Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Source Information Disclosure in Ontology-Based Data Integration

Michael Benedikt, Bernardo Cuenca Grau, and **Egor V. Kostylev**
Department of Computer Science
University of Oxford

Ontology-based data integration systems allow users to effectively access data sitting in multiple sources by means of queries over a global schema described by an ontology. In practice, datasources often contain sensitive information that the data owners want to keep inaccessible to users. In this paper, we formalize and study the problem of determining whether a given data integration system discloses a source query to an attacker. We consider disclosure on a particular dataset, and also whether a schema admits a dataset on which disclosure occurs. We provide lower and upper bounds on disclosure analysis, in the process introducing a number of techniques for analyzing logical privacy issues in ontology-based data integration.

1 Introduction

Data integration systems expose information from multiple, heterogeneous datasources by means of a *global schema*, in which the mismatches between the individual schemas of the datasources have been reconciled (Lenzerini 2002). The relationships between the datasources and the global schema are determined by *mappings*, which declaratively specify how each term in the global schema relates to the data.

In addition to reconciling the structure of the datasources, the global schema also enables uniform access to the data by providing users with the vocabulary for query formulation. Queries issued against the global schema are typically answered by one of two approaches. In the first approach, an instance of the global schema is initially materialized using the mappings and the data in the sources; then, the query is answered over the materialized instance. In the second approach, no data is exported from the sources and the global schema remains virtual; this is achieved by first reformulating the user query on-the-fly into a set of queries over the sources, and then assembling back their results.

In *ontology-based data integration* (Poggi et al. 2008) the global schema is realized using an *ontology*. In addition to a vocabulary, the ontology also specifies how the terms in the vocabulary relate to each other, thus providing valuable background knowledge about the domain. In this setting, queries are typically answered following the virtual approach, where the ontology axioms must now also be taken into account during query reformulation.

In practice, datasources often contain sensitive information to be protected against unauthorized disclosure. It is well-known that information integration and linkage poses major threats to the confidentiality of such sensitive data, even if it is only made available in an anonymized form (Sweeney 2002). In the setting of ontology-based data integration, the risks of unauthorized information disclosure quickly become apparent; indeed, the information exposed to users depends on a complex combination of schema reconciliation, reasoning over the ontology, and access to chunks of data in the sources via the mappings.

Example 1. A hospital has a number of information systems storing data about appointments. For instance, the oncology department relies on the following schema consisting of a table $\mathsf{OncAppt}(\mathsf{TreatId}, \mathsf{PatId}, \mathsf{DocId}, \mathsf{Date}, \ldots)$, where TreatId, PatId, DocId represent treatment, patient and doctor IDs. Although other departments, such as cardiology, may store appointment data using different schemas, they all share some basic attributes, such as the IDs for treatments, patients, and doctors, as well as the appointment times. To integrate this data, the hospital relies on a global schema capturing the common terminology in all types of appointments. Such global schema would include predicates such as $\mathsf{Appt}(\mathsf{PatId}, \mathsf{DocId}, \mathsf{Date})$, $\mathsf{Doctor}(\mathsf{DocId}, \mathsf{Date})$, and $\mathsf{SpecialistRecord}(\mathsf{DocId}, \mathsf{Date})$. The following simple mappings translate from the source to the global schema, where in each case $\mathbf{t}_i$, $1 \leq i \leq 4$, represents sets of attributes occurring only in the source:

$\mathsf{OncAppt}(\mathbf{t}_1, \mathsf{PatId}, \mathsf{DocId}, \mathsf{Date}) \rightarrow \mathsf{Appt}(\mathsf{PatId}, \mathsf{DocId}, \mathsf{Date})$,
$\mathsf{OncAppt}(\mathbf{t}_2, \mathsf{DocId}, \mathsf{Date}) \rightarrow \mathsf{SpecialistRecord}(\mathsf{DocId}, \mathsf{Date})$,
$\mathsf{CardAppt}(\mathbf{t}_3, \mathsf{PatId}, \mathsf{DocId}, \mathsf{Date}) \rightarrow \mathsf{Appt}(\mathsf{PatId}, \mathsf{DocId}, \mathsf{Date})$,
$\mathsf{CardAppt}(\mathbf{t}_4, \mathsf{DocId}, \mathsf{Date}) \rightarrow \mathsf{Doctor}(\mathsf{DocId}, \mathsf{Date})$.

The schema designers may not want to disclose the relationship between patients and the departments they have visited. However, the confidentiality of such information is at risk: by querying SpecialistRecord an attacker can determine which doctors had some oncology appointment on a given date. From Appt, the attacker has access to a list of the appointments a doctor had on a given date, and if the data contains only one oncology appointment for some doctor on a given date, then the attacker could infer that the patient involved had an oncology appointment.

In this case, the unauthorized disclosure depends on the ability of the attacker to "trace back" (using the mappings) the exact relation in the source that exported each tuple in the extension of the global predicates SpecialistRecord and Doctor. An ontology, however, could be used to represent that these predicates have the same meaning and hence have the same extension; then, an attacker would no longer be able to determine the origin of the exported data tuples and no disclosure would occur, regardless of the source data. $\Diamond$

Our goal in this paper is to lay the logical foundations of information disclosure in ontology-based data integration. Our focus is on the semantic requirements that a data integration system and dataset should satisfy before it is made available to users for querying, as well as on the complexity of checking whether such requirements are fulfilled. These are fundamental steps towards the development of algorithms suitable for applications.

Our framework for information disclosure builds on work in the database community by Nash and Deutsch (2006). The sensitive information is represented by a query over the source schema (the *policy*). The schema-level information in the system (ontology, mappings, source schemas, and policy specification) is assumed publicly available (a worst-case scenario for confidentiality enforcement). In contrast, the actual data is not made available directly, but rather only by means of queries over the global schema. Disclosure of sensitive information occurs when a user is able to uncover an answer to the policy over the datasource by just querying the global schema and exploiting the full availability of schema-level information. If no such disclosure is possible given the current data in the sources, we say that the data integration system *complies to the policy*. There is a natural data-independent variant of this notion, where compliance must hold regardless of the specific source data.

We study the computational properties of compliance checking, both in its instance-dependent and data-independent variants. We consider arbitrary first-order ontology languages and parametrize our main results in terms of their complexity for standard query answering. Concerning mappings, we consider the general case of *GLAV* mappings as well as well-known special cases (Lenzerini 2002). Our contributions are as follows.

- We show that checking instance-based compliance is decidable whenever the ontology language of choice has decidable query answering problem. Then, we isolate its precise complexity for many of the most common cases, ranging from NEXPTIME to P.
- We study the data-independent version of compliance and show that the problem is undecidable even if the ontology is empty. We then isolate a decidable case and study a further restriction ensuring tractability.
- Our notions of compliance depend on the ability of an attacker to distinguish between difference datasources. Hence, we also study the *source indistinguishability* problem and provide tight complexity bounds for many cases.
- Our results have implications on related work. On the one hand, they correct some of the complexity bounds claimed by Nash and Deutsch (2006); on the other hand, our work also closes an open problem in *data pricing* (Koutris et al. 2015), by showing a Π_2^p lower bound to the so-called *instance-based determinacy* problem.
- We introduce a "repair" process that ensures tractability of instance-based compliance in certain cases. For the data-independent compliance problem, we give refinements of methods from earlier work, particularly the "critical instance method" (Gogacz and Marcinkowski 2014; Cuenca Grau et al. 2013a; Benedikt et al. 2016; Baader et al. 2016; Shmueli 1993; Marnette 2010) for obtaining decidability.

2 Preliminaries

Tuple-Generating Dependencies and Ontologies. We adopt standard notions from function-free first-order logic over a vocabulary of relational names and constants. An *instance* is a finite set of facts. 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 for the head and body, are called the *frontier variables*. A TGD is *linear* if its body consists of a single atom; it is *Datalog* if its head consists of a single atom and there are no existential variables $\mathbf{y}$. An *ontology* is a finite set of first-order sentences; an ontology is *linear* if it consists of linear TGDs. A *conjunctive query* (*CQ*) with *free* variables $\mathbf{x}$ is a formula $q(\mathbf{x}) = \exists \mathbf{y}.\varphi(\mathbf{x},\mathbf{y})$, where $\varphi(\mathbf{x},\mathbf{y})$ is a conjunction of atoms with each term either a constant or a variable from $\mathbf{x} \cup \mathbf{y}$; *arity* of a CQ is the number of its free variables, and CQs of arity 0 are *Boolean*.

Let $\mathcal{O}$ be an ontology, let q be a Boolean CQ, and let $\mathcal{D}$ be an instance. We recall the standard query entailment problem: $\mathsf{CQEnt}(\mathcal{O},\mathcal{D},q) = \mathsf{true}$ if and only if $\mathcal{O} \cup \mathcal{D} \models q$.

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. A *GLAV mapping* is a TGD where the body is over the source schema and the head is over the global schema. Datalog mappings are called *GAV*. A set of *CQ views* is a set of GAV mappings with different head predicates.

A *data integration setting* is a tuple $(\mathcal{O},\mathcal{M},\mathcal{D})$, where $\mathcal{O}$ is an ontology over the global schema, $\mathcal{M}$ is a finite set of GLAV mappings, and $\mathcal{D}$ is an instance over the source schema. For $q(\mathbf{x})$ a CQ over the global schema, we say that a tuple $\mathbf{a}$ of constants is a *certain answer* to $q(\mathbf{x})$ with respect to $(\mathcal{O},\mathcal{M},\mathcal{D})$ if $I \models q(\mathbf{a})$ for all models I of $\mathcal{O}$ such that, for every mapping $\varphi(\mathbf{x},\mathbf{z}) \rightarrow \exists \mathbf{y}.\psi(\mathbf{x},\mathbf{y})$ in $\mathcal{M}$ and each tuple of constants $\mathbf{c}$ it holds that $I \models \exists \mathbf{y}.\psi(\mathbf{c},\mathbf{y})$ whenever $\mathcal{D} \models \varphi(\mathbf{c},\mathbf{z})$. The *virtual image* of $\mathcal{M}$ and $\mathcal{D}$, denoted $\mathcal{V}_{\mathcal{M},\mathcal{D}}$, is the following set of Boolean CQs:

$$\{\exists \mathbf{y}.\psi(\mathbf{c},\mathbf{y}) \mid \varphi(\mathbf{x},\mathbf{z}) \rightarrow \exists \mathbf{y}.\psi(\mathbf{x},\mathbf{y}) \text{ in } \mathcal{M}, \text{ and } \mathcal{D} \models \exists \mathbf{z}.\varphi(\mathbf{c},\mathbf{z})\}.$$

It is routine to check that $\mathbf{a}$ is a certain answer to a CQ $q(\mathbf{x})$ with respect to $(\mathcal{O},\mathcal{M},\mathcal{D})$ if and only if $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models q(\mathbf{a})$.

3 Basic Framework

In this section we present our framework for information disclosure and define its associated reasoning problems.

In a data integration setting, users (including malicious attackers) can only interact with the system by posing queries against the global schema. Users have no direct access to the source instances and hence the information they can gather about the source data is inherently incomplete. As a result of such incompleteness, many different source instances may be *indistinguishable*, in the sense that users cannot tell the difference between them by just querying the system.

Definition 2. *Source instances $\mathcal{D}$ and $\mathcal{D}'$ are* indistinguishable *with respect to an ontology $\mathcal{O}$ over the global schema and mappings $\mathcal{M}$ if, for every query $q(\mathbf{x})$ over the global schema, the certain answers to $q(\mathbf{x})$ over the data integration settings $(\mathcal{O}, \mathcal{M}, \mathcal{D})$ and $(\mathcal{O}, \mathcal{M}, \mathcal{D}')$ coincide.*

Informally, all a malicious attacker can gather from the source instance $\mathcal{D}$ is that it must be one of the (possibly infinitely many) source instances $\mathcal{D}'$ indistinguishable from $\mathcal{D}$.

The sensitive information in a data integration setting is given by a CQ over the source schema, which we refer to as the *policy*. Intuitively, disclosure of sensitive information occurs whenever there is an answer to the policy that holds in *all* the sources that are indistinguishable from the point of view of the attacker. Indeed, in such situation the attacker would be able to uncover the aforementioned answer without a shadow of a doubt. If no such disclosure can occur, then the data integration setting *complies* to the policy.

Definition 3. *Let $(\mathcal{O}, \mathcal{M}, \mathcal{D})$ be a data integration setting, and let $p(\mathbf{x})$ be a CQ over the source schema (called* policy*). Setting $(\mathcal{O}, \mathcal{M}, \mathcal{D})$* complies to *$p(\mathbf{x})$ if, for every tuple of constants $\mathbf{a}$ such that $\mathcal{D} \models p(\mathbf{a})$, there is a source instance $\mathcal{D}_\mathbf{a}$ indistinguishable from $\mathcal{D}$ with respect to $\mathcal{O}$ and $\mathcal{M}$ such that $\mathcal{D}_\mathbf{a} \not\models p(\mathbf{a})$.*

Returning to Example 1, the security need for the schema might include the requirement that the schema complies with the following policy with free variable PatId:

$$\exists \mathbf{t}_1.\exists \mathsf{DocId}.\exists \mathsf{Date}.\mathsf{OncAppt}(\mathbf{t}_1, \mathsf{PatId}, \mathsf{DocId}, \mathsf{Date}).$$

With these definitions in hand, we are ready to present the computational problems considered in our work.

Definition 4. *Let $\mathcal{O}$ be an ontology, $\mathcal{M}$ be mappings, $\mathcal{D}$ and $\mathcal{D}'$ be source instances, and p be a policy. Consider the following decision problems:*

- $\mathsf{SourceInd}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \mathcal{D}')$ *is* `true` *iff $\mathcal{D}$ and $\mathcal{D}'$ are indistinguishable with respect to $\mathcal{O}$ and $\mathcal{M}$;*
- $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$ *is* `true` *iff $(\mathcal{O},\mathcal{M},\mathcal{D})$ complies to p;*
- $\mathsf{ComplyAll}(\mathcal{O}, \mathcal{M}, p)$ *is* `true` *iff* $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$ *is* `true` *for every source instance $\mathcal{D}$.*

4 Source Indistinguishability

In this section we study the complexity of checking whether two given sources are indistinguishable from the point of view of users of a data integration system. The results in this section will be relevant to the study of policy compliance later on. Furthermore, source indistinguishability is an interesting problem in its own right; for instance, it can be used to determine whether given changes in the source instances can affect applications that query the system.

The following lemma extends Theorem 1 of (Nash and Deutsch 2006) to the setting with an ontology, providing a fundamental characterization of source indistinguishability.

Lemma 5. *The following are equivalent for any ontology $\mathcal{O}$, mappings $\mathcal{M}$, and source instances $\mathcal{D}$ and $\mathcal{D}'$:*

1. $\mathsf{SourceInd}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \mathcal{D}')$ *is* `true`*;*
2. *for each mapping with the head CQ q, the certain answers to q with respect to $(\mathcal{O}, \mathcal{M}, \mathcal{D})$ and $(\mathcal{O}, \mathcal{M}, \mathcal{D}')$ coincide;*
3. *$\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ are logically equivalent.*

The lemma suggests a basic high-level algorithm that decides $\mathsf{SourceInd}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \mathcal{D}')$ for any ontology language with decidable entailment problem: *(i)* construct the virtual images $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$; *(ii)* check whether $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ are equivalent.

Checking indistinguishability is potentially harder than query entailment since precomputing the images of the sources can lead to an exponential blowup. Analysis of our algorithm reveals that SourceInd is no harder than query entailment in many cases: e.g., if the mappings are linear then no such blowup occurs, or if the ontology language has sufficiently high complexity for entailment (at least ExpTime) while retaining tractability in the size of the data. In other cases, however, source indistinguishability is indeed harder than entailment. For example, when the input ontology is empty and the mappings are GAV, determining equivalence of the source images amounts to a syntactic check, whereas we prove SourceInd to be Π_2^p-hard. Additionally, if the arity of the global schema is bounded (as in Description Logic ontologies, where arity is at most two), the problem stays hard for $\mathrm{P}^{\|\mathrm{NP}}$: the class of problems solvable in P with non-adaptive calls to an NP oracle (Wagner 1987).

Theorem 6. *Problem $\mathsf{SourceInd}(\emptyset, \mathcal{M}, \mathcal{D}, \mathcal{D}')$ is Π_2^p-hard for sets of GAV mappings $\mathcal{M}$; it is $\mathrm{P}^{\|\mathrm{NP}}$-hard if, additionally, the arity of the global schema is bounded by 2.*

In such cases, our basic algorithm only provides an ExpTime upper bound, which stems from the cost of materializing the images of the sources.

If the ontology consists of linear TGDs, however, we can do better. We can avoid explicit construction of the virtual images of the sources by exploiting the following property of linear ontologies: to check whether $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models q$ with Boolean CQ q in $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$ it suffices to consider only a set of instantiations of the frontier of $\mathcal{M}$ over $\mathcal{D}$ that is polynomially bounded in the size of q. This allows us to obtain matching upper bounds for the lower bounds in Theorem 6.

Theorem 7. *Problem $\mathsf{SourceInd}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \mathcal{D}')$ for $\mathcal{O}$ in an ontology language $\mathbb{O}$ and $\mathcal{M}$ in a mappings language $\mathbb{M}$ is*

1. *C-complete, for a complexity class C with ExpTime $\subseteq$ C, and in P in the size $|\mathcal{D} \cup \mathcal{D}'|$ of $\mathcal{D} \cup \mathcal{D}'$ for $\mathbb{O}$ such that $\mathsf{CQEnt}(\mathcal{O}, \mathcal{D}, q)$ is C-complete and in P in $|\mathcal{D}|$;*
2. *PSpace-complete and in AC^0 in $|\mathcal{D} \cup \mathcal{D}'|$ for linear $\mathbb{O}$;*
3. *Π_2^p-complete for the empty $\mathcal{O}$;*
4. *$\mathrm{P}^{\|\mathrm{NP}}$-complete for linear $\mathbb{O}$ (i.e., $\mathbb{O}$ consisting of linear ontologies), $\mathbb{M}$ consisting of sets of mappings with bounded numbers of frontier variables, and the arity of the global schema bounded by 2;*

5. NP-*complete and in* AC^0 *in* $|\mathcal{D} \cup \mathcal{D}'|$ *for linear* $\mathbb{O}$*, linear* $\mathbb{M}$*, and the arity of the global schema bounded by 2;*
6. *in* P *for linear* $\mathbb{O}$*, linear GAV* $\mathbb{M}$*, and the arity of the global schema bounded by 2.*

Case 4 is of particular interest because it covers OBDA settings with DL-Lite$_\mathcal{R}$ ontologies (Calvanese et al. 2007).

5 Policy Compliance

We now turn our attention to the Comply problem and show that it is decidable for any ontology language with decidable query entailment problem. Furthermore, we establish its precise complexity for the most common cases.

5.1 Decidability and Upper Bounds

In what follows, let us consider a fixed, but arbitrary, input $(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$ to Comply; let $\mathsf{Dom}(\mathcal{D})$ be the set of constants in $\mathcal{D}$. By Definition 3, a correct procedure must return `true` if and only if, for every tuple $\mathbf{a}$ with $\mathcal{D} \models p(\mathbf{a})$, there exists $\mathcal{D}_\mathbf{a}$ indistinguishable from $\mathcal{D}$ such that $\mathcal{D}_\mathbf{a} \not\models p(\mathbf{a})$.

We start with a basic observation: for a source instance to be indistinguishable from $\mathcal{D}$, its image via $\mathcal{M}$ can only contain constants from $\mathsf{Dom}(\mathcal{D})$. The following definition formalises such notion of a "candidate" source image.

Definition 8. *For a set of constants* $\mathcal{C}$*, a* $\mathcal{C}$-source type τ *is a function assigning* `true` *or* `false` *to each sentence of the form* $\exists \mathbf{z}.\varphi(\mathbf{a}, \mathbf{z})$*, with* $\mathbf{a}$ *a tuple of constants from* $\mathcal{C}$ *and* $\varphi(\mathbf{x}, \mathbf{z})$ *the body of a mapping in* $\mathcal{M}$*. The* image *of* τ*, denoted* $\mathcal{V}_\tau$*, is the set of sentences* $\exists \mathbf{y}.\psi(\mathbf{a}, \mathbf{y})$ *such that* $\varphi(\mathbf{x}, \mathbf{z}) \rightarrow \exists \mathbf{y}.\psi(\mathbf{x}, \mathbf{y})$ *is a mapping in* $\mathcal{M}$ *and* τ *returns* `true` *when applied to* $\exists \mathbf{z}.\varphi(\mathbf{a}, \mathbf{z})$*.*

Intuitively, each $\mathcal{V}_\tau$ associated to a type τ represents a candidate source image. We will be interested only in *realizable* $\mathcal{C}$-types τ: those having a witness source instance $\mathcal{D}_\tau$ that refutes some answer to the policy.

Definition 9. *Let* $\mathbf{a}$ *be a tuple of constants from a set* $\mathcal{C}$*. A* $\mathcal{C}$*-source type* τ *is* $\mathbf{a}$-realizable *if there is a source instance* $\mathcal{D}_\tau$ *such that (i)* $\mathcal{V}_{\mathcal{M},\mathcal{D}_\tau} = \mathcal{V}_\tau$*, and (ii)* $\mathcal{D}_\tau \not\models p(\mathbf{a})$*.*

The following lemma shows that realizability can be characterized as a logical satisfiability problem.

Lemma 10. *Let* $\mathbf{a}$ *be a tuple of constants from a set* $\mathcal{C}$*, and* τ *be a* $\mathcal{C}$*-source type. Let* ρ *be the conjunction of the sentences*

$$\begin{array}{ll} \neg p(\mathbf{a}), & \\ \varphi, & \text{for all } \varphi \text{ with } \tau(\varphi) = \texttt{true}, \\ \neg\varphi, & \text{for all } \varphi \text{ with } \tau(\varphi) = \texttt{false}, \\ \forall \mathbf{x}.\forall \mathbf{y}.\big(\varphi(\mathbf{x},\mathbf{y}) \rightarrow \bigwedge_{x \in \mathbf{x}} \bigvee_{c \in \mathsf{Dom}(\mathcal{D})} x = c\big), & \\ \text{for any mapping in } \mathcal{M} \text{ with body } \varphi(\mathbf{x},\mathbf{y}) \text{ and frontier } \mathbf{x}. & \end{array}$$

Then, τ *is* $\mathbf{a}$-realizable *if and only if* ρ *is satisfiable.*

Note that the formula in Lemma 10 is a Boolean combination of existentially quantified sentences; hence, whenever it is satisfiable, it has a model polynomial in its size.

Finally, by Lemma 5 in the previous section, a realizable type τ must satisfy an additional property to witness compliance, namely that $\mathcal{O} \cup \mathcal{V}_\tau$ must be equivalent to $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$.

With these ingredients, we are ready to present an alternating procedure for checking $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$:

1. universally guess a tuple $\mathbf{a}$ of constants from $\mathsf{Dom}(\mathcal{D})$ of the size equal to the arity of p;
2. existentially guess a $\mathsf{Dom}(\mathcal{D})$-source type τ;
3. verify whether τ is $\mathbf{a}$-realizable and reject if it is not;
4. verify whether $\mathcal{O} \cup \mathcal{V}_\tau$ is equivalent to $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$; accept if yes and reject otherwise.

Correctness of this algorithm follows from Lemma 5 and the definition of realizable type. Furthermore, by analysing the algorithm, we can obtain decidability and complexity upper bounds for a range of ontology languages. In particular, cases 2 and 4 in the following theorem are applicable to DL-Lite$_\mathcal{R}$ ontologies, whereas case 3 is relevant to the more general case of ontologies consisting of linear TGDs.

Theorem 11. *Problem* $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$ *for* $\mathcal{O}$ *in an ontology language* $\mathbb{O}$ *is*

1. *decidable if* $\mathsf{CQEnt}(\mathcal{O}', \mathcal{D}', q)$ *is decidable as* $\mathcal{O}'$ *ranges over* $\mathbb{O}$*;*
2. *in* NEXPTIME *if* $\mathsf{CQEnt}(\mathcal{O}', \mathcal{D}', q)$ *is in* NP *as* $\mathcal{O}'$ *ranges over* $\mathbb{O}$*;*
3. *in* PSPACE *if* $\mathsf{CQEnt}(\mathcal{O}', \mathcal{D}', q)$ *is in* PSPACE *as* $\mathcal{O}'$ *ranges over* $\mathbb{O}$*, when* $\mathcal{M}$ *ranges over sets of mappings with bounded number of frontier variables;*
4. *in* Σ_2^p *if* $\mathsf{CQEnt}(\mathcal{O}', \mathcal{D}', q)$ *is in* NP *as* $\mathcal{O}'$ *ranges over* $\mathbb{O}$*,* $\mathcal{M}$ *ranges over sets of mappings with bounded number of frontier variables, and* p *over queries with bounded arity;*
5. *in* NP *in* $|\mathcal{D}|$ *if* $\mathsf{CQEnt}(\mathcal{O}', \mathcal{D}', q)$ *is in* NP *in* $|\mathcal{D}'|$ *for* $\mathcal{O}'$ *ranging over* $\mathbb{O}$*.*

The proof of the theorem is a consequence of the correctness of our generic algorithm and the following remarks. Case 1 in the theorem follows from the fact that realizability is decidable and equivalence checking is also decidable for $\mathbb{O}$ if so is CQEnt. In all cases but the fourth one, we can iterate over the possible bindings of the free variables in p within the required complexity class; in case 4, however, this is possible only if the arity of p is assumed bounded.

For case 2, guessing a source type and finding a witness instance can be done in NEXPTIME, with the size of the witness instance being bounded by an exponential. The verification of equivalence can be done with exponentially many calls to CQEnt, which is feasible in exponential time under the assumption that CQEnt is in NP for $\mathbb{O}$.

For cases 3 and 4, the bound on the frontier allows us to guess a source type τ in NP and then also a witness source instance $\mathcal{D}_\tau$ of polynomial size (Lemma 10). Then, we can use an NP oracle to check that $\mathcal{D}_\tau$ satisfies the required properties in Definition 9. The equivalence check can then be done with polynomially many calls to CQEnt, each of which is feasible in PSPACE (case 3) or in NP (case 4).

Finally, in case 5, the ontology, policy and mappings are considered to be fixed; as a result, the verification that the guessed witness instance satisfies the source type can be done in polynomial time, bringing complexity down to NP.

5.2 Lower Bounds

The main drawback of our generic algorithm for Comply is the need to guess a source type, given that the number of source-types is exponential, even when the schema is fixed. Unfortunately, this algorithm cannot be improved in general.

Theorem 12. *Problem* $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$ *for* $\mathcal{O}$ *in a language* $\mathbb{O}$ *and* $\mathcal{M}$ *in a language* $\mathbb{M}$ *is*

1. NEXPTIME*-hard if* $\mathcal{O}$ *is empty and* $\mathbb{M}$ *consists of sets of CQ views;*
2. PSPACE*-hard if* $\mathsf{CQEnt}(\mathcal{O}, \mathcal{D}, q)$ *is* PSPACE*-hard for* $\mathbb{O}$*, and all the mappings in* $\mathbb{M}$ *have no frontier variables;*
3. Σ_2^p*-hard if* $\mathcal{O}$ *is empty,* $\mathbb{M}$ *consists of sets of linear CQ views, and the arity of the global schema is bounded by 2;*
4. NP*-hard in* $|\mathcal{D}|$ *if* $\mathcal{O}$ *is empty and* $\mathbb{M}$ *consists of sets of linear CQ views.*

All these bounds hold even if p *is Boolean.*

Case 1 uses an encoding of an NEXPTIME-complete version of the tiling problem. In the source, there are relations associating "cell objects" with vertical and horizontal coordinates, and also with tile types. The only exported information is that adjacent coordinates are associated with some cells and with some compatible tile type assignments. In the source instance $\mathcal{D}$, a cell with coordinates $(\mathbf{x}, \mathbf{y})$ will be associated with each tile type, since there is only one cell object; this information is not exported, and thus sources that are indistinguishable from $\mathcal{D}$ may be better behaved. The policy p is chosen so that indistinguishable sources where p fails will correspond to ones where coordinates are assigned a unique tiling type. Case 2 relies on an easy reduction from CQ entailment. Case 3 uses a non-trivial encoding of the well-known Σ_2^p-hard variant of QBF validity; as discussed later on, a variant of our Σ_2^p-hardness result closes a problem on instance-based determinacy left open in (Koutris et al. 2015). Case 4 follows from the proof of hardness of instance-based determinacy in (Koutris et al. 2015).

5.3 Tractable Case

The lower bounds in Theorem 12 are rather discouraging: even with the empty ontology and linear CQ views, the compliance problem is Σ_2^p-hard and NP-hard in data complexity. We next show that tractability can be obtained if we restrict ourselves to linear mappings and require also the policy to be *ground*, that is, to be a conjunction of facts. It is easy to see, however, that the upper bounds implied by our generic algorithm in Section 5.1 do not improve if we restrict ourselves to ground policies. Hence, we next describe a new algorithm that deals with ground policies explicitly.

Let us fix an arbitrary input $(\emptyset, \mathcal{M}, \mathcal{D}, p)$ to Comply, where $\mathcal{M}$ is linear and GAV, and p is ground. For simplicity, let us assume also that p consists of a single fact (the extension to the general case is straightforward). Our algorithm proceeds as follows:

1. construct the image $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ of $\mathcal{D}$;
2. construct $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$, where $\mathcal{D}' = \mathcal{D} \setminus \{p\}$;
3. for each "uncovered" fact $U(\mathbf{c}) \in \mathcal{V}_{\mathcal{M},\mathcal{D}} \setminus \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ and each mapping $R(\mathbf{x}, \mathbf{z}) \to U(\mathbf{x})$ in $\mathcal{M}$
   - look for a fact $R(\mathbf{c}, \mathbf{d})$, where $\mathbf{d}$ can include constants from $\mathcal{D}$ or fresh constants, such that $R(\mathbf{c}, \mathbf{d}) \neq p$ and the application of all mappings to $R(\mathbf{c}, \mathbf{d})$ yields only facts in $\mathcal{V}_{\mathcal{M},\mathcal{D}}$; if no such fact exists, return `false`, otherwise, add $R(\mathbf{c}, \mathbf{d})$ to $\mathcal{D}'$;
4. return `true` and witnessing $\mathcal{D}'$.

The algorithm attempts to construct a witness to compliance by first removing the policy fact p from $\mathcal{D}$. The resulting $\mathcal{D}'$, however, may not be indistinguishable from $\mathcal{D}$. The algorithm proceeds to "repair" $\mathcal{D}'$ by recovering each fact $U(\mathbf{c})$ that was lost from the image after removing p from the source. For this, it attempts to find a fact (different from p) which, when added to $\mathcal{D}'$, brings $U(\mathbf{c})$ back into the image without generating other facts not already in $\mathcal{V}_{\mathcal{M},\mathcal{D}}$.

This algorithm justifies the following theorem.

Theorem 13. *If the arity of the source schema is bounded, then* $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ *is in* P *for linear GAV sets of mappings* $\mathcal{M}$ *and ground policies* p*.*

6 Data-Independent Compliance

We now turn to problem ComplyAll, which requires that all possible source instances comply to the policy. This is a very desirable property for (the schema of) a data integration system to satisfy: it ensures that none of the tuples in the extension of the policy is revealed to a malicious attacker, regardless of the underlying source data.

Unfortunately, ComplyAll can be shown undecidable even under very strong restrictions on the input.

Theorem 14. *Problem* $\mathsf{ComplyAll}(\emptyset, \mathcal{M}, p)$ *is undecidable even for GAV mappings* $\mathcal{M}$ *and the arity of the global schema is bounded by 2.*

The proof is via an involved reduction from the well-known tiling problem (Berger 1966) into the complement of ComplyAll. Our reduction exploits a variant of the "challenge method" by Benedikt et al. (2016), where special "challenge" predicates are introduces in the mappings and query to ensure confluence and hence close the grid. The construction relies on GAV mappings and an empty ontology. But it is easy to see that with a non-trivial ontology we can simulate arbitrary GAV using CQ views. Thus, our undecidability result extends to the case of CQ views, provided a very simple ontology is present.

Corollary 15. *Problem* $\mathsf{ComplyAll}(\mathcal{O}, \mathcal{M}, p)$ *is undecidable even for linear Datalog ontologies* $\mathcal{O}$*, sets of CQ views* $\mathcal{M}$*, and the arity of the global schema bounded by 2.*

We now complete the picture for ComplyAll by showing that it is decidable when the mappings are CQ views and there is no ontology. Here, we exploit the *critical instance* method which has been used for both decidability and undecidability results (Gogacz and Marcinkowski 2014; Cuenca Grau et al. 2013a; Benedikt et al. 2016; Baader et al. 2016; Shmueli 1993; Marnette 2010). We show that if there is any non-compliant source instance, then the *critical instance of the source schema* is also a witness to non-compliance. The critical instance $\mathsf{Crit}_{\mathbf{R}}$ for a schema $\mathbf{R}$ is the instance whose domain has one single constant a and whose facts are $R(a, \ldots, a)$ for all $R \in \mathbf{R}$. Note that every CQ holds on the critical instance, and thus it is (intuitively) the "hardest" instance to get to comply.

Theorem 16. *Let* $\mathbf{R}$ *be a source schema,* $\mathcal{M}$ *be a set of CQ views,* p *be a Boolean policy, and both* $\mathcal{M}$ *and* p *be constant-free. Then* $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathsf{Crit}_{\mathbf{R}}, p) = \mathtt{true}$ *if and only if* $\mathsf{ComplyAll}(\emptyset, \mathcal{M}, p) = \mathtt{true}$*.*

From this theorem and the results for Comply in Section 5, we immediately obtain decidability in Σ_2^p of ComplyAll for

the case of CQ views. This upper bound is, however, not tight since we can exploit the special structure of the critical instance to obtain more favourable complexity.

Theorem 17. *The problem* $\mathsf{ComplyAll}(\emptyset, \mathcal{M}, p)$ *for constant-free policies* p*, and sets of constant-free CQ views* $\mathcal{M}$ *is* coNP*-complete; it is in* P *if the CQ views are linear.*

7 Implications of Our Results

We discuss the implications of our work on the literature.

Nash and Deutsch (2006) study similar problems to ours in the context of data integration via GLAV mappings and no ontology. In the discussion below, we focus for simplicity on the case of Boolean policies p. Nash and Deutsch (2006) consider privacy guarantees for Boolean policies that are stricter than ours: they require that neither the policy *nor its negation* can be inferred by an attacker. In Example 1, we could require that the attacker can neither learn that a specific patient has an oncology appointment or that they do not have such an appointment. Following (Benedikt et al. 2016), we can extend the compliance guarantee in (Nash and Deutsch 2006) to account for an ontology as given next. We let $\mathsf{ComplyBoth}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$ be `true` if and only if both $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p)$ and $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \neg p)$ are `true`. Then, a variation of our hardness proof for Comply in case 3 of Theorem 12 gives us also hardness for ComplyBoth.

Theorem 18. *Problem* $\mathsf{ComplyBoth}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ *is* NExpTime*-hard for sets of CQ views* $\mathcal{M}$*; it is* Σ_2^p*-hard for sets of linear CQ views.*

The second result contradicts (modulo standard complexity-theoretic assumptions) a prior NP upper bound established by Corollary 3 of Theorem 3 in (Nash and Deutsch 2006). The bound of Nash and Deutsch (2006) is given in terms of the size of $\mathcal{D}$ and the *rewriting* of the global relations in $\mathcal{M}$ over the source relations. Indeed, our Σ_2^p lower bound holds already for linear views, in which case such rewriting is of linear size in $|\mathcal{M}|$.

We conclude this section by discussing the *instance-based determinacy* problem studied in Koutris et al. (2015).

Let $\mathcal{V}$ be a set of CQ views, $\mathcal{D}$ be a source instance, and p be a CQ over the source schema. We say that $\mathcal{V}$ *determines* p given $\mathcal{D}$ if, for each $\mathcal{D}'$ such that the extension of $\mathcal{V}$ over $\mathcal{D}'$ coincides with the extension of $\mathcal{V}$ over $\mathcal{D}$, the answers to p over $\mathcal{D}$ and $\mathcal{D}'$ also coincide. Then, $\mathsf{Determinacy}(\mathcal{V}, p, \mathcal{D})$ is `true` if and only if $\mathcal{V}$ determines p given $\mathcal{D}$.

Koutris et al. (2015) show that Determinacy is in Π_2^p in the combined size of the views, mappings, source instance and its global extension, but leave the lower bound open. We can observe, however, that, for Boolean queries and the empty ontology, Determinacy is precisely the complement of ComplyBoth. Thus, the following holds by Theorem 18.

Corollary 19. $\mathsf{Determinacy}(\mathcal{V}, p, \mathcal{D})$ *is* coNExpTime *hard; it is* Π_2^p*-hard if the extension of* $\mathcal{D}$ *over* $\mathcal{V}$ *is also part of the input.*

8 Related Work

The problem of preventing information disclosure in information systems has received significant attention in recent years. We focus our discussion on logic-based approaches, which are the closest to our work, and leave out probabilistic techniques such as those in (Dalvi, Miklau, and Suciu 2005; Miklau and Suciu 2007). We also leave out anonymization approaches, which involve modification of the source data (Cuenca Grau et al. 2015; Cuenca Grau et al. 2013b).

Disclosure in the setting where data is materialized is related to "querying with closed predicates", which has drawn much recent attention in the KR community (Lutz, Seylan, and Wolter 2015; Ahmetaj, Ortiz, and Simkus 2016). Our work takes ideas from one paper in this line, Benedikt et al. (2016), which considers the scenario where the materialized contents of visible relations in a relational schema are known to users, whereas the contents of all other tables are hidden. A background theory provides semantic information about both visible and invisible relations. The secret information is provided by a query, and the goal is to determine whether (positive or negative) information about the query can be answered by looking only at the contents of the visible tables. Our instance-level problems for GAV mappings are subsumed by this setting, since we can consider the targets instead of the sources, and can generate a background theory from the mappings and constraints. However, even for GAV mappings, the complexity of our problem is difficult to align with the problems of (Lutz, Seylan, and Wolter 2015; Ahmetaj, Ortiz, and Simkus 2016; Benedikt et al. 2016). Our input is the source instance, whose size may be larger or smaller than the target, while our background theory considers only mappings coupled with an ontology over the global vocabulary, quite different from the assumptions in (Lutz, Seylan, and Wolter 2015; Ahmetaj, Ortiz, and Simkus 2016; Benedikt et al. 2016).

A number of works focus not on policy analysis at design time, as we do, but on policy enforcement at query time. Calvanese et al. (2012) study privacy-aware data access in the presence of ontologies, by extending the database authorization framework by Zhang and Mendelzon (2005). In their setting, users are assigned a set of authorization views; every query is then answered by the system using only the information that follows from the ontology and their respective views. In the *Controlled Query Evaluation* (CQE) framework, a *censor* ensures that query answers that may compromise the policy are either distorted, or not returned to users. CQE was introduced by (Sicherman, de Jonge, and van de Riet 1983) for databases and has received significant attention since (e.g., see (Biskup and Bonatti 2004; Biskup and Weibert 2008; Bonatti, Kraus, and Subrahmanian 1995)) CQE has been recently extended to ontologies in (Cuenca Grau et al. 2015; Bonatti and Sauro 2013; Cuenca Grau et al. 2013b; Studer and Werner 2014). (Guarnieri and Basin 2014) compares policy enforcement and policy restriction based approaches, in the absence of an ontology but for richer query languages (e.g., full relational calculus).

Finally, source indistinguishability is related to query inseparability in knowledge bases as studied by (Botoeva et al. 2016). However, the emphasis in query inseparability is on having distinct ontologies (and not data) and mappings are not present; as a result, the techniques applied are different.

9 Future Work

In this paper, we have provided an analysis of disclosure of source data in an ontology-based integration scenario.

Most of our decidability results are likely to extend to the setting where the sources come with integrity constraints. In future work, we will study the impact of source constraints on the complexity of our problems. We also leave for future work an extended study of the ComplyBoth problem in the presence ontologies and its data-independent version.

Our notion of compliance does not limit the computational resources of the attacker. Although Lemma 5 shows that the attacker can always make due with polynomially many queries, Theorem 12 suggests that it is hard in general for an attacker to determine if the policy holds. Thus, a main open issue is to distinguish the schema/query combinations that are computationally easy (as data varies) for the attacker from those that are hard. Lutz, Seylan, and Wolter (2015) and Lutz, Seylan, and Wolter (2012) did a similar analysis for hybrid closed-and-open world query answering, and their techniques may be directly relevant.

Acknowledgements

Work supported by a Royal Society University Research Fellowship, and EPSRC projects DBonto (EP/L012138/1), ED3 (EP/N014359/1) and PDQ (EP/M005852/1).

References

Ahmetaj, S.; Ortiz, M.; and Simkus, M. 2016. Polynomial datalog rewritings for expressive description logics with closed predicates. In *IJCAI*.

Baader, F.; Bienvenu, M.; Lutz, C.; and Wolter, F. 2016. Query and predicate emptiness in ontology-based data access. *J. Artif. Intell. Res. (JAIR)* 56:1–59.

Benedikt, M.; Bourhis, P.; Puppis, G.; and ten Cate, B. 2016. Querying visible and invisible information. In *LICS*.

Berger, R. 1966. The undecidability of the domino problem. *Memoirs of the American Mathematical Society* 66(72).

Biskup, J., and Bonatti, P. 2004. Controlled query evaluation for enforcing confidentiality in complete information systems. *Int. J. Inf. Sec.* 3(1):14–27.

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

Bonatti, P., and Sauro, L. 2013. A confidentiality model for ontologies. In *ISWC*.

Bonatti, P.; Kraus, S.; and Subrahmanian, V. S. 1995. Foundations of Secure Deductive Databases. *TKDE* 7(3):406–422.

Botoeva, E.; Kontchakov, R.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2016. Games for query inseparability of description logic knowledge bases. *Artif. Intell.* 234:78–119.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning* 39(3):385–429.

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2012. View-based Query Answering in Description Logics: Semantics and Complexity. *J. Comput. Syst. Sci.* 78(1):26–46.

Cuenca Grau, B.; Horrocks, I.; Krötzsch, M.; Kupke, C.; Magka, D.; Motik, B.; and Wang, Z. 2013a. Acyclicity notions for existential rules and their application to query answering in ontologies. *JAIR* 47:741–808.

Cuenca Grau, B.; Kharlamov, E.; Kostylev, E. V.; and Zheleznyakov, D. 2013b. Controlled query evaluation over owl 2 rl ontologies. In *ISWC*.

Cuenca Grau, B.; Kharlamov, E.; Kostylev, E. V.; and Zheleznyakov, D. 2015. Controlled query evaluation for datalog and OWL 2 profile ontologies. In *IJCAI*.

Dalvi, N. N.; Miklau, G.; and Suciu, D. 2005. Asymptotic conditional probabilities for conjunctive queries. In *ICDT*.

Gogacz, T., and Marcinkowski, J. 2014. All-instances termination of chase is undecidable. In *ICALP*.

Guarnieri, M., and Basin, D. A. 2014. Optimal security-aware query processing. *PVLDB* 7(12):1307–1318.

Koutris, P.; Upadhyaya, P.; Balazinska, M.; Howe, B.; and Suciu, D. 2015. Query-based data pricing. *J. ACM* 62(5).

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *PODS*.

Lutz, C.; Seylan, I.; and Wolter, F. 2012. Mixing open and closed world assumption in ontology-based data access: Non-uniform data complexity. In *Description Logics*.

Lutz, C.; Seylan, I.; and Wolter, F. 2015. Ontology-mediated queries with closed predicates. In *IJCAI*.

Marnette, B. 2010. *Tractable schema mappings under oblivious termination*. Ph.D. Dissertation, Oxford Univ. , UK.

Miklau, G., and Suciu, D. 2007. A Formal analysis of information disclosure in data exchange. *J. Comput. Syst. Sci.* 73(3):507–534.

Nash, A., and Deutsch, A. 2006. Privacy in GLAV information integration. In *ICDT*.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking Data to Ontologies. *J. Data Semantics* 10:133–173.

Shmueli, O. 1993. Equivalence of DATALOG queries is undecidable. *J. Log. Program.* 15(3):231–241.

Sicherman, G. L.; de Jonge, W.; and van de Riet, R. P. 1983. Answering queries without revealing secrets. *ACM Trans. Database Syst.* 8(1):41–59.

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

Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(5):557–570.

Wagner, K. W. 1987. More complicated questions about maxima and minima, and some closures of NP. *Theor. Comput. Sci.* 51:53–80.

Zhang, Z., and Mendelzon, A. O. 2005. Authorization views and conditional query containment. In *ICDT*.