two-valued) case, the complexity of CSP has been successfully studied from the above perspective [28]. It is widely acknowledged that, compared to the Boolean case, one needs more advanced tools to make progress with non-Boolean constraint satisfaction problems. Such tools based on algebra, logic, and graph theory were developed in [4,5,12,15,17,18,22]. The algebraic method [4,5,17,18], which has proved to be quite powerful, builds on the fact that one can extract much information about the structure and the complexity of restricted constraint satisfaction problems from knowing certain operations, called polymorphisms, connected with the constraint relations. More exactly, polymorphisms provide a convenient 'dual' language for describing relations and, more importantly, they allow one to show that one constraint can be simulated by other constraints without giving explicit constructions.

The paper is organized as follows. In Section 2, we give basic definitions and discuss the algebraic method that will be used in the paper. In Section 3, we show that the algebraic technique is applicable to the auditing problem. Section 4 contains a proof of our main result and a number of examples are collected in Section 5. Finally, Section 6 contains some conclusions about the work we have done.

2 Preliminaries

Throughout the paper we use the standard correspondence between predicates and relations: a relation consists of all tuples of values for which the corresponding predicate holds. We will use the same symbol for a predicate and its corresponding relation, since the meaning will always be clear from the context. We will use $R_D^{(m)}$ to denote the set of all m-ary relations (or predicates) over a *fixed finite* set D, and R_D to denote the set $\bigcup_{m=1}^{\infty} R_D^{(m)}$. Note that unary relations on D are simply the subsets of D.

2.1 Constraint satisfaction problems

Definition 2.1 *A* constraint language *over D is an arbitrary subset of R_D. The* constraint satisfaction problem *over the constraint language $\Gamma \subseteq R_D$, denoted* CSP(Γ)*, is the decision problem with instance $I = (V, D, \mathcal{C})$, where*

- *V is a finite set of variables,*
- *D is a finite set of values (known as the domain) such that $|D| > 1$, and*
- *$\mathcal{C}$ is a finite set of constraints $\{C_1, \ldots, C_q\}$, in which each constraint C_i is a pair (s_i, ϱ_i) with s_i a list of variables of length m_i, called* the constraint scope*, and ϱ_i an m_i-ary relation over the set D, belonging to Γ, called* the constraint relation*.*

The question is whether there exists a solution *to I, that is, a function $\varphi : V \to D$ such that, for each constraint in $\mathcal{C}$, the image of the constraint scope is a member of the constraint relation. If I has a solution then we also say that I is* satisfiable*.*

Given an instance I of CSP(Γ), let $Sol(I) = \{\varphi \mid \varphi \text{ is a solution to } I\}$. We define the size of a problem instance as the length of the encoding of all tuples in all constraints. Note that this is a sound definition even if Γ is infinite, since an instance can only contain a finite number of different relations from Γ. We say that $\mathrm{CSP}(\Gamma)$ is *tractable* if $\mathrm{CSP}(\Gamma)$ is in **PTIME**. Throughout this paper we assume that **PTIME** $\neq$ **NP**.

Example 2.2 *Let N and N' be the following ternary relations on $\{0,1\}$:*

$$N = \{(1,0,0),(0,1,0),(0,0,1)\}, \quad N' = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}.$$

It is easy to see that the 1-IN-3-SAT *and the* NOT-ALL-EQUAL-SAT *problems (as defined in [28]) can be expressed as* $\mathrm{CSP}(\{N\})$ *and* $\mathrm{CSP}(\{N'\})$*, respectively. Both problems are known to be* **NP***-complete [28].*

Example 2.3 *Let $\neq_D$ be the binary disequality relation on any finite D. Then* $\mathrm{CSP}(\neq_D)$ *is exactly the* GRAPH $|D|$-COLORING *problem. It is known to be tractable if $|D| = 2$ and* **NP***-complete otherwise [13].*

2.2 Statistical databases and the auditing problem

A *statistical database* B can be viewed as a set of *records* $\{r_1, \ldots, r_n\}$ where each record r_i is a list $(a_1, \ldots, a_m)$ of *attribute values*. The i:th position of these tuples is denoted *attribute* A_i. The domain D_i of an attribute A_i is the set of values from which attribute A_i draws its values. Throughout this article, we assume that all domains that may be audited are finite. Loosely speaking, a *statistical query* finds those records in the database that satisfy a certain condition and returns the result of some test or computation on them (but does *not* return the records themselves). The standard *auditing problem* is to decide if the answers to such a set of statistical queries uniquely determine the attribute value of some record. We will now define a slight generalisation of the auditing problem in terms of constraint satisfaction:

AUDIT(Γ)
INSTANCE: A tuple $((V, D, \mathcal{C}), v, k)$ where $I = (V, D, \mathcal{C})$ is an instance of $\mathrm{CSP}(\Gamma)$, $v \in V$ and an integer $1 \leq k \leq |D|$.
QUESTION: Is $|\{\varphi(v) \mid \varphi \in Sol(I)\}| \leq k$?

Our definition of AUDIT generalises the more standard notions of auditing in two ways: (1) we put no restriction on the set Γ of relations that can be used in queries; and (2) we do not only consider the problem to check whether an attribute value is completely revealed – we can also check whether an attribute value is narrowed down to a set of specified size. Our definition of statistical queries is very broad; the answer to a statistical query is an expression $\varrho(x_{i_1}, \ldots, x_{i_n})$ where the variables denotes attributes in the selected records and ϱ is a predicate. Let $\mathcal{C}$ be a collection of such expressions and assume that variable v corresponds to a certain attribute a in a certain record r. If v can take only one value when considering all solutions to $\mathcal{C}$, we know that $r[a]$ must have this value, i.e. $r[a]$ is compromised. Our ultimate goal is to distinguish those queries that make auditing tractable from those for which it is hard.

Example 2.4 *We reconsider the example given in the introduction. Thus, we have a statistical database with attributes* $(\mathsf{name}, \mathsf{age}, \mathsf{salary})$ *supporting queries of the form 'give me the sum of salaries of all individuals whose age satisifes a certain condition'. Assume a large number of such questions have been asked and we have a knowledge-base containing information like 'the sum of the salaries of those being 35 years old is \$2.200.000' and 'the sum of the salaries of those being older than 55 years is \$50.000.000'. Obviously, this knowledge can easily be cast into an* AUDIT *problem as defined above: for instance, the first piece of information can be transformed into the constraint* $\varrho(x_1, \ldots, x_k)$ *where we assume that there are* k *individuals being 35 years old, variable* x_i *denotes the salary of individual* i *and* ϱ *is the relation*

$$\{(a_1, \ldots, a_k) \in D^k \mid a_1 + \ldots + a_k = 2.200.000\}.$$

We let K-AUDIT denote the subproblem of AUDIT such that (I, v, k) is an instance of K-AUDIT if and only if $0 < k \leq K$. The problem 1-AUDIT is consequently equivalent to the 'usual' auditing problem. Given a CSP instance $I = (V, D, \mathcal{C})$ and a variable $v \in V$ that has at most k different values under all solutions to I, we say that v is *k-compromised* in I. We want to emphasise that if I has no solution, then the given K-AUDIT instance is a 'yes'-instance. We would like to point out that there is a slight difference in our definition of auditing and the one by Kleinberg et al. [21] (which was presented in the introduction). In their formulation, one checks if *at least* one variable is compromised while we check whether a given variable is compromised or not – there is an obvious Turing reduction from their problem to ours so our problem is always at least as hard as theirs. Furthermore, we believe our computational problem is more realistic from a practical point of view; the identification of compromised attributes may very well be nontrivial if only given the information that they exist.

Obviously, AUDIT(Γ) is in **coNP** for every choice of Γ. The following reduction will turn out to be important throughout the article.

Proposition 2.5 *For any* Γ*,* CSP(Γ) *is polynomial-time reducible to the complement of* K-AUDIT(Γ)*.*

Proof. Given an instance $I = (V, D, \mathcal{C})$ of CSP(Γ), let v denote an variable not in V and consider the instance $I' = ((V \cup \{v\}, D, \mathcal{C}), v, 1)$ of AUDIT(Γ). If I has a solution, then $(V \cup \{v\}, D, \mathcal{C})$ has a solution but v can be assigned $|D| > 1$ different values so I' is a 'no'-instance. Otherwise, I' is a 'yes'-instance. ∎

Let us now consider the 1-AUDIT($\{N\}$) problem (where N is defined as in Example 2.2). First, the fact that $N(x, y, z)$ holds if and only if $x+y+z=1$ suggests that 1-AUDIT($\{N\}$) is a subproblem of the sum query auditing problem. Secondly, CSP($\{N\}$) is **NP**-complete which implies that 1-AUDIT($\{N\}$) is **coNP**-complete by Proposition 2.5. The **coNP**-completeness of the sum query auditing problem [21] follows immediately.

A related problem that has been studied both from theoretical and practical viewpoints is the *frozen variable* problem FV-CSP [3,10,23,25,29]. In this problem, we are given a CSP instance I and a variable v, and the question is whether variable v has the same value in all models – if I has no solution, then the FV-CSP instance is considered a 'no'-instance. The auditing problem and the frozen variable may appear to be very similar but there are at least two very important differences that the reader should be aware of: first, FV-CSP is complete for **DP** while the auditing problem is complete for **coNP**, and there exists sets of relations such that FV-CSP is **DP**-, **NP**- or **coNP**-complete, or tractable [23]. Secondly, for any set of relations Γ, the auditing problem over Γ cannot be easier than CSP(Γ) (due to the reduction presented in Proposition 2.5) but this is not true for FV-CSP, though.

2.3 Algebraic framework

In addition to predicates and relations we will also consider arbitrary *operations* on the set of values. We will use $O_D^{(n)}$ to denote the set of all n-ary operations on a set D (that is, the set of mappings $f\colon D^n \to D$), and O_D to denote the set $\bigcup_{n=1}^{\infty} O_D^{(n)}$.

Any operation on D can be extended in a standard way to an operation on tuples over D, as follows. For any operation $f \in O_D^{(n)}$, and any collection of tuples $\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n \in D^m$, where $\vec{a}_i = (\vec{a}_i(1), \ldots, \vec{a}_i(m))$, $i = 1, \ldots, n$, define

$$f(\vec{a}_1, \ldots, \vec{a}_n) = (\, f(\vec{a}_1(1), \ldots, \vec{a}_n(1)), \ldots, f(\vec{a}_1(m), \ldots, \vec{a}_n(m))\,).$$

Definition 2.6 *For any relation $\varrho \in R_D^{(m)}$, and any operation $f \in O_D^{(n)}$, if $f(\vec{a}_1, \ldots, \vec{a}_n) \in \varrho$ for all $\vec{a}_1, \ldots, \vec{a}_n \in \varrho$, then ϱ is said to be* invariant *under f, and f is called a* polymorphism *of ϱ.*

The set of all relations that are invariant under each operation from some set $C \subseteq O_D$ will be denoted $\mathsf{Inv}(C)$. The set of all operations that are polymorphisms of every relation from some set $\Gamma \subseteq R_D$ will be denoted $\mathsf{Pol}(\Gamma)$. By $\mathsf{Pol}_n(\Gamma)$ we will denote the set of all n-ary members of $\mathsf{Pol}(\Gamma)$. We remark that the operators Inv and Pol form a Galois correspondence between R_D and O_D (see Proposition 1.1.14 in [26]).

It is easy to see that $\mathrm{CSP}(\Gamma)$ can be expressed as a logical problem as follows: is it true that a first-order formula $\varrho_1(s_1) \wedge \ldots \wedge \varrho_q(s_q)$, where each ϱ_i is an atomic formula involving a predicate from Γ, is satisfiable?

Definition 2.7 *For any set $\Gamma \subseteq R_D$ the set $\langle\Gamma\rangle$ consists of all predicates that can be expressed using*

(1) predicates from $\Gamma \cup \{=_D\}$,
(2) conjunction,
(3) existential quantification.

A relation belongs to $\langle\Gamma\rangle$ if and only if it can be represented as the projection of the set of all solutions to some $\mathrm{CSP}(\Gamma)$-instance onto some subset of variables [19]. Stated differently, $\varrho \in \langle\Gamma\rangle$ if and only if ϱ can be expressed by a conjunctive query over $\Gamma \cup \{=\}$. Intuitively, constraints using relations from $\langle\Gamma\rangle$ are exactly those which can be 'simulated' by constraints using relations in Γ. In fact, $\langle\Gamma\rangle$ can be characterized in a number of ways [26], and one of them is most important for our purposes.

Theorem 2.8 ([26]) *For every set $\Gamma \subseteq R_D$, $\langle\Gamma\rangle = \mathsf{Inv}(\mathsf{Pol}(\Gamma))$.*

Theorem 2.8 is the corner-stone of the algebraic method, since it shows that the expressive power of constraints is determined by polymorphisms. In particular, in order to show that a relation ϱ can be expressed by relations in Γ, one does not have to give an explicit construction, but instead one can show that ϱ is invariant under all polymorphisms of Γ, which often turns out to be significantly easier. An operation $e_n^i : D^n \to D$ is called the i-th n-ary *projection* if $e_n^i(a_1, \ldots, a_i, \ldots, a_n) = a_i$ for all $a_1, \ldots, a_n \in D$. It is easy to check that any projection is a polymorphism of every relation. We will use the following result from [27].

Proposition 2.9 *Let Γ be a set of relations on $\{0,1\}$. Either $\mathsf{Pol}(\Gamma)$ consists of all projections (and then $\mathsf{Inv}(\mathsf{Pol}(\Gamma)) = R_{\{0,1\}}$), or else $\mathsf{Pol}(\Gamma)$ contains at least one of the following 7 operations:*

(a) the constant operation 0,
(b) the constant operation 1,
(c) the negation operation $\neg x$,
(d) the disjunction operation $x \vee y$,
(e) the conjunction operation $x \wedge y$,
(f) the majority operation $(x \vee y) \wedge (x \vee z) \wedge (y \vee z)$,
(g) the affine operation $x - y + z \pmod 2$.

Example 2.10 *Reconsider the relation N from Example 2.2. It is easy to check that none of the 7 operations from Proposition 2.9 is a polymorphism of N. Hence, $\mathsf{Pol}(\{N\})$ consists of all projections and $\langle\{N\}\rangle = R_{\{0,1\}}$.*

Example 2.10 illustrates how Theorem 2.8 allows one to make use of known algebraic results. A number of results on the complexity of constraint satisfaction problems have been obtained via the algebraic approach (e.g., [4–6,17,18]). For example, it is shown in [17] that Schaefer's Dichotomy Theorem [28], when appropriately re-stated, easily follows from well-known algebraic results [27].

Theorem 2.11 ([28]) *For any set* $\Gamma \subseteq R_{\{0,1\}}$, $\mathrm{CSP}(\Gamma)$ *is tractable when* $\mathsf{Pol}(\Gamma)$ *contains at least one of the operations (a)-(b) or (d)-(g) from Proposition 2.9. In all other cases* $\mathrm{CSP}(\Gamma)$ *is* $\mathbf{NP}$*-complete.*

3 Algebraic results

In this section, we prove that the complexity of K-AUDIT(Γ) is determined by the polymorphisms of Γ, and hence the algebraic technique is applicable. This implies that the algebraic techniques are applicable to the unrestricted AUDIT(Γ) problem, too, since this problem is $|D|$-AUDIT(Γ). We also show how the complexity of K-AUDIT(Γ) depends on the set $\mathsf{Pol}_1(\Gamma)$ of unary polymorphisms of Γ and on the complexity of $\mathrm{CSP}(\Gamma)$.

Lemma 3.1 *Let* $\Gamma \subseteq R_D$ *and* $\varrho \in \langle\Gamma\rangle$ *for some* $\varrho \in R_D$*. Then, the problems* K-AUDIT($\Gamma \cup \{\varrho\}$) *and* K-AUDIT(Γ) *are polynomial-time equivalent.*

Proof. By the remark after Definition 2.7, each occurence of ϱ in every instance I of $\mathrm{CSP}(\Gamma \cup \{\varrho\})$ can be replaced by the corresponding collection of constraints involving only relations from $\Gamma \cup \{=_D\}$ (with possible renaming of variables to avoid name clashes). The equality constraint can then be removed by identifying variables. It is easy to see that transforming an arbitrary instance (I, v) of K-AUDIT($\Gamma \cup \{\varrho\}$) in the same way and keeping v the same gives us a polynomial-time reduction from K-AUDIT($\Gamma \cup \{\varrho\}$) to K-AUDIT(Γ). The reduction in the other direction is trivial. ∎

Theorem 3.2 *Arbitrarily choose* $\Gamma_1, \Gamma_2 \subseteq R_D$ *and assume that* Γ_1 *is finite. If* $\mathsf{Pol}(\Gamma_2) \subseteq \mathsf{Pol}(\Gamma_1)$ *then* K-AUDIT(Γ_1) *is polynomial-time reducible to* K-AUDIT(Γ_2).

Proof. Follows from Lemma 3.1, Theorem 2.8, and the obvious fact that the operator Inv is antimonotone (i.e. inclusion-reversing). ∎

Theorem 3.2 shows that the complexity of K-AUDIT(Γ) is determined by the polymorphisms of Γ. The unary polymorphisms are of special interest as is suggested by the following lemma.

Lemma 3.3 *If φ is a solution to an instance I of* $\mathrm{CSP}(\Gamma)$ *then so is $f\varphi$ for every $f \in \mathsf{Pol}_1(\Gamma)$.*

Proof. Every relation $\varrho \in \Gamma$ is invariant under f, so $(a_1, \ldots, a_k) \in \varrho$ implies $(f(a_1), \ldots, f(a_k)) \in \varrho$. Thus, $f\varphi$ is a solution whenever φ is a solution. ∎

It follows that if $\varphi(x) = a$ for some variable x in I then, for every $b \in D$ with $b = f(a)$ for some $f \in \mathsf{Pol}_1(\Gamma)$, there is another solution that maps x to b. This shows that unary polymorphisms are important in the recognition of compromised variables. For example, if $f(d) \neq d$ for some $f \in \mathsf{Pol}_1(\Gamma)$ then d cannot be the value taken by a 1-compromised variable in an instance of $\mathrm{CSP}(\Gamma)$.

To be able to state the results in forthcoming sections, we need some notation and some basic results. Let $\sqsubseteq$ denote the order on D defined by the following rule: $a \sqsubseteq b$ if and only if $f(a) = b$ for some $f \in \mathsf{Pol}_1(\Gamma)$. It is easy to see that $\sqsubseteq$ is a quasi-order (i.e. reflexive and transitive) since $\mathsf{Pol}_1(\Gamma)$ is closed under composition and contains the identity operation. It is well known and easy to show that the relation θ, such that $a\ \theta\ b$ if and only if $a \sqsubseteq b$ and $b \sqsubseteq a$, is an equivalence relation on D. Let $[a]$ denote the θ-class containing a. It is also well known and easy to show that the relation $\leq$, on the set of all θ-classes, such that $[a] \leq [b]$ if and only if $a \sqsubseteq b$, is well-defined and is a partial order. Let P denote the corresponding poset. We will often omit θ and call the elements of P classes. The intuition behind the poset P is simple: if, in some instance, a variable can take some value a in a solution then by Lemma 3.3 it also takes, in some other solution, any other value lying in the same class as a or in a class that is above $[a]$ in P. In particular, values taken by compromised variables must belong to maximal classes in P that are one-element. We have the following results:

Lemma 3.4 *Let I be an arbitrary instance of* $\mathrm{CSP}(\Gamma)$ *and $\{t_1, t_2\} \in T$ for some class T in P. If there exists a solution φ to I such that $\varphi(v) = t_1$, then there exists another solution φ' such that $\varphi'(v) = t_2$.*

Proof. By the definition of P and T, there is $f \in \mathsf{Pol}_1(\Gamma)$ such that $f(t_1) = t_2$. Therefore, $f\varphi$ is a solution to I by Proposition 2.5 and $f\varphi(v) = t_2$. ∎

Lemma 3.5 *For every class T in P, the unary relation $R_T = \bigcup\{T' \mid T \leq T'\}$ is in $\langle\Gamma\rangle$.*

Proof. Let $t' = f(t_1, \ldots, t_n)$ for some $f \in \mathsf{Pol}_n(\Gamma)$ and $t_1, \ldots, t_n \in R_T$. By the definition of R_T, there exist $f_1 \ldots f_n \in \mathsf{Pol}_1(\Gamma)$ such that $f_i(t) = t_i$ for some $t \in T$. It is easy to see that the function $f'(x) = f(f_1(x), \ldots, f_n(x))$ is a member of $\mathsf{Pol}_1(\Gamma)$ and $f'(t) = t'$. By the definition of P, we infer that t'

belongs to some class T' such that $T \leq T'$ in P, that is, $t' \in R_T$. Therefore, $R_T \in \mathsf{Inv}(\mathsf{Pol}(\Gamma))$, and, by Theorem 2.8, the result follows. ∎

A consequence of Lemma 3.5 is the following:

Corollary 3.6 *If T is a maximal class in P, then $T \in \langle \Gamma \rangle$.*

4 The Audit problem

In this section, we characterise all Γ such that K-Audit(Γ) (for fixed K) and Audit(Γ) is tractable. Note that Proposition 2.5 implies that any such characterization must, for all problems Audit(Γ), contain the tractability condition for the corresponding CSP(Γ).

Let K be fixed and let Γ be an arbitrary set of relations over some finite domain D. Let the partial order P be defined as in Section 3 and assume P contains the classes $T_1, \ldots, T_q$. For every $T \in P$, define $U(T) = \sum_{T<T'} |T'|$ and, for every $1 \leq k \leq K$, let

$$\mathcal{Z}_k = \{T \in P \mid U(T) \leq k\}.$$

If $\mathcal{Z}_k = \{Z_1, \ldots, Z_m\}$, then let $z_1, \ldots, z_m$ denote arbitrarily chosen elements in $Z_1, \ldots, Z_m$, respectively, and let $\Gamma_i = \Gamma \cup \{\{z_i\}\}$. The exact choice of $z_i \in Z_i$ is not important since if CSP($\Gamma \cup \{\{z_i\}\}$) is tractable, then CSP($\Gamma \cup \{\{z_i'\}\}$) is tractable for every $z_i' \in Z_i$ (which follows from Lemma 3.3). We also note that $\mathcal{Z}_k \subseteq \mathcal{Z}_K$ whenever $k \leq K$.

Next, we present a transformation on CSP instances that will facilitate the forthcoming proof. Loosely speaking, this transformation replaces all unary constraints of the type (v, ϱ_1) with the unary constraints (v, ϱ_2) and, moreover, forces the variables affected by this change to take the same value. Let $I = (V, D, C)$ be an arbitrary instance of CSP(Γ), and let ϱ_1, ϱ_2 be unary relations over D. We construct the CSP instance $I[\varrho_1 \to \varrho_2]$ as follows:

(1) introduce a new variable v and a constraint (v, ϱ_2),
(2) for every constraint of the form (x, ϱ_1) in C,
- remove this constraint from C,
- identify all occurences of x in C (if they exist) with v.

Clearly, the resulting instance as an instance of CSP($(\Gamma - \{\varrho_1\}) \cup \{\varrho_2\}$).

Input: An instance (I, v, k) of K-AUDIT(Γ).
Output: 'Yes' if v is k-compromised in I and 'No' otherwise.

- if $\mathcal{A} \neq \emptyset$ then answer 'no' and stop
- if $\sum_{Z_j \in \mathcal{B}} |Z_j| > k$ then answer 'no' else answer 'yes'

Fig. 1. Algorithm for solving K-AUDIT(Γ)

Theorem 4.1 *K-AUDIT(Γ) is tractable if and only if*

(a) $\mathcal{Z}_K = \emptyset$ and CSP(Γ) *is tractable; or*
(b) $\mathcal{Z}_K \neq \emptyset$ and CSP(Γ_i) *is tractable for all $1 \leq i \leq m$.*

*Furthermore, K-AUDIT(Γ) is **coNP**-complete if* CSP(Γ) *is **NP**-complete or there exists some Γ_i, $1 \leq i \leq m$, such that* CSP(Γ_i) *is **NP**-complete.*

Proof. In case (a), we see that if $\mathcal{Z}_K = \emptyset$, then a variable can be k-compromised, $k \leq K$, if and only if the given CSP instance is not satisfiable (and this can be checked in polynomial time). If CSP(Γ) is not tractable, then K-AUDIT(Γ) is not tractable by Lemma 2.5

To prove case (b), define

$$\mathcal{W}_k = \{T \in \mathcal{Z}_k \mid |T| + U(T) > k\}$$

for $1 \leq k \leq K$ and assume that $I = ((V, D, C), v, k)$ is an arbitrary instance of K-AUDIT(Γ), i.e. $k \leq K$. Let $I_j = (V, D, C \cup \{(v, \{z_j\})\})$ and define the sets $\mathcal{A}$ and $\mathcal{B}$ such that

$$\mathcal{A} = \{Z_j \in \mathcal{W}_k \mid \mathrm{Sol}(I_j) \neq \emptyset\}$$

and

$$\mathcal{B} = \{Z_j \in \mathcal{Z}_k \setminus \mathcal{W}_k \mid \mathrm{Sol}(I_j) \neq \emptyset\}.$$

We claim that the algorithm in Fig. 1 solves I in polynomial time. By assumption, CSP(Γ_i), $1 \leq i \leq m$, are tractable problems so the algorithm runs in polynomial time. The correctness of line (1) follows from the fact that if I_j is satisfiable, then v can take $|Z_j| + U(Z_j) > k$ different values. If the inequality in line (2) of the algorithm holds then, obviously, the answer is 'no'. Assume that it does not hold and, in some solution to I, v takes a value in some class $T \notin \mathcal{Z}_k$. Choose T to be maximal with this property. By the definition of $\mathcal{Z}_k$, all maximal classes in P belongs to $\mathcal{Z}_k$. Hence, there are classes in P above T, and they all belong to $\mathcal{Z}_k$, by the choice of T. Due to the test in line (1) of the algorithm, all classes above T belong to $\mathcal{Z}_k \setminus \mathcal{W}_k$, and, since $T \notin \mathcal{Z}_k$, we have $U(T) > k$. Now, by Lemma 3.3 and by the definition of P, it follows that v can take all values from classes above T. But then the inequality in line (2) of the algorithm holds, contrary to our assumption. This means that after the check in line (1) and provided the inequlaity does not hold, the values taken by v (if there any) belong to $\bigcup(\mathcal{Z}_k \setminus \mathcal{W}_k)$. Now correctness of line (2) follows from the facts that the classes in $\mathcal{Z}_k \setminus \mathcal{W}_k$ are pairwise disjoint, and that if I_j is satisfiable then v can take any of the values in Z_j.

We now prove the necessity of the condition. Assume, without loss of generality, that $\mathrm{CSP}(\Gamma_1)$ is intractable. We consider two cases:

$\underline{Z_1 \text{ is maximal.}}$ We make a polynomial-time reduction from $\mathrm{CSP}(\Gamma_1)$ to $\mathrm{CSP}(\Gamma)$ which, by Proposition 2.5, implies intractability of K-AUDIT(Γ). We observe that $Z_1 \in \langle\Gamma\rangle$ by Corollary 3.6. Let I be an arbitrary instance of $\mathrm{CSP}(\Gamma_1)$ and let $I' = I[\{z_1\} \to Z_1]$ and assume variable v is introduced by the transformation. If I' is not satisfiable, then I is not satisfiable since $\{z_1\} \subseteq Z_1$. Otherwise, $\varphi(v) = z \in Z_1$ for some $\varphi \in Sol(I')$ and I has a solution by Lemma 3.4.

$\underline{Z_1 \text{ is not maximal.}}$ We make a polynomial-time reduction from $\mathrm{CSP}(\Gamma_1)$ to the complement of K-AUDIT(Γ). We begin by defining the unary relation $R = \bigcup\{T \mid Z_1 \leq T\}$ and we note that $R \neq Z_1$ since Z_1 is not maximal. By Lemma 3.5, $R \in \langle\Gamma\rangle$, and by Lemma 3.1, we may assume that R is in Γ. Let I be an arbitrary instance of $\mathrm{CSP}(\Gamma_1)$ and let $I' = I[\{z_1\} \to R]$ and assume v to be the variable introduced by the transformation. Map I to the instance $(I', v, |R| - |Z_1|)$ of K-AUDIT(Γ) and let $k = |R| - |Z_1|$. Note that $0 < k \leq K$ by the choice of elements in $\mathcal{Z}_K$. We show that I is not satisfiable if and only if v is k-compromised in I'. If v is k-compromised in I' then v cannot be assigned any value in Z_1 (and hence not z_1) since this would imply that v could take $|R|$ different values. We conclude that I is not satisfiable. If v is not k-compromised in I', then $\varphi(v) = z \in Z_1$ for some $\varphi \in Sol(I')$. By Lemma 3.4, I has a solution.

The second part of the theorem holds since the reductions above prove **coNP**-completeness of K-AUDIT(Γ) whenever $\mathrm{CSP}(\Gamma)$ is **NP**-complete or $\mathrm{CSP}(\Gamma_i)$ is **NP**-complete for some i. ∎

We note that when $K = |D|$, i.e. when we do not assume K to be a fixed constant, the previous theorem can be modified to yield the next result.

Corollary 4.2 AUDIT(Γ) *is a tractable problem if and only if* $\mathrm{CSP}(\Gamma \cup \{\{d\}\})$ *is tractable for all* $d \in D$. *Furthermore,* AUDIT(Γ) *is* **coNP***-complete if there exists some* $d \in D$ *such that* $\mathrm{CSP}(\Gamma \cup \{\{d\}\})$ *is* **NP***-complete.*

The next corollary says that, whenever K-AUDIT(Γ) is tractable, not only can the compromised variables be recognized efficiently, but also the possible values for them can be found in polynomial time.

Corollary 4.3 *Choose* Γ *such that* K-AUDIT(Γ) *is tractable. Then, the values for all* k*-compromised,* $k \leq K$ *variables in any instance of* K-AUDIT(Γ) *can be found in polynomial time.*

Proof. Assume the algorithm in Fig. 1 has indicated that variable v is k-compromised. Then, $\{\varphi(v) \mid \varphi \in Sol(I)\} = \bigcup \mathcal{B}$. ∎

Note that if the conjecture that every CSP(Γ) is either tractable or **NP**-complete holds (and there is strong evidence that it does [4–6,12,16,17,28]), then Theorem 4.1 also gives a complete characterization of the **coNP**-complete subproblems of K-AUDIT(Γ). It was proved in [4] that, for $|D| \leq 3$, this conjecture is true and that there exists a polynomial-time algorithm which determines, for a given finite $\Gamma \subseteq R_D$, whether CSP(Γ) is tractable or **NP**-complete. We get the following dichotomy result.

Corollary 4.4 *Let* $|D| \leq 3$*. Then, for every* $\Gamma \subseteq R_D$*,* K-AUDIT(Γ) *is either tractable or* **coNP***-complete. Moreover, there is a polynomial-time algorithm which determines, for a given finite* $\Gamma \subseteq R_D$*, into which case the problem* K-AUDIT(Γ) *falls.*

A natural question at this point is whether it exists problems such that K-AUDIT is tractable and $(K+1)$-AUDIT is computationally hard. We answer this question affirmatively below.

Theorem 4.5 *For any* D *such that* $|D| \geq 3$*, and any* K *with* $1 \leq K < |D|-1$*, there exist a finite set* $\Gamma \subseteq R_D$ *such that* K-AUDIT(Γ) *is tractable, but* $(K+1)$-AUDIT(Γ) *is* **coNP***-complete.*

Proof. Assume $D = \{0, \ldots, l-1\}$ and $1 \leq K < l-1$. Let $\Gamma = \{\varrho_1, \varrho_2\}$ where the relations ϱ_1 (l-ary) and ϱ_2 (ternary) are defined as follows:

- $(x_1, \ldots, x_l) \in \varrho_1$ if and only if 1) $x_1 = 0, \ldots, x_K = K-1, x_{K+1} = K$, and 2) either $x_{K+2} = \cdots = x_l \in \{0, \ldots K\}$ or $(x_{K+2} \ldots, x_l)$ is a permutation of $\{K+1 \ldots, l-1\}$;
- $(x_1, x_2, x_3) \in \varrho_2$ if and only if either both $x_1 \neq x_2$ and $x_3 \in \{K+1, \ldots, l-1\}$, or else $x_1, x_2, x_3 \in \{0, \ldots K\}$

We will show that Γ has the required property. First, we compute $\mathsf{Pol}_1(\Gamma)$. Take an arbitrary $f \in \mathsf{Pol}_1(\Gamma)$. Since f is a polymorphism of ϱ_1 and the tuple $(0, 1, \ldots, l-1)$ belongs to ϱ_1, the tuple $(f(0), f(1), \ldots, f(l-1))$ must also belong to ϱ_1. On the other hand, it is easy to check that any unary operation f with this property is polymorphism of both ϱ_1 and ϱ_2. Hence, $\mathsf{Pol}_1(\Gamma)$ consists of all unary operations with the above property.

Then, the poset P has the following structure: it has $K+2$ classes, where the classes $\{0\}, \ldots, \{K\}$ are maximal, and $\{K+1, \ldots, l-1\}$ is the only class below them all. Note that $\mathcal{Z}_K$ consists precisely of the maximal classes in P.

Fix $f \in \mathsf{Pol}_1(\Gamma)$ such that $f(K+1) = \cdots f(l-1) = 0$ and let $f(\Gamma) = \{f(\varrho_1), f(\varrho_2)\}$ where $f(\varrho_i) = \{f(\vec{a}) \mid \vec{a} \in \varrho_i\}$, $i = 1, 2$. By [17], CSP(Γ) is polynomial-time equivalent to CSP($f(\Gamma)$). Note that, for $1 \leq i \leq K+1$, $f(\Gamma_i)$ is a set of relations on $\{0, \ldots K\}$. It is straightforward to verify that that the binary operation $\min(x, y)$ on $\{0, \ldots, K\}$ is a polymorphism of every $f(\Gamma_i)$. Hence, CSP($f(\Gamma_i)$) is tractable by [18]. This implies that CSP(Γ_i) is tractable for all $1 \leq i \leq K+1$, and thus K-AUDIT(Γ) is tractable by Theorem 4.1.

It remains to prove that $(K+1)$-AUDIT(Γ) is **coNP**-complete. Note that the class $\{K+1, \ldots, l-1\}$ belongs to $\mathcal{Z}_{K+1}$. Hence, one of the sets Γ_i that needs to be checked in Theorem 4.1 is $\Gamma \cup \{\{a\}\}$ where $a \in \{K+2, \ldots, l-1\}$. Note that the disequality relation $\neq_D$ on D belongs to $\langle \Gamma \cup \{\{a\}\} \rangle$, since

$$x \neq_D y \equiv \exists z(\varrho_2(x, y, z) \wedge z = a).$$

It is obvious that CSP($\neq_D$) corresponds precisely to the GRAPH $|D|$-COLORING problem so it is **NP**-complete. Now, the problem $(K+1)$-AUDIT(Γ) is **coNP**-complete by Lemma 3.1 and Theorem 4.1. ∎

5 Examples

As concrete examples, we will study the complexity of auditing a number of different statistical queries and give a complete classification for the Boolean auditing problem. Most examples of statistical queries are taken from [7]. Note that whenever we are considering Boolean domains, it is sufficient to consider the 1-auditing problem. Before we begin, define the relation $\text{SUM}_3^1 \subseteq \{0,1\}^3$ such that $(x, y, z) \in \text{SUM}_3^1$ if and only if $x + y + z = 1$. The example at the end of Subsection 2.2 implies **coNP**-completeness of 1-AUDIT($\{\text{SUM}_3^1\}$).

Max and Min queries

Kleinberg et al. [21] have shown that the 1-auditing problem for MAX queries over real-valued data is tractable. We complement this result by showing that the K-auditing problem is tractable over arbitrary finite domains $D =$

$\{1,\ldots,d\}$. Define the relation $\text{MAX}_m^t \subseteq D^m$ such that

$$(x_1,\ldots,x_m) \in \text{MAX}_m^t \text{ if and only if } \text{MAX}\{x_1,\ldots,x_m\} = t.$$

Assume that Γ consists of $\{\text{MAX}_m^t \mid m > 0 \text{ and } 1 \leq t \leq d\}$ together with the unary relations $\{1\},\ldots,\{d\}$. If $\text{CSP}(\Gamma)$ is tractable, then K-AUDIT(Γ) is tractable by Corollary 4.2. To prove tractability of $\text{CSP}(\Gamma)$, we note that if the binary max operation is in $\mathsf{Pol}(\Gamma)$, then $\text{CSP}(\Gamma)$ is tractable [18]. The unary relations in Γ are obviously invariant under max since it is an idempotent function. Now, arbitrarily choose a function $\text{MAX}_m^t \in \Gamma$ and arbitrarily choose two tuples $(x_1,\ldots,x_m),(x'_1,\ldots,x'_m)$ in MAX_m^t. If MAX_m^t is invariant under max, then the tuple $(\text{MAX}(x_1,x'_1),\ldots,\text{MAX}(x_m,x'_m))$ must be in MAX_m^t. This is obviously true since the largest element among $x_1,\ldots,x_m,x'_1,\ldots,x'_m$ have value t.

By using similar techniques, it follows that auditing MIN queries is also tractable. However, if MAX and MIN queries are mixed, then auditing is hard: Consider the Boolean domain $\{0,1\}$. The $\text{MAX}_3^1(x,y,z)$ relation holds if and only if the Boolean clause $(x \vee y \vee z)$ is satisfied. Similarly, $\text{MIN}_3^0(x,y,z)$ holds if and only if the clause $(\neg x \vee \neg y \vee \neg z)$ is satisfied. Consequently, Theorem 4.1 and Theorem 2.11 imply **coNP**-completeness of 1-AUDIT$(\{\text{MAX}_3^1,\text{MIN}_3^0\})$.

Sum queries modulo $|D|$

We know that the auditing problem for sum queries is **coNP**-complete, cf. Kleinberg et al [21]. We will now show that the K-auditing problem is tractable if we consider sum queries modulo $|D|$. Assume $D = \{0,\ldots,|D|-1\}$. Define the relation $\text{MSUM}_m^t \subseteq D^m$ such that

$$(x_1,\ldots,x_m) \in \text{MSUM}_m^t \text{ if and only if} x_1+\ldots+x_m \equiv t \pmod{|D|}.$$

Assume that Γ consists of $\{\text{MSUM}_m^t \mid m > 0 \text{ and } t \geq 0\}$ together with the unary relations $\{0\},\ldots,\{|D|-1\}$. To prove tractability of K-AUDIT(Γ), we note that $(D,+)$ is an abelian group and tractability of $\text{CSP}(\Gamma)$ (and consequently K-AUDIT(Γ)) follows from [14]. The tractability of $\text{CSP}(\Gamma)$ can also be proved by noting that the affine operation $M(x,y,z) = x - y + z$ is in $\mathsf{Pol}(\Gamma)$ and using results from [18].

Mean queries

We will study the complexity of auditing three different mean queries. Let $S = (a_1,\ldots,a_n)$ be a finite sequence of real numbers and define the *arithmetical mean* (AMEAN(S)), the *geometrical mean* (GMEAN(S)) and the *harmonic mean*

($\textsc{hmean}(S)$) as

$$\frac{1}{n}\sum_{i=1}^{n} a_i, \quad \left(\prod_{i=1}^{n} a_i\right)^{1/n} \quad \text{and} \quad \frac{1}{(1/n)\sum_{i=1}^{n} 1/a_i},$$

respectively. We define the relation $\textsc{amean}_m^t \subseteq D^m$ such that

$$(x_1,\ldots,x_m) \in \textsc{amean}_m^t \text{ if and only if } \textsc{amean}(x_1,\ldots,x_m) = t$$

and relations $\textsc{gmean}_m^t$ and $\textsc{hmean}_m^t$ analogously.

We begin by considering $\textsc{amean}$ on the domain $D = \{0,1\}$. It is obvious that $\textsc{amean}_3^{1/3} = \textsc{sum}_3^1$ and **coNP**-completeness of 1-auditing $\textsc{amean}$ queries follows immediately. In the case of $\textsc{gmean}$, we will consider two examples with different domains leading to different complexities. First, let $D = \{0,1\}$ and note that $\textsc{gmean}(a_1,\ldots,a_n) \in \{0,1\}$ for all choices of $a_1,\ldots,a_n$. The relation $\textsc{gmean}_m^1(a_1,\ldots,a_n)$ holds if and only if $a_1 = \ldots = a_n = 1$ while $\textsc{gmean}_m^0(a_1,\ldots,a_n)$ holds if and only if at least one $a_i = 0$. Interpreting these relations as Boolean clauses give us the following: $\textsc{gmean}_m^1(a_1,\ldots,a_n) \Leftrightarrow a_1 \wedge \ldots \wedge a_n$ and $\textsc{gmean}_m^0(a_1,\ldots,a_n) \Leftrightarrow (\neg a_1 \vee \ldots \vee \neg a_n)$. Thus, the $\textsc{gmean}_m^t$ relations together with the unary relations $\{0\}$ and $\{1\}$ is contained in the Horn fragment of propositional logic (and is consequently tractable). Corollary 4.2 implies tractability of the 1-auditing problem. Let us now consider the domain $D = \{1,2\}$ instead. Let $\alpha = \sqrt[3]{2}$. It is easy to see that the relation $\textsc{gmean}_3^\alpha$ is isomorphic to the relation $\textsc{sum}_3^1$ under the isomorphism $f(1) = 0$ and $f(2) = 1$ and **coNP**-completeness of 1-auditing follows. Since $\textsc{hmean}$ is not defined for domains containing zero, we continue using the domain $D = \{1,2\}$. It is easy to see that $\textsc{hmean}_3^{5/4}$ is isomorphic to $\textsc{sum}_3^1$ under the same isomorphism f so **coNP**-completeness follows in this case, too.

Median queries
Define the relation $\textsc{median}_m^t \subseteq D^m$ such that $(x_1,\ldots,x_m) \in \textsc{median}_m^t$ if and only if the $\lceil m/2 \rceil$th largest element in $(x_1,\ldots,x_m)$ equals t. Let us consider the case $D = \{0,1\}$; then, $\textsc{median}_3^0 = \{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$ and $\textsc{median}_3^1 = \{(0,1,1),(1,0,1),(1,1,0),(1,1,1)\}$. By Schaefer's dichotomy result (Theorem 2.11), $\mathrm{CSP}(\{\textsc{median}_3^0,\textsc{median}_3^1\})$ is **NP**-complete and the 1-auditing problem for median queries in **coNP**-complete by Theorem 4.1.

Boolean attributes
We give a complete classification of the 1-auditing problem when the domain

is Boolean, i.e. $|D| = 2$. Let $\Gamma \subseteq R_{\{0,1\}}$. We claim that

(1) if $\mathsf{Pol}(\Gamma)$ contains both constant operations, 0 and 1, or one of the operations (d)-(g) from Proposition 2.9 then 1-AUDIT(Γ) is tractable;
(2) otherwise 1-AUDIT(Γ) is **coNP**-complete.

Before the proof, we observe that the conditions above can be verified efficiently for any finite $\Gamma \subseteq R_{\{0,1\}}$. We prove the two cases as follows:

1) If $\mathsf{Pol}(\Gamma)$ contains both constant operations, then $\mathsf{Pol}_1(\Gamma) = \{0, 1\}$ and the quasi-order defined in Section 3 satisfies $0 \not\sqsubseteq 1$ and $1 \not\sqsubseteq 0$. Thus, $\mathcal{Z} = \emptyset$ and no variable can be compromised. In cases (d)-(g), the problem $\mathrm{CSP}(\Gamma \cup \{\{0\}, \{1\}\})$ is tractable by Theorem 2.11, and tractability of 1-AUDIT follows from Corollary 4.2.

2) Assume that $1 \in \mathsf{Pol}(\Gamma)$, $0 \notin \mathsf{Pol}(\Gamma)$ and no operation from cases (d)-(g) is in $\mathsf{Pol}(\Gamma)$. In this case, $\mathrm{CSP}(\Gamma)$ is tractable by Theorem 2.11. We have $\mathsf{Pol}_1(\Gamma) = \{\mathrm{id}_{\{0,1\}}, 1\}$. Therefore the quasi-order defined before Theorem 4.1 satisfies $0 \sqsubseteq 1$ and $1 \not\sqsubseteq 0$, and we have $\mathcal{Z} = \{\{0\}, \{1\}\}$. It follows from Theorem 2.11 that $\mathrm{CSP}(\Gamma \cup \{\{0\}\})$ is **NP**-complete since the relation $\{0\}$ is not invariant under constant operation 1. Consequently, Theorem 4.1 implies that 1-AUDIT(Γ) is **coNP**-complete.

In the remaining cases, $\mathrm{CSP}(\Gamma)$ is **NP**-complete by Theorem 2.11 and we conclude that 1-AUDIT(Γ) is **coNP**-complete by Theorem 4.1.

6 Conclusion

We have studied the auditing problem for databases supporting statistical queries. Under the assumption that the attributes are finite, we have identified all classes of statistical queries having a tractable auditing problem. We have also proved that the problem is **coNP**-complete in all other cases if a certain conjecture is true. The results were obtained by exploiting connections between auditing and constraint satisfaction, and using certain algebraic techniques.

Acknowledgements

Peter Jonsson is partially supported by the Center for Industrial Information Technology (CENIIT) under grant 04.??, and the Swedish Research Council (VR) under grant 621-2002-4126. Andrei Krokhin is partially supported by the UK EPSRC grant GR/R29598 and by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

References

[1] N. Adam and J. Wortmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys*, 5(3):515–556, 1989.

[2] L. Beck. A security mechanism for statistical databases. *ACM Transactions on Database Systems*, 5(3):316–338, 1980.

[3] B. Bollobás, C. Borgs, J. Chayes, J. Kim, and D. Wilson. The scaling window of the 2-SAT transition. *Random Structures and Algorithms*, 18(3):201–256, 2001.

[4] A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings of 43rd IEEE Symposium on Foundations of Computer Science, FOCS'02*, pages 649–658, 2002.

[5] A. Bulatov, A. Krokhin, and P. Jeavons. Constraint satisfaction problems and finite algebras. In *Proceedings 27th International Colloquium on Automata, Languages and Programming, ICALP'00*, volume 1853 of *Lecture Notes in Computer Science*, pages 272–282. Springer-Verlag, 2000.

[6] A. Bulatov, A. Krokhin, and P. Jeavons. The complexity of maximal constraint languages. In *Proceedings 33rd ACM Symposium on Theory of Computing, STOC'01*, pages 667–674, 2001.

[7] S. Castano, M. Fugini, G. Martella, and P. Samarati. *Database Security*. ACM Press/Addison-Wesley, New York, NY, USA, 1995.

[8] F. Chin and G. Ösoyoglu. Statistical database design. *ACM Transactions on Database Systems*, 6(1):113–139, 1981.

[9] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. 2001.

[10] J. Culberson and I. Gent. Frozen development in graph coloring. *Theoretical Computer Science*, 265(1-2):227–264, 2001.

[11] D. Dobkin, A. Jones, and R. Lipton. Secure databases: protection against user influence. *ACM Transactions on Database Systems*, 4(1):97–106, 1979.

[12] T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1998.

[13] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[14] M. Goldmann and A. Russell. The complexity of solving equations over finite groups. *Information and Computation*, 178(1):253–262, 2002.

[15] G. Gottlob, L. Leone, and F. Scarcello. Hypertree decomposition and tractable queries. *Journal of Computer and System Sciences*, 64(3):579–627, 2002.

[16] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Ser.B*, 48:92–110, 1990.

[17] P. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

[18] P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.

[19] P. Jeavons, D. Cohen, and M. Gyssens. How to determine the expressive power of constraints. *Constraints*, 4:113–131, 1999.

[20] J. B. Kam and J. Ullman. A model of statistical databases and their security. *ACM Transactions on Database Systems*, 2(1):1–10, 1977.

[21] J. Kleinberg, C. H. Papadimitriou, and P. Raghavan. Auditing Boolean attributes. *Journal of Computer and System Sciences*, 66:244–253, 2003.

[22] P. Kolaitis and M. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61:302–332, 2000.

[23] A. Krokhin and P. Jonsson. Recognizing frozen variables in constraint satisfaction problems. Research Report 62, Electronic Colloquium on Computational Complexity, 2003.

[24] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems*, 10(3):395–411, 1985.

[25] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. 2+p-SAT: Relation of typical-case complexity to the nature of phase transition. *Random Structures and Algorithms*, 15(3-4):414–440, 1999.

[26] R. Pöschel and L. Kalužnin. *Funktionen- und Relationenalgebren.* DVW, Berlin, 1979.

[27] E. Post. *The two-valued iterative systems of mathematical logic*, volume 5 of *Annals Mathematical Studies.* Princeton University Press, 1941.

[28] T. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th ACM Symposium on Theory of Computing, STOC'78*, pages 216–226, 1978.

[29] J. Singer, I. Gent, and A. Smaill. Backbone fragility and the local search cost peak. *Journal of Artificial Intelligence Research*, 12:235–270, 2000.