Reasoning about disclosure in data integration in the presence of source constraints

Anonymous

Abstract

Data integration systems allow users to access data sitting in multiple sources by means of queries over a global schema, related to the sources via mappings. Datasources often contain sensitive information, and thus an analysis is needed to verify that a schema satisfies a privacy policy, given as a set of queries whose answers should not be accessible to users. Such an analysis should take into account not only knowledge that an attacker may have about the mappings, but also what they may know about the semantics of the sources. In this paper, we show that *source constraints* can have a dramatic impact on disclosure analysis. We study the problem of determining whether a given data integration system discloses a source query to an attacker in the presence of constraints, providing both lower and upper bounds on source-aware disclosure analysis.

1 Introduction

In data integration, users are shielded from the heterogeneity of multiple datasources by querying via a *global schema*, which provides a unified vocabulary. The relationship between sources and the user-facing schema are specified declaratively via *mapping rules*. In data integration systems based on knowledge representation techniques, users pose queries against the global schema, and these queries are answered using data in the sources and background knowledge. The computation of the answers involves reasoning based on the query, the mappings, and any additional semantic information that is known on the global schema.

Data integration brings with it the danger of disclosing information that data owners wish to keep confidential. In declarative data integration, detection of privacy violations is complex: although explicit access to source information may be masked by the global schema, an attacker can infer source facts via reasoning with schema and mapping information.

Example 1. *We consider an information integration setting for a hospital, which internally stores the following data:*

Predicate	*Meaning*
$\mathsf{IsOpen}(b, t)$	*building b is open on date t*
$\mathsf{PatBdlg}(p, b)$	*patient p is present in building b*
$\mathsf{PatSpec}(p, s)$	*patient p was treated for specialty s*
$\mathsf{PatDoc}(p, d)$	*patient p was treated by doctor d*
$\mathsf{DocBldg}(d, b)$	*doctor d is associated with building b*
$\mathsf{DocSpec}(d, s)$	*doctor d is associated with specialty s*

The hospital publishes the following data: $\mathsf{OpenHours}(b, t)$ *giving opening times t for building b,* $\mathsf{VisitingHours}(p, t)$ *giving times t when a given patient p can be visited, and* $\mathsf{DocList}(d, s, b)$ *listing the doctors d with their specialty s and their building b. Formally the data being exposed is given by the following mappings:*

$$\begin{aligned} \mathsf{IsOpen}(b, t) &\rightarrow \mathsf{OpenHours}(b, t) \\ \mathsf{PatBdlg}(p, b) \wedge \mathsf{IsOpen}(b, t) &\rightarrow \mathsf{VisitingHours}(p, t) \\ \mathsf{DocSpec}(d, s) \wedge \mathsf{DocBldg}(d, b) &\rightarrow \mathsf{DocList}(d, s, b) \end{aligned}$$

Prior work [Benedikt *et al.*, 2018] has studied disclosure in knowledge-based data integration, with an emphasis on the role of semantic information on the *global schema* – in the form of ontological rules that relate the global schema vocabulary. The presence of an ontology can assist in privacy, since distinctions in the source data may become indistinguishable in the ontology. More dangerous from the point of view of protecting information is *semantic information about sources*. For example, the sources in a data integration setting will generally overlap: that is, they will satisfy *referential integrity constraints*, saying that data items in one source link to items in another source. Such constraints should be assumed as public knowledge, and with that knowledge the attacker may be able to infer information that was intended to be secret.

Example 2. *Continuing Example 1, suppose that we know that each patient has a doctor specialized in their condition, which can be formalized as:*

$$\mathsf{PatDoc}(p, d) \rightarrow \exists s\, \mathsf{PatSpec}(p, s) \wedge \mathsf{DocSpec}(d, s)$$

And that we also know that when a patient is in a building, they must have a doctor there:

$$\mathsf{PatBdlg}(p, b) \rightarrow \exists d\, \mathsf{PatDoc}(p, d) \wedge \mathsf{DocBldg}(d, b)$$

Due to the presence of these constraints, there can be a disclosure of the relationship of patient to speciality

$\mathsf{PatSpec}(p, s)$. *Indeed, an attacker can see the* VisitingHours *for* p*, and from this, along with* OpenHours*, they can sometimes infer the building* b *where* p *is treated (e.g. if* b *has a unique set of open hours). From this they may be able to infer, using* DocList*, the specialty that* p *has been treated for – for example, if all the doctors in* b *share a specialty.*

In this work, we perform a detailed examination of the role of source constraints in disclosing information in the context of data integration. We focus on mappings from the sources given by universal Horn rules, where the global schema comes with no constraints. Since our disclosure problem requires reasoning over all sources satisfying the constraints, we need a constraint formalism that admits effective reasoning. We will look at a variety of well-studied rule-based formalisms, with the simplest being referential constraints, and the most complex being the *frontier-guarded rules* [Baget *et al.*, 2011]. While decidability of our disclosure problems will follow from prior work [Benedikt *et al.*, 2016], we will need new tools to analyze the complexity of the problem. In Section 3, we give reductions of disclosure problems to the *query entailment problem* that is heavily-studied in knowledge representation. While a naïve application of the reduction allows us only to conclude very pessimistic bounds, a more fine-grained analysis, combined with some recent results on CQ entailment, will allow us to get much better bounds, in some cases ensuring tractability. In Section 4, we complement these results with lower bounds. Both the upper and lower bounds revolve around a complexity analysis for reasoning with *guarded existential rules and a restricted class of equality rules, where the rule head compares a variable and a distinguished constant*. We believe this exploration of limited equality rules can be productive for other reasoning problems.

Overall we get a complete picture of the complexity of disclosure in the presence of source constraints for many natural classes: see Tables 1 in Section 6 for a summary of our bounds. Full proofs are available at the address https://hal.inria.fr/hal-02145369.

2 Preliminaries

We adopt standard notions from function-free first-order logic over a vocabulary of relational names. An *instance* is a finite set of facts. By a *query* we always mean a *conjunctive query* (CQ), which is a first-order formula of the form $\exists \vec{x} \bigwedge A_i$, where each A_i is an atom. The *arity* of a CQ is the number of its free variables, and CQs of arity 0 are *Boolean*.

Data integration. Assume that the relational names in the vocabulary are split into two disjoint subsets: *source* and *global schema*. The *arity* of such a schema is the maximal arity of its relational names. We consider a set $\mathcal{M}$ of *mapping rules* between source relations and a global schema relation $\mathcal{T}$ given. We focus on rules $\phi(\vec{x}, \vec{y}) \rightarrow \mathcal{T}(\vec{x})$ where ϕ is a conjunctive query, there are no repeated variables in $\mathcal{T}(\vec{x})$, and where each global schema relation $\mathcal{T}$ is associated with *exactly one rule*. Such rules are sometimes called "GAV mappings" in the database literature [Lenzerini, 2002], and the unique ϕ associated to a global relation $\mathcal{T}$ is referred to as the *definition* of $\mathcal{T}$. The rules are *guarded* ($\mathcal{M} \in \mathsf{GuardedMap}$) if for every rule, there exists an atom in the antecedent ϕ that contains all the variables of ϕ. The rules are *atomic* ($\mathcal{M} \in \mathsf{AtomMap}$) if each ϕ consists of a single atom, and they are *projection maps* ($\mathcal{M} \in \mathsf{ProjMap}$) if each ϕ is a single atom with no repeated variables. Given an instance $\mathcal{D}$ for the source relations, the *image of* $\mathcal{D}$ *under mapping* $\mathcal{M}$, denoted $\mathcal{M}(\mathcal{D})$, is the instance for the global schema consisting of all facts $\{\mathcal{T}(\vec{c}) \mid \mathcal{D} \models \exists \vec{y}\ \phi(\vec{c})\}$, where ϕ is the definition of $\mathcal{T}$.

Source constraints. We consider restrictions on the sources in the form of rules. A *tuple-generating dependency (TGD)* is a universally quantified sentence of the form $\varphi(\mathbf{x}, \mathbf{z}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$, where the *body* $\varphi(\mathbf{x}, \mathbf{z})$ and the *head* $\psi(\mathbf{x}, \mathbf{y})$ are conjunctions of atoms such that each term is either a constant or a variable in $\mathbf{x} \cup \mathbf{z}$ and $\mathbf{x} \cup \mathbf{y}$, respectively. Variables $\mathbf{x}$, common to the head and body, are called the *frontier variables*. A *frontier-guarded TGD* (FGTGD) is a TGD in which there is an atom of the body that contains every frontier variable. We focus on FGTGDs because they have been heavily studied in the database and knowledge representation community, and it is known that many computational problems involving FGTGDs are decidable [Baget *et al.*, 2011]. In particular this is true of the *query entailment problem*, which asks, given a finite collection of facts $\mathcal{F}$, a finite set Σ of sentences, and a CQ Q, whether $\mathcal{F} \wedge \Sigma$ entails Q. We use $\mathsf{QEntail}(\mathcal{F}, \Sigma, Q)$ to denote an instance of this problem and also say that "$\mathcal{F}$ entails Q w.r.t. constraints Σ". A special case of FGTGDs are *Guarded TGDs* (GTGDs), in which there is an atom containing all body variables. These specialize further to *linear TGDs* (LTGDs), whose body consists of a single atom; and even further to *inclusion dependencies* (IncDeps), a linear TGD with a single atom in the head, in which no variable occurs multiple times in the body, and no variable occurs multiple times in the head. Even IncDeps occur quite commonly: for example, the source constraints of Example 2 can be rewritten as IncDeps. The most specialized class we study are the *unary* IncDep*s*:(UIDs), which are IncDeps with at most one frontier variable.

Queries and disclosure. The sensitive information in a data integration setting is given by a CQ p over the source schema, which we refer to as the *policy*. Intuitively, disclosure of sensitive information occurs in a source instance $\mathcal{D}$ whenever the attacker can infer from the image $\mathcal{M}(\mathcal{D})$ that p holds of a tuple in $\mathcal{D}$. Formally, we say an instance $\mathcal{V}$ for the global schema is *realizable*, with respect to mappings $\mathcal{M}$ and source constraints Σ_{Source} if there is some source instance $\mathcal{D}$ that satisfies Σ_{Source} such that $\mathcal{M}(\mathcal{D}) = \mathcal{V}$. For a realizable $\mathcal{V}$, the set of such $\mathcal{D}$ are the *possible source instances for* $\mathcal{V}$. A query result $p(\vec{t})$ is *disclosed* at $\mathcal{V}$ if $p(\vec{t})$ holds on all possible source instances for $\mathcal{V}$. A query p *admits a disclosure* (for mappings $\mathcal{M}$ and source constraints Σ_{Source}) if there is some realizable instance $\mathcal{V}$ and binding $\vec{t}$ for the free variables of p for which $p(\vec{t})$ is disclosed. In this terminology, the conclusion of Example 2 was that policy $\mathsf{PatSpec}(p, s)$ admits a disclosure with respect to the constraints and mappings. For a class of constraints $\mathcal{C}$, a class of mappings Map, a class of policies Policy, we write $\mathsf{Disclose}_{\mathsf{C}}(\mathcal{C}, \mathsf{Map})$ to denote the problem of determining whether a policy (a CQ, unless other-

wise stated) admits a disclosure for a set of mappings in Map and a set of source constraints in $\mathcal{C}$. Given $\Sigma_{\mathsf{Source}}, \mathcal{M}$ and a CQ p, the corresponding instance of this problem is denoted by $\mathsf{Disclose}(\mathcal{C}, \mathcal{M}, p)$. In this paper we will focus on disclosure for queries and constraints *without constants*, although our techniques extend to the setting with constants, as long as distinct constants are not assumed to be unequal.

3 Reducing Disclosure to Query Entailment

Our first goal is to provide a reduction from $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{TGD}, \mathsf{Map})$ to a finite collection of standard query entailment problems. For simplicity we will restrict to Boolean queries p in stating the results, but it is straightforward to extend the reductions and results to the non-Boolean case. We first recall a prior reduction of $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{TGD}, \mathsf{Map})$ to a more complex problem, the *hybrid open and closed world query answering problem* [Lutz *et al.*, 2013; Lutz *et al.*, 2015; Franconi *et al.*, 2011], denoted HOCWQ. HOCWQ takes as input a set of facts $\mathcal{F}$, a collection of constraints Σ, a Boolean query Q, and additionally a subset $\mathcal{C}$ of the vocabulary. A *possible world* for such $\mathsf{HOCWQ}(\mathcal{F}, \Sigma, Q, \mathcal{C})$ is any instance $\mathcal{D}$ containing $\mathcal{F}$, satisfying Σ, and such that for each relation $C \in \mathcal{C}$, the C-facts in $\mathcal{D}$ are the same as the C-facts in $\mathcal{F}$. $\mathsf{HOCWQ}(\mathcal{F}, \Sigma, Q, \mathcal{C})$ holds if Q holds in every possible world. Note that the query entailment problem is a special case of HOCWQ, where $\mathcal{C}$ is empty.

Given a set of mapping rules $\mathcal{M}$ of the form $\phi(\vec{y}, \vec{x}) \rightarrow \mathcal{T}(\vec{x})$, we let $\mathcal{G}(\mathcal{M})$ be the set of global schema predicates, and let $\Sigma_{\mathcal{M}}(\mathcal{M})$ be the mapping rules, considered as bidirectional constraints between global schema predicates and sources.

We now recall one of the main results of [Benedikt *et al.*, 2016]:

Theorem 1. *There is an instance $\mathcal{D}'$ computable in linear time from $\Sigma_{\mathsf{Source}}, \mathcal{M}, p$, such that $\mathsf{Disclose}(\Sigma_{\mathsf{Source}}, \mathcal{M}, p)$ holds if and only if $\mathsf{HOCWQ}(\mathcal{D}', \Sigma_{\mathsf{Source}} \cup \Sigma_{\mathcal{M}}(\mathcal{M}), p, \mathcal{G}(\mathcal{M}))$ holds.*

In fact, the arguments in [Benedikt *et al.*, 2016] show that $\mathcal{D}'$ can be taken to be a very simple instance, the *critical instance* over the global schema $\mathcal{G}(\mathcal{M})$ denoted $\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}}$ where $\mathcal{D}^{\mathcal{S}}_{\mathsf{Crit}}$, for $\mathcal{S}$ a set of predicates, denotes the instance that mentions only a single element c_{Crit}, and contains, for each relation R in $\mathcal{S}$ of arity n, the fact $R(c_{\mathsf{Crit}}, \ldots c_{\mathsf{Crit}})$.

Corollary 1. $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{FGTGD}, \mathsf{CQMap})$ *is in* 2ExpTime.

Proof. The non-classical aspect of HOCWQ comes into play with rules of $\Sigma_{\mathcal{M}}(\mathcal{M})$ of form $\phi(\vec{x}, \vec{y}) \rightarrow \mathcal{T}(\vec{x})$. But in the context of $\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}}$, these can be rewritten as *single-constant equality rules* (SCEQrules) $\phi(\vec{x}, \vec{y}) \rightarrow \bigwedge_i x_i = c_{\mathsf{Crit}}$. Such rules remain in the Guarded Negation Fragment of first-order logic, which also subsumes FGTGDs, while having a query entailment problem in 2ExpTime [Bárány *et al.*, 2015]. □

We now want to conduct a finer-grained analysis, looking for cases that give lower complexity. To do this we will transform further into a classical query entailment problem. This will require a transformation of our query p, a transformation of our source constraints and mappings into a new set of constraints, and a transformation of the instance $\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}}$. The idea of the transformation is that we remove the SCEQrules that are implicit in the HOCWQ problem, replacing them with constraints and queries that reflect all the possible impacts the rules might have on identifying two variables.

We first describe the transformation of the query and the constraints. They will involve introducing a new unary predicate $\mathsf{IsCrit}(x)$; informally this states that x is equal to c_{Crit}. Consider a CQ $Q = \exists \vec{y} \bigwedge A_i$. An *annotation* of Q is a subset of Q's variables. Given an annotation Annot of Q, we let Q_{Annot} be the query obtained from Q by performing the following operation for each v in Annot: for all occurrences j of v except the first one, replacing v with a fresh variable v_j; and adding conjuncts $\mathsf{IsCrit}(v_j)$ as well as $\mathsf{IsCrit}(v)$ to Q_{Annot}. A *critical-instance rewriting* of a CQ Q is a CQ obtained by applying the above process to Q for any annotation. We write $Q_{\mathsf{Annot}} \in \mathsf{CritRewrite}(Q)$ to indicate that Q_{Annot} is such a rewriting.

To transform the mapping rules and constraints to a new set of constraints using $\mathsf{IsCrit}(x)$, we lift the notion of critical-instance rewriting to TGDs in the obvious way: a critical-instance rewriting of a TGD σ (either in Σ_{Source} or $\Sigma_{\mathcal{M}}(\mathcal{M})$), is the set of TGDs formed by applying the above process to the body of σ. We write $\sigma_{\mathsf{Annot}} \in \mathsf{CritRewrite}(\Sigma)$ to indicate that σ_{Annot} is a critical-instance rewriting for a $\sigma \in \Sigma$, and similarly for mappings. For example, the second mapping rule in Example 1 has several rewritings; one of them will change the rule body to $\mathsf{PatBdlg}(p, b) \wedge \mathsf{IsOpen}(b', d) \wedge \mathsf{IsCrit}(b) \wedge \mathsf{IsCrit}(b')$.

Our transformed constraints will additionally use the set of constraints $\mathsf{IsCrit}(\mathcal{M})$, including all rules:

$$\mathcal{T}(x_1 \ldots x_n) \rightarrow \mathsf{IsCrit}(x_i)$$

where $\mathcal{T}$ ranges over the global schema and $1 \leq i \leq n$. Informally $\mathsf{IsCrit}(\mathcal{M})$ states that all elements in the mapping image must be c_{Crit}. We also need to transform the instance, using a source instance with "witnesses for the target facts". Consider a fact $\mathcal{T}(c_{\mathsf{Crit}} \ldots c_{\mathsf{Crit}})$ in $\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}}$ formed by applying a mapping rule $\bigwedge_i A_i(\vec{x}_i, \vec{y}_i) \rightarrow \mathcal{T}(\vec{x})$ in $\mathcal{M}$. The *set of witness tuples for $\mathcal{T}(\vec{x})$* is the set $A_i(\vec{c})$, where $\vec{c}$ contains c_{Crit} in each position containing a variable x_j and containing a constant c_{y_j} in every position containing a variable y_j. That is the witness tuples are witnesses for the fact $\mathcal{T}(c_{\mathsf{Crit}} \ldots c_{\mathsf{Crit}})$, where each existential witness is chosen fresh. Let $\mathsf{Hide}_{\mathcal{M}}(\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}})$ be the instance formed by taking the witness tuples for every fact $\mathcal{T}(c_{\mathsf{Crit}} \ldots c_{\mathsf{Crit}}) \in \mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}}$.

We are now ready to state the reduction of the disclosure problem to query entailment:

Theorem 2. $\mathsf{Disclose}(\Sigma_{\mathsf{Source}}, \mathcal{M}, p)$ *holds exactly when there is a $p_{\mathsf{Annot}} \in \mathsf{CritRewrite}(p)$ such that $\mathsf{Hide}_{\mathcal{M}}(\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}})$ entails p_{Annot} w.r.t. constraints:*

$$\mathsf{CritRewrite}(\Sigma_{\mathsf{Source}}) \cup \mathsf{CritRewrite}(\mathcal{M}) \cup \mathsf{IsCrit}(\mathcal{M})$$

Note that Theorem 2 does not give a polynomial time reduction: both $\mathsf{CritRewrite}(\Sigma_{\mathsf{Source}})$ and $\mathsf{CritRewrite}(\mathcal{M})$ can

contain exponentially many rewritings, and further there can be exponentially many rewritings in $\mathsf{CritRewrite}(p)$.

However, the algorithm does give us a better bound in the case of Guarded TGDs with bounded arity.

Corollary 2. *If we bound the arity of schema relations, then* $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{GTGD}, \mathsf{GuardedMap})$ *is in* EXPTIME.

Proof. First, by introducing additional intermediate relations and source constraints, we can assume that $\mathcal{M}$ contains only projection mappings. Thus we can guarantee that $\mathsf{CritRewrite}(\mathcal{M})$ just contains the rules in $\mathcal{M}$. By introducing intermediate relations and additional source constraints, we can also assume that each $\mathsf{GTGD} \in \Sigma_{\mathsf{Source}}$ has a body with at most two atoms. Since the arity of relations is fixed, the size of such 1- or 2-atom bodies is fixed as well. From this we see that the number of constraints in any $\mathsf{CritRewrite}(\sigma)$ is polynomial. The reduction in Theorem 2 thus gives us exponentially many GTGD entailment problems of polynomial size. Since entailment over Guarded TGDs with bounded arity is in EXPTIME [Calì *et al.*, 2013], we can conclude. □

3.1 Refinements of the Reduction to Identify Lower Complexity Cases

In order to lower the complexity to EXPTIME *without* bounding the arity, we refine the construction of the function $\mathsf{CritRewrite}(\sigma)$ in the case where σ is a linear TGD, providing a function $\mathsf{CritRewrite}_{\text{PTIME}}(\sigma)$ that constructs only *polynomially many rewritten constraints*.

Let $\sigma = B(\vec{x}) \rightarrow \exists \vec{y}\, H(\vec{z})$ be a linear TGD with relation B of arity k, and suppose $\vec{x}$ contains d distinct free variables $V = \{v_1 \dots v_d\}$. Let P be the set of pairs (e, f) with $e < f \leq k$ such that the same variable v_i sits at positions e and f in $\vec{x}$. We order P as $(e_0, f_0) \dots (e_h, f_h)$; for each (e, f) that is not the initial pair (e_0, f_0), we let $(e, f)^-$ be its predecessor in the linear order.

We let $B_{e,f}$ denote new predicates of arity k for each $(e, f) \in P$. Let $\vec{w}$ be a set of k distinct variables, and $\vec{w}^{i=j}$ be formed from $\vec{w}$ by replacing w_j with w_i. We begin the construction of $\mathsf{CritRewrite}_{\text{PTIME}}(\sigma)$ with the constraints: $B(\vec{w}^{e_0=f_0}) \rightarrow B_{e_0,f_0}(\vec{w}^{e_0=f_0})$ and $B(\vec{w}) \wedge \mathsf{IsCrit}(w_{e_0}) \wedge \mathsf{IsCrit}(w_{f_0}) \rightarrow B_{e_0,f_0}(\vec{w})$.

For each (e, f) with a predecessor $(e, f)^- = (e', f')$, we add to $\mathsf{CritRewrite}_{\text{PTIME}}(\sigma)$ the following constraints: $B_{e',f'}(\vec{w}^{e=f}) \rightarrow B_{e,f}(\vec{w}^{e=f})$ and $B_{e',f'}(\vec{w}) \wedge \mathsf{IsCrit}(w_e) \wedge \mathsf{IsCrit}(w_f) \rightarrow B_{e,f}(\vec{w})$.

Letting e_h, f_h the final pair in P, we add to $\mathsf{CritRewrite}_{\text{PTIME}}(\sigma)$ the constraint $B_{e_h,f_h}(\vec{x}') \rightarrow \exists \vec{y}\, H(\vec{z})$ where $\vec{x}'$ is obtained from $\vec{x}$ by replacing all but the first occurrence of each variable v by a fresh variable.

If Σ_{Source} consists of LTGDs, we let $\mathsf{CritRewrite}_{\text{PTIME}}(\Sigma_{\mathsf{Source}})$ be the result of applying this process to every $\sigma \in \Sigma_{\mathsf{Source}}$. Similarly, if $\mathcal{M}$ consists of atomic mappings (implying that the associated rules are LTGDs), then we let $\mathsf{CritRewrite}_{\text{PTIME}}(\mathcal{M})$ the result of applying the process above to the rule going from source relation to global schema relation associated to $m \in \mathcal{M}$. Then we have:

Theorem 3. *When* Σ_{Source} *consists of* LTGD*s,* $\mathsf{Disclose}(\Sigma_{\mathsf{Source}}, \mathcal{M}, p)$ *holds exactly when there is a* $p_{\mathsf{Annot}} \in \mathsf{CritRewrite}(p)$ *such that* $\mathsf{Hide}_{\mathcal{M}}(\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}})$ *entails* p_{Annot} *w.r.t. to the constraints*

$$\mathsf{CritRewrite}_{\text{PTIME}}(\Sigma_{\mathsf{Source}}) \cup \mathsf{CritRewrite}_{\text{PTIME}}(\mathcal{M}) \cup \mathsf{IsCrit}(\mathcal{M})$$

We can combine this result with recent work on fine-grained complexity of GTGDs to improve the doubly exponential upper bound of Corollary 1 for linear TGD source constraints and atomic mappings:

Theorem 4. $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{LTGD}, \mathsf{AtomMap})$ *is in* EXPTIME. *If the arity of relations in the source schema is bounded, then the complexity drops to* NP*, while if further the policy is atomic, the problem is in* PTIME.

Proof. It is sufficient to get an EXPTIME algorithm for the entailment problem produced by Theorem 3, since then we can apply it to each p_{Annot} in EXPTIME. The constraints in $\mathsf{CritRewrite}_{\text{PTIME}}(\Sigma_{\mathsf{Source}}) \cup \mathsf{CritRewrite}_{\text{PTIME}}(\mathcal{M})$ are Guarded TGDs that are not necessarily LTGDs. But the bodies of these guarded TGDs consist of a guard predicate and atoms over a fixed "side signature", namely the unary predicate IsCrit. It is known that the query entailment for IncDeps and guarded TGDs with a fixed side signature is in EXPTIME, with the complexity dropping to NP (resp. PTIME) when the arity is fixed (resp. fixed and the query is atomic) [Amarilli and Benedikt, 2018]. □

Can we do better than EXPTIME? We can note that when the constraints $\sigma \in \Sigma_{\mathsf{Source}}$ are IncDeps, $\mathsf{CritRewrite}(\sigma)$ consists only of σ; similarly if a mapping $m \in \mathcal{M}$ is a projection, then $\mathsf{CritRewrite}(m)$ consists only of m. This gives us a good upper bound in one of the most basic cases:

Corollary 3. $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{IncDep}, \mathsf{ProjMap})$ *is in* PSPACE. *If further a bound is fixed on the arity of relations in the source schema, then the problem becomes* NP*, dropping to* PTIME *when the policy is atomic.*

Proof. Our algorithm will guess a p_{Annot} in $\mathsf{CritRewrite}(Q)$ and checks the entailment of Theorem 2. This gives an entailment problem for IncDeps, known to be in PSPACE in general, in NP for bounded arity, and in PTIME for bounded arity and atomic queries [Johnson and Klug, 1984]. □

3.2 Obtaining Tractability

Thus far we have seen cases where the complexity drops to PSPACE in the general case and NP in the bounded arity case, and PTIME for atomic queries. We now present a case where we obtain tractability for arbitrary queries and arity. Recall that a UID is an IncDep where at most one variable is exported. They are actually quite common, capturing referential integrity when data is identified by a single attribute. We can show that restricting to UIDs while having only projection maps leads to tractability:

Theorem 5. $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{UID}, \mathsf{ProjMap})$ *is in* PTIME.

Proof. The first step is to refine the reduction of Theorem 2 to get an entailment problem with only UIDs, over an instance consisting of a single unary fact $\mathsf{IsCrit}(c_{\mathsf{Crit}})$. The main issue is avoid the constraints in $\Sigma_{\mathcal{M}}(\mathcal{M})$, corresponding to the mapping rules. The intuition for this is that on $\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}}$,

the only impact of the backward and forward implications of $\Sigma_{\mathcal{M}}(\mathcal{M})$ is to create new facts among the source relations. In these new facts only c_{Crit}, is propagated. Rather than creating SCEQrules (implicitly what happens in the HOCWQ reduction) or generating classical constraints where the impact of the equalities are "baked in" (as in the critical-instance rewritings of Theorems 2 and 3), we truncate the source relations to the positions where non-visible elements occur, while generating UIDs on these truncated relations that simulate the impact of back-and-forth using $\Sigma_{\mathcal{M}}(\mathcal{M})$.

The second step is to show that query entailment with UIDs over the instance consisting only of $\mathsf{IsCrit}(c_{\mathsf{Crit}})$ is in PTIME. This can be seen as an extension of the PTIME inference algorithm for UIDs [Cosmadakis *et al.*, 1990]. The idea behind this result is to analyze the classical "chase procedure" for query entailment with TGDs [Fagin *et al.*, 2005]. In the case of UIDs over a unary fact, the shape of the chase model is very restricted; roughly speaking, it is a tree where only a single fact connects two values. Based on this, we can simplify the query dramatically, making it into an acyclic query where any two variables co-occur in at most one predicate. Once query simplification is performed, we can reduce query entailment to polynomial many entailment problems involving individual atoms in the query. This in turn can be solved using the UID inference procedure of [Cosmadakis *et al.*, 1990]. □

4 Lower Bounds

We now focus on providing lower bounds for $\mathsf{Disclose}_{\mathsf{C}}(\mathcal{C}, \mathsf{Map})$, showing in particular that the upper bounds provided in Section 3 can not be substantially improved. For many classes of constraints it is easy to see that the complexity of disclosure inherits the lower bounds for the classical entailment problem for the class. From this we get a number of matching lower bounds; e.g. 2EXPTIME for GTGD constraints, PSPACE for IncDep constraints. But note that in some cases the upper bounds we have provided for disclosure in Section 3 are higher than the complexity of entailment over the source constraints. For example, for IncDeps we have provided only a 2EXPTIME upper bound for guarded mappings (from Corollary 1), and only an exponential bound for atomic mappings (from Theorem 4). This suggests that the form of the mappings influences the complexity as well, as we now show.

Most of our proofs for hardness above the entailment bound for source constraints rely on the encoding of a Turing machine. Source constraints are used to generate the underlying structures (tree of configurations, tape of a Turing machine) while mappings are used to ensure consistency (a universal configuration is accepting if and only if all its successor configurations are accepting, the content of the tape is consistently represented,...). To illustrate our approach, we sketch the proof of the following result.

Theorem 6. $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{IncDep}, \mathsf{GuardedMap})$ *and* $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{GTGD}, \mathsf{ProjMap})$ *are* 2EXPTIME*-hard, and are* EXPTIME*-hard even in bounded arity.*

Proof. Recall that Theorem 1 relates disclosure to a HOCWQ problem on $\mathcal{D}^{\mathcal{G}(\mathcal{M})}_{\mathsf{Crit}}$. Also recall from Section 3 the intuition that such a problem amounts to a classical entailment problem for a CQ over a very simple instance, using the source dependencies and SCEQrules: of the form $\phi(\vec{x}) \rightarrow x = c_{\mathsf{Crit}}$, where ϕ will be the body of a mapping. We will sketch how to simulate an alternating EXPSPACE Turing machine $\mathcal{M}$ using a QEntail problem using IncDeps and *guarded* SCEQrules. This can in turn be simulated using our HOCWQ problem.

We first build a tree of configurations using IncDeps, such that each node has a type (existential or universal) and is the parent of two nodes (called α-successor and β-successor) of the opposite type. This tree structure is represented, together with additional information, by atoms such as:

$$\mathsf{Children}_{\forall}(c, c_\alpha, c_\beta, ac, ac_\alpha, ac_\beta, \vec{y_0}, \vec{y_1}, r).$$

Intuitively, this states that c is a universal configuration, parent of c_α and c_β. ac (resp. ac_α, resp. ac_β) is the acceptance bit for c (resp. c_α, resp. c_β), which will be made equal to c_{Crit} if and only if the configuration represented by c (resp. c_α, resp. c_β) is accepting. $\vec{y_0}, \vec{y_1}$ will be used to represent cell addresses, while r is the identifier of the root of the configuration tree. The initial instance is such an atom, where the first position and the last position are the same constant, $\vec{y_0}$ is a vector of n 0's, $\vec{y_1}$ is a vector of n 1's, and all other arguments are distinct constants.

We use SCEQrules to propagate acceptance information up in the tree. For instance, a universal configuration is accepting if both its successors are accepting. This is simulated by the following SCEQrule:

$$\mathsf{Children}_{\forall}(c, c_\alpha, c_\beta, ac_c, c_{\mathsf{Crit}}, c_{\mathsf{Crit}}, \vec{y_0}, \vec{y_1}, r) \rightarrow ac_c = c_{\mathsf{Crit}}.$$

To simulate $\mathcal{M}$, we need access to an exponential number of cells for each configuration. We identify a cell by the configuration it belongs to and an address, which is a vector, generated by IncDeps, of length n whose arguments are either 0 or 1. The atom for representing a cell is thus $\mathsf{Cell}(c_p, c, \vec{addr}, \vec{v}, \vec{v}_{prev}, \vec{v}_{next})$, where c_p is the parent configuration of c, which is the configuration to which the represented cell belongs, $\vec{addr}$ is the address of the cell, $\vec{v}$ its content, $\vec{v}_{prev}$ the content of the previous cell, and $\vec{v}_{next}$ the content of the next cell. Note that this representation is redundant, and we need to use SCEQrules to ensure its consistency.

Note that $\vec{v}$ is a tuple of length the size of $(\Sigma \cup \{\flat\}) \times (Q \cup \{\perp\})$. Each position corresponds to an element of that set, and the content of a represented cell is the element which corresponds to the unique position in which c_{Crit} appears.

We now explain how to build the representation of the initial tape, and simulate the transition function. Both steps are done by unifying some nulls with c_{Crit}. W.l.o.g., we assume that the initial tape contains a l in the first cell, on which points the head of $\mathcal{M}$ in a state s, and that (l, s) corresponds to the first bit of $\vec{v}$. We thus use a SCEQrule to set this bit to c_{Crit} in the first cell of the first configuration. We then set (w.l.o.g.) the second bit of all the other cells of that configuration to c_{Crit} (assuming this represents $(\flat, \perp)$).

To simulate the transitions, we note that the content of a cell in a configuration depends only on the content of the same cell in the parent configuration, along with the content of parent's previous and next cells. We thus add a SCEQrule

Σ_{Source} \ $\mathcal{M}$	Unbounded arity				Bounded arity			
	ProjMap	AtomMap	GuardedMap	CQMap	ProjMap	AtomMap	GuardedMap	CQMap
IncDep	$\text{PSPACE}^{U=C3}_{L=\mathsf{QEntail}}$	$\text{EXPTIME}_{L=T7}$	$\text{2EXPTIME}_{L=T6}$	2EXPTIME	$\text{NP}_{L=\mathsf{QEntail}}$	NP	$\text{EXPTIME}_{L=T6}$	$\text{2EXPTIME}_{L=T8}$
LTGD	$\text{EXPTIME}_{L=T7}$	$\text{EXPTIME}^{U=T4}$	2EXPTIME	2EXPTIME	NP	$\text{NP}^{U=T4}$	EXPTIME	2EXPTIME
GTGD	$\text{2EXPTIME}_{L=T6}$	2EXPTIME	2EXPTIME	2EXPTIME	$\text{EXPTIME}_{L=T6}$	EXPTIME	$\text{EXPTIME}^{U=C2}$	2EXPTIME
FGTGD	2EXPTIME	2EXPTIME	2EXPTIME	$\text{2EXPTIME}^{U=C1}$	$\text{2EXPTIME}_{L=\mathsf{QEntail}}$	2EXPTIME	2EXPTIME	$\text{2EXPTIME}^{U=C1}$

Table 1: Complexity of disclosure: $\text{PSPACE}^{U=C3}_{L=\mathsf{QEntail}}$ means the corresponding problem is PSPACE-complete, where the Upper bound is given by Corollary 3 (U=C3) and the Lower bound is inherited from entailment. We omit bounds inferred from inclusion ($\mathcal{M}$ or Σ_{Source}).

that checks for the presence of c_{Crit} specifying the content of three consecutive cells in a configuration, and unify a null with c_{Crit} to specify the content of the corresponding cell of a child configuration.

The argument above uses IncDeps and GuardedMaps, but we can simplify the mappings to ProjMap using GTGDs. □

A simple variation of the construction used for PSPACE-hardness of entailment with IncDeps [Casanova *et al.*, 1984] shows that our upper bounds for IncDep source constraints and atomic maps are tight. The case of LTGD source constraints and projection maps can be done via reduction to that of IncDep source constraints and atomic maps:

Theorem 7. $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{IncDep}, \mathsf{AtomMap})$ *and* $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{LTGD}, \mathsf{ProjMap})$ *are both* EXPTIME*-hard.*

The above results, coupled with argument that the lower bounds for entailment are inherited by disclosure, show tightness of all upper bounds from Table 1 in the unbounded arity case. Another variation of the encoding in Theorem 6 shows that with no restriction on the mappings one can not do better than the 2EXPTIME upper bound of Corollary 1 even for IncDep constraints in bounded arity,

Theorem 8. $\mathsf{Disclose}_{\mathsf{C}}(\mathsf{IncDep}, \mathsf{CQMap})$ *is* 2EXPTIME*-hard in bounded arity.*

The theorem above, again combined with results showing that the lower bounds for entailment are inherited, suffice to show tightness of all upper bounds from Table 1 in the case of bounded arity.

We can also show that our tractability result for UID constraints and projection maps does not extend when either the maps or the constraints are broadened. Informally, this is because with these extensions we can generate an instance on which CQ querying is NP-hard.

5 Related Work

Disclosure analysis has been approached from many angles. We do not compare with the vast amount of work that analyzes probabilistic mechanisms for releasing information, providing probabilistic guarantees on disclosure [Dwork, 2006]. Our work focuses on the impact of reasoning on mapping-based mechanisms used in knowledge-based information integration, which are deterministic; thus one would prefer, and can hope for, deterministic guarantees on disclosure. We deal here with the *analysis* of disclosure, while there is a complementary literature on how to *enforce* privacy [Biskup and Weibert, 2008; Bonatti *et al.*, 1995; Bonatti and Sauro, 2013; Studer and Werner, 2014].

The problem of whether information is disclosed on a particular instance (variation of HOCWQ introduced in Section 3) has been studied in both the knowledge representation [Lutz *et al.*, 2013; Lutz *et al.*, 2015; Franconi *et al.*, 2011; Ahmetaj *et al.*, 2016; Amendola *et al.*, 2018] and database community [Abiteboul and Duschka, 1998]. The corresponding schema-level problem was defined in [Benedikt *et al.*, 2016], which allows arbitrary constraints relating the source and the global schema. However, results are provided only for constraints in guarded logics, which does not subsume the case of mappings given here. Our results clarify some issues in prior work: [Benedikt *et al.*, 2016] claimed that disclosure with IncDep source constraints and atomic maps is in PSPACE, while our Theorem 7 shows that the problem is EXPTIME-hard. Our notion of disclosure corresponds to the complement of [Benedikt *et al.*, 2018]'s "data-independent compliance". The formal framework of [Benedikt *et al.*, 2018] is orthogonal to ours. On the one hand, source constraints are absent; on the other hand a more powerful mapping language is considered, with existentials in the head of rules, while constraints on the global schema, given by ontological axioms, are now allowed. [Benedikt *et al.*, 2018] assume that the attacker has an interface for posing queries against the global schema, with the queries being answered under entailment semantics. In general, the semantic information on the global schema makes disclosure harder, since the outputs of different mapping rules may be indistinguishable by an attacker who only sees the results of reasoning. In contrast, source constraints make disclosure of secrets easier, since they provide additional information to the attacker.

6 Summary and Conclusion

We have isolated the complexity of information disclosure from a schema in the presence of commonly-studied sets of source constraints. A summary of many combinations of mappings $\mathcal{M}$ and source constraints Σ_{Source} is given in Table 1: note that *all problems are complete for the complexity classes listed*. We have shown tractability in the case of UIDs and projection maps (omitted in the tables), while showing that lifting the restriction leads to intractability. But we leave open a finer-grained analysis of complexity for frontier-one constraints with more general mappings. Our results depend on a fine-grained analysis of reasoning with TGDs and SCEQrules, a topic we think is of independent interest.

References

[Abiteboul and Duschka, 1998] Serge Abiteboul and Olivier Duschka. Complexity of answering queries using materialized views. In *PODS*, pages 254–263, 1998.

[Ahmetaj *et al.*, 2016] Shqiponja Ahmetaj, Magdalena Ortiz, and Mantas Šimkus. Polynomial datalog rewritings for expressive description logics with closed predicates. In *IJCAI*, pages 878–885, 2016.

[Amarilli and Benedikt, 2018] Antoine Amarilli and Michael Benedikt. When Can We Answer Queries Using Result-Bounded Data Interfaces? In *PODS*, pages 281–293, 2018.

[Amendola *et al.*, 2018] Giovanni Amendola, Nicola Leone, Marco Manna, and Pierfrancesco Veltri. Enhancing existential rules by closed-world variables. In *IJCAI*, pages 1676–1682, 2018.

[Baget *et al.*, 2011] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10), 2011.

[Bárány *et al.*, 2015] Vince Bárány, Balder Ten Cate, and Luc Segoufin. Guarded negation. *J. ACM*, 62(3):356–367, 2015.

[Benedikt *et al.*, 2016] Michael Benedikt, Pierre Bourhis, Balder ten Cate, and Gabriele Puppis. Querying visible and invisible information. In *LICS*, pages 297–306, 2016.

[Benedikt *et al.*, 2018] Michael Benedikt, Bernardo Cuenca Grau, and Egor V. Kostylev. Logical foundations of information disclosure in ontology-based data integration. *Artif. Intell.*, 262:52–95, 2018.

[Biskup and Weibert, 2008] Joachim Biskup and Torben Weibert. Keeping Secrets in Incomplete Databases. *Int. J. Inf. Sec.*, 7(3):199–217, 2008.

[Bonatti and Sauro, 2013] Piero A. Bonatti and Luigi Sauro. A confidentiality model for ontologies. In *ISWC*, pages 17–32, 2013.

[Bonatti *et al.*, 1995] Piero Bonatti, Sarit Kraus, and V. S. Subrahmanian. Foundations of Secure Deductive Databases. *TKDE*, 7(3):406–422, 1995.

[Calì *et al.*, 2013] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *JAIR*, pages 70–80, 2013.

[Casanova *et al.*, 1984] Marco Casanova, Ronald Fagin, and Christos Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *JCSS*, 28(1):29–59, 1984.

[Cosmadakis *et al.*, 1990] Stavros S. Cosmadakis, Paris C. Kanellakis, and Moshe Y. Vardi. Polynomial-time implication problems for unary inclusion dependencies. *J. ACM*, 37(1):15–46, 1990.

[Dwork, 2006] Cynthia Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[Fagin *et al.*, 2005] Ronald Fagin, Phokion G. Kolaitis, Renee J. Miller, and Lucian Popa. Data Exchange: Semantics and Query Answering. *Theoretical Computer Science*, 336(1):89–124, 2005.

[Franconi *et al.*, 2011] Enrico Franconi, Yasmin Ibáñez-García, and Inanç Seylan. Query answering with DBoxes is hard. *ENTCS*, 278:71–84, 2011.

[Johnson and Klug, 1984] David S. Johnson and Anthony C. Klug. Testing Containment of Conjunctive Queries under Functional and Inclusion Dependencies. *JCSS*, 28(1), 1984.

[Lenzerini, 2002] Maurizio Lenzerini. Data integration: A theoretical perspective. In *PODS*, pages 233–246, 2002.

[Lutz *et al.*, 2013] Carsten Lutz, Inanç Seylan, and Frank Wolter. Ontology-based data access with closed predicates is inherently intractable (sometimes). In *IJCAI*, pages 1024–1030, 2013.

[Lutz *et al.*, 2015] Carsten Lutz, Inanç Seylan, and Frank Wolter. Ontology-mediated queries with closed predicates. In *IJCAI*, pages 3120–3126, 2015.

[Studer and Werner, 2014] Thomas Studer and Johannes Werner. Censors for Boolean Description Logic. *Trans. on Data Privacy*, 7(3):223–252, 2014.