Source Information Disclosure in Ontology-Based Data Integration (with proofs)

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

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.

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

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.

Organization. The background and problem definition are introduced in Section 2 and Section 3. The problem of deciding whether two source instances are indistinguishable in an ontology-based data integration setting is studied in Section 4, as a prelude to the study of instance-level compliance in Section 5. The schema-level compliance problem (compliance over all instances) is studied in Section 6. Connections of our work with problems studied previously in the literature are discussed in Section 7, and additional related work is overviewed in Section 8. Finally, conclusions and future work are examined in Section 9. All proofs of the results are either given in the text or in the appendix.

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*

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

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.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.

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*.

Deutsch, A.; Nash, A.; and Remmel, J. 2008. The chase revisited. In *PODS*.

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.

Spakowski, H. 2005. *Completeness for parallel access to NP and counting class separations*. Ph.D. Dissertation, Universität Düsseldorf.

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*.

Appendix A: Proofs of Results in Section 4

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.

Proof. The first statement implies the second by definition.

If the second statement holds, then clearly the third holds, by the observation after the definition of virtual image.

We show that the third statement implies the first.

Assume that $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ are logically equivalent. Let $q(\mathbf{x})$ be an arbitrary CQ over the global relations, and let $\mathbf{a}$ be a binding for the variables of q such that $q(\mathbf{a})$ is a certain answer with respect to $(\mathcal{O}, \mathcal{M}, \mathcal{D})$. We show that the same holds with respect to $(\mathcal{O}, \mathcal{M}, \mathcal{D}')$ as well. If not, then there is an instance $\mathcal{F}'$ over both source and global relations such that it satisfies both $\mathcal{M}$ and $\mathcal{O}$, its source part is precisely $\mathcal{D}'$, and it does not satisfy $q(\mathbf{a})$. Let $\mathcal{F}$ be formed by replacing the interpretation of the source relations in $\mathcal{F}'$ by $\mathcal{D}$. Clearly $\mathcal{F}$ satisfies $\mathcal{O} \wedge \neg q(\mathbf{a})$, since these mention only the global relations, which are unchanged from $\mathcal{F}'$. Also the source part of $\mathcal{F}$ is $\mathcal{D}$ by definition. To see that $\mathcal{F}$ satisfies $\mathcal{M}$, consider any trigger h for the body of a mapping in $\mathcal{D}$. Then the head of the mapping is in $\mathcal{V}_{\mathcal{M},\mathcal{D}}$. Thus by hypothesis the head is implied by $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$. But $\mathcal{F}'$ satisfies $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$, and thus $\mathcal{F}'$ satisfies the head. Since $\mathcal{F}$ agrees with $\mathcal{F}'$ on the global relations, the conclusion follows.

Arguing symmetrically, we see that a certain answer of q with respect to $(\mathcal{O}, \mathcal{M}, \mathcal{D}')$ is a certain answer of q with respect to $(\mathcal{O}, \mathcal{M}, \mathcal{D})$. This completes the proof. □

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.*

Proof.

First statement. We show Π_2^p-hardness by reduction from $\forall\exists$SAT. Let $\varphi = \forall\mathbf{u}.\exists\mathbf{v}.\psi$, where ψ is a conjunction of clauses γ of the form $\ell_1 \vee \ell_2 \vee \ell_3$ for ℓ_j either a propositional variable from $\mathbf{u} \cup \mathbf{v}$ or the negation of such a variable; for each such γ, we denote w_γ^j, for $j = 1, 2, 3$, the variable in ℓ_j. Let also $\mathbf{u} = u_1, \ldots, u_n$.

Let Bool be a unary source relation and let Clause_γ be a unary source relation for each clause γ in ψ. Furthermore, let Arg_j for $j = 1, 2, 3$ and UValue be binary source relations. Let also Target be an n-ary global relation. We define the set of GAV mappings $\mathcal{M}$ that consists of mappings

$$\mathsf{Bool}(y_1) \wedge \cdots \wedge \mathsf{Bool}(y_n) \rightarrow \mathsf{Target}(y_1, \ldots, y_n)$$

and

$$\bigwedge_{\gamma \text{ in } \psi} \left(\mathsf{Clause}_\gamma(x_\gamma) \wedge \bigwedge_{j=1}^{3} \mathsf{Arg}_j(x_\gamma, x_{w_\gamma^j}) \right) \wedge \bigwedge_{i=1}^{n} \mathsf{UValue}(x_{u_i}, y_{u_i}) \rightarrow \mathsf{Target}(y_{u_1}, \ldots, y_{u_n}).$$

Finally, let $\mathcal{D} = \{\mathsf{Bool}(0), \mathsf{Bool}(1)\}$ and let $\mathcal{D}'$ consist of the following facts:

- $\mathsf{Clause}_\gamma(a_\gamma^\pi), \mathsf{Arg}_1(a_\gamma^\pi, b_1), \mathsf{Arg}_2(a_\gamma^\pi, b_2), \mathsf{Arg}_3(a_\gamma^\pi, b_3)$ for each clause $\gamma = \ell_1 \vee \ell_2 \vee \ell_3$ in ψ and each satisfying assignment π of γ, where, for each $j = 1, 2, 3$, $b_j = t_{w_\gamma^j}$ if $\pi(w_\gamma^j) =$ `true` and $b_j = f_{w_\gamma^j}$ if $\pi(w_\gamma^j) =$ `false` (each π assigns only variables in γ, so there are at most 7 of them for γ);
- $\mathsf{UValue}(f_{u_i}, 0)$ and $\mathsf{UValue}(t_{u_i}, 1)$ for all (universally quantified) variables u_i in $\mathbf{u}$.

We argue that formula φ is `true` if and only if $\mathsf{SourceInd}(\emptyset, \mathcal{M}, \mathcal{D}, \mathcal{D}') =$ `true`. Observe that, due to the first mapping of $\mathcal{M}$, $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ contains all facts $\mathsf{Target}(c_1, \ldots, c_n)$ where $c_i \in \{0, 1\}$ for all $i = 1, \ldots, n$. It is now routine to check, using the construction of the second mapping, that $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$ consists of exactly the same facts (thus making $\mathcal{D}$ and $\mathcal{D}'$ indistinguishable) if and only if φ is `true`.

Second statement. We show $\mathrm{P}^{\|\mathrm{NP}}$-hardness by reduction from Max-True-3SAT-Equality problem, which is known to be $\mathrm{P}^{\|\mathrm{NP}}$-complete (see (Spakowski 2005)). Its input is two satisfiable propositional formulas ψ_1 and ψ_2 in 3CNF and the question is whether $\max_1(\psi_1) = \max_1(\psi_2)$, where $\max_1(\psi)$ is the maximum of the number of variables assigned to `true` over all satisfying assignments of ψ.

Consider a formula ψ in 3CNF that is a conjunction of clauses γ of the form $\ell_1 \vee \ell_2 \vee \ell_3$ for ℓ_j either a propositional variable from $\mathbf{u} = u_1, \ldots, u_n$ or the negation of such a variable; for each such γ, denote u_γ^j, for $j = 1, 2, 3$, the variable in ℓ_j. Note that in this proof the order of $u_1, \ldots, u_n$ plays a technical role: it can be arbitrary, but it is important that it is fixed. Next we define a set of mappings $\mathcal{M}_\psi$ and a source instance $\mathcal{D}_\psi$ for ψ.

Let $\mathcal{M}_\psi$ consist of the following mappings, for each $m = 0, \dots, n$:

$$\bigwedge_{\gamma \text{ in } \psi} \left(\mathsf{Clause}^\psi_\gamma(x_\gamma) \wedge \bigwedge_{j=1}^{3} \mathsf{Arg}_j(x_\gamma, x_{u^j_\gamma}) \right) \wedge$$

$$\mathsf{PartialSum}(x_{u_1}, s_1) \wedge \bigwedge_{i=2}^{n} \Big(\mathsf{PreviousValue}(s_{i-1}, r_i) \wedge \mathsf{CurrentValue}(x_{u_i}, r_i) \wedge \mathsf{PartialSum}(r_i, s_i) \Big) \wedge$$

$$\mathsf{TotalSum}_m(s_n) \rightarrow \mathsf{TotalSum}'_m(),$$

and let $\mathcal{D}_\psi$ consist of

– the facts
$$\mathsf{Clause}^\psi_\gamma(a^\pi_\gamma), \mathsf{Arg}_1(a^\pi_\gamma, b_1), \mathsf{Arg}_2(a^\pi_\gamma, b_2), \mathsf{Arg}_3(a^\pi_\gamma, b_3)$$
for each clause $\gamma = \ell_1 \vee \ell_2 \vee \ell_3$ in ψ and each satisfying assignment π of γ, where, for each $j = 1, 2, 3$, $b_j = t_{u^j_\gamma}$ if $\pi(u^j_\gamma) = \texttt{true}$ and $b_j = f_{u^j_\gamma}$ if $\pi(u^j_\gamma) = \texttt{false}$ (as before each π assigns only variables in γ);

– the facts
$$\mathsf{PartialSum}(f_{u_1}, d^0_1), \mathsf{PartialSum}(t_{u_1}, d^1_1), \mathsf{PartialSum}(t_{u_1}, d^0_1),$$

– for each $i = 2, \dots, n$ and $k = 0, \dots, i-1$, the facts
$$\begin{array}{ll} \mathsf{PreviousValue}(d^k_{i-1}, c^{k0}_i), \mathsf{CurrentValue}(f_{u_i}, c^{k0}_i), \mathsf{PartialSum}(c^{k0}_i, d^m_i), & \text{for all } m = 0, \dots, k, \\ \mathsf{PreviousValue}(d^k_{i-1}, c^{k1}_i), \mathsf{CurrentValue}(t_{u_i}, c^{k1}_i), \mathsf{PartialSum}(c^{k1}_i, d^m_i), & \text{for all } m = 0, \dots, k+1, \end{array}$$

– for each $m = 0, \dots, n$, the fact
$$\mathsf{TotalSum}_m(d^m_n).$$

The key property of $\mathcal{M}_\psi$ and $\mathcal{D}_\psi$ is that $\mathcal{V}_{\mathcal{M}_\psi, \mathcal{D}_\psi} \models \mathsf{TotalSum}'_k()$ for a number k if and only if $k \leq \max_1(\psi)$. Indeed, consider a satisfying assignment σ of ψ. As before, the $\mathsf{Clause}^\psi_\gamma$ and Arg_j atoms of the body of every homomorphism can be uniquely mapped to $\mathcal{D}_\psi$ in such that way that each x_{u_i} is mapped to f_{u_i} or t_{u_i} depending of the Boolean value $\sigma(u_i)$. If x_{u_1} is mapped to f_{u_1}, then variable s_1 can be mapped only to d^0_1: intuitively, $\mathsf{PartialSum}(f_{u_i}, d^0_1)$ in the image of the homomorphism represents the fact that at least 0 propositional variables are assigned to $\texttt{true}$ among the first 1 variables. If x_{u_1} is sent to t_{u_1} then s_1 can be sent either to d^0_1 or to d^1_1, representing similar facts. Having a homomorphism defined for all the variables up to s_{i-1}, which is sent to d^k_{i-1}, we know that at least k propositional variables are assigned to $\texttt{true}$ among the first $i-1$ ones. If the current propositional variable u_i is assigned to $\texttt{false}$ by σ, that is, x_{u_i} is sent to f_{u_i}, then r_i can be sent only to c^{k0}_i, and, hence, s_i can be sent to any of $d^0_i, \dots, d^k_i$. Each option d^m_i represents the fact that at least m variables are assigned to $\texttt{true}$ among the first i ones. Similarly, if u_i is assigned to $\texttt{true}$, then s_i can be sent to any of $d^0_i, \dots, d^{k+1}_i$. At the end, s_n can be sent to one of $d^0_n, \dots, d^k_n$ where k is the total number of propositional variables assigned to $\texttt{true}$ by σ, so all $\mathsf{TotalSum}'_0(), \dots, \mathsf{TotalSum}'_k()$ are present in $\mathcal{V}_{\mathcal{M}_\psi, \mathcal{D}_\psi}$. This process goes through for all satisfying assignments, so the key property indeed holds.

Therefore, $\mathsf{Max\text{-}True\text{-}3SAT\text{-}Equality}(\psi_1, \psi_2)$ is $\texttt{true}$ for two satisfiable formulas ψ_1 and ψ_2 in 3CNF if and only if $\mathsf{SourceInd}(\emptyset, \mathcal{M}_{\psi_1} \cup \mathcal{M}_{\psi_2}, \mathcal{D}_{\psi_1}, \mathcal{D}_{\psi_2})$ is $\texttt{true}$. □

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^p_2*-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.*

Proof. We start with the lower bounds. Cases 3 and 4 (Π^p_2-hardness and $\mathrm{P}^{\|\mathrm{NP}}$-hardness, respectively) follow from Theorem 6. Hardness for cases 1, 2, and 5 follow by the following reduction from CQEnt (recall that CQEnt is PSpace-complete for linear ontologies and NP-complete if, additionally, the arity of predicates is bounded). Note that the reduction uses only linear mappings.

Consider an instance of CQEnt consisting of an ontology $\mathcal{O}_0$, an instance $\mathcal{D}_0$, and a Boolean CQ q_0. Let θ be a predicate renaming which maps each relation R in $\mathcal{D}_0$ to a distinct source relation $R\theta$ of the same arity. Furthermore, let Q be a fresh unary predicate. We now define an instance of SourceInd consisting of the following ontology $\mathcal{O}$, mappings $\mathcal{M}$, and source instances $\mathcal{D}$ and $\mathcal{D}'$:

1. $\mathcal{O} = \mathcal{O}_0$,
2. $\mathcal{D} = \mathcal{D}_0\theta$ and $\mathcal{D}' = \mathcal{D} \cup \{Q(a)\}$,
3. $\mathcal{M}$ consists of the set of all linear GAV mappings $R\theta(\mathbf{x}) \to R(\mathbf{x})$, for each relation R in $\mathcal{D}_0$, extended with a linear GLAV mapping $Q(x) \to q_0$.

We argue that $\mathsf{SourceInd}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \mathcal{D}') = \texttt{true}$ if and only if $\mathcal{O}_0 \cup \mathcal{D}_0 \models q_0$. Assume that $\mathsf{SourceInd}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \mathcal{D}') = \texttt{true}$. Then, $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ must be logically equivalent. By construction of $\mathcal{D}'$ and $\mathcal{M}$ we have that $\mathcal{V}_{\mathcal{M},\mathcal{D}'} = \mathcal{V}_{\mathcal{M},\mathcal{D}} \cup \{q_0\}$ and the equivalence implies that $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models q_0$. Since $\mathcal{O} = \mathcal{O}_0$ and $\mathcal{V}_{\mathcal{M},\mathcal{D}} = \mathcal{D}_0$ we have that $\mathcal{O}_0 \cup \mathcal{D}_0 \models q_0$, as required. For the converse, assume that $\mathcal{O}_0 \cup \mathcal{D}_0 \models q_0$. Since $\mathcal{O} = \mathcal{O}_0$ and $\mathcal{V}_{\mathcal{M},\mathcal{D}} = \mathcal{D}_0$, we have that $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models q_0$. Since $\mathcal{V}_{\mathcal{M},\mathcal{D}'} = \mathcal{V}_{\mathcal{M},\mathcal{D}} \cup \{q_0\}$ we then have that $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ and hence $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ must be equivalent, which implies that $\mathsf{SourceInd}(\mathcal{O}, \mathcal{M}, \mathcal{D}, \mathcal{D}') = \texttt{true}$.

We now argue the upper bounds. For cases 1, 5, and 6 we analyse the basic algorithm for indistinguishability:

1. construct the source images $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$;
2. check equivalence of $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$.

Assume that the mappings are linear (which covers cases 5 and 6). Then, the first step of the algorithm requires polynomial time. If we fix $\mathcal{M}$ (i.e., consider data complexity), then it is feasible in AC^0. In turn, the second step requires polynomially many entailment tests if the arity of global predicates is bounded, each of which is feasible in NP in case 5, and in P in case 6 (note that fact entailment under the conditions of case 6 is tractable).

If the mappings are not linear (case 1), then the first step takes exponential time (it requires exponentially many CQ evaluation tests over the source instance in the size of the frontier of $\mathcal{M}$, each of which is feasible in NP). If $\mathcal{M}$ is fixed, then again the process is feasible in AC^0. The (exponential) size of the images $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$ determines the number of entailment tests to be performed. Each test is feasible in C in the size of $\mathcal{O}$ and in P in the size of data. Hence the overall process is feasible in C and in P in data.

The aforementioned basic algorithm does not give us tight upper bounds for cases 2, 3, and 4, which involve linear ontologies or restrictions thereof. For this, we need a specialised algorithm that takes into account the specific properties of linear TGDs. We use the following claim.

Claim 20. *Let $\mathcal{O}$ be a linear ontology and let $\mathcal{M}$ be a set of mappings. Furthermore, let $\mathcal{D}$ and $\mathcal{D}'$ be arbitrary source instances. Then, $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models \mathcal{V}_{\mathcal{M},\mathcal{D}'}$ if and only if the following condition holds: for every mapping $\varphi \to \exists\mathbf{y}.\psi$ in $\mathcal{M}$ and every grounding θ' such that $\varphi\theta' \subseteq \mathcal{D}'$ there are $k \leq |\psi|$ groundings $\theta_1, \dots \theta_k$ of mappings $m_i = \varphi_i \to \exists\mathbf{y}.\psi_i$ in $\mathcal{M}$, with $1 \leq i \leq k$, such that $\varphi_i\theta_i \subseteq \mathcal{D}$ and $\mathcal{O} \cup \{\exists\mathbf{y}.\psi_i\theta_i\}_{i=1}^k \models \exists\mathbf{y}.\psi\theta'$.*

Proof. Assume that the condition in the claim holds. Pick an existentially quantified sentence $\exists\mathbf{y}.\psi(\mathbf{a}, \mathbf{y})$ in $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$. By the definition of $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$ there must exist a mapping $\varphi(\mathbf{x}, \mathbf{z}) \to \exists\mathbf{y}.\psi(\mathbf{x}, \mathbf{y})$ in $\mathcal{M}$ such that $\mathcal{D}' \models \exists\mathbf{z}.\varphi(\mathbf{a}, \mathbf{z})$. But then, there must exist a grounding $\theta' = \{\mathbf{x} \to \mathbf{a}, \mathbf{z} \to \mathbf{c}\}$ such that $\varphi\theta' \subseteq \mathcal{D}'$. By the condition of the lemma, groundings $\theta_1, \dots \theta_k$ of mappings $m_1, \dots, m_k$ must exist such that $\varphi_i\theta_i \subseteq \mathcal{D}$. But then, by the definition of $\mathcal{V}_{\mathcal{M},\mathcal{D}}$, we have that $\{\exists\mathbf{y}.\psi_i\theta_i\}_{i=1}^k \subseteq \mathcal{V}_{\mathcal{M},\mathcal{D}}$ and hence the condition of the lemma ensures that $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models \exists\mathbf{y}.\psi(\mathbf{a}, \mathbf{y})$, as required.

Assume now that $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models \mathcal{V}_{\mathcal{M},\mathcal{D}'}$. Let $\varphi \to \exists\mathbf{y}.\psi$ be a mapping in $\mathcal{M}$ and let θ' be a grounding such that $\varphi\theta' \subseteq \mathcal{D}'$. By the definition of $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$ we have that $\exists\mathbf{y}.\psi\theta' \in \mathcal{V}_{\mathcal{M},\mathcal{D}}$, and hence by our assumption we have that $\mathcal{O} \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models \exists\mathbf{y}.\psi\theta'$. This implies that there exists a substitution $\sigma = \{\mathbf{y} \to \mathbf{b}\}$ such that the set of $|\psi|$ facts $\psi\theta'\sigma$ is contained in the chase of $\mathcal{O}$ union the "freezing" of $\mathcal{V}_{\mathcal{M},\mathcal{D}}$. Since $\mathcal{O}$ consists of linear TGDs, each inference step in the chase depends of at most one fact. It is convenient to think of the chase as a directed graph where nodes represent facts in the chase and the edges describe inference steps; in this setting, if the TGDs are linear, all nodes in the chase are reachable from some root and every node in the chase has at most one incoming edge. Now, each fact α in $\psi\theta'\sigma$ is contained in the chase and has a single root $\mathsf{root}(\alpha)$ associated to it in the chase. Each such $\mathsf{root}(\alpha)$ must be contained in the "freezing" of $\mathcal{V}_{\mathcal{M},\mathcal{D}}$; by the definition of $\mathcal{V}_{\mathcal{M},\mathcal{D}}$, the presence of a fact in the freezing of $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ is justified by a grounding θ_i of a mapping m_i in $\mathcal{M}$ such that $\varphi_i\theta_i \subseteq \mathcal{D}$, which implies the claim. □

For the PSPACE and Π_2^p upper bounds we analyse the following non-deterministic algorithm, which is correct by Claim 20 for linear ontologies:

1. universally guess a mapping $m = \varphi \to \exists\mathbf{y}.\psi$ in $\mathcal{M}$ and a grounding θ' of φ such that $\varphi\theta' \subseteq \mathcal{D}'$;
2. for all (polynomially many) sets $\{\theta_1, \dots, \theta_k\}$ of $k \leq |\psi|$ groundings over $\mathcal{D}$ of mappings $m_1, \dots, m_k$ in $\mathcal{M}$ such that $\varphi_i\theta_i \subseteq \mathcal{D}$ for each $1 \leq i \leq k$, check whether $\mathcal{O} \cup \{\exists\mathbf{y}.\psi_i\theta_i\}_{i=1}^k \models \exists\mathbf{y}.\psi\theta'$; reject the branch if none of the entailment tests hold;
3. repeat Steps 1 and 2 with $\mathcal{D}$ and $\mathcal{D}'$ swapped;
4. accept the branch.

The algorithm works in non-deterministic polynomial space. In particular, each CQ entailment is feasible in polynomial space if $\mathcal{O}$ consists of linear TGDs. If $\mathcal{O} = \emptyset$, then each entailment test in the algorithm amounts to query containment, which is feasible by a call to an NP oracle thus giving us decidability in Π_2^p.

For the $\mathrm{P}^{\|\mathrm{NP}}$ upper bound we analyse the deterministic variant of the previous algorithm:

1. for each mapping $m = \varphi(\mathbf{x}, \mathbf{z}) \rightarrow \exists \mathbf{y}.\psi(\mathbf{x}, \mathbf{y})$ in $\mathcal{M}$ and each grounding θ' of the frontier $\mathbf{x}$ of m
   - set $b :=$ `false`;
   - for all (polynomially many) sets $\{\theta_1, \ldots, \theta_k\}$ of $k \leq |\psi|$ groundings over $\mathcal{D}$ of mappings $m_1, \ldots, m_k$ in $\mathcal{M}$ such that $\varphi_i\theta_i \subseteq \mathcal{D}$ for each $1 \leq i \leq k$
     - if $\mathcal{D}' \models \exists \mathbf{z}.(\varphi\theta')$ and $\mathcal{O} \cup \{\exists \mathbf{y}.\psi_i\theta_i\}_{i=1}^k \models \exists \mathbf{y}.\psi\theta'$, set $b :=$ `true`;
   - if $b =$ `false`, return `false`;
2. repeat Step 1 with $\mathcal{D}$ and $\mathcal{D}'$ swapped;
3. return `true`.

If the frontier of $\mathcal{M}$ is bounded, then the outermost loop has at most polynomially many iterations, and so do the nested loops taken together. The semantic condition inside the loops is feasible with two oracle calls. None of the calls depend on each other, which shows membership in $\mathrm{P}^{\|\mathrm{NP}}$.

Finally, data complexity in AC^0 for all cases 2, 3, and 4 follows from the analysis of the deterministic algorithm for $\mathcal{O}$ and $\mathcal{M}$ fixed; indeed, the data complexity of Boolean CQ entailment for linear TGDs is in AC^0 for fixed $\mathcal{O}$ and the number of iterations of the nested loops is also polynomial if we fix $\mathcal{M}$. □

Appendix B: Proofs of Results in Section 5

Note that Lemma 10 is straightforward, while a proof of Theorem 11 is given after its statement in the main part of the paper.

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.*

Proof. We next show each of the statements of the theorem.

Statement 1. We show NEXPTIME-hardness of $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ for the case where $\mathcal{M}$ is a set of CQ views. Let $(\mathcal{T}, R_H, R_V)$ be a tiling instance, where $\mathcal{T}$ is a finite set of tile types, while R_H and R_V are horizontal and vertical compatibility relations, respectively. We will first show how to construct a set of GAV mappings $\mathcal{M}$, a Boolean UCQ p and a source instance $\mathcal{D}$ such that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) =$ `true` if and only if it is possible to tile a square $2^n \times 2^n$, for $n = |\mathcal{T}|$. After it we explain how to modify this construction to make $\mathcal{M}$ consist of CQ views and p be a single Boolean CQ. For the sake of readability, we allow for (safe) equalities of variables in the bodies of mappings in $\mathcal{M}$, which is clearly just a syntactic sugar.

Let $\mathcal{D}$ consist of the facts

$$\begin{array}{ll} \mathsf{Zero}(0), \mathsf{One}(1), & \\ \mathsf{Tiled}_t(e), & \text{for } t \in \mathcal{T}, \\ \mathsf{HBit}_i(e,0), \mathsf{HBit}_i(e,1), & \text{for } n \geq i \geq 1, \\ \mathsf{VBit}_i(e,0), \mathsf{VBit}_i(e,1), & \text{for } n \geq i \geq 1. \end{array}$$

For $\mathbf{u} = u_n, \ldots, u_1$ and $\mathbf{v} = v_n, \ldots, v_1$, we introduce the following abbreviations:

$$\begin{array}{l} \mathsf{NextCoord}_1(\mathbf{u}, \mathbf{v}) = (u_n = v_n) \wedge \cdots \wedge (u_2 = v_2) \wedge \mathsf{Zero}(u_1) \wedge \mathsf{One}(v_1), \\ \quad \ldots \\ \mathsf{NextCoord}_i(\mathbf{u}, \mathbf{v}) = (u_n = v_n) \wedge \cdots \wedge (u_{i+1} = v_{i+1}) \wedge \\ \quad \mathsf{Zero}(u_i) \wedge \mathsf{One}(v_i) \wedge \mathsf{One}(u_{i-1}) \wedge \mathsf{Zero}(v_{i-1}) \wedge \cdots \wedge \mathsf{One}(u_1) \wedge \mathsf{Zero}(v_1), \\ \quad \ldots \\ \mathsf{NextCoord}_n(\mathbf{u}, \mathbf{v}) = \\ \quad \mathsf{Zero}(u_n) \wedge \mathsf{One}(v_n) \wedge \mathsf{One}(u_{n-1}) \wedge \mathsf{Zero}(v_{n-1}) \wedge \cdots \wedge \mathsf{One}(u_1) \wedge \mathsf{Zero}(v_1). \end{array}$$

Essentially, $\mathsf{NextCoord}_i(\mathbf{u}, \mathbf{v})$ is true when $\mathbf{u}$ and $\mathbf{v}$ represent consecutive binary numbers of the form $b_n \ldots b_{i+1}01 \ldots 1$ and $b_n \ldots b_{i+1}10 \ldots 0$, respectively.

Next, let, for $\mathbf{x} = x_n, \ldots, x_1$ and $\mathbf{y} = y_n, \ldots, y_1$,

$$\mathsf{CoordsOf}(z, \mathbf{x}, \mathbf{y}) = \mathsf{HBit}_n(z, x_n) \wedge \ldots \wedge \mathsf{HBit}_1(z, x_1) \wedge \mathsf{VBit}_n(z, y_n) \wedge \ldots \wedge \mathsf{VBit}_1(z, y_1).$$

Let $\mathcal{M}$ include, for $n \geq i \geq 1$ and $(t_1, t_2) \in R_H$, the following mapping, which checks that all horizontally adjacent cells are assigned with the tile types according to R_H:

$$\mathsf{NextCoord}_i(\mathbf{x}_1, \mathbf{x}_2) \wedge \mathsf{CoordsOf}(z_1, \mathbf{x}_1, \mathbf{y}) \wedge \mathsf{CoordsOf}(z_2, \mathbf{x}_2, \mathbf{y}) \wedge \mathsf{Tiled}_{t_1}(z_1) \wedge \mathsf{Tiled}_{t_2}(z_2) \rightarrow \mathsf{HValid}_i(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}).$$

Similarly, let $\mathcal{M}$ include, for the same i and $(t_1, t_2) \in R_V$, the following mapping, which checks that all vertically adjacent cells are assigned according to R_V:

$$\mathsf{NextCoord}_i(\mathbf{y}_1, \mathbf{y}_2) \wedge \mathsf{CoordsOf}(z_1, \mathbf{x}, \mathbf{y}_1) \wedge \mathsf{CoordsOf}(z_2, \mathbf{x}, \mathbf{y}_2) \wedge \mathsf{Tiled}_{t_1}(z_1) \wedge \mathsf{Tiled}_{t_2}(z_2) \rightarrow \mathsf{VValid}_i(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2).$$

Finally, let p be the union of the following Boolean CQs, for all $t_1, t_2 \in \mathcal{T}$ with $t_1 \neq t_2$:

$$\mathsf{CoordsOf}(z_1, \mathbf{x}, \mathbf{y}) \wedge \mathsf{CoordsOf}(z_2, \mathbf{x}, \mathbf{y}) \wedge \mathsf{Tiled}_{t_1}(z_1) \wedge \mathsf{Tiled}_{t_2}(z_2).$$

Intuitively, in the source there are relations that associated "cell objects" with vertical and horizontal co-ordinates, and also with tile types. The only information exported to the attacker is that adjacent co-ordinates are associated with some cells and with some tile type assignments which are compatible. In fact, in the input source $\mathcal{D}$, a cell with coordinates $(\mathbf{x}, \mathbf{y})$ will be associated with every tile type, since there is only one cell object. But this information is not exported, and thus sources that are indistinguishable from the source $\mathcal{D}$ may be better behaved. In particular, indistinguishable sources where the query p fails will correspond to ones where co-ordinates are associated with a unique tiling type.

Formally, suppose we have a source $\mathcal{D}'$ indistinguishable from $\mathcal{D}$ and satisfying $\neg p$. From indistinguishability we know that for each bit vectors $\mathbf{x}, \mathbf{y}$ there is at least one cell object c and tile type t such that $\mathsf{CoordsOf}(c, \mathbf{x}, \mathbf{y})$ and $\mathsf{Tiled}_t(c)$. Further, the fact that $\neg p$ holds guarantees that there is at most one such t. We define a tiling by giving $\mathbf{x}, \mathbf{y}$ the corresponding tile type, and indistinguishability further guarantees that the compatibility relations are satisfied. Conversely, if we have a tiling, we can define a source $\mathcal{D}'$ by creating a unique c for each bit vectors $\mathbf{x}, \mathbf{y}$, associating c with $\mathbf{x}, \mathbf{y}$ via $\mathsf{CoordsOf}$, and setting $\mathsf{Tiled}_t(c)$ to true only for the tile type given to $\mathbf{x}, \mathbf{y}$ in the tiling. It is easy to see that the resulting $\mathcal{D}'$ is indistinguishable from $\mathcal{D}$ and satisfies $\neg p$.

The mappings in $\mathcal{M}$ are not CQ views, because there is a mapping with the head predicate HValid_i for every pair (t_1, t_2) of compatible types (and, similar mappings for VValid_i). Next we show how to transform these disjunctions to conjunctions. The idea will be to have tile types as objects in a new "type storage" relation, along with a relation storing the compatibility relations. This will allow us to replace a union of mappings ver the different pairs of tile types by an existential quantification.

Let $\mathcal{D}'$ be the same as $\mathcal{D}$, except that instead of the facts $\mathsf{Tiled}_t(e)$, for $t \in \mathcal{T}$, it contains the facts

$$\begin{array}{ll} \mathsf{TypeOf}(e, d_t), \mathsf{TileType}(d_t), & \text{for all } t \in \mathcal{T}, \\ \mathsf{DiffTypes}(t, t') & \text{for all } t \neq t' \in \mathcal{T}, \\ \mathsf{HCompat}(d_{t_1}, d_{t_2}), & \text{for all } (t_1, t_2) \in R_H, \\ \mathsf{VCompat}(d_{t_1}, d_{t_2}), & \text{for all } (t_1, t_2) \in R_V. \end{array}$$

Let $\mathcal{M}'$ contain the mappings

$$\begin{array}{rcll} \mathsf{TileType}_t(w) & \rightarrow & \mathsf{TileType}'_t(w), & \text{for all } t \in \mathcal{T}, \\ \mathsf{DiffTypes}(x, y) & \rightarrow & \mathsf{DiffTypes}'(x, y), & \\ \mathsf{HCompat}(w_1, w_2) & \rightarrow & \mathsf{HCompat}'(w_1, w_2), & \\ \mathsf{VCompat}(w_1, w_2) & \rightarrow & \mathsf{VCompat}'(w_1, w_2). & \end{array}$$

That is, we export the information about what the tiles are, which tiles are different, and which ones are compatible.

Let $\mathcal{M}'$ also contain, for each $n \geq i \geq 1$, the mapping

$$\mathsf{NextCoord}_i(\mathbf{x}_1, \mathbf{x}_2) \wedge \mathsf{CoordsOf}(z_1, \mathbf{x}_1, \mathbf{y}) \wedge \mathsf{CoordsOf}(z_2, \mathbf{x}_2, \mathbf{y}) \wedge \mathsf{TypeOf}(z_1, w_1) \wedge \mathsf{TypeOf}(z_2, w_2) \wedge \mathsf{HCompat}(w_1, w_2) \rightarrow \mathsf{HValid}'_i(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y})$$

and the mapping

$$\mathsf{NextCoord}_i(\mathbf{y}_1, \mathbf{y}_2) \wedge \mathsf{CoordsOf}(z_1, \mathbf{x}, \mathbf{y}_1) \wedge \mathsf{CoordsOf}(z_2, \mathbf{x}, \mathbf{y}_2) \wedge \mathsf{TypeOf}(z_1, w_1) \wedge \mathsf{TypeOf}(z_2, w_2) \wedge \mathsf{VCompat}(w_1, w_2) \rightarrow \mathsf{VValid}'_i(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}).$$

Finally, let p' be the Boolean CQ:

$$\mathsf{CoordsOf}(z_1, \mathbf{x}, \mathbf{y}) \wedge \mathsf{CoordsOf}(z_2, \mathbf{x}, \mathbf{y}) \wedge \mathsf{TypeOf}(z_1, w_1) \wedge \mathsf{TypeOf}(z_2, w_2) \wedge \mathsf{TileType}(w_1) \wedge \mathsf{TileType}(w_2) \wedge \mathsf{DiffTypes}(w_1, w_2).$$

Mappings $\mathcal{M}'$ are CQ views as required.

Assume that we have an indistinguishable source $\mathcal{D}'$ satisfying $\neg p'$. Then $\mathcal{D}'$ must agree with $\mathcal{D}$ on the tiles and their compatibility relations, and as before each pair of vectors is associated with some tile type. Since $\mathcal{D}'$ satisfies $\neg p'$, this type is unique. Indistinguishability guarantees that the tilings satisfy the required compatibility. The other direction is also straightforward.

Statement 2. We prove that CQEnt is reducible to Comply, where the reduction does not change the ontology, and it only involves mappings with empty frontier. The statement then follows for every ontology language with PSPACE-hard CQEnt problem. Consider an instance of CQEnt consisting of an ontology $\mathcal{O}_0$ an instance $\mathcal{D}_0$ and a Boolean conjunctive query q_0. We assume that $\mathcal{O}_0$ and q_0 are both constant-free. Let θ be a predicate renaming which maps each relation R in $\mathcal{D}_0$ to a distinct source relation $R\theta$ of the same arity. Let σ_1 and σ_2 be renamings mapping each constant a in $\mathcal{D}_0$ to a fresh variable x_a and y_a, respectively. Let also $\mathbf{y}$ be the set of all y_a defined above. Finally, let Q be a source unary relation not contained in the range of θ. We then define the instance of Comply consisting of the following ontology $\mathcal{O}$, mappings $\mathcal{M}$, source instance $\mathcal{D}$ and policy p:

- $\mathcal{O} = \mathcal{O}_0$,
- $\mathcal{M} = \{m_1, m_2\}$, where m_1 is $(\mathcal{D}_0\theta)\sigma_1 \to \exists \mathbf{y}.(\mathcal{D}_0\sigma_2)$ and m_2 is defined as $Q(x) \to q_0$,
- $\mathcal{D} = \mathcal{D}_0\theta \cup \{Q(a)\}$,
- $p = \exists x.Q(x)$.

Note that both mappings export no variables and hence have no frontier.

By construction, $\mathcal{V}_{\mathcal{M},D} = \mathcal{D}'_0 \cup \{q_0\}$ where now the freezing of $\mathcal{D}'_0$ is isomorphic to $\mathcal{D}_0$. The implication $\mathsf{CQEnt}(\mathcal{O}_0, \mathcal{D}_0, q_0) = \mathsf{true}$ implies $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p) = \mathsf{true}$ holds easily. Now, assume $\mathsf{Comply}(\mathcal{O}, \mathcal{M}, \mathcal{D}, p) = \mathtt{true}$. This implies that there exists a source instance $\mathcal{D}'$ indistinguishable with $\mathcal{D}$ for which the policy does not hold. But now, by the way we defined the mappings, it must be the case that $\mathcal{V}_{\mathcal{M},\mathcal{D}} = \mathcal{V}_{\mathcal{M},\mathcal{D}'} \cup \{q_0\}$. As a result, $\mathcal{O}_0 \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models \mathcal{O}_0 \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'}$. Furthermore, $\mathcal{O}_0 \cup \mathcal{V}_{\mathcal{M},\mathcal{D}'} \models q_0$ and hence we have $\mathcal{O}_0 \cup \mathcal{V}_{\mathcal{M},\mathcal{D}} \models q_0$. Since $\mathcal{D}_0$ and the freezing od $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ are isomorphic, we also have $\mathcal{O}_0 \cup \mathcal{D}_0 \models q_0$, as required.

Statement 3. We show Π^p_2-hardness of the complement of Comply by reduction of $\forall\exists$SAT. Let $\varphi = \forall \mathbf{u}.\exists \mathbf{v}.\psi$, where ψ is a conjunction of clauses of the form $\ell_1 \vee \ell_2 \vee \ell_3$ for ℓ_i either a propositional variable from $\mathbf{u} \cup \mathbf{v}$ or the negation of such a variable. Let $\mathcal{M}$ consist of mappings

$$\begin{array}{rcll}
\mathsf{Clause}_\gamma(x) & \to & \mathsf{Clause}'_\gamma(x), & \text{for all clauses } \gamma \text{ in } \psi,\\
\mathsf{Arg}_i(x,y) & \to & \mathsf{Arg}'_i(x,y), & \text{for all } i = 1,2,3,\\
\mathsf{UVar}_u(x,y) & \to & \mathsf{UVar}'_u(x), & \text{for all } u \in \mathbf{u},\\
\mathsf{UVar}_u(x,y) & \to & \mathsf{Choice}'_u(y), & \text{for all } u \in \mathbf{u},\\
\mathsf{Chosen}_u(x) & \to & \mathsf{Chosen}'_u(x), & \text{for all } u \in \mathbf{u},\\
\mathsf{EVar}_v(x) & \to & \mathsf{EVar}'_v(x), & \text{for all } v \in \mathbf{v}.
\end{array}$$

Let p be the Boolean CQ that has the following atoms:

$$\begin{array}{rl}
\mathsf{UVar}_u(x_u, y_u), \mathsf{Chosen}_u(y_u), & \text{for all } u \in \mathbf{u},\\
\mathsf{EVar}_v(x_v), & \text{for all } v \in \mathbf{v};
\end{array}$$

and the following atoms for each clause $\gamma = \ell_1 \vee \ell_2 \vee \ell_3$ in ψ:

$$\begin{array}{rl}
\mathsf{Clause}_\gamma(x_\gamma),\\
\mathsf{Arg}_i(x_\gamma, x_w), & \text{for all } i = 1,2,3 \text{ and for the variable } w \in \mathbf{u} \cup \mathbf{v} \text{ of } \ell_i.
\end{array}$$

Let $\mathcal{D}$ consist of

- the facts
$$\mathsf{Clause}_\gamma(a^\pi_\gamma), \mathsf{Arg}_1(a^\pi_\gamma, b_1), \mathsf{Arg}_2(a^\pi_\gamma, b_2), \mathsf{Arg}_3(a^\pi_\gamma, b_3),$$
for each $\gamma = \ell_1 \vee \ell_2 \vee \ell_3$ in ψ and each satisfying assignment π of γ (π assigns only variables of γ, so there are at most 7 assignments π for each γ), where, for each $i = 1,2,3$ and for the variable w of ℓ_i, $b_i = t_w$ if $\pi(w) = \mathtt{true}$ and $b_i = f_w$ if $\pi(w) = \mathtt{false}$;
- the facts
$$\begin{array}{l}
\mathsf{UVar}_u(t_u, c^+_u), \mathsf{UVar}_u(f_u, c^+_u), \mathsf{UVar}_u(t_u, c^-_u), \mathsf{UVar}_u(f_u, c^-_u),\\
\mathsf{Chosen}_u(c^+_u),
\end{array}$$
for all variables $u \in \mathbf{u}$;
- the facts
$$\mathsf{EVar}_v(t_v), \mathsf{EVar}_v(f_v),$$
for all variables $v \in \mathbf{v}$.

Next we give an intuition for the reduction. Source database $\mathcal{D}$ encodes all the satisfying assignments of all the clauses. In particular, it has a constant a_γ^π for each assignment π of variables of each clause γ, and constants t_w, f_w for `true` and `false` values of each propositional variable w (both universally and existentially quantified). Each universally quantified variable u is additionally associated with two constants c_u^+ and c_u^-. Each constant a_γ^π is connected to the corresponding values of its variables by means of binary predicates Arg_i. Both constants t_u and f_u for values of each universally quantified variable u are additionally connected to both of the corresponding constants c_u^+ and c_u^- by predicate UVar_u. Mappings $\mathcal{M}$ copy all the source database, except UVar_u, which is exported only by means of projections UVar'_u and Choice'_u on the first and the second argument, respectively. Therefore, all the source databases indistinguishable from $\mathcal{D}$ differ from $\mathcal{D}$ only in UVar_u: the attacker knows that each of t_u and f_u, for all universally quantified u, is connected by UVar_u to at least one of c_u^+ and c_u^-, and, conversely, each of c_u^+ and c_u^- is connected to at least one of t_u and f_u. In other words, these indistinguishable sources represent all possible assignments of universally quantified variables by means of the value constants connected to corresponding c_u^+ (the sources with both value constants of some u connected to c_u^+ may be seen as representing several assignments). The task of the attacker is to check that the policy query p holds in all these sources, which correspond to all possible assignments of $\mathbf{u}$. A homomorphism from p to any source sends each clause variable x_γ to one of its assignment variables a_γ^π, and each x_w to one of its value constants t_w, f_w. If w is existential then the choice is free. However, if w is universal, then x_w must be sent to the value constant that is connected to c_w^+, because only c_w^+ is in Chosen_w as required by p. Therefore, such a homomorphism exists if and only if for all assignments of $\mathbf{u}$ (i.e., for all indistinguishable sources) there exists an assignment of $\mathbf{v}$ that satisfies all the clauses of ψ.

Next we formally prove that the reduction is correct.

Suppose first that φ holds. That is, for any assignment of $\mathbf{u}$ there exists an assignment of $\mathbf{v}$ such that ψ evaluates to `true`. We need to show that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ does not hold. That is, for each source database $\mathcal{D}'$ indistinguishable from $\mathcal{D}$ there is a homomorphism from p to $\mathcal{D}'$. Consider any such $\mathcal{D}'$. By construction, for each $u \in \mathbf{u}$ at least one of $\mathsf{UVar}_u(t_u, c_u^+)$ and $\mathsf{UVar}_u(f_u, c_u^+)$ is in $\mathcal{D}'$. Consider the assignment σ of $\mathbf{u}$ such that, for every u, $\sigma(u) = \texttt{true}$ if $\mathsf{UVar}_u(t_u, c_u^+)$ is in $\mathcal{D}'$ and $\sigma(u) = \texttt{false}$ otherwise. Since φ holds, σ can be extended to $\mathbf{v}$ such that ψ is `true`. Taking such an extension, let, for every $w \in \mathbf{u} \cup \mathbf{v}$, s_w be t_w if $\sigma(w) = \texttt{true}$ and f_w otherwise. Consider now the mapping h of variables of p to constants of $\mathcal{D}'$ such that

- for each clause $\gamma = \ell_1 \vee \ell_2 \vee \ell_3$, $h(x_\gamma) = a_\gamma^\pi$, where π is the assignment that sends w of each ℓ_i to $\sigma(w)$,
- for each $w \in \mathbf{u} \cup \mathbf{v}$, $h(x_w) = s_w$,
- for each $u \in \mathbf{u}$, $h(y_u) = c_u^+$.

By construction, h is a homomorphism from p to $\mathcal{D}'$.

Suppose now that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ does not hold. That is, for each $\mathcal{D}'$ indistinguishable from $\mathcal{D}$ there is a homomorphism from p to $\mathcal{D}'$. We need to show that for any assignment of $\mathbf{u}$ there exists an assignment of $\mathbf{v}$ such that ψ evaluates to `true`. Consider any assignment σ of $\mathbf{u}$. By construction, there is an indistinguishable $\mathcal{D}'$ such that, for any $u \in \mathbf{u}$, $\mathsf{UVar}_u(t_u, c_u^+)$ is in $\mathcal{D}'$ and $\mathsf{UVar}_u(f_u, c_u^+)$ is not if $\sigma(u) = \texttt{true}$ and vice versa otherwise. Consider a homomorphism h from p to $\mathcal{D}'$. It agrees with σ in the sense that $h(x_u) = t_u$ if $\sigma(u) = \texttt{true}$ and $h(x_u) = f_u$ otherwise. Extend σ to $\mathbf{v}$ in the same way: let, for each $v \in \mathbf{v}$, $\sigma(v) = \texttt{true}$ if $h(x_v) = t_v$ and $\sigma(v) = \texttt{false}$ otherwise. By construction, σ is a satisfying assignment of ψ, as required.

Statement 4. We argue that there is a set of linear CQ view mappings $\mathcal{M}$ and a CQ p such that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ is NP-hard as $\mathcal{D}$ varies over instances. This follows directly from the proof of Lemma 3.8 in (Koutris et al. 2015), which involves only CQ mappings. □

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*.*

Proof. We first argue correctness of our repair algorithm. If it succeeds, the constructed $\mathcal{D}'$ is clearly a witness to compliance since it does not contain the policy and satisfies $\mathcal{V}_{\mathcal{M},\mathcal{D}} = \mathcal{V}_{\mathcal{M},\mathcal{D}'}$. Conversely, suppose that some $\mathcal{D}_0$ witnesses compliance. Without loss of generality, we can assume that $\mathcal{D}_0$ contains all facts of $\mathcal{D} \setminus \{p\}$; otherwise, we can union $\mathcal{D}_0$ with $\mathcal{D} \setminus \{p\}$ and the result would still be a witness to compliance. Consider an uncovered fact. Since $\mathcal{D}_0$ witnesses compliance, there must be some fact $R(\mathbf{c}, \mathbf{d})$ in $\mathcal{D}_0$ which is different from p and generates $U(\mathbf{c})$. By replacing with nulls all the constants from $\mathbf{d}$ not already in $\mathcal{D}$, we obtain a fact satisfying both requirements in the algorithm. Thus, the algorithm returns true.

Finally, we analyse the algorithm's running time. Since $\mathcal{M}$ is linear, the images $\mathcal{V}_{\mathcal{M},\mathcal{D}}$ and $\mathcal{V}_{\mathcal{M},\mathcal{D}'}$ can be constructed in polynomial time. Also, if the source arity is bounded, then the algorithm considers only polynomially many candidate facts in Step 3. Thus, the overall process runs in polynomial time. □

Appendix C: Proofs of Results in Section 6

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.*

Proof. We start with reducing the problem of tiling an infinite grid, which is known to be undecidable, to the complement of a relaxed version of $\mathsf{ComplyAll}(\emptyset, \mathcal{M}, p)$, in which p may be a UCQ and the source instance is allowed to be infinite, and then discuss how to adapt the proof to get rid of the relaxations.

The tiling problem takes as input a finite set $\mathcal{T}$ of tile types, along with sets $R_H, R_V \subseteq \mathcal{T} \times \mathcal{T}$, which are the horizontal and vertical compatibility relations. The goal is to assign elements of $\mathcal{T}$ to each pair of numbers (m, n) such that the tile types assigned to (m, n) and $(m + 1, n)$ are in the relation R_H while the types assigned to (m, n) and $(m, n + 1)$ are in the relation R_V. Equivalently, it suffices to create a structure containing unary relations Tiled_t for $t \in \mathcal{T}$ and binary relations Hor and Ver, such that:

- elements are assigned a unique relation Tiled_t, thus associating elements with tile types,
- Hor-related elements have their associated tile types in the relation in R_H, while Ver-related elements have associated tile types in R_V,
- relations Hor and Ver are functional,
- from any element traversing a Hor edge and then a Ver edge leads to the same element as traversing first a Ver and then a Hor.

Let $(\mathcal{T}, R_H, R_V)$ be a tiling instance. We will show how to construct a set of mappings $\mathcal{M}$ and a Boolean UCQ p such that there exists a (possibly infinite) source database $\mathcal{D}$ with $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{false}$ if and only if it is possible to tile an infinite plane with the tiling instance. Intuitively, the only possibility for any such source database will be to "represent" the tiling of the plane.

We construct mappings $\mathcal{M}$ and query p in several steps, proving in parallel the properties of the corresponding components. Boolean query p is in fact going to be a conjunction of Boolean UCQs, and its transformation to a single UCQ is standard. To avoid multiple indexes, we will use the same variables in the conjuncts of p. Since we also omit existential quantifiers as usual, the reader should keep in mind that the variables are local for the conjuncts.

We first show how to enforce the existence of a "square of successors" for each element. Let the first block $\mathcal{M}_1$ of $\mathcal{M}$ be

$$\begin{aligned} \mathsf{Hor}(x, y) &\rightarrow \mathsf{Hor}'(x, y), \\ \mathsf{Ver}(x, y) &\rightarrow \mathsf{Ver}'(x, y), \\ \mathsf{ConfCh}(x) \wedge \mathsf{Hor}(x, y) &\rightarrow \mathsf{ConfCh}^*(), \end{aligned}$$

where Hor and Ver are binary relations responsible for horizontal and vertical successors, respectively, Hor' and Ver' are their copies in the global schema, while ConfCh and ConfCh^* are special "challenge" predicates. Let the first conjunct p_1 of p be the Boolean CQ formed by existentially quantifying the following formula:

$$\mathsf{ConfCh}(x) \wedge \mathsf{Hor}(x, y) \wedge \mathsf{Ver}(x, z) \wedge \mathsf{Hor}(z, u) \wedge \mathsf{Ver}(y, u) \wedge \mathsf{Hor}(y, v) \wedge \mathsf{Hor}(u, w).$$

Claim 21. *Given a source instance* $\mathcal{D}$ *such that* $\mathcal{D} \models p_1$, $\mathsf{Comply}(\emptyset, \mathcal{M}_1, \mathcal{D}, p_1) = \texttt{false}$ *if and only if for each fact* $\mathsf{Hor}(a_x, a)$ *in* $\mathcal{D}$ *there are also facts*

$$\mathsf{Hor}(a_x, a_y), \mathsf{Ver}(a_x, a_z), \mathsf{Hor}(a_z, a_u), \mathsf{Ver}(a_y, a_u), \mathsf{Hor}(a_y, a_v), \mathsf{Hor}(a_u, a_w) \tag{1}$$

in $\mathcal{D}$ *for some constants* a_y, a_z, a_u, a_v *and* a_w.

Proof. We start with the forward direction. Since $\mathcal{D} \models p_1$, $\mathcal{M}_1 \cup \mathcal{D} \models \mathsf{ConfCh}^*()$. On the one hand, all source instances $\mathcal{D}'$ indistinguishable from $\mathcal{D}$ differ from $\mathcal{D}$ only in the precise element a_x such that $\mathsf{Hor}(a_x, a)$ and $\mathsf{ConfCh}(a_x)$ (and, possibly, some irrelevant atoms). On the other, since $\mathsf{Comply}(\emptyset, \mathcal{M}_1, \mathcal{D}, p_1) = \texttt{false}$, for all indistinguishable $\mathcal{D}'$ it holds that $\mathcal{D}' \models p_1$. Therefore, each element a_x with $\mathsf{Hor}(a_x, a)$ in $\mathcal{D}'$ for some a must also have atoms (1) in $\mathcal{D}'$ as required.

Next we show the backward direction. Again, since $\mathcal{D} \models p_1$, $\mathcal{M}_1 \cup \mathcal{D} \models \mathsf{ConfCh}^*()$. Therefore, each indistinguishable $\mathcal{D}'$ contains atoms $\mathsf{Hor}(a_x, a)$ and $\mathsf{ConfCh}(a_x)$ for some a_x. Since it also contains atoms (1) for this a_x, we have that $\mathcal{D}' \models p_1$, as required. □

Next, we enforce functionality of Hor and Ver, which together with the previous property guarantees that they form a grid-like structure. Let the second block $\mathcal{M}_2$ of mappings in $\mathcal{M}$ be

$$\begin{aligned} \mathsf{Hor}(x, y_1) \wedge \mathsf{Hor}(x, y_2) \wedge \mathsf{HorCh}_1(y_1) \wedge \mathsf{HorCh}_2(y_2) &\rightarrow \mathsf{H}^*(), \\ \mathsf{Ver}(x, y_1) \wedge \mathsf{Ver}(x, y_2) \wedge \mathsf{VerCh}_1(y_1) \wedge \mathsf{VerCh}_2(y_2) &\rightarrow \mathsf{V}^*(), \end{aligned}$$

where HorCh_i, VerCh_i, H^* and V^* are again special challenge predicates. Let the second conjunct p_2 of p be the conjunction of the following Boolean CQs (recall that variables x and y are local for both CQs):

$$\mathsf{Hor}(x,y) \land \mathsf{HorCh}_1(y) \land \mathsf{HorCh}_2(y),$$
$$\mathsf{Ver}(x,y) \land \mathsf{VerCh}_1(y) \land \mathsf{VerCh}_2(y).$$

The intuition is that the challenge predicates represent a test of a pair of x, y_1 and x, y_2 that are both in the Hor relation or both in the Ver relation. By the setting the challenge predicates with only this particular y_1 and y_2 we get an indistinguishable instance, and non-compliance would imply that this instance must satisfy the query, which would imply that $y_1 = y_2$.

A predicate P is *functional* in an instance $\mathcal{I}$ if for no element a there are distinct a_1 and a_2 with both $\mathsf{P}(a, a_1)$ and $\mathsf{P}(a, a_2)$ in $\mathcal{I}$.

Claim 22. *Given a source instance $\mathcal{D}$ such that $\mathcal{D} \models p_1 \land p_2$, $\mathsf{Comply}(\emptyset, \mathcal{M}_1 \cup \mathcal{M}_2, \mathcal{D}, p_1 \land p_2) = \texttt{false}$ if and only if*

- *the condition in Claim 21 holds;*
- Hor *and* Ver *are functional in $\mathcal{D}$.*

Proof. In one direction, suppose $\mathsf{Comply}(\emptyset, \mathcal{M}_1 \cup \mathcal{M}_2, \mathcal{D}, p_1 \land p_2) = \texttt{false}$ and one of the conditions above fails. By the previous claim we see that it cannot be the first item. If Hor is not functional then for some x and distinct y_1, y_2 we have $\mathsf{Hor}(x, y_1) \land \mathsf{Hor}(x, y_2)$. We create $\mathcal{D}'$ by letting HorCh_1 hold only of y_1 and HorCh_2 hold only of y_2. Thus $\mathcal{D}'$ does not satisfy p_2 (and will not satisfy the other conjuncts as before). The second item implies that $\mathcal{D}$ will have $\mathsf{H}^*()$ in its virtual instance, and the first mapping will imply that $\mathcal{D}'$ will have $\mathsf{H}^*()$ as well. Using this, we can see that $\mathcal{D}'$ and $\mathcal{D}$ are indistinguishable, contradicting the hypothesis that $\mathsf{Comply}(\emptyset, \mathcal{M}_1 \cup \mathcal{M}_2, \mathcal{D}, p_1 \land p_2) = \texttt{false}$. The case of Ver is argued similarly.

In the other direction, suppose the conditions above hold, but $\mathsf{Comply}(\emptyset, \mathcal{M}_1 \cup \mathcal{M}_2, \mathcal{D}, p_1 \land p_2) = \texttt{true}$ with a witness $\mathcal{D}'$ indistinguishable from $\mathcal{D}$. Indistinguishability of $\mathcal{D}'$ and $\mathcal{D}$ coupled with the fact that $\mathcal{D} \models p_1 \land p_2$ imply that $\mathcal{D}'$ contains a_x, a_{y_1}, a_{y_2} with $\mathsf{Hor}(a_x, a_{y_1}), \mathsf{Hor}(a_x, a_{y_2}), \mathsf{HorCh}_1(a_{y_1}), \mathsf{HorCh}_2(a_{y_2})$ as well as b_x, b_{y_1}, b_{y_2} with $\mathsf{Ver}(b_x, b_{y_1}), \mathsf{Ver}(b_x, b_{y_2}), \mathsf{VerCh}_1(b_{y_1}), \mathsf{VerCh}_2(b_{y_2})$. Further, since $\mathcal{D}'$ cannot satisfy p_2 we must have either $a_{y_1} \neq a_{y_2}$ or $b_{y_1} \neq b_{y_2}$. But either of these contradicts the second item above. □

The immediate corollary is that a non-compliant $\mathcal{D}$ contains a positive quadrant of a plane. That is, there exists a homomorphism from the infinite grid on H and V with a start point to $\mathcal{D}$.

The next step is to guarantee that each node in the grid is assigned with a tile type. Let the third block $\mathcal{M}_3$ of $\mathcal{M}$ consist of the mappings

$$\begin{array}{rcll} \mathsf{Tiled}_t(x) & \to & \mathsf{Tiled}'_t(x), & \text{for } t \in \mathcal{T}, \\ \mathsf{TileCh}(x) \land \mathsf{Hor}(x,y) & \to & \mathsf{TileCh}^*(), & \end{array}$$

and let the third conjunct p_3 of p be the Boolean query

$$\mathsf{TileCh}(x) \land \mathsf{Hor}(x,y) \land \left(\bigvee\nolimits_{t \in \mathcal{T}} \mathsf{Tiled}_t(x) \right).$$

Intuitively, TileCh is a predicate challenging that a particular node has some tile type.

Claim 23. *Given a source instance $\mathcal{D}$ such that $\mathcal{D} \models p_1 \land p_2 \land p_3$, $\mathsf{Comply}(\emptyset, \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3, \mathcal{D}, p_1 \land p_2 \land p_3) = \texttt{false}$ if and only if*

- *the conditions in Claim 22 hold (including the one from Claim 21);*
- *for every a_x such that $\mathsf{Hor}(a_x, a_y)$ in $\mathcal{D}$ for some a_y there is $t \in \mathcal{T}$ with $\mathsf{Tiled}_t(a_x)$ in $\mathcal{D}$.*

Proof. In one direction, suppose $\mathsf{Comply}(\emptyset, \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3, \mathcal{D}, p_1 \land p_2 \land p_3)$ is $\texttt{false}$. The first item holds as before. To see the second item, given $\mathsf{Hor}(a_x, a_y) \in \mathcal{D}$ we modify $\mathcal{D}$ to $\mathcal{D}'$ by letting TileCh hold only on a_x. Then $\mathcal{D}'$ is indistinguishable from $\mathcal{D}$, and hence satisfies p_3 by assumption on $\mathcal{D}$, which is only possible if $\mathsf{Tiled}_t(a_x)$ holds in $\mathcal{D}$.

In the other direction, suppose $\mathcal{D}$ satisfies the properties above and consider any $\mathcal{D}'$ indistinguishable from $\mathcal{D}$. The fact that $\mathcal{D} \models p_1 \land p_2 \land p_3$ implies that $\mathsf{TileCh}^*()$ is exported from $\mathcal{D}$, and hence must be exported from $\mathcal{D}'$ as well. Thus there are $a_x, a_y \in D'$ satisfying $\mathsf{TileCh}(a_x) \land \mathsf{Hor}(a_x, a_y)$. Further since in the other mappings Hor is exported, we must have $\mathsf{Hor}(a_x, a_y)$ holding in $\mathcal{D}$ as well. By the second item $\mathsf{Tiled}_t(a_x)$ holds in $\mathcal{D}$ for some $t \in \mathcal{T}$. Since other mappings export Tiled_t, we know that $\mathsf{Tiled}_t(a_x)$ holds in $\mathcal{D}'$, which guarantees that p_3 holds in $\mathcal{D}'$ as required. □

The final step is to guarantee that no node is assigned with two differenttile types and the assignment is compatibility-preserving. Let the forth block $\mathcal{M}_4$ of $\mathcal{M}$ be

$$\begin{array}{rcll} \mathsf{OverlapCh}() & \to & \mathsf{OverlapCh}^*(), & \\ \mathsf{Tiled}_{t_1}(x) \land \mathsf{Tiled}_{t_2}(x) & \to & \mathsf{OverlapCh}^*(), & \text{for } t_1, t_2 \in \mathcal{T}, t_1 \neq t_2, \\ \mathsf{Tiled}_{t_1}(x) \land \mathsf{Hor}(x,y) \land \mathsf{Tiled}_{t_2}(y) & \to & \mathsf{OverlapCh}^*(), & \text{for } t_1, t_2 \in \mathcal{T}, (t_1, t_2) \notin R_H, \\ \mathsf{Tiled}_{t_1}(x) \land \mathsf{Ver}(x,y) \land \mathsf{Tiled}_{t_2}(y) & \to & \mathsf{OverlapCh}^*(), & \text{for } t_1, t_2 \in \mathcal{T}, (t_1, t_2) \notin R_V, \end{array}$$

and let the fourth conjunct p_4 of p be the Boolean CQ

$$\mathsf{OverlapCh}().$$

Let $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3 \cup \mathcal{M}_4$ and $p = p_1 \wedge p_2 \wedge p_3 \wedge p_4$.

Claim 24. *Given a source instance $\mathcal{D}$ such that $\mathcal{D} \models p$, $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{false}$ for a source instance $\mathcal{D}$ if and only if*

- *the conditions in Claim 23 hold; and*
- *the assignment of tile types is unique and agrees with compatibility relations R_H and R_V.*

Proof. In one direction, suppose $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{false}$ and that one of the conditions above fails. By the prior arguments we know it is not the first. Therefore, there is a node assigned multiple tiles in $\mathcal{D}$ or an edge, horizontal or vertical, with an incompatible assignment. Since $\mathcal{D} \models p$, it contains $\mathsf{OverlapCh}()$. Create $\mathcal{D}'$ from $\mathcal{D}$ by removing $\mathsf{OverlapCh}()$. The mappings imply that $\mathsf{OverlapCh}^*()$ still holds in the virtual image of $\mathcal{D}$, and from this we can see that $\mathcal{D}'$ is indistinguishable from $\mathcal{D}$. This contradicts $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{false}$.

In the other direction, suppose that the conditions hold, and, for the sake of contradiction, that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{true}$ with a witness instance $\mathcal{D}'$. In $\mathcal{D}'$ $\mathsf{OverlapCh}()$ must be $\texttt{false}$, but the fact that $\mathcal{D} \models p$ implies that $\mathsf{OverlapCh}^*()$ is $\texttt{true}$ in the target. Thus in $\mathcal{D}'$ one of the other mappings leading to $\mathsf{OverlapCh}^*()$ must fire. This contradicts the third item. □

We conclude that there exists a source database $\mathcal{D}$ with $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{false}$ if and only if it is possible to tile an infinite plane with the tiling instance, that is, Comply is undecidable for UCQs as policies.

Next we show how to modify this construction to guarantee that p is a Boolean CQ. We will first extend $\mathcal{M}_1$ and p_1 to guarantee that for each fact $\mathsf{Hor}(a_x, a)$ in a non-compliant $\mathcal{D}$ each node a_x has attached to it a "path gadget": for each tile type t, a_x has a connection to an element e_t that in turn connects back to elements with all types beside t. Thus if a_x is itself tiled, it will have a connection to an element e that connects back to elements tiled with all tile types. By arranging this we convert the disjunctive property that a_x has some tile type to the conjunctive property that a_x connects up and back to elements of all tile types. The details of this will be a bit subtle, since we need to ensure that there are not "spurious occurrences" of the desired conjunctive property.

For the first step, let $\mathcal{M}_1'$ extend $\mathcal{M}_1$ with the mappings

$$\begin{aligned}
\mathsf{GadgetConnect}_t(x, y) &\rightarrow \mathsf{GadgetConnect}'_t(x, y), && \text{for } t \in \mathcal{T},\\
\mathsf{Tiled}_t(x) &\rightarrow \mathsf{Tiled}'_t(x), && \text{for } t \in \mathcal{T},\\
\mathsf{Hor}(x, y) \wedge \mathsf{GadgetConnect}_t(x, z) \wedge & \\
\mathsf{GadgetConnect}_s(u, z) \wedge \mathsf{STileCh}(u, z) &\rightarrow \mathsf{STileCh}'(u, z), && \text{for } t, s \in \mathcal{T}.
\end{aligned}$$

Here $\mathsf{GadgetConnect}_t$ are our connection predicates. Note that the second set of mappings in $\mathcal{M}_1'$ are a part of $\mathcal{M}_3$; we will see later that $\mathcal{M}_3'$ does not include this set.

Let p_1' be the Boolean CQ formed by conjoining all the atoms of p_1 and

$$\bigwedge_{t \in \mathcal{T}} \Big(\mathsf{GadgetConnect}_t(x, x_t) \wedge \bigwedge_{s \in \mathcal{T} \setminus \{t\}} \big(\mathsf{GadgetConnect}_s(x_t^s, x_t) \wedge \mathsf{STileCh}(x_t^s, x_t) \wedge \mathsf{Tiled}_s(x_t^s) \big) \Big).$$

The following claim establishes that in non-compliant instances each element has the "path gadget" attached to it.

Claim 25. *Given a source instance $\mathcal{D}$ such that $\mathcal{D} \models p_1'$, $\mathsf{Comply}(\emptyset, \mathcal{M}_1', \mathcal{D}, p_1') = \texttt{false}$ if and only if for each fact $\mathsf{Hor}(a_x, a)$ in $\mathcal{D}$ there are also facts*

$$\begin{aligned}
&\mathsf{Hor}(a_x, a_y), \mathsf{Ver}(a_x, a_z), \mathsf{Hor}(a_z, a_u), \mathsf{Ver}(a_y, a_u), \mathsf{Hor}(a_y, a_v), \mathsf{Hor}(a_u, a_w),\\
&\mathsf{GadgetConnect}_t(a_x, a_{x_t}), \mathsf{GadgetConnect}_s(a_{x_t^s}, a_{x_t}), \mathsf{STileCh}(a_{x_t^s}, a_{x_t}), \mathsf{Tiled}_s(a_{x_t^s}), \qquad \text{for all } t \in \mathcal{T}, s \in \mathcal{T} \setminus \{t\},
\end{aligned}$$

in $\mathcal{D}$ for some constants a_y, a_z, a_u, a_v, a_w, as well as a_{x_t} and $a_{x_t^s}$ for all $t, s \in \mathcal{T}$, $s \neq t$.

Note that the first set of facts is as in Claim 21.

Proof. We start with the forward direction. Since $\mathcal{D} \models p_1'$, and $\mathcal{M}_1'$ includes $\mathcal{M}_1$, the virtual image of $\mathcal{D}$ satisfies $\mathsf{ConfCh}^*()$. All source instances $\mathcal{D}'$ indistinguishable from $\mathcal{D}$ differ from $\mathcal{D}$ only in the precise element a_x such that $\mathsf{Hor}(a_x, a)$ and $\mathsf{ConfCh}(a_x)$ (and, possibly, some atoms that will be irrelevant for the remainder of the argument). Thus since $\mathsf{Comply}(\emptyset, \mathcal{M}_1', \mathcal{D}, p_1') = \texttt{false}$, for all indistinguishable $\mathcal{D}'$, we know that $\mathcal{D}' \models p_1'$. Therefore, keeping in mind that p_1' contains all atoms of p_1, each element a_x with $\mathsf{Hor}(a_x, a)$ in $\mathcal{D}'$ for some a must also have the required atoms in $\mathcal{D}'$.

Next we show the reverse direction. Again, since $\mathcal{D} \models p_1'$, $\mathcal{M}_1' \cup \mathcal{D} \models \mathsf{ConfCh}^*()$. Therefore, each indistinguishable $\mathcal{D}'$ contains atoms $\mathsf{Hor}(a_x, a)$ and $\mathsf{ConfCh}(a_x)$ for some a_x. Since it also contains the required atoms ($\mathsf{GadgetConnect}_s$ atoms and the confluence-related atoms) for this a_x, we have that $\mathcal{D}' \models p_1'$, as required. □

Next, we use these additions to guarantee that every a_x with $\mathsf{Hor}(a_x, a)$ is indeed assigned with a type. Let $\mathcal{M}'_3$ be

$$\begin{array}{rcl} \mathsf{Hor}(x,y) \wedge \mathsf{GadgetConnect}_t(x,z) \wedge \mathsf{GadgetConnect}_t(x',z) & \rightarrow & \mathsf{STileCh}'(x',z), \text{ for all } t \in \mathcal{T}, \\ \mathsf{Hor}(x,y) \wedge \mathsf{STileCh}(x,z) & \rightarrow & \mathsf{STileCh}^*(), \end{array}$$

and let p'_3 be the Boolean CQ

$$\bigwedge_{s \in \mathcal{T}} \big(\mathsf{GadgetConnect}_s(x_s, x) \wedge \mathsf{STileCh}(x_s, x) \wedge \mathsf{Tiled}_s(x_s)\big).$$

Claim 26. *Given a source instance $\mathcal{D}$ such that $\mathcal{D} \models p'_1 \wedge p_2 \wedge p'_3$, $\mathsf{Comply}(\emptyset, \mathcal{M}'_1 \cup \mathcal{M}_2 \cup \mathcal{M}'_3, \mathcal{D}, p'_1 \wedge p_2 \wedge p'_3) = \texttt{false}$ if and only if*

- *the condition in Claim 25 and the second condition in Claim 22 hold;*
- *for every a_x such that $\mathsf{Hor}(a_x, a) \in \mathcal{D}$ for some a there is $t \in \mathcal{T}$ with $\mathsf{Tiled}_t(a_x) \in \mathcal{D}$.*

Proof. In one direction, suppose $\mathsf{Comply}(\emptyset, \mathcal{M}'_1 \cup \mathcal{M}_2 \cup M'_3, \mathcal{D}, p'_1 \wedge p_2 \wedge p'_3)$ is `false`. The first item holds as before, so we focus on the second item. Given $\mathsf{Hor}(a_x, a) \in \mathcal{D}$ we modify $\mathcal{D}$ to $\mathcal{D}'$ as follows:

1. we remove from $\mathsf{STileCh}$ any pair $(a_{x'}, a_z)$ such that
$$\bigvee_{t \in \mathcal{T}} \exists x. \exists y. \mathsf{Hor}(x,y) \wedge \mathsf{GadgetConnect}_t(x, a_z) \wedge \mathsf{GadgetConnect}_t(a_{x'}, a_z)$$
holds in $\mathcal{D}$;
2. we add to $\mathsf{STileCh}$ all the pairs (a_x, a_z), for some a_z, such that $\mathsf{GadgetConnect}_s(a_x, a_z)$ holds in $\mathcal{D}$ for $s \in \mathcal{T}$;
3. we remove from $\mathsf{STileCh}$ any pair $(a_{x'}, a_z)$ not added in the previous step, such that
$$\bigwedge_{t,s \in \mathcal{T}, t \neq s} \neg \exists x. \exists y. \mathsf{Hor}(x,y) \wedge \mathsf{GadgetConnect}_t(x, a_z) \wedge \mathsf{GadgetConnect}_s(a_{x'}, a_z)$$
holds in $\mathcal{D}$.

We first claim that $\mathcal{D}'$ is indistinguishable from $\mathcal{D}$. The mapping $\mathsf{Hor}(x,y) \wedge \mathsf{STileCh}(x,z) \rightarrow \mathsf{STileCh}^*()$ is clearly not impacted by any of the changes above, since the second item ensures that this still fires. We may worry about the impact of all the changes above on the third family of mappings in $\mathcal{M}'_1$:

$$\mathsf{Hor}(x,y) \wedge \mathsf{GadgetConnect}_t(x,z) \wedge \mathsf{GadgetConnect}_s(u,z) \wedge \mathsf{STileCh}(u,z) \rightarrow \mathsf{STileCh}'(u,z).$$

However, the removal in the first item does not have any impact, since although we are removing elements where this mapping fires, all these elements also trigger one of the mappings in the first family of $\mathcal{M}'_3$

$$\mathsf{Hor}(x,y) \wedge \mathsf{GadgetConnect}_t(x,z) \wedge \mathsf{GadgetConnect}_t(x',z) \rightarrow \mathsf{STileCh}'(x',z),$$

so the exported information is the same. Similarly, the addition in the second item does not have any impact, since $\mathsf{STileCh}'(a_x, a_z)$ was already exported due to the same mappings in $\mathcal{M}'_3$. Finally, the removal in the third item does not have any impact, since by definition we are removing elements where the mapping did not fire.

Hence $\mathcal{D}'$ satisfies p'_3 by the assumption on $\mathcal{D}$. Thus there are (not necessary different) a_c and a_s, $s \in \mathcal{T}$, such that

$$\bigwedge_{s \in \mathcal{T}} \big(\mathsf{GadgetConnect}_s(a_s, a_c) \wedge \mathsf{STileCh}(a_s, a_c) \wedge \mathsf{Tiled}_s(a_s)\big)$$

holds in $\mathcal{D}'$. In particular all $\mathsf{Tiled}_s(a_s)$ hold in $\mathcal{D}$ as well, since $\mathcal{D}'$ agrees with $\mathcal{D}$ on the relations Tiled_s. We claim that one of these a_s must be a_x, which would suffice to prove the forward direction of the claim. We first note that no a_s other than a_x can be in the first position of Hor. This is because any other element in that position would have been removed from $\mathsf{STileCh}$ in the first removal step above, and would not have been replaced. Thus it suffices to argue that there is an a_s that is in the first position of Hor. Now for each a_s other than a_x there are $a_{x'}$ and $a_{y'}$ such that $\mathsf{Hor}(a_{x'}, a_{y'}) \wedge \mathsf{GadgetConnect}_{t'}(a_{x'}, a_c) \wedge \mathsf{GadgetConnect}_{s'}(a_s, a_c)$ holds in $\mathcal{D}'$ for some $s', t' \in \mathcal{T}$, since otherwise the pair (a_s, a_c) would have been removed at the third step. Take one of these elements a_s different from a_x (if all of them equal a_x, then the claim is automatic), and consider the corresponding t' and $a_{t'}$. If $a_{t'}$ is not a_x, then we would have removed $\mathsf{STileCh}(a_{t'}, a_c)$ in $\mathcal{D}'$ in the first removal step above, which is a contradiction. Hence we have $a_{t'}$ is a_x, and we are done.

In the other direction, suppose $\mathcal{D}$ satisfies the properties above and consider any $\mathcal{D}'$ indistinguishable from $\mathcal{D}$. The fact that $\mathcal{D} \models p'_1 \wedge p_2 \wedge p'_3$ implies that $\mathsf{STileCh}^*()$ is exported from $\mathcal{D}$, and hence must be exported from $\mathcal{D}'$. Thus there are $a_x, a_y, a_z \in \mathcal{D}'$ satisfying $\mathsf{Hor}(a_x, a_y) \wedge \mathsf{STileCh}(a_x, a_z)$. Further since in the other mappings Hor is exported, we must have $\mathsf{Hor}(a_x, a_y)$ holding in $\mathcal{D}$ as well. By the second item $\mathsf{Tiled}_t(a_x)$ holds in $\mathcal{D}$ for some $t \in \mathcal{T}$. Since other mappings export Tiled_t, we know that $\mathsf{Tiled}_t(a_x)$ holds in $\mathcal{D}'$, which guarantees that p'_3 holds in $\mathcal{D}'$ as required. □

Let $\mathcal{M}' = \mathcal{M}'_1 \cup \mathcal{M}_2 \cup \mathcal{M}'_3 \cup \mathcal{M}_4$ and $p' = p'_1 \wedge p_2 \wedge p'_3 \wedge p_4$.

We conclude that there exists a source database $\mathcal{D}$ with $\mathsf{Comply}(\emptyset, \mathcal{M}', \mathcal{D}, p') = \texttt{false}$ if and only if it is possible to tile an infinite plane with the tiling instance, that is, $\mathsf{ComplyAll}(\emptyset, \mathcal{M}', p') = \texttt{true}$ is undecidable.

Finally, we note that the argument above allows witnesses to non-compliance to be infinite. But the same construction works for finite instances, reducing the existence of a periodic tiling (also known to be undecidable) to the complement of ComplyAll. □

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.*

Proof. To prove this, by the prior undecidability results it suffices to simulate an arbitrary set of GAV mapping without an ontology by CQ view mappings with an ontology. Given a GAV mapping in which global relation G is associated with bodies $\varphi_1 \ldots \varphi_n$, we create a new scenario in which G is replaced by global relations $G_1 \ldots G_n$ of the same arity as G, and the ontology consinsts of the mappings

$$G_i(\mathbf{x}) \rightarrow G_j(\mathbf{x})$$

for all $i, j = 1, \ldots, n$, $i \neq j$. □

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* $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathsf{Crit}_{\mathbf{R}}, p) = \texttt{true}$ *if and only if* $\mathsf{ComplyAll}(\emptyset, \mathcal{M}, p) = \texttt{true}$.

Proof. We start with an alternative characterization of the Comply problem in the case where the mappings are CQ views and the ontology is empty. We show that we can always take a witness to Comply to be of a special form. This will be useful in dealing with the ComplyAll problem later. Given a GAV mapping of the form

$$\varphi(\mathbf{x}, \mathbf{y}) \rightarrow A(\mathbf{x}),$$

the *inverse mapping* of this mapping is the TGD

$$A(\mathbf{x}) \rightarrow \exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y}).$$

Given $\mathcal{M}$ a set of CQ views, we let $\mathcal{M}^{-1}$ be the set of inverses. Note that an instance $\mathcal{F}$ over both source and global schemas satisfies $\mathcal{M} \cup \mathcal{M}^{-1}$ if and only if the global relations in $\mathcal{F}$ are exactly what one gets by applying $\mathcal{M}$ to the source relations. If $\mathcal{TI}$ is an instance over the global schema, we say that $\mathcal{F}$ is a *realizer* for $(\mathcal{TI}, \mathcal{M})$ if applying $\mathcal{M}$ to the source relations in $\mathcal{F}$ gives $\mathcal{TI}$.

We use an idea from (Benedikt et al. 2016): a version of the "classical" chase procedure (Abiteboul, Hull, and Vianu 1995) tailored for the variant of the certain answer problem with closed relations. This version returns a collection of instances, along the lines of the "disjunctive chase" of (Deutsch, Nash, and Remmel 2008). The procedure receives as input a set of CQ views $\mathcal{M}$, and an initial instance $\mathcal{D}$ for the source schema. The procedure first produces the corresponding global instance $\mathcal{TI}_0$. In then chases with the mappings and their inverses starting from the instance $\mathcal{TI}_0$, guaranteeing at the same time that the global relations of the constructed instances agree with those of $\mathcal{TI}_0$.

In the first step (after producing $\mathcal{TI}_0$), we chase $\mathcal{TI}_0$ with $\mathcal{M}^{-1}$ to get a new source instance $\mathcal{D}'$. In the second step we look at each trigger for a mapping in $\mathcal{M}$, and *choose an existing witness in* $\mathcal{TI}_0$ *to satisfy it*. If this requires identifying nulls (values produced in the first step) within the source instance, we do so. If it requires identifying constants in $\mathcal{TI}_0$, then the chase fails. As a result of the identifications in this second step, new triggers can fire. We iterate the second step until no new triggers fire. Since no constants or nulls are created after the first step, the process must terminate. This process is well-defined only once the ordering of steps is chosen, but for the results below which order is chosen will not matter, so we abuse notation by referring to $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D})$ as a single collection.

It is clear that every instance in $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D})$ satisfies the constraints in $\mathcal{M} \cup \mathcal{M}^{-1}$ and agrees with $\mathcal{TI}_0$ on the target schema. In particular, the restriction of this instance to the source relations is indistinguishable from $\mathcal{D}$. We claim that $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D})$ satisfies the following universality property.

Claim 27. *Let* $\mathcal{M}$ *consist of CQ views and* $\mathcal{D}$ *be a source instance. For any instance* $\mathcal{D}''$ *indistinguishable from* $\mathcal{D}$ *with respect to* $\mathcal{M}$*, there exist an instance* $\mathcal{K} \in \mathsf{ClosedChase}(\mathcal{M}, \mathcal{D})$ *and a homomorphism* h *from* $\mathcal{K}$ *to* $\mathcal{D}'' \cup \mathcal{TI}_0$.

The proof is a straightforward generalization of earlier universality results (and in particular follows easily from (Deutsch, Nash, and Remmel 2008)). But we include it for completeness.

Proof. We consider the sequence of instances created in each trigger of the process of building for $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D})$ and, based on the instance $\mathcal{D}'' \cup \mathcal{TI}_0$, we identify inside this sequence a suitable path $\mathcal{K}_0, \mathcal{K}_1, \ldots$ and a corresponding sequence of homomorphisms $h_0, h_1, \ldots$ such that, for all $i \in \mathbb{N}$, h_i maps $\mathcal{K}_i$ to $\mathcal{D}'' \cup \mathcal{TI}_0$.

The base step is easy, as we simply let $\mathcal{K}_0$ be the initial global instance $\mathcal{TI}_0$, and let h_0 be the identity. In the first step we chased with the inverse mappings. Since $\mathcal{D}'' \cup \mathcal{TI}_0$ also satisfied the inverse mappings, we can extend the homomorphism in this step.

In the remaining steps, we consider mappings $R_1(\mathbf{x}_1, \mathbf{z}_1) \wedge \ldots \wedge R_m(\mathbf{x}_m, \mathbf{z}_m) \rightarrow \exists \mathbf{y}.S(\mathbf{x}, \mathbf{y})$, with $\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_m$, from $\mathcal{M}$ which have a trigger τ in $\mathcal{K}_i$. Since h_i is a homomorphism from K_i into $\mathcal{D}'' \cup \mathcal{TI}_0$ we know that $R_1(h(\tau(\mathbf{x}_1, \mathbf{z}_1))) \wedge \ldots \wedge R_m(h(\tau(\mathbf{x}_m, \mathbf{z}_m)))$ holds in $\mathcal{D}''$. Note that $\mathcal{D}'' \cup \mathcal{TI}_0$ satisfies the same mappings by assumption, so $\mathcal{TI}_0$ must contain a fact of the form $S\big(h(\tau(\mathbf{x}, \mathbf{y}))\big)$. We extend K_i by identifying, for any frontier variable x of the mapping, $\tau(x)$ with $h(\tau(x))$. One can see that this is one of the valid choices available. We can then revise h to be the identity on the identified nodes. □

From Claim 27 we obtain the following fact.

Claim 28. *Let $(\emptyset, \mathcal{M}, \mathcal{D}, p)$ be an input to* Comply *where p is Boolean and $\mathcal{M}$ consists of CQ views. Then* $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{true}$ *if and only if $\neg p$ holds on some instance in* $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D})$.

Proof. Clearly the left-to-right direction holds. Suppose that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) = \texttt{true}$ with instance $\mathcal{D}'$ as a witness. Let $\mathcal{TI}_0$ be formed by chasing $\mathcal{D}$ with $\mathcal{M}$. Let $\mathcal{F}'$ be formed from $\mathcal{TI}_0$ by interpreting the source relations as in $\mathcal{D}'$. From Lemma 27 we know that there is a homomorphism of some instance $\mathcal{K} \in \mathsf{ClosedChase}(\mathcal{M}, \mathcal{D})$ to $\mathcal{F}'$. Thus $\mathcal{K}$ cannot satisfy p, which completes the argument. □

We are now ready to prove the theorem.

The "only if" direction is trivial, so we focus on the "if" direction. Assume that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}_1, p) = \mathsf{true}$, where $\mathcal{D}_1 = \mathsf{Crit}_{\mathbf{R}}$. By Claim 28 we must have $\neg p$ holding on some instance within $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_1)$. Note that because $\mathcal{M}$ is GAV, the corresponding virtual instance is a subinstance of the critical instance over the global schema. Since the choices we make in the non-deterministic process of $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_1)$ are only choices as to which global instance domain element to choose, and there is only one global instance domain element, we conclude that *there is only one instance* $\mathcal{K}_1$ in $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_1)$.

Let $\mathcal{D}_2$ be an arbitrary source instance. We show that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}_2, p)$ is $\texttt{true}$. If not, then in particular we must have p holding on all instances within $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_2)$. Although some branches of ClosedChase can fail, there must always be at least one instance $\mathcal{K}_2$ in $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_2)$. This follows from Claim 27, since we know there is some instance indistinguishable from $\mathcal{D}_2$, and the claim says that any such instance there must be an element of $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_2)$ that maps homomorphically into it. In particular, we have a $\mathcal{K}_2$ in $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_2)$ that satisfies p.

We claim that there is a homomorphism from $\mathcal{K}_2$ to some instance in $\mathsf{ClosedChase}(\mathcal{M}, \mathcal{D}_1)$ (which can be none other than $\mathcal{K}_1$ as noted above). This suffices to get a contradiction.

The homomorphism h will map every element of $\mathcal{K}_2$ that occurs in a global relation to the critical element. For each element e occurring in a source relation but not in a global relation, there must be an inverse mapping m^{-1} that generated the element e as a null, based on some trigger τ mapping to facts on a global relation. Collapsing those facts to facts on the global relation of $\mathcal{K}_1$ (by mapping each element to the critical element) we get a trigger τ' for m^{-1} in $\mathcal{K}_1$, and we take the corresponding element e' as $h(e)$. □

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.*

Proof. In the proof we deal with p Boolean for simplicity.

By Theorem 16, the problem is equivalent to $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathsf{Crit}_{\mathbf{R}}, p)$, where $\mathbf{R}$ is the source schema. Recall also from the proof of Theorem 16 that to check this we need to see that the unique instance in $\mathsf{ClosedChase}(\mathcal{M}, \mathsf{Crit}_{\mathbf{R}})$ satisfies $\neg p$. Thus, to determine that $\neg\mathsf{ComplyAll}$ holds we need to form the instance in $\mathsf{ClosedChase}(\mathcal{M}, \mathsf{Crit}_{\mathbf{R}})$ and then guess a homomorphism of p in it. Clearly this can be done in NP, giving the desired coNP bound.

Conversely, we prove coNP-hardness by reducing conjunctive query containment, known to be NP-hard, to the complement of ComplyAll. We employ a variation of an argument in (Benedikt et al. 2016) to do this. Given CQs q_1 and q_2, we create a source schema containing all relations of q_1 and q_2. The global schema contains one 0-ary relational name good, and the associated mapping $\mathcal{M}$ simply states

$$q_1 \rightarrow \mathsf{good}().$$

The policy query p is q_2. We claim that q_1 is contained in q_2 exactly when $\neg\mathsf{ComplyAll}(\emptyset, \mathcal{M}, p)$ holds.

First, suppose there is a non-compliant instance $\mathcal{D}$. We claim q_1 must hold on $\mathcal{D}$. If not, then $\mathcal{D}$ would export $\neg\mathsf{good}()$, and the empty instance would then be an indistinguishable from $\mathcal{D}$, and clearly satisfies $\neg p$, contradicting the assumption that $\mathcal{D}$ is non-compliant. Non-compliance of $\mathcal{D}$ implies that every $\mathcal{D}'$ having q_1 $\texttt{true}$ also has q_2 $\texttt{true}$, and thus containment holds.

Conversely, suppose q_1 is contained in q_2. Take any instance satisfying q_1 as $\mathcal{D}$. Then, $\mathsf{good}()$ will hold of its image, and every $\mathcal{D}'$ indistinguishable to $\mathcal{D}$ must then satisfy q_1, and hence q_2.

Finally, we consider the linear case. As mentioned above, to decide ComplyAll we need only form the instance $\mathcal{D}' = \mathsf{ClosedChase}(\mathsf{Crit}_{\mathbf{R}})$, in P, and then evaluate p on it. In evaluating p, we can leverage that any joined variable of p can hold in $\mathcal{D}'$ only of the domain element of $\mathsf{Crit}_{\mathbf{R}}$, and thus after substituting this we need only check the individual atoms separately. □

Appendix D: Proofs of Results in Section 7

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.*

Proof. For brevity, we only show how to adapt the Σ_2^p-hardness argument from statement 3 of Theorem 12 to ComplyBoth; the adaptation of the NEXPTIME-hardness from statement 1 of the same theorem is similar.

We give a reduction of $\forall\exists$SAT to the complement of ComplyBoth. Given $\varphi = \forall\mathbf{u}.\exists\mathbf{v}.\psi$, where ψ is a conjunction of clauses of the form $\ell_1 \vee \ell_2 \vee \ell_3$ for ℓ_i either a variable from $\mathbf{u} \cup \mathbf{v}$ or the negation of such a variable. Without loss of generality we assume that ψ is satisfiable as a formula in CNF.

We construct an instance $(\emptyset, \mathcal{M}, \mathcal{D}, p)$ of ComplyBoth, where $\mathcal{M}$ consists of linear CQ views, such that the following property holds:

$$\varphi \text{ is valid if and only if either } \mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p) \text{ is } \texttt{false} \text{ or } \mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, \neg p) \text{ is } \texttt{false}.$$

The construction of $\mathcal{M}$, $\mathcal{D}$, and p is exactly the same as that of statement 3 of Theorem 12. We show that the reduction works. Assume that formula φ is valid; then, in the aforementioned proof of Theorem 12 we already established that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ is `false`, and hence property above holds.

Conversely, assume that property holds. Then, either $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ is `false` or $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, \neg p)$ is `false`. By our construction, and the assumption that ψ is satisfiable as a formula in CNF, we have that p holds in $\mathcal{D}$. This implies that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, \neg p)$ must be `true`; indeed, there exists $\mathcal{D}'$ (namely, $\mathcal{D}$ itself) such that $\mathcal{D}'$ is indistinguishable from $\mathcal{D}$ and $\mathcal{D}' \not\models \neg p$ (and hence such that $\mathcal{D}' \models p$ under closed world assumption). As a result, the fact that the property holds implies that $\mathsf{Comply}(\emptyset, \mathcal{M}, \mathcal{D}, p)$ must be `false`. But then, in such a case, we already proved in Theorem 12 that φ is valid, as required. □