Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Controlled Query Evaluation for Datalog and OWL 2 Profile Ontologies

Bernardo Cuenca Grau, Evgeny Kharlamov, Egor V. Kostylev, Dmitriy Zheleznyakov
Department of Computer Science, University of Oxford, UK
{*f_name.l_name*}@cs.ox.ac.uk

Abstract

We study confidentiality enforcement in ontologies under the Controlled Query Evaluation framework, where a policy specifies the sensitive information and a censor ensures that query answers that may compromise the policy are not returned. We focus on censors that ensure confidentiality while maximising information access, and consider both Datalog and the OWL 2 profiles as ontology languages.

1 Introduction

As semantic technologies are becoming increasingly mature, there is a need for mechanisms to ensure that confidential data is only accessible by authorised users.

Controlled Query Evaluation (CQE) is a prominent confidentiality enforcement framework, in which sensitive information is declaratively specified by means of a *policy* and confidentiality is enforced by a *censor*. When given a query, the censor checks whether returning the correct answer may lead to a policy violation, in which case it returns a distorted answer. The CQE framework was introduced in [Sicherman *et al.*, 1983], and studied in [Biskup and Bonatti, 2001; 2004; Bonatti *et al.*, 1995; Biskup and Weibert, 2008] for propositional databases. It has been recently extended to ontologies, where different formalisations have been proposed [Bonatti and Sauro, 2013; Cuenca Grau *et al.*, 2013; Studer and Werner, 2014].

We study CQE for ontologies expressed in the rule language Datalog as well as in the lightweight description logics (DLs) underpinning the profiles of OWL 2 [Motik *et al.*, 2012]. We assume that data is hidden and that users access the system by means of a query interface. An ontology, which is known to users, provides the vocabulary and background knowledge needed for users to formulate queries, as well as to enrich query answers with implicit information. Policies, formalised as conjunctive queries, are available to system administrators, but not to ordinary users. The role of the censor is to preserve confidentiality by filtering out those answers to user queries that could lead to a policy violation.

In this setting, there is a danger that confidentiality enforcement may over-restrict the access of the user. Thus, we focus on *optimal* censors, which maximise answers to queries while ensuring confidentiality of the policy. We are especially interested in censors that can be realised by off-the-shelf reasoning infrastructure. To fullfil this requirement, we introduce in Section 4 *view* and *obstruction* censors.

View censors return only answers that follow from the ontology and an anonymised dataset (a view) where some occurrences of constants may have been replaced with labelled nulls. The censor answers faithfully all queries against the view; thus, any information not captured by the view is inaccessible by default. View censors may require materialisation of implicit data, and hence are well-suited for applications where materialisation is feasible.

Obstruction censors are defined by a set of "forbidden query patterns" (an obstruction), where all answers instantiating such patterns are not returned to users. These censors do not require data modification and are well-suited for applications such as Ontology Based Data Access (OBDA), where data is managed by an RDBMS. Obstruction censors are dual to view censors in the sense that they specify the information that users are denied access to. We formally characterise this duality, and show that their capabilities are incomparable.

In Section 5 we investigate the limitations of view censors and show that checking existence of a view realising an optimal censor is undecidable for Datalog ontologies. We then study fragments of Datalog for which such views always exist and extend our results to OWL 2 profile ontologies.

In Section 6 we focus on obstruction censors, and provide sufficient and necessary conditions for an optimal censor based on an obstruction to exist. Then, we propose a polynomial time algorithm for computing such obstructions realising optimal views for linear Datalog ontologies, and apply our results to OWL 2 QL ontologies.

Compete proofs of all our results are delegated to an extended version (see [Cuenca Grau *et al.*, 2015]).

2 Preliminaries

We adopt standard notions in first order logic over function-free finite signatures. Our focus is on ontologies, so we assume signatures with constants $a, \ldots$, unary predicates $A, \ldots$, and binary predicates $R, \ldots$. We treat equality $\approx$ as an ordinary predicate, but assume that any set of formulae containing $\approx$ also contains all the axioms of $\approx$ for its signature.

Datasets and Ontologies A *dataset* is a finite set of facts (i.e., ground atoms). An *ontology* is a finite set of *rules*, that

(1) $A(x) \wedge R(x,y_1) \wedge B(y_1) \wedge R(x,y_2) \wedge B(y_2) \rightarrow y_1 \approx y_2$,
(2) $R(x,y) \rightarrow S(x,y)$, (3) $A(x) \rightarrow \exists y.[R(x,y) \wedge B(y)]$,
(4) $A(x) \rightarrow x \approx a$, (5) $R(x,y) \wedge S(y,z) \rightarrow T(x,z)$,
(6) $A(x) \wedge B(x) \rightarrow C(x)$, (7) $A(x) \wedge R(x,y) \rightarrow B(y)$,
(8) $R(x,y) \rightarrow S(y,x)$, (9) $R(x,a) \rightarrow B(x)$,
(10) $R(x,y) \rightarrow A(y)$, (11) $A(x) \rightarrow R(x,a)$,
(12) $A(x) \rightarrow B(x)$, (13) $R(x,y) \wedge B(y) \rightarrow A(x)$.

Table 1: OWL 2 profile axioms as rules

is, universally quantified sentences of the form

$$\varphi(\vec{x}) \rightarrow \exists \vec{y}.\psi(\vec{x},\vec{y}),$$

where the *body* $\varphi(\vec{x})$ and the *head* $\psi(\vec{x},\vec{y})$ are conjunctions of atoms, and variables $\vec{x}$ are implicitly universally quantified. We restrict ourselves to ontologies $\mathcal{O}$ and datasets $\mathcal{D}$ such that $\mathcal{O} \cup \mathcal{D}$ is satisfiable, which ensures that answers to queries are meaningful. A rule is

– *Datalog* if the head has a single atom and $\vec{y}$ is empty;
– *guarded* if the body has an atom (*guard*) with all $\vec{x}$;
– *linear* if the body has a single atom;
– *multi-linear* if the body contains only guards; and
– *tree-shaped* if the undirected multigraph with an edge $\{t_1, t_2\}$ for each binary body atom $R(t_1, t_2)$ is a tree.

An ontology is of a type above if so are all the rules in it.

OWL 2 Profiles Table 1 provides the types of rules sufficient to capture the axioms in the OWL 2 RL, EL, and QL profiles. We treat the $\top$ concept in DLs as a unary predicate and assume that each ontology contains the rule $S(\vec{x}) \rightarrow \top(x)$ for each predicate S and variable x from $\vec{x}$. An ontology consisting of rules in Table 1 is

– *RL* if it has no rules of type (3);
– *QL* if it only has rules of types (2), (3), (8), (10), and (12);
– *EL* if it has no rules of types (1), (7), (8).

Queries A *conjunctive query* (*CQ*) with *free* variables $\vec{x}$ is a formula of the form $\exists \vec{y}.\varphi(\vec{x},\vec{y})$, with the *body* $\varphi(\vec{x},\vec{y})$ a conjunction of atoms. A *union* of CQs (*UCQ*) is disjunction of CQs with same free variables. Queries with no free variables are *Boolean*. A tuple of constants $\vec{a}$ is a (*certain*) *answer* to a (U)CQ $Q(\vec{x})$ over ontology $\mathcal{O}$ and dataset $\mathcal{D}$ if $\mathcal{O} \cup \mathcal{D} \models Q(\vec{a})$. The set of answers to $Q(\vec{x})$ over $\mathcal{O}$ and $\mathcal{D}$ is denoted by $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$.

3 Basic Framework

We assume that data $\mathcal{D}$ is hidden while the ontology $\mathcal{O}$ is known to all users. It is assumed that system administrators are in charge of specifying policies (i.e., sensitive information) as CQs, and that policies are assigned to (groups of) users by standard mechanisms such as role-based access control [Sandhu *et al.*, 1996]. To simplify the exposition, we assume that all users are assigned with the same policy; and the lifting to the general case is straightforward.

Definition 1. *A* CQE instance $\mathbf{I}$ *is a triple* $(\mathcal{O}, \mathcal{D}, P)$*, with* $\mathcal{O}$ *an ontology,* $\mathcal{D}$ *a dataset, and* P *a CQ, which is called* policy*. The instance* $\mathbf{I}$ *is Datalog, guarded, etc. if so is the ontology* $\mathcal{O} \cup \{\varphi(\vec{x},\vec{y}) \rightarrow A_p(\vec{x})\}$*, where* $\varphi(\vec{x},\vec{y})$ *is the body of* P *and* A_p *a fresh predicate.*

Example 2. Consider the following ontology and dataset that describe an excerpt of a social network:

$$\mathcal{O}_{\mathsf{ex}} = \{ \; \mathit{Likes}(x,y) \wedge \mathit{Thriller}(y) \rightarrow \mathit{ThrillerFan}(x), \; \mathit{Suspense}(x) \wedge \mathit{Crime}(x) \rightarrow \mathit{Thriller}(x), \; \mathit{FrOf}(x,y) \rightarrow \mathit{FrOf}(y,x) \; \},$$

$$\mathcal{D}_{\mathsf{ex}} = \{ \; \mathit{FrOf}(\mathsf{John},\mathsf{Bob}), \mathit{FrOf}(\mathsf{Bob},\mathsf{Mary}), \; \mathit{Crime}(\mathsf{Seven}), \mathit{Suspense}(\mathsf{Seven}), \; \mathit{Likes}(\mathsf{John},\mathsf{Seven}), \mathit{Likes}(\mathsf{Bob},\mathsf{Seven}) \; \}.$$

Here, the ontology $\mathcal{O}_{\mathsf{ex}}$ states, for example, that people who like thrillers are thriller fans, or that friendship is a symmetric relation; the dataset $\mathcal{D}_{\mathsf{ex}}$ states, for example, that Bob is John's friend. Then, a policy $P_{\mathsf{ex}} = \mathit{FrOf}(\mathsf{Bob}, x)$ forbids access to Bob's friend list. $\Diamond$

A key component of a CQE system is the *censor*, whose goal is to decide according to the policy which query answers can be safely returned to users.

Definition 3. *A* censor *for a CQE instance* $(\mathcal{O}, \mathcal{D}, P)$ *is a function* cens *mapping each CQ* Q *to a subset of* $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$*. The* theory $\mathsf{Th}_{\mathsf{cens}}$ *of* cens *is the set*

$$\{Q(\vec{a}) \mid \vec{a} \in \mathsf{cens}(Q) \text{ and } Q(\vec{x}) \text{ is a CQ}\}.$$

Censor cens *is* confidentiality preserving *if for any tuple of constants* $\vec{a}$ *it holds that* $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \not\models P(\vec{a})$*. It is* optimal *if*

– *it is confidentiality preserving, and*
– *no confidentiality preserving censor* $\mathsf{cens}' \neq \mathsf{cens}$ *exists such that* $\mathsf{cens}(Q) \subseteq \mathsf{cens}'(Q)$ *for every CQ* Q*.*

Intuitively, the theory $\mathsf{Th}_{\mathsf{cens}}$ represents all the information that a user can gather by asking CQs to the system. If the censor is confidentiality preserving, then no information can be obtained about the policy, regardless of the number of CQs asked. In this way, optimal censors maximise information accessibility without compromising the policy.

4 View and Obstruction Censors

As already mentioned, we are interested in censors implementable by off-the-shelf tools. In this section we discuss *view* and *obstruction* censors, which satisfy this requirement.

The idea behind *view censors*, is as follows. First, the dataset is modified by anonymising occurrences of constants as well as by adding or removing facts, whenever needed. Such modified dataset constitutes an *(anonymisation) view*. Then, the view censor returns only the answers that follow from the ontology and view; in this way, the main workload of the censor amounts to the computation of certain answers, which can be delegated to a query answering engine.

Definition 4. *A* view $\mathcal{V}$ *for a CQE instance* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ *is a dataset over the signature of* $\mathbf{I}$ *extended with a set of fresh constants. The* view censor $\mathsf{vcens}^{\mathcal{V}}_{\mathbf{I}}$ *based on* $\mathcal{V}$ *is the censor mapping each CQ* $Q(\vec{x})$ *to* $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) \cap \mathsf{cert}(Q, \mathcal{O}, \mathcal{V})$*. The view is* optimal *if so is its corresponding censor.*

Clearly, for a view censor to be confidentiality preserving $\mathcal{O} \cup \mathcal{V}$ must not entail any answer to the policy. On the other hand, to ensure optimality a view must encode as much information from the hidden dataset as possible.

Example 5. Consider the view $\mathcal{V}_{\mathsf{ex}}$ obtained from $\mathcal{D}_{\mathsf{ex}}$ in Example 2 by replacing Bob with a fresh an_b. Intuitively, $\mathcal{V}_{\mathsf{ex}}$ is the result of "anonymising" the constant Bob, while keeping the structure of the data intact. Since $\mathcal{V}_{\mathsf{ex}}$ contains no information about Bob, we have $\mathsf{cert}(P_{\mathsf{ex}}, \mathcal{O}_{\mathsf{ex}}, \mathcal{V}_{\mathsf{ex}}) = \emptyset$ and the censor based on $\mathcal{V}_{\mathsf{ex}}$ is confidentiality preserving. View $\mathcal{V}_{\mathsf{ex}}$, however, is not optimal: for instance, $\mathcal{O}_{\mathsf{ex}} \cup \mathcal{V}_{\mathsf{ex}}$ does not entail the fact $Likes(\mathsf{Bob}, \mathsf{Seven})$, which can be added to the view without violating confidentiality. ◊

The idea behind *obstruction censors* is to associate to a CQE instance a Boolean UCQ U such that the censor returns an answer $\vec{a}$ to a CQ $Q(\vec{x})$ only if no CQ in U follows from $Q(\vec{a})$. Thus, the obstruction can be seen as a set of forbidden query patterns, which should not be disclosed.

Definition 6. *An* obstruction *U for a CQE instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ is a Boolean UCQ. The* obstruction censor $\mathsf{ocens}^U_{\mathbf{I}}$ *based on U is the censor that maps each CQ $Q(\vec{x})$ to the set*

$$\{\vec{a} \mid \vec{a} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) \text{ and } Q(\vec{a}) \not\models U\}.$$

The obstruction is optimal *if so is its censor* $\mathsf{ocens}^U_{\mathbf{I}}$.

Similarly to view censors, obstruction censors do not require dedicated algorithms: checking whether $Q(\vec{a}) \models U$ can be delegated to an RDBMS. Obstructions can be virtually maintained and do not require data materialisation.

Example 7. The censor based on $\mathcal{V}_{\mathsf{ex}}$ from Example 5 can also be realised with the following obstruction U_{ex}:

$$\exists x.FrOf(x, \mathsf{Bob}) \vee \exists x.FrOf(\mathsf{Bob}, x) \vee$$
$$\exists x.Likes(\mathsf{Bob}, x) \vee ThrillerFan(\mathsf{Bob}).$$

Intuitively, U_{ex} "blocks" query answers involving Bob; and all other answers are the same as over $\mathcal{O}_{\mathsf{ex}} \cup \mathcal{D}_{\mathsf{ex}}$. ◊

Examples 5 and 7 show that the same censor may be based on both a view and an obstruction. These censors, however, behave *dually*: a view explicitly encodes the information accessible to users, whereas obstructions specify information which users are denied access to. It is not obvious whether (and how) a view can be realised by an obstruction, or vice-versa. We next focus on Datalog ontologies and characterise when a view $\mathcal{V}$ and obstruction U yield the same censor. We start with few definitions.

Each Datalog ontology $\mathcal{O}$ and dataset $\mathcal{D}$ have a unique *least Herbrand model* $\mathcal{H}_{\mathcal{O},\mathcal{D}}$ that is the finite structure satisfying $\vec{a} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ if and only if $\mathcal{H}_{\mathcal{O},\mathcal{D}} \models Q(\vec{a})$ for every CQ Q. Thus, this model captures all the information relevant to CQ answering. A natural specification of the duality between views and obstructions is then as follows: U and $\mathcal{V}$ implement the same censor if and only if U captures the structures *not* homomorphically embeddable into $\mathcal{H}_{\mathcal{O},\mathcal{V}}$. To formalise this statement, we recall the central problem in the (non-uniform) constraint satisfaction theory.

Definition 8 (Kolaitis and Vardi, 2008). *Let $\mathbb{C}$ be a class of finite structures and let $\mathbb{C}'$ be a subset of $\mathbb{C}$. A first-order sentence ψ* defines *$\mathbb{C}'$ in $\mathbb{C}$ if $\mathcal{I} \in \mathbb{C}'$ is equivalent to $\mathcal{I} \models \psi$ for every structure $\mathcal{I} \in \mathbb{C}$.*

Let $\mathcal{J} \hookrightarrow \mathcal{J}'$ denote the fact that there is a homomorphism from a structure $\mathcal{J}$ to a structure $\mathcal{J}'$. The correspondence is given in the following theorem.

Theorem 9. *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ be a Datalog CQE instance and let $\mathbb{C}$ consist of all finite $\mathcal{I}$ such that $\mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{D}}$. Then, $\mathsf{vcens}^{\mathcal{V}}_{\mathbf{I}} = \mathsf{ocens}^U_{\mathbf{I}}$ iff U defines $\mathbb{C} \setminus \{\mathcal{I} \mid \mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$ in $\mathbb{C}$.*

Using this theorem together with definability results in Finite Model Theory, we can show that views and obstructions cannot simulate one another in general.

Theorem 10. *The following statements hold.*

1. *There is a Datalog CQE instance admitting a confidentiality preserving view censor not based on any obstruction.*
2. *Conversely, there is a Datalog CQE instance admitting a confidentiality preserving obstruction censor that is not based on any view.*

5 Optimal View Censors

Our discussion in Section 4 suggests that view and obstruction censors must be studied independently. In this section we focus on view censors and start by establishing their theoretical limitations. The following example shows that optimal view censors may not exist, even if we restrict ourselves to empty ontologies.

Example 11. Consider a CQE instance with an empty ontology, a dataset consisting of a fact $R(a, a)$, and a policy $P = \exists x \exists y \exists z.R(x, y) \wedge R(y, z) \wedge R(z, x)$. Consider also the family of Boolean CQs $Q_n = \exists x_1 \dots \exists x_n. \bigwedge_{i<j} R(x_i, x_j)$, which represent strict total orders on n elements. Answering these queries positively is harmless: $\mathcal{V} \cup \{Q_n\}_{n \geq 1} \not\models P$ for any confidentiality preserving view $\mathcal{V}$. Assume now that $\mathcal{V}$ is optimal, and let m be the number of constants in $\mathcal{V}$. Then, $\mathcal{V} \not\models Q_{m+1}$ since otherwise $\mathcal{V}$ would entail $\exists x.\, R(x, x)$ and violate the policy. This contradicts the optimality of $\mathcal{V}$, and hence no optimal view exists. ◊

Furthermore, determining the existence of an optimal view is undecidable even for Datalog CQE instances.

Theorem 12. *The problem of checking whether a Datalog CQE instance admits an optimal view is undecidable.*

Proof (idea). The proof is by reduction to the undecidable problem of checking whether a deterministic Turing machine without a final state has a repeated configuration in a run on the empty tape. For each such machine we construct a CQE instance such that the run corresponds to an infinite grid-like "view" with axes for the tape and time. The ontology of the instance is constructed to guarantee that representations of adjacent configurations agree with the transition function, while the policy forbids invalid configurations (e.g., with many symbols in a cell). Then, coinciding configurations appear in the run if and only if the grid can be "folded" to a finite view on all sides (i.e., if the representations of these configurations can be merged). If the first pair of such configurations is merged, then the view is optimal. □

In what follows we identify classes of CQE instances that guarantee existence of optimal view censors. We start by studying restrictions on Datalog ontologies and then adapt the obtained results to the OWL 2 profiles.

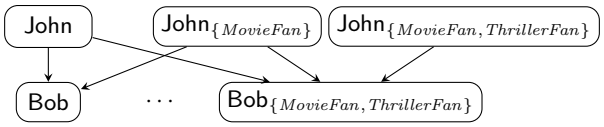

Figure 1: Part of optimal view in Example 13 (labels of nodes coincide to subscripts and omitted, same as labels of arrows, which represent $FrOf$ relation)

5.1 Guarded Tree-Shaped Datalog

The idea behind view censors is to anonymise information in the original data in such a way that the policy cannot be violated. For instance, in Example 5 we replaced the atom $FrOf(\mathsf{John}, \mathsf{Bob})$ with $FrOf(\mathsf{John}, an_b)$, where an_b is a fresh anonymised copy of Bob. In general, however, many such anonymous copies may be required for each data constant to encode all the information required for ensuring optimality. The limit case is illustrated by Example 11, where no finite number of fresh constants suffices for optimality.

Observe that the CQE instance used in Example 11 is neither guarded nor tree-shaped due to the form of the policy. In what follows we show that an optimal view can always be constructed using at most exponentially many anonymous constants if we restrict ourselves to Datalog CQE instances that are guarded and tree-shaped.

We next provide an intuitive idea of the construction. Consider the view for a CQE instance $(\mathcal{O}, \mathcal{D}, P)$ consisting of the following three components $\mathcal{V}_1$–$\mathcal{V}_3$.

(1) Component $\mathcal{V}_1$ is any maximal set of unary atoms in $\mathcal{H}_{\mathcal{O},\mathcal{D}}$ that does not compromise the policy.

(2) To construct $\mathcal{V}_2$, we consider an anonymised copy $a_{\mathcal{B}}$ of each constant a and each set $\mathcal{B}$ of unary predicates B such that $\mathcal{H}_{\mathcal{O},\mathcal{D}} \models B(a)$. The corresponding set of all unary atoms $B(a_{\mathcal{B}})$ for $B \in \mathcal{B}$ is a part of $\mathcal{V}_2$ if and only if it is "safe", that is, neither discloses the policy nor entail new facts together with $\mathcal{O} \cup \mathcal{V}_1$.

(3) Finally, $\mathcal{V}_3$ consists of a maximal set of binary atoms on all the constants (including the copies) that are justified by $\mathcal{H}_{\mathcal{O},\mathcal{D}}$ and do not disclose the policy.

Optimality of this view follows immediately from the construction. The view, however, may require exponentially many anonymised copies of data constants. The need for them is illustrated by the following example.

Example 13. Consider the CQE instance with ontology consisting of rules

$$\begin{aligned} \mathit{ThrillerFan}(y) \land \mathit{FrOf}(x,y) &\rightarrow \mathit{MovieFan}(x) \text{ and} \\ \mathit{ThrillerFan}(x) &\rightarrow \mathit{MovieFan}(x), \end{aligned}$$

dataset consisting of facts

$$\mathit{FrOf}(\mathsf{John}, \mathsf{Bob}),\ \mathit{ThrillerFan}(\mathsf{John}),\ \mathit{ThrillerFan}(\mathsf{Bob}),$$

and policy $\mathit{MovieFan}(x)$. The essential part of the optimal view obtained using the aforementioned construction is given in Figure 1. According to the construction, $\mathcal{V}_1$ is empty, $\mathcal{V}_2$ contains unary atoms over the anonymised copies $\mathsf{John}_{\{\mathit{MovieFan}\}}$ and $\mathsf{John}_{\{\mathit{MovieFan}, \mathit{ThrillerFan}\}}$ of John, and $\mathsf{Bob}_{\{\mathit{MovieFan}, \mathit{ThrillerFan}\}}$ of Bob, while $\mathcal{V}_3$ contains the $FrOf$ atoms represented by arrows. Note that at least two anonymised copies of John are necessary in any optimal view to answer correctly "harmless" queries such as

$$\begin{aligned} \exists x\, \exists y\, \exists z.\, &\mathit{ThrillerFan}(x) \land \mathit{FrOf}(x,y) \land \\ &\mathit{ThrillerFan}(y) \land \mathit{FrOf}(z,y) \land \\ &\mathit{MovieFan}(z) \land \mathit{FrOf}(z, \mathsf{Bob}). \end{aligned}$$

$\diamondsuit$

This example shows that, in order to avoid the exponential blow up in the number of anonymised copies, we need further restrictions on the ontology. In particular, in the case of multi-linear CQE instances we can guarantee that just one copy suffices for every constant.

The following theorem formalises the intuition above.

Theorem 14. *Let* $\mathbf{I}$ *be a Datalog tree-shaped CQE instance.*

1. *If* $\mathbf{I}$ *is guarded, then it admits an optimal view that can be computed in time exponential in* $|\mathbf{I}|$ *and polynomial in data size.*
2. *If* $\mathbf{I}$ *is multi-linear, then it admits an optimal view that can be computed in time polynomial in* $|\mathbf{I}|$.

Additionally, if $\mathbf{I}$ *is linear the it has a unique optimal censor.*

5.2 OWL 2 Profiles

The result in Theorem 14 is immediately applicable to RL ontologies, with the only restriction that they do not contain rules of types (1), (4), or (5) in Table 1. In contrast to RL, the QL and EL profiles provide means for capturing existentially quantified knowledge. To bridge this gap, we show that every (guarded) QL or EL CQE instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ can be polynomially trasformed into a Datalog CQE instance $\mathbf{I}' = (\mathcal{O}', \mathcal{D}, P)$ by rewriting $\mathcal{O}$ into a (guarded and tree-shaped) Datalog ontology $\mathcal{O}'$ such that optimal views for $\mathbf{I}$ can be directly obtained from those for $\mathbf{I}'$. We start by specifying what constitutes an acceptable rewriting $\mathcal{O}'$ of $\mathcal{O}$.

Definition 15. *Let* σ *be a set of constants.*[1] *A Datalog ontology* $\mathcal{O}'$ *is a* σ-rewriting *of an ontology* $\mathcal{O}$ *if* $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) = \mathsf{cert}(Q, \mathcal{O}', \mathcal{D})$ *for each tree-shaped CQ* Q *and dataset* $\mathcal{D}$ *over constants from* σ.

The following proposition provides the mechanism to reduce optimal view computation for arbitrary ontologies to the case of Datalog.

Proposition 16. *Let* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ *be a CQE instance over constants* σ *with* P *tree-shaped, and* $\mathcal{O}'$ *a* σ*-rewriting of* $\mathcal{O}$ *such that* $\mathcal{O}' \models \mathcal{O}$. *If* $\mathcal{V}'$ *is an optimal view for* $\mathbf{I}' = (\mathcal{O}', \mathcal{D}, P)$, *then* $\mathcal{H}_{\mathcal{O}',\mathcal{V}'}$ *is an optimal view for* $\mathbf{I}$.

With this proposition at hand, we just need to devise a technique for rewriting any QL (or guarded EL) ontology into a stronger Datalog ontology, which, however, preserves the answers to all tree-shaped queries. To this end, we exploit techniques developed for the so-called *combined approach* to query answering [Kontchakov *et al.*, 2011; Lutz *et al.*, 2009; 2013; Stefanoni *et al.*, 2013]. The idea is to transform rules of type (3) into Datalog by Skolemising existentially quantified variables into globally fresh constants. Such transformation strengthens the ontology; however, if applied to a QL

[1] The role of the set σ is purely technical—it allows us to pick fresh constants in Definition 17.

or guarded EL ontology, it preserves answers to tree-shaped CQs for any dataset over σ [Stefanoni *et al.*, 2013].

Definition 17. *Let $\mathcal{O}$ be an ontology and σ be a set of constants. The ontology $\Xi_\sigma(\mathcal{O})$ is obtained from $\mathcal{O}$ by replacing each rule $A(x) \to \exists y.[R(x,y) \wedge B(y)]$ with*

$$A(x) \to R'(x,a), R'(x,y) \to R(x,y), R'(x,y) \to B(y),$$

where R' is a fresh binary predicate, uniquely associated to the original rule, and a is a globally fresh constant not from σ, uniquely associated to A and R.

Theorem 18. *For any ontology $\mathcal{O}$ we have $\Xi_\sigma(\mathcal{O}) \models \mathcal{O}$. Furthermore, if $\mathcal{O}$ is either a QL or guarded EL ontology, then $\Xi_\sigma(\mathcal{O})$ is a σ-rewriting of $\mathcal{O}$.*

Proposition 16 and Theorem 18 ensure that $\mathcal{H}_{\Xi_\sigma(\mathcal{O}),\mathcal{V}}$ is an optimal view for $\mathbf{I}$ whenever $\mathcal{V}$ is such a view for $\mathbf{I}' = (\Xi_\sigma(\mathcal{O}), \mathcal{D}, P)$. The transformation of $\mathcal{O}$ to $\Xi_\sigma(\mathcal{O})$ preserves linearity, guardedness, and tree-shapedness, so the results of Section 5.1 are applicable to $\mathbf{I}'$.

Theorem 19. *Every guarded EL CQE instance admits an optimal view that can be computed in exponential time. Every QL instance admits a unique optimal censor, which is implementable by a view of polynomial size.*

6 Optimal Obstruction Censors

Similarly to Section 5, we start the study of optimal obstruction censors with their limitations. The following example shows that such a censor may not exist even if we restrict ourselves to ontologies with only one rule.

Example 20. Consider a CQE instance with an ontology $\{R(x,y) \wedge A(y) \to A(x)\}$, dataset $\{R(a,a), A(a)\}$, and policy $A(a)$. Let Q_n, for $n > 0$, be a family of Boolean CQs

$$\exists x_1 \dots \exists x_n. \\ R(a,x_1) \wedge R(x_1,x_2) \wedge \dots \wedge R(x_{n-1},x_n) \wedge A(x_n).$$

With the help of the ontology each of Q_n discloses the policy. Thus, each Q_n should entail a Boolean CQ in any optimal obstruction. Consider now the set of all CQs that are entailed by queries Q_n but not equivalent to any of them. On the one hand, this set is "harmless", that is, any obstruction censor should answer all these queries positively. On the other hand, the CQs Q_n do not entail each other. Hence, any optimal obstruction should contain a CQ equivalent to every Q_n, which is however not possible, because n is unbounded. $\Diamond$

We leave the question of decidability of checking the existence of an optimal obstruction for a CQE instance open. In fact, answering this question positively would imply a solution to a long-standing open problem on uniform boundedness for Datalog programs over binary signatures (see [Marcinkowski, 1999] for details of the problem and the extended version [Cuenca Grau *et al.*, 2015] of this paper for the reduction).

In the rest of this section we give a characterisation of optimal obstructions for Datalog instances in terms of resolution proofs and identify restrictions for which this characterisation guarantees existence of such obstructions.

6.1 Characterisation of Optimal Obstructions

We first recall the standard notion of SLD resolution.

A *goal* is a conjunction of atoms. An *SLD resolution step* takes a goal $\beta \wedge \varphi$ with a selected atom β and a sentence r that is either a Datalog rule $\psi \to \delta$ or a fact δ, and produces a new goal $\varphi\theta \wedge \psi\theta$, where θ is a most general unifier of β and δ (assuming that ψ is empty in the case when r is a fact). An *(SLD) proof* of a goal G_0 in a Datalog ontology $\mathcal{O}$ and dataset $\mathcal{D}$ is a sequence of goals $G_0, G_1, \dots, G_n$, where G_n is empty, and each G_i is produced from G_{i-1} and a sentence (rule or fact) in $\mathcal{O} \cup \mathcal{D}$ by an SLD resolution step.

Resolution is sound and complete: for any Datalog ontology $\mathcal{O}$, dataset $\mathcal{D}$, and goal G (such that $\mathcal{O} \cup \mathcal{D}$ is satisfiable) there is a proof of G in $\mathcal{O}$ and $\mathcal{D}$ if and only if $\mathcal{O} \cup \mathcal{D} \models \exists^* G$ for the existential closure $\exists^* G$ of G.

We next characterise optimal obstructions using SLD proofs. Intuitively, if an obstruction censor answers positively sufficient number of Boolean CQs $\exists^* G$ for goals G in a proof of a policy, then a user could reconstruct (a part of) this proof and compromise the policy. Also, there can be many proofs, and a user may compromise the policy by reconstructing any of them. Thus, to ensure that a censor is confidentiality preserving, we must guarantee that the obstruction contains enough CQs to prevent reconstruction of any proof. If we also want the censor to be optimal, the obstruction should not block too many CQs. As we will see later on, these requirements may be in conflict and lead to an infinite "obstruction". Next definitions formalise this intuition.

Definition 21. *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ be a Datalog CQE instance, $\mathbb{Q}$ be the set of all Boolean CQs $\exists^* G$ for goals G in proofs of $P(\vec{a})$ in $\mathcal{O}$ and $\mathcal{D}$ for some tuple of constants $\vec{a}$, and $\mathbb{S}$ be a maximal subset of $\mathbb{Q}$ such that $\mathcal{O} \cup \mathbb{S} \not\models P(\vec{a})$ for any $\vec{a}$. Then, a* pseudo-obstruction *for $\mathbf{I}$ is a subset of $\mathbb{Q} \setminus \mathbb{S}$ that contains a CQ Q' for any Q in $\mathbb{Q} \setminus \mathbb{S}$ with $Q \models Q'$.*

The next theorem establishes the connection between pseudo-obstructions and optimality.

Theorem 22. *Let $\mathbf{I}$ be a Datalog CQE instance.*

1. *If Υ is a finite pseudo-obstruction for $\mathbf{I}$, then $\bigvee_{Q \in \Upsilon} Q$ is an optimal obstruction for $\mathbf{I}$.*
2. *If each pseudo-obstruction for $\mathbf{I}$ is infinite, then no optimal obstruction censor for $\mathbf{I}$ exists.*

This theorem has implications on the expressive power of obstructions. In particular, we can now extend the result in Theorem 10, which applies to censors that are not necessarily optimal, to capture also optimality.

Theorem 23. *The following statements hold.*

1. *There is a CQE instance, which is both RL and EL, admitting an optimal view, but no optimal obstruction.*
2. *Conversely, there exists an RL CQE instance that admits an optimal obstruction, but no optimal view.*

6.2 Linear Datalog and OWL 2 QL

We now show how to apply resolution-based techniques to compute optimal obstructions for linear Datalog CQE instances and then adapt the results to QL. In fact, we can guarantee not only existence of optimal obstructions for such

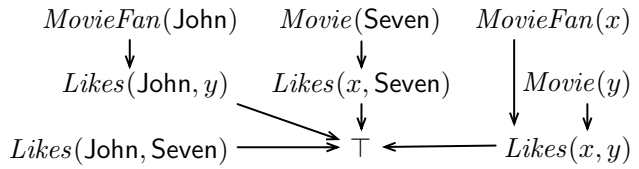

Figure 2: Fragment of proof graph from Example 24

instances, but also uniqueness and polynomiality of corresponding censors.

Our solution for linear Datalog instances is based on the computation of the set $\mathbb{Q}$ of existential closures of goals in the proofs of policies. However, since all the rules in the ontology are linear and the body of the policy is an atom (recall that the rule corresponding to the policy should be linear as well), each of these goals consists of a single atom, except the last goal in each proof, which is empty. There are only polynomial number of such atoms (up to renaming of variables). So, all the proofs can be represented by a single finite *proof graph* with atoms and the empty conjunction (denoted by $\top$) as nodes, and SLD resolution steps as edges. This is illustrated by the following example.

Example 24. Consider a CQE instance with an ontology

$$\{Likes(x,y) \to Movie(y),\, Likes(x,y) \to MovieFan(x)\},$$

dataset $Likes(\mathsf{John}, \mathsf{Seven})$, and policy $MovieFan(\mathsf{John})$. A fragment of the proof graph is given in Figure 2. ◊

Using proof graphs we can compute optimal censors.

Theorem 25. *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ be a linear Datalog CQE instance, and let S be the set of all nodes in the proof graph of $\mathcal{O} \cup \mathcal{D}$ on the paths from facts $P(\vec{a})$ with any tuple of constants $\vec{a}$ to $\top$. Then, the Boolean UCQ*

$$U = \bigvee\nolimits_{G \in S \setminus \{\top\}} \exists^* G$$

is an optimal obstruction computable in polynomial time, and $\mathsf{ocens}^U_{\mathbf{I}}$ is the unique optimal censor for $\mathbf{I}$.

Example 26. For the instance in Example 24 there is only one path in the proof graph from the policy to $\top$, and $S = \{MovieFan(\mathsf{John}), Likes(\mathsf{John}, y), \top\}$. Thus, $movieFan(\mathsf{John}) \vee \exists y.Likes(\mathsf{John}, y)$ is optimal. ◊

Finally, note that the transformation of a QL ontology $\mathcal{O}$ to an RL ontology $\Xi_\sigma(\mathcal{O})$ given in Definition 17, preserves linearity of rules. Hence, Proposition 18 with Theorem 25 yield the following result.

Theorem 27. *Every QL CQE instance admits a unique optimal censor based on an obstruction that can be computed in polynomial time.*

7 Related Work

The formal study of privacy in databases has received significant attention. CQE for propositional databases with complete information has been studied in [Sicherman *et al.*, 1983; Bonatti *et al.*, 1995; Biskup and Bonatti, 2001; 2004]. The framework was extended to (propositional) incomplete databases in [Biskup and Weibert, 2008]. Miklau and Suciu (2007) studied *perfect privacy*. Perfect privacy, however, is very strict and may preclude publishing of any meaningful information when extended to ontologies [Cuenca Grau and Horrocks, 2008]. View-based authorisation was investigated in [Rizvi *et al.*, 2004; Zhang and Mendelzon, 2005], while Deutsch and Papakonstantinou (2005) analysed the implications to privacy derived from publishing database views.

Privacy in the context of ontologies is a growing area of research. Information hiding at the schema level was studied in [Konev *et al.*, 2009; Cuenca Grau and Motik, 2012]. Data privacy for $\mathcal{EL}$ and $\mathcal{ALC}$ DLs was investigated in [Stouppa and Studer, 2007; Tao *et al.*, 2010], and the notion of a privacy-preserving reasoner was introduced in [Bao *et al.*, 2007]. Calvanese *et al.* (2012) extended the view-based authorisation framework by Zhang and Mendelzon (2005) to DL ontologies.

An early work on non-propositional CQE is [Biskup and Bonatti, 2007]. CQE for ontologies has been studied in [Cuenca Grau *et al.*, 2013; Bonatti and Sauro, 2013; Studer and Werner, 2014]. We extend Cuenca Grau *et al.* (2013) with a wide range of new results: *(i)* we consider arbitrary CQs as policies rather than just ground facts; *(ii)* we introduce obstruction censors, compare their expressive power with that of view censors, characterise their optimality, and show how to compute obstructions for linear Datalog and QL ontologies; *(iii)* we show undecidability of checking existence of an optimal view censor and provide algorithms for guarded Datalog and all the OWL 2 profiles. We see our work as complementary to Bonatti and Sauro (2013) and Studer and Werner (2014). The former focuses on situations where attackers have access to external sources of background knowledge; they identify vulnerabilities and propose solutions within the CQE framework. The latter focuses on meta-properties of general censors that, in contrast to ours, can also provide unsound answers or refuse queries.

8 Conclusions

In this paper, we have studied CQE in the context of ontologies. Our results provide insights on the fundamental trade-off between accessibility and confidentiality of information. Moreover, they yield a flexible way for system designers to ensure selective access to data.

We have proposed tractable view based solutions for CQE instances with tree-shaped and linear Datalog and QL ontologies, and tractable obstruction based solutions for linear Datalog and QL ontologies. Our solutions can be implemented using off-the-shelf query answering infrastructure and provide a starting point for CQE system development.

Acknowledgements

This research has been partially supported by the Royal Society, the EPSRC grants Score!, DBonto, and MaSI^3, and the FP7 project Optique.

References

[Bao *et al.*, 2007] Jie Bao, Giora Slutzki, and Vasant Honavar. Privacy-Preserving Reasoning on the Semantic Web. In *WI*, pages 791–797, 2007.

[Biskup and Bonatti, 2001] Joachim Biskup and Piero Bonatti. Lying Versus Refusal for Known Potential Secrets. *Data Knowl. Eng.*, 38(2):199–222, 2001.

[Biskup and Bonatti, 2004] Joachim Biskup and Piero Bonatti. Controlled Query Evaluation for Enforcing Confidentiality in Complete Information Systems. *Int. J. Inf. Sec.*, 3(1):14–27, 2004.

[Biskup and Bonatti, 2007] Joachim Biskup and Piero Bonatti. Controlled Query Evaluation with Open Queries for a Decidable Relational Submodel. *Ann. Math. and Artif. Intell.*, 50(1-2):39–77, 2007.

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

[Bonatti and Sauro, 2013] Piero Bonatti and Luigi Sauro. A Confidentiality Model for Ontologies. In *ISWC*, pages 17–32, 2013.

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

[Calvanese *et al.*, 2012] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. View-based Query Answering in Description Logics: Semantics and Complexity. *J. Comput. Syst. Sci.*, 78(1):26–46, 2012.

[Cuenca Grau and Horrocks, 2008] Bernardo Cuenca Grau and Ian Horrocks. Privacy-Preserving Query Answering in Logic-based Information Systems. In *ECAI*, pages 40–44, 2008.

[Cuenca Grau and Motik, 2012] Bernardo Cuenca Grau and Boris Motik. Reasoning over Ontologies with Hidden Content: The Import-by-Query Approach. *J. Artif. Intell. Res.*, 45:197–255, 2012.

[Cuenca Grau *et al.*, 2013] Bernardo Cuenca Grau, Evgeny Kharlamov, Egor V. Kostylev, and Dmitriy Zheleznyakov. Controlled Query Evaluation over OWL 2 RL Ontologies. In *ISWC*, pages 49–65, 2013.

[Cuenca Grau *et al.*, 2015] Bernardo Cuenca Grau, Evgeny Kharlamov, Egor V. Kostylev, and Dmitriy Zheleznyakov. Controlled Query Evaluation for Datalog and OWL 2 Profile Ontologies (Extended Version). *CoRR*, abs/1504.06529, 2015.

[Deutsch and Papakonstantinou, 2005] Alin Deutsch and Yannis Papakonstantinou. Privacy in Database Publishing. In *ICDT*, pages 230–245, 2005.

[Kolaitis and Vardi, 2008] Phokion G. Kolaitis and Moshe Y. Vardi. A Logical Approach to Constraint Satisfaction. In *Complexity of Constraints*, pages 125–155, 2008.

[Konev *et al.*, 2009] Boris Konev, Dirk Walther, and Frank Wolter. Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In *IJCAI*, pages 830–835, 2009.

[Kontchakov *et al.*, 2011] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The Combined Approach to Ontology-Based Data Access. In *IJCAI*, pages 2656–2661, 2011.

[Lutz *et al.*, 2009] Carsten Lutz, David Toman, and Frank Wolter. Conjunctive Query Answering in the Description Logic EL Using a Relational Database System. In *IJCAI*, pages 2070–2075, 2009.

[Lutz *et al.*, 2013] Carsten Lutz, Inanç Seylan, David Toman, and Frank Wolter. The Combined Approach to OBDA: Taming Role Hierarchies Using Filters. In *ISWC*, pages 314–330, 2013.

[Marcinkowski, 1999] Jerzy Marcinkowski. Achilles, Turtle, and Undecidable Boundedness Problems for Small DATALOG Programs. *SIAM J. Comput.*, 29(1):231–257, 1999.

[Miklau and Suciu, 2007] Gerome Miklau and Dan Suciu. A Formal Analysis of Information Disclosure in Data Exchange. *J. Comput. Syst. Sci.*, 73(3):507–534, 2007.

[Motik *et al.*, 2012] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. OWL 2 Web Ontology Language Profiles (2nd Edition), 2012. W3C Recommendation.

[Rizvi *et al.*, 2004] Shariq Rizvi, Alberto O. Mendelzon, S. Sudarshan, and Prasan Roy. Extending Query Rewriting Techniques for Fine-Grained Access Control. In *SIGMOD*, pages 551–562. ACM, 2004.

[Sandhu *et al.*, 1996] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.

[Sicherman *et al.*, 1983] George L. Sicherman, Wiebren de Jonge, and Reind P. van de Riet. Answering Queries Without Revealing Secrets. *ACM Trans. Database Syst.*, 8(1):41–59, 1983.

[Stefanoni *et al.*, 2013] Giorgio Stefanoni, Boris Motik, and Ian Horrocks. Introducing Nominals to the Combined Query Answering Approaches for EL. In *AAAI*, pages 1177–1183, 2013.

[Stouppa and Studer, 2007] Phiniki Stouppa and Thomas Studer. A Formal Model of Data Privacy. In *PSI*, pages 400–408, 2007.

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

[Tao *et al.*, 2010] Jia Tao, Giora Slutzki, and Vasant Honavar. Secrecy-Preserving Query Answering for Instance Checking in $\mathcal{EL}$. In *RR*, pages 195–203, 2010.

[Zhang and Mendelzon, 2005] Zheng Zhang and Alberto O. Mendelzon. Authorization Views and Conditional Query Containment. In *ICDT*, pages 259–273, 2005.