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

Table 1: OWL 2 profile axioms as rules

is, universally quantified sentences of the form

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

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 $S(\vec{x}) \to \top(x)$ 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* $\mathcal{O} \cup \{\varphi(\vec{x}, \vec{y}) \to A_p(\vec{x})\}$*, 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:

$$\begin{aligned}\mathcal{O}_{\mathsf{ex}} = \{ \quad & \mathit{Likes}(x, y) \wedge \mathit{Thriller}(y) \to \mathit{ThrillerFan}(x),\\ & \mathit{Suspense}(x) \wedge \mathit{Crime}(x) \to \mathit{Thriller}(x),\\ & \mathit{FrOf}(x, y) \to \mathit{FrOf}(y, x) \qquad \},\\ \mathcal{D}_{\mathsf{ex}} = \{ \quad & \mathit{FrOf}(\mathsf{John}, \mathsf{Bob}), \mathit{FrOf}(\mathsf{Bob}, \mathsf{Mary}),\\ & \mathit{Crime}(\mathsf{Seven}), \mathit{Suspense}(\mathsf{Seven}),\\ & \mathit{Likes}(\mathsf{John}, \mathsf{Seven}), \mathit{Likes}(\mathsf{Bob}, \mathsf{Seven}) \qquad \}.\end{aligned}$$

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. ◊

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 $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$ *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 $\mathit{Likes}(\mathsf{Bob}, \mathsf{Seven})$, 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 $\mathsf{ocens}_{\mathbf{I}}^{U}$ *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* $\mathsf{ocens}_{\mathbf{I}}^{U}$.

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}:

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

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,* $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}} = \mathsf{ocens}_{\mathbf{I}}^{U}$ *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 $Q_n = \exists x_1 \ldots \exists x_n. \bigwedge_{i<j} R(x_i, x_j)$, 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.

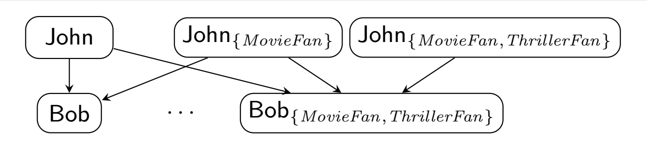
John
John$_{\{MovieFan\}}$
John$_{\{MovieFan,\,ThrillerFan\}}$
Bob
. . .
Bob$_{\{MovieFan,\,ThrillerFan\}}$

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

$$\begin{array}{rcl} ThrillerFan(y) \wedge FrOf(x,y) & \rightarrow & MovieFan(x) \text{ and} \\ ThrillerFan(x) & \rightarrow & MovieFan(x), \end{array}$$

dataset consisting of facts

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

and policy $MovieFan(x)$. 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 $\mathsf{John}_{\{MovieFan\}}$ and $\mathsf{John}_{\{MovieFan,\, ThrillerFan\}}$ of John, and $\mathsf{Bob}_{\{MovieFan,\, ThrillerFan\}}$ 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

$$\begin{array}{l} \exists x\, \exists y\, \exists z.\, ThrillerFan(x) \wedge FrOf(x,y) \wedge \\ \qquad ThrillerFan(y) \wedge FrOf(z,y) \wedge \\ \qquad MovieFan(z) \wedge FrOf(z, \mathsf{Bob}). \qquad \diamondsuit \end{array}$$

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 $A(x) \rightarrow \exists y.[R(x,y) \wedge B(y)]$ with*

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

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 $\{R(x,y) \wedge A(y) \rightarrow A(x)\}$, dataset $\{R(a,a), A(a)\}$, and policy $A(a)$. Let Q_n, for $n > 0$, be a family of Boolean CQs

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

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. $\diamondsuit$

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 $\psi \rightarrow \delta$ 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 $G_0, G_1, \ldots, G_n$, 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

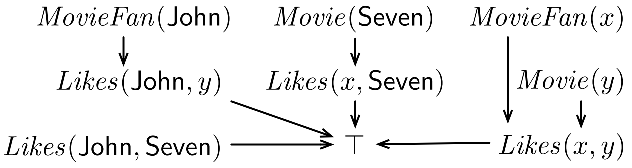
$\mathit{MovieFan}(\mathsf{John})$
$\mathit{Movie}(\mathsf{Seven})$
$\mathit{MovieFan}(x)$
$\mathit{Likes}(\mathsf{John}, y)$
$\mathit{Likes}(x, \mathsf{Seven})$
$\mathit{Movie}(y)$
$\mathit{Likes}(\mathsf{John}, \mathsf{Seven})$
$\top$
$\mathit{Likes}(x, y)$

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

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

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

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 $\mathsf{ocens}_{\mathbf{I}}^{U}$ *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 $S = \{\mathit{MovieFan}(\mathsf{John}), \mathit{Likes}(\mathsf{John}, y), \top\}$. Thus, $\mathit{movieFan}(\mathsf{John}) \vee \exists y.\mathit{Likes}(\mathsf{John}, y)$ is optimal. $\Diamond$

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 tradeoff 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.

A Appendix (Proofs)

A.1 Proofs for Section 4

Before proving Theorem 9 we present the following notation and a lemma. Let $\mathcal{I}$ be a finite structure and f a function associating a fresh variable to each domain element of $\mathcal{I}$. The query $Q^{\mathcal{I}}$ for $\mathcal{I}$ is the Boolean CQ defined as follows, with $R_1, \ldots R_n$ the predicates interpreted by $\mathcal{I}$:

$$Q^{\mathcal{I}} = \exists^* \bigwedge_{1 \leq i \leq n} \{R_i(f(u_1), \ldots, f(u_{m_i})) \mid (u_1, \ldots, u_{m_i}) \in R_i^{\mathcal{I}}\}.$$

Given a BCQ Q, denote $[Q]$ the structure interpreting each R, occurring in Q, with $(f(u_1), \ldots f(u_n))$ for every atom $R(u_1, \ldots, u_n)$ in Q, where f maps each constant in Q to itself and each variable y to a fresh constant d_y.

Lemma 28. *Let $\mathcal{J}$ be a finite structure and let $\mathbb{C}$ be a class of finite structures. Then, the following holds:*

$$\{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \not\hookrightarrow \mathcal{J}\} = \{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \models \bigvee_{\mathcal{K} \in \mathbb{C}, \mathcal{K} \not\hookrightarrow \mathcal{J}} Q^{\mathcal{K}}\}.$$

Proof. Let $\mathcal{I} \in \mathbb{C}$ be such that $\mathcal{I} \not\hookrightarrow \mathcal{J}$; clearly, $\mathcal{I} \models Q^{\mathcal{I}}$ and hence $\mathcal{I} \models \bigvee_{\mathcal{K} \in \mathbb{C}, \mathcal{K} \not\hookrightarrow \mathcal{J}} Q^{\mathcal{K}}$, as required. Conversely, assume that $\mathcal{I} \in \mathbb{C}$ is such that $\mathcal{I} \models \bigvee_{\mathcal{K} \in \mathbb{C}, \mathcal{K} \not\hookrightarrow \mathcal{J}} Q^{\mathcal{K}}$; then, there exists $\mathcal{K}$ such that $\mathcal{K} \in \mathbb{C}$, $\mathcal{K} \not\hookrightarrow \mathcal{J}$ and $\mathcal{I} \models Q^{\mathcal{K}}$. The latter implies that $\mathcal{K} \hookrightarrow \mathcal{I}$ and hence we can deduce $\mathcal{I} \not\hookrightarrow \mathcal{J}$, as required (otherwise, we would have by composition of homomorphisms that $\mathcal{K} \hookrightarrow \mathcal{J}$, which is a contradiction). □

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, $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}} = \mathsf{ocens}_{\mathbf{I}}^{U}$ iff U defines $\mathbb{C} \setminus \{\mathcal{I} \mid \mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$ in $\mathbb{C}$.*

Proof.
($\Leftarrow$) Assume that U defines $\mathbb{C} \setminus \{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$, which is equal to $\{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$. Then, for each $\mathcal{I} \in \mathbb{C}$ we have that $\mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$ iff $\mathcal{I} \models U$. By Lemma 28, the following holds for each $\mathcal{I} \in \mathbb{C}$:

$$\mathcal{I} \models U \quad \text{iff} \quad \mathcal{I} \models \bigvee_{\mathcal{K} \in \mathbb{C}, \mathcal{K} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}} Q^{\mathcal{K}}. \tag{1}$$

Let $Q(\vec{x})$ be a CQ, and let $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$, which implies that $[Q(\vec{t})] \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{D}}$ and hence $[Q(\vec{t})] \in \mathbb{C}$. We show that $\vec{t} \in \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q)$ iff $\vec{t} \in \mathsf{ocens}_{\mathbf{I}}^{U}(Q)$.

For the forward direction, assume that $\vec{t} \in \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q)$; then, $\mathcal{O} \cup \mathcal{V} \models Q(\vec{t})$ and hence $[Q(\vec{t})] \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$. We can then conclude $[Q(\vec{t})] \not\models \bigvee_{\mathcal{K} \in \mathbb{C}, \mathcal{K} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}} Q^{\mathcal{K}}$ (otherwise, $\mathcal{K} \hookrightarrow [Q(\vec{t})]$ for some $Q^{\mathcal{K}}$ in U and since we have established that $[Q(\vec{t})] \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$ and homomorphism compose we would have $\mathcal{K} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$ which is a contradiction). But then, Equation (1) implies that $[Q(\vec{t})] \not\models U$ and by the definition of obstruction-censor that $\vec{t} \in \mathsf{ocens}_{\mathbf{I}}^{U}(Q)$, as required.

For the backward direction, assume now that $\vec{t} \in \mathsf{ocens}_{\mathbf{I}}^{U}(Q)$. Then, by the definition of obstruction censor we have $[Q(\vec{t})] \not\models U$. By Equation (1) we then have $[Q(\vec{t})] \not\models \bigvee_{\mathcal{K} \in \mathbb{C}, \mathcal{K} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}} Q^{\mathcal{K}}$. Lemma 28 immediately implies that $[Q(\vec{t})] \notin \{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$. From this, we must conclude that $[Q(\vec{t})] \in \{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$ and hence $[Q(\vec{t})] \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$, which implies $\mathcal{O} \cup \mathcal{V} \models Q(\vec{t})$ and $\vec{t} \in \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q)$, as required.

($\Rightarrow$) Assume that $\mathsf{ocens}_{\mathbf{I}}^{U} = \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$. To show that U defines $\{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$, we prove that $\mathcal{I} \models U$ iff $\mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$ for every structure $\mathcal{I}$ in $\mathbb{C}$. If $\mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{D}}$ and $\mathcal{I} \models U$, then $\mathsf{ocens}_{\mathbf{I}}^{U}(Q^{\mathcal{I}}) = \mathtt{False}$. Since $\mathsf{ocens}_{\mathbf{I}}^{U} = \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$, we also have that $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q^{\mathcal{I}}) = \mathtt{False}$ and hence $\mathcal{O} \cup \mathcal{V} \not\models Q^{\mathcal{I}}$. Consequently, $\mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$, as required. If $\mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$, then $\mathcal{O} \cup \mathcal{V} \not\models Q^{\mathcal{I}}$; consequently, $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q^{\mathcal{I}}) = \mathtt{False}$. Since $\mathsf{ocens}_{\mathbf{I}}^{U} = \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$, we have $\mathsf{ocens}_{\mathbf{I}}^{U}(Q^{\mathcal{I}}) = \mathtt{False}$ and hence, since $\mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{D}}$, we necessarily have $\mathcal{I} \models U$. □

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.*

Proof. First we illustrate that obstruction censors cannot always simulate view censors. Consider CQE instance $\mathbf{I} = (\emptyset, \mathcal{D}, \emptyset)$, where $\mathcal{D}$ represents an undirected graph with nodes “green” g and “blue” b, which are connected by edge in all possible ways:

$$\mathcal{D} = \{\mathit{edge}(g, b), \mathit{edge}(b, g), \mathit{edge}(b, b), \mathit{edge}(g, g)\}.$$

Clearly, $\mathcal{D}$ entails every Boolean CQ over the edge relation and thus every graph can be homomorphically embedded into $\mathcal{D}$. Consider $\mathcal{V} = \{\mathit{edge}(g, b), \mathit{edge}(b, g)\}$. Since the ontology is empty, $\mathcal{H}_{\emptyset,\mathcal{V}} = \mathcal{V}$ and $\{\mathcal{I} \mid \mathcal{I} \text{ is finite}, \mathcal{I} \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{D}}, \text{ and } \mathcal{I} \not\hookrightarrow \mathcal{V}\}$

is the class of all graphs that are not 2-colourable. It is well-known that this class of graphs is not first-order definable and hence cannot be captured by a UCQ.

Next we construct an obstruction censor which cannot be simulated by a view censor. Consider the instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \emptyset)$, where $\mathcal{D} = \{edge(a, a)\}$ and $\mathcal{O}$ consists of the single transitivity rule

$$edge(x, y) \wedge edge(y, z) \to edge(x, z).$$

Clearly, $\mathcal{O} \cup \mathcal{D}$ entails each Boolean CQ over the $edge$ relation. Consider obstruction $U = \exists y.edge(y, y)$, which defines the class of directed graphs with self loops. Suppose that some view $\mathcal{V}$ realises $\mathsf{ocens}^U_{\mathbf{I}}$. By Theorem 9, the obstruction U must define $\{\mathcal{I} \in \mathbb{C} \mid \mathcal{I} \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}\}$ where $\mathcal{C}$ is the class of all directed graphs. Thus, any graph G must satisfy the property

$$G \text{ has no self loops iff } G \hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}.$$

Due to the rule in $\mathcal{O}$, we conclude that $\mathcal{V}$ is a DAG, that is, it has no $edge$-loops. Take a DAG G extending (a graph isomorphic to) $\mathcal{H}_{\mathcal{O},\mathcal{V}}$ with a new node v and edges connecting all its sink nodes to v. Clearly G has no self loops, but $G \not\hookrightarrow \mathcal{H}_{\mathcal{O},\mathcal{V}}$, which is a contradiction. □

A.2 Proofs for Section 5

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

Proof. The proof is by reduction from the following problem: does a deterministic Turing machine without a final state have a repeated configuration? This problem is undecidable by Rice's Theorem.

Formally, for every such Turing machine $M = (\Gamma, \mathcal{Q}, q_0, \delta)$ with Γ a tape alphabet, which include the blank symbol 0, $\mathcal{Q}$ a set of states, $q_0 \in \mathcal{Q}$ an initial state, and $\delta : \Gamma \times \mathcal{Q} \to \Gamma \times (\mathcal{Q} \setminus \{q_0\}) \times \{+, -\}$ a transition function, we construct an Datalog CQE instance $\mathbf{I}_M = (\mathcal{O}, \mathcal{D}, P)$ such that it admits an optimal view if and only if M starting on the empty tape has a repeated configuration. The notion of configuration is as usual—it is the content of the tape and the head pointer to a cell on the tape. Note that the transition function δ is defined is such a way that the initial state does not appear in a computation anywhere except the initial configuration. This clearly does not affect the undecidability of the problem. We also assume, that the tape of the machine is infinite in both directions, and all of it can freely be used for computations.

We start the construction of $\mathbf{I}_M$ from the dataset $\mathcal{D}$. It uses only one constant a and consists of three binary atoms

$$R(a, a), S(a, a), T(a, a).$$

The predicate T is intended to point to the next cell on the tape, the predicate S points to the same cell in the following configuration, and the predicate R is responsible for initialisation. We start the definition of the ontology $\mathcal{O}$ with the description of the role of R. Let $\mathcal{O}$ contain rules

$$R(x, x) \to I(x), \tag{2}$$

$$R(x, y) \wedge R(y, z) \to R(x, z), \tag{3}$$

$$R(x, y) \wedge I(y) \to I(x). \tag{4}$$

As we will see formally later, these rules guarantee that if $\mathbf{I}_M$ admits an optimal view, then this view contains the fact $I(a)$. This fact initialises the tape by means of the following rules (conjunction in heads is just a syntactic sugar):

$$I(x) \to I^+(x) \wedge I^-(x) \wedge C_{q_0}(x) \wedge A_0(x), \tag{5}$$

$$I^+(x) \wedge T(x, y) \to I^+(y) \wedge C_\emptyset(y) \wedge A_0(y), \tag{6}$$

$$I^-(y) \wedge T(x, y) \to I^-(x) \wedge C_\emptyset(x) \wedge A_0(x). \tag{7}$$

In these rules C_{q_0} is a unary predicate indicating that the head is pointing to the first cell and the state is q_0. For each other state q in $\mathcal{Q}$ the vocabulary contains the corresponding predicate C_q. The rest of the tape should always be marked by predicates $C_\emptyset$ indicating that the head does not point to this cell. Similarly, if in some configuration a cell contains an alphabet symbol $g \in \Gamma$, then this is indicated by the predicate A_g; for example, the rules above ensure that the tape is initialised by the symbol 0. To ensure the consistency of the computation grid, constructed by means of tape and time predicates T and S, the ontology contains the rules

$$T(x, y) \wedge T(z, u) \wedge S(y, u) \to S(x, z), \tag{8}$$

$$T(x, y) \wedge T(z, u) \wedge S(x, z) \to S(y, u). \tag{9}$$

Finally, we need to make the adjacent configurations consistent. In particular, the content of each cell, as well as the fact that the head is pointing to this cell in some particular state, that is, the cell's C_q and A_g labels, is completely defined by the labels of the three cells in the previous configuration. So, abbreviating $T(x,y) \wedge T(y,z) \wedge S(y,u) \wedge A_{g^-}(x) \wedge A_g(y) \wedge A_{g^+}(z)$ by $\varphi(x,y,z,u)$, the ontology $\mathcal{O}$ contains the rules

$$\begin{array}{ll}
\varphi(x,y,z,u) \wedge C_\emptyset(x) \wedge C_q(y) \wedge C_\emptyset(z) \rightarrow C_\emptyset(u) \wedge A_{g'}(u), & \text{for all } g^-, g^+ \in \Gamma, \text{ if } \delta(g,q) = (g',q',d) \text{ for some } q', d, \\
\varphi(x,y,z,u) \wedge C_q(x) \wedge C_\emptyset(y) \wedge C_\emptyset(z) \rightarrow C_\emptyset(u) \wedge A_g(u), & \text{for all } g, g^+ \in \Gamma, \text{ if } \delta(g^-,q) = (g',q',-) \text{ for some } g', q', \\
\varphi(x,y,z,u) \wedge C_\emptyset(x) \wedge C_\emptyset(y) \wedge C_q(z) \rightarrow C_\emptyset(u) \wedge A_g(u), & \text{for all } g, g^- \in \Gamma, \text{ if } \delta(g^+,q) = (g',q',+) \text{ for some } g', q', \\
\varphi(x,y,z,u) \wedge C_q(x) \wedge C_\emptyset(y) \wedge C_\emptyset(z) \rightarrow C_{q'}(u) \wedge A_g(u), & \text{for all } g, g^+ \in \Gamma, \text{ if } \delta(g^-,q) = (g',q',+) \text{ for some } g', \\
\varphi(x,y,z,u) \wedge C_\emptyset(x) \wedge C_\emptyset(y) \wedge C_q(z) \rightarrow C_{q'}(u) \wedge A_g(u), & \text{for all } g, g^- \in \Gamma, \text{ if } \delta(g^+,q) = (g',q',-) \text{ for some } g', \\
\varphi(x,y,z,u) \wedge C_\emptyset(x) \wedge C_\emptyset(y) \wedge C_\emptyset(z) \rightarrow C_\emptyset(u) \wedge A_g(u), & \text{for all } g^-, g, g^+ \in \Gamma.
\end{array}$$

Having the ontology defined, we complete the construction with specifying the policy. It consists of several BCQs, but the translation to a single CQ by means of several rules in the ontology is straightforward. The policy P guarantees that a cell cannot contain several alphabet symbols, the machine cannot be in several states, and the head cannot simultaneously point and not point to a cell. This is formalised as the following set of BCQs:

$$\begin{array}{ll}
\exists x.\, A_g(x) \wedge A_{g'}(x), & \text{for all } g, g' \in \Gamma \text{ such that } g \neq g', \\
\exists x.\, C_q(x) \wedge C_{q'}(x), & \text{for all } q, q' \in \mathcal{Q} \cup \{\emptyset\} \text{ such that } q \neq q'.
\end{array}$$

Completed the construction, next we formally prove that M has a repeated configuration if and only if $\mathbf{I}_M$ has a (finite) optimal view. We start with forward direction.

($\Rightarrow$)

Let the first pair of repeated configurations of M have numbers m and n, while the smallest (non-positive) number of a cell whose content was changed during the computation is $k+1$, and the biggest (non-negative) such number is $\ell - 1$ (we assume that initially the head is pointing to the cell number 0). Note that k and ℓ are finite, because a computation cannot use infinite number of cells in finite number of steps. In fact, $k > -n$ and $\ell < n$.

The view $\mathcal{V}$ makes use of constants a_{ij} with $-1 \leq i < n$ and $k \leq j \leq \ell$, such that $a_{00} = a$ and all others are anonymous copies of a. By means of binary predicates S and T these constants form a grid, that is the view contains atoms

$$\begin{array}{ll}
S(a_{(i-1)j}, a_{ij}), & \text{for all } 0 \leq i < n, k \leq j \leq \ell, \\
T(a_{i(j-1)}, a_{ij}), & \text{for all } -1 \leq i < n, k < j \leq \ell.
\end{array}$$

The grid is "folded" on all the sides, in the configuration number $i = -1$ and cells number k and ℓ by means of self loops, and on repeated configurations m and n:

$$\begin{array}{ll}
S(a_{(-1)j}, a_{(-1)j}), & \text{for all } k \leq j \leq \ell, \\
T(a_{ik}, a_{ik}), & \text{for all } -1 \leq i < n, \\
T(a_{i\ell}, a_{i\ell}), & \text{for all } -1 \leq i < n, \\
S(a_{(n-1)j}, a_{mj}), & \text{for all } k \leq j \leq \ell.
\end{array}$$

Each configuration with number $0 \leq i < n$ with the word $g_k \dots g_\ell$ written on the part of the tape with cell numbers from k to ℓ, the state q, and the head pointing to the cell number h is represented by means of the following facts:

$$\begin{array}{ll}
A_{g_j}(a_{ij}), & \text{for all } k \leq j \leq \ell, \\
C_q(a_{ih}), & \\
C_\emptyset(a_{ij}), & \text{for all } k \leq j \leq \ell, j \neq h.
\end{array}$$

The auxiliary "configuration" number -1 is the same as a usual configuration with the empty tape, except that the head does not point anywhere:

$$\begin{array}{ll}
A_0(a_{(-1)j}), & \text{for all } k \leq j \leq \ell, \\
C_\emptyset(a_{(-1)j}), & \text{for all } k \leq j \leq \ell, j \neq h.
\end{array}$$

The constant a is in the initialisation predicates:

$$R(a,a), I(a).$$

Finally, each configuration with number $-1 \leq i < n$ (i.e., including the auxiliary one) has cells with numbers k' and ℓ' such that all the cells between k and k', as well as all the cells between ℓ' and ℓ contain 0 and do not have the head pointing on them. the first group is marked by I^- and the second by I^+:

$$\begin{array}{ll}
I^-(a_{(ij)}), & \text{for all } k \leq j \leq k', \\
I^+(a_{(ij)}), & \text{for all } \ell' \leq j \leq \ell.
\end{array}$$

It is straightforward to see that $\mathcal{V} \models \mathcal{O}$ and $\mathcal{O} \cup \mathcal{V} \not\models P$, that is, $\mathcal{V}$ is a confidentiality preserving view for $\mathbf{I}_M$. Also, it is a matter of technicality to check that the view is indeed optimal.

($\Leftarrow$)

Next we show that if the machine M does not have a repeated configuration, then there is no optimal view for the instance $\mathbf{I}_M$. Assume for the sake of contradiction that such a view $\mathcal{V}$ exists. Without loss of generality we may assume that $\mathcal{V} \models \mathcal{O}$. The first fact we need is the following claim.

Claim 29. The view $\mathcal{V}$ contains the atom $I(a)$.

Proof. Whatever is the shape of $\mathcal{V}$, it entails the BCQs

$$Q_i^R = \exists x_1 \dots \exists x_i.\, R(a, x_1) \wedge R(x_1, x_2) \wedge \cdots \wedge R(x_{i-1}, x_i) \text{ for all } i \geq 1.$$

Since i is unbounded, but $\mathcal{V}$ is finite, there exists i_0 such that there is a homomorphism from the body of $Q_{i_0}^R$ to $\mathcal{V}$ which sends different x_j and x_k to the same constant. This means that there is an R-loop of some length in $\mathcal{V}$, which is connected by an R-chain from a. By the rules (2)–(4) this implies that $I(a)$ is a fact in $\mathcal{V}$. □

Similarly to the proof of the claim above, whatever is the shape of $\mathcal{V}$, it entails the BCQs Whatever is the shape of $\mathcal{V}$, it entails the BCQs

$$Q_i^S = \exists x_1 \dots \exists x_i.\, S(a, x_1) \wedge S(x_1, x_2) \wedge \cdots \wedge S(x_{i-1}, x_i) \text{ for all } i \geq 1.$$

Since $\mathcal{V}$ is finite, this implies that there is the (finite) biggest number $n - 1$ such that the body of Q_n^S has a homomorphism to $\mathcal{V}$ which sends different x_j to different constants.

Consider now a "grid" BCQ $Q^{S,T}$ that consists of the following atoms:

$$\begin{array}{ll} S(x_{(i-1)j}, x_{ij}), & \text{for all } 0 < i \leq n, -n \leq j \leq n, \\ T(x_{i(j-1)}, x_{ij}), & \text{for all } 0 \leq i \leq n, -n < j \leq n, \\ x_{00} = a. & \end{array}$$

This query is also "harmless", that is, should be entailed by $\mathcal{V}$ whatever is its shape. Since this BCQ has a chain of S starting from a of length greater than $n - 1$, for any homomorphism from the body of $Q^{S,T}$ to $\mathcal{V}$ there are numbers k and ℓ such that this homomorphism sends x_{k0} and $x_{\ell 0}$ to the same constant. Let h be such a homomorphism, and k, ℓ be the numbers corresponding to h. By rules (8) and (9) we have that $\mathcal{V}$ contains atoms

$$S(x_{(\ell-1)j}, x_{kj}), \quad \text{for all } -n \leq j \leq n. \tag{10}$$

On the other hand, by the fact that $I(a)$ is in $\mathcal{V}$ and the rules (5)–(7) we have that the constants $h(x_{0j})$ for $-n \leq j \leq n$ represent the part of the initial configuration on cells with numbers from $-n$ to n. Furthermore, by means of the rules corresponding to the transition function of the machine, the constants $h(x_{ij})$ form the part of the configuration with number i for all $0 < i < \ell$. By the same rules and atoms (10) we conclude that the constants $h(x_{kj})$ represent not only the part of the configuration number k, but also the part of the configuration number ℓ. If these parts are different, then this discloses the policy, so they are the same. But the rest of the configuration, that is the content of the tape beyond the cells with numbers from $-n$ to n, is also the same for the configurations, because they are just full of symbols 0 (the head cannot reach this part of the tape because it is too far). So, we come to the fact that M has a repeated computation, which contradicts the precondition. □

Proposition 30. *The censor* $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$ *based on a view* $\mathcal{V}$ *is confidentiality preserving for a CQE instance* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ *if and only if* $\mathcal{O} \cup \mathcal{V} \not\models P(\vec{s})$ *for each* $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$*. Additionally, it is optimal if and only if for each CQ* $Q(\vec{x})$ *and each* $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$*, the fact that* $\mathcal{O} \cup \mathcal{V} \cup \{Q(\vec{t})\} \not\models P(\vec{s})$ *for any* $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ *implies that* $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{V})$*.*

Proof. Assume that $\mathcal{O} \cup \mathcal{V} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. Trivially, $\mathcal{O} \cup \mathcal{V} \models \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}}$ and hence we have $\mathcal{O} \cup \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$, as required.

Assume now that cens is confidentiality preserving, in which case $\mathcal{O} \cup \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. Next, assume for the sake of contradiction that $\mathcal{O} \cup \mathcal{V} \models P(\vec{s})$ for some $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. Since $\mathcal{O} \cup \mathcal{D} \models P(\vec{s})$, by the definition of policy we have that $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(P(\vec{s})) = \mathtt{True}$ and thus $P(\vec{s}) \in \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}}$; therefore, $\mathcal{O} \cup \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}} \models P(\vec{s})$, which is a contradiction.

We next focus on the optimality statement. Assume that $\mathcal{O} \cup \mathcal{V} \cup \{Q(\vec{t})\} \not\models P(\vec{s})$ for any $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ implies that $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{V})$, while $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$ is not optimal. Then, there is a confidentiality preserving censor cens that extends $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$; this means that for some CQ $Q(\vec{x})$ and, $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ we have $\vec{t} \in \mathsf{cens}(Q)$, but $\vec{t} \notin \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q)$. The fact that $\vec{t} \notin \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q)$ and $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ implies that $\vec{t} \notin \mathsf{cert}(Q, \mathcal{O}, \mathcal{V})$. Furthermore, the fact that cens is confidentiality-preserving implies that $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \cup \{Q(\vec{t})\} \not\models P(\vec{s})$ for any $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. But then, since cens extends $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$, we have that $\mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}} \subseteq \mathsf{Th}_{\mathsf{cens}}$ and hence $\mathcal{O} \cup \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}} \cup \{Q(\vec{t})\} \not\models P(\vec{s})$, and therefore $\vec{t} \in \mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}(Q)$, which is a contradiction.

Finally, assume that there exists some CQ $Q(\vec{x})$ and $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ such that $\mathcal{O} \cup \mathcal{V} \cup \{Q(\vec{t})\} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$, but $\mathcal{O} \cup \mathcal{V} \not\models Q(\vec{t})$. Then, we can define a censor cens that behaves exactly like $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$, with the exception of answering $Q(\vec{t})$ positively. Thus, $\mathsf{Th}_{\mathsf{cens}} = \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}} \cup \{Q(\vec{t})\}$. But then, since $\mathcal{O} \cup \mathcal{V} \cup \{Q(\vec{t})\} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ and $\mathcal{O} \cup \mathcal{V} \models \mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}}$ we have that $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \not\models P(\vec{s})$, which implies that cens is confidentiality preserving and $\mathsf{Th}_{\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}}$ is not optimal, as required. □

We say that a rule is *normalised* if it has at most two atoms in its body; an ontology is *normalised* if it is a set of normalised rules. Clearly, any guarded ontology can be normalised.

Definition 31. *Let Σ be a signature, $\mathcal{O}$ an ontology over Σ, and a subset S of Σ is a set of unary predicates. S is* closed under *$\mathcal{O}$ if (i) $\mathcal{O} \cup \{A(x) \mid A \in S\} \models C(x)$ implies that $C \in S$ and (ii) if A does not occur in $\mathcal{O}$, then $A \in S$.*

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.

Proof.
Guarded, tree-shaped CQE instance. Algorithm 1 presents a procedure that builds a view for a given CQE instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$. We are going to show that if $\mathbf{I}$ is tree-shaped and guarded, then the algorithm returns an optimal view for $\mathbf{I}$. By its construction, the constructed dataset $\mathcal{V}$ is safe, so it remains to prove its optimality. Due to Proposition 30, it suffices to show that for each CQ Q and a tuple $\vec{t}$ such that $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$:

$$\text{if } \mathcal{O} \cup \mathcal{V} \cup [Q(\vec{t})] \not\models [P(\vec{s})] \text{ for each } \vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D}), \text{ then } \vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{V}). \tag{11}$$

Observe the following.

(O1) W.l.g. we can assume that $\mathcal{V} \cap \mathcal{H}_{\mathcal{O},\mathcal{D}} \subseteq [Q(\vec{t})]$.

(O2) If $\mathcal{H}$ is as defined in Algorithm 1, $\mathcal{H}_{\mathcal{O},\mathcal{D}} \subseteq \mathcal{H}$ and $\mathcal{H} \setminus \mathcal{H}_{\mathcal{O},\mathcal{D}}$ consists of unary atoms only over fresh predicates introduced into $\mathcal{O}_E$ at Line 1.

(O3) No rule of $\mathcal{O}_E$ can be applied to $\mathcal{V}$.

Assume that $Q(\vec{t})$ satisfies the "if"-clause of Equation (11). Since by the assumption $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$, then there is a homomorphism h from $\mathcal{H}_{\mathcal{O},[Q(\vec{t})]}$ into $\mathcal{H}_{\mathcal{O},\mathcal{D}}$. It is easy to see that

$$h : \mathcal{H}_{\mathcal{O},[Q(\vec{t})]} \to \mathcal{H}_{\mathcal{O},\mathcal{D}} \text{ iff } h : \mathcal{H}_{\mathcal{O}_E,[Q(\vec{t})]} \to \mathcal{H}_{\mathcal{O}_E,\mathcal{D}}.$$

We are going to use the following notations.

- Denote $\mathcal{H}_{\mathcal{O}_E,[Q(\vec{t})]}$ as $\mathcal{B}$.
- Let $\mathcal{X}$ be a dataset and d an element occurring in $\mathcal{X}$. Then we define the set $\mathtt{conc}_{\mathcal{X}}(d)$ as $\{A \mid A(d) \in \mathcal{X}\}$.

We are going to show the existence of a homomorphism $g : \mathcal{B} \to \mathcal{V}$, which would prove that Q satisfies the "then"-clause of Equation (11). Let $d_1, \ldots, d_m$ be all the fresh constants from $[Q(\vec{t})]$, let d be an element from $[Q(\vec{t})]$, and let h be a homomorphism from $\mathcal{B}$ into $\mathcal{H}$. We claim that there exists g that satisfies the following properties:
1. If d is from $\mathcal{O} \cup \mathcal{D}$, then $g(d) = d$.
2. Let $d = d_i$ and $h(d_i) = a$. Then $g(d_i) = a'$ such that $a' \in \sigma_a$, $a' \neq a$ and $\mathtt{conc}_{\mathcal{S}}(a') = \mathtt{conc}_{\mathcal{B}}(d)$, where σ_a is a set of all "copies" of a introduced by the algorithm (for example, see sub-routines in Algorithm 2).

It remains to show that g does indeed exist and map $\mathcal{B}$ into $\mathcal{V}$. To this end, we need to show that
1. for each element d from $[Q(\vec{t})]$, there is an element a' in $\mathcal{V}$ satisfying the second property of g, and
2. for each binary atom $R(d_1, d_2) \in [Q(\vec{t})]$, there exists a corresponding binary atom $R(g(d_1), g(d_2)) \in \mathcal{V}$.

The former requirement follows from the construction of $\mathcal{V}$. The latter one requires that

$$\mathsf{cert}(P, \mathcal{O}_E, \mathcal{V} \cup g(\mathcal{B})) = \emptyset. \tag{12}$$

Note that Equation 11 implies that $\mathsf{cert}(P, \mathcal{O}_E, \mathcal{V} \cup \mathcal{B}) = \emptyset$. Also observe that *no rule from $\mathcal{O}_E$ is applicable to $\mathcal{V} \cup \mathcal{B}$*. Indeed, no rule is applicable to $\mathcal{V}$ nor to $\mathcal{B}$ by construction. Assume that a rule r is applicable to $\mathcal{V} \cup \mathcal{B}$. If the body of r contains one atom, then we immediately obtain a contradiction. If the body of r contains two atoms then there exist an atom $f_1 \in \mathcal{V}$ and an atom $f_2 \in \mathcal{B}$ such that $f_1 \wedge f_2$ is an instantiation of the body of r. Assume that the atom in the body of r corresponding to f_1 is a guard of the rule; then all constants occurring in f_2 occur in f_1 too. Since $\mathcal{B}$ and $\mathcal{V}$ share only "active" constants (i.e., the ones from $\mathbf{I}$), we have that $f_2 \in \mathcal{V} \cap \mathcal{B}$ (due to Observation (O1)), and thus r is applicable to $\mathcal{V}$, which gives a contradiction.

Assume that Equation 12 does not hold. Hence, there is a rule $r \in \mathcal{O}$ applicable to $\mathcal{V} \cup g(\mathcal{B})$. Recall that r is not applicable to $\mathcal{V}$. We have the following cases depending on the shape of r.

1. r is of the form $A(x) \to C(x)$, $A(x) \wedge B(x) \to C(x)$, or $A \wedge B(x) \to C(x)$. Clearly, in this case r is applicable to $g(\mathcal{B})$. It is easy to see that r is then applicable to $\mathcal{B}$ since $\mathtt{conc}_{\mathcal{B}}(d) = \mathtt{conc}_{\mathcal{V}}(g(d))$ for every d in $\mathcal{B}$, which contradicts the observation above.

2. r is of the form $R(x,y) \to \mathit{Head}(\vec{x})$ or $A \wedge R(x,y) \to \mathit{Head}(\vec{x})$, where $\mathit{Head}(\vec{x})$ is of one of the following forms for some unary C of binary Q predicate: $C(x)$, $C(y)$, $Q(x,y)$, or $Q(y,x)$. Here we obtain a contradiction similarly to the previous case.
3. r is of the form $R(x,y) \wedge A(x) \to \mathit{Head}(\vec{x})$. There are three cases.
   (a) There are a, b, and d_i such that $R(a,b) \in \mathcal{V}$ and $A(d_i) \in \mathcal{B}$, where $g(d_i) = a$. Since r is not applicable to $\mathcal{B}$, then for any element c occurring in $\mathcal{B}$, it is the case that $R(d_i, c) \notin \mathcal{B}$. Thus, $\delta_R(d_i) \notin \mathcal{B}$ and consequently $\delta_R(a) \notin \mathcal{V}$. The latter statement contradicts the assumption that $R(a,b) \in \mathcal{V}$.
   (b) There are a, b', and d_i such that $R(d_i, b') \in \mathcal{B}$ and $A(a) \in \mathcal{V}$, where $g(d_i) = a$. Since r is not applicable to $\mathcal{B}$, then $A(d_i) \notin \mathcal{B}$ and thus $A(g(d_i)) \notin \mathcal{V}$. This contradicts that $A(a) \in \mathcal{V}$.
   (c) there are a, b, b', d_i, and d_j such that $R(d_i, b') \in \mathcal{B}$, $A(d_j) \in \mathcal{B}$, $g(d_i) = g(d_j) = a$, and $g(b') = b$. Then we conclude that $\mathtt{conc}_{\mathcal{B}}(d_i) = \mathtt{conc}_{\mathcal{B}}(d_j)$ and consequently $A(d_i) \in \mathcal{B}$. If $\mathit{Head}(d_i, b')$ is equal to $C(d_i)$ or $C(b')$ for some unary predicate C, then $C \in \mathtt{conc}_{\mathcal{B}}(d_i)$ or $C \in \mathtt{conc}_{\mathcal{B}}(b')$, respectively, and thus $C(g(d_i)) \in \mathcal{V}$ or $C(g(b')) \in \mathcal{V}$, respectively. If $\mathit{Head}(d_i, b')$ is equal to $Q(d_i, b')$ for some binary predicate Q, then $\delta_Q \in \mathtt{conc}_{\mathcal{B}}(d_i)$ and $\rho_Q \in \mathtt{conc}_{\mathcal{B}}(b')$, and thus $\delta_Q(g(d_i))$ and $\rho_Q(g(b'))$ are in $\mathcal{V}$; therefore, `CheckRole` sub-routine of the algorithm would return `True` on input $(Q(g(d_i), g(b')), \mathcal{V})$, and thus $Q(g(d_i), g(b')) \in \mathcal{V}$. Anyway, the obtained contradictions conclude the case.
4. r is of the form $R(x,y) \wedge A(y) \to \mathit{Head}(\vec{x})$. This case is analogous to the previous one.

Finally, $g(f)$ should be in $\mathcal{V}$ for each binary atom $f \in \mathcal{B}$, since *(i)* Equation (12) holds and *(ii)* binary atoms that do not discover the policy were exhaustively added to $\mathcal{V}$.

Regarding the size of the $\mathcal{V}$, if a is a constant occurring in $\mathbf{I}$ and C a set of unary predicates A such that $A(a) \in \mathcal{H}$, then the number of "copies" of a added by the algorithm is equal to a number of subsets of C closed under $\mathcal{O}_E$ (see Algorithm 2). Clearly, this number is exponential in $|\mathcal{O}|$ and polynomial in $|\mathcal{D}|$ (see Definition 31).

Multi-linear, tree-shaped CQE instance. Let a DPI $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ be such that $\mathcal{O}$ is multi-linear Datalog. Let $\mathcal{V}$ be a dataset returned by Algorithm 1. For every constant a, the set σ_a contains the constant $a_{\mathcal{A}^*}$ such that $\mathcal{A}^*$ is a maximal subset of $\{A \mid A(a) \in \mathcal{H}_{\mathcal{O},\mathcal{D}}\}$ closed under $\mathcal{O}_E$. It is easy to check that the number of such subsets is polynomial in the size of $\mathcal{O}$. The set $\mathcal{A}^*$ is a maximal set of labels (i.e., unary predicates) among all constants in σ_a, i.e., if $a' \in \sigma_a$, then $\{A \mid A(a') \in \mathcal{V}\} \subseteq \mathcal{A}^*$ for some $\mathcal{A}^*$. We will also denote as a^* an element of σ_a such that $\mathtt{conc}_{\mathcal{V}}(a^*) = \mathcal{A}^*$.

Let b be a constant from $\mathbf{I}$ and let a' be from σ_a such that $R(a,b)$ is in $\mathcal{H}_{\mathcal{O}_E,\mathcal{D}}$. Since $\mathbf{I}$ is multi-linear, $\mathcal{O}$ does not include rules with bodies of the form $R(x,y) \wedge A(x)$ and thus whatever unary atoms a' participates in, they cannot affect the atoms b participates in. Hence we conclude that *(i)* if $R(a', b)$ is in $\mathcal{V}$ for some $a' \in \sigma_a$ and b from $\mathbf{I}$, then so is $R(a^*, b)$ for a corresponding element a^* from σ_a; *(ii)* if $R(a', b')$ is in $\mathcal{V}$ for some $a' \in \sigma_a$ and $b' \in \sigma_b$, then so is $R(a^*, b^*)$ for corresponding elements a^* and b^* from σ_a and σ_b, respectively. Let $\mathcal{V}^*$ be a subset of $\mathcal{V}$ which is based on constants a from $\mathbf{I}$ and their copies a^*. Clearly if, for some CQ $Q(\vec{x})$, $\vec{a} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ and $\vec{a} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{V})$, then $\vec{a} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{V}^*)$, which proves optimality of $\mathcal{V}^*$.

The polynomial size of $\mathcal{V}^*$ follows from the observation that the sub-routine `AddUnPredicates` introduces only linearly many copies of a constant a for each set of labels, including $\mathcal{A}^*$.

Linear, tree-shaped CQE instance. Finally, assume that $\mathcal{O}$ is linear. Then, there is the unique maximal subset $\mathcal{V}_0$ of $\mathcal{H}_{\mathcal{O}_E,\mathcal{D}}$ such that $\mathsf{cert}(P, \mathcal{O}_E, \mathcal{V}_0) = \emptyset$, which gives the uniqueness of $\mathcal{V}$. □

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 $\mathbf{I}$.*

Proof. First we show the confidentiality preservation of the censor. Since $\mathsf{vcens}_{\mathbf{I}'}^{\mathcal{V}'}$ is confidentiality-preserving, we have that $\mathcal{O}' \cup \mathcal{V}' \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. Since $\mathcal{O}'$ is Datalog, it is clear that $\mathcal{H}_{\mathcal{O}',\mathcal{V}} = \mathcal{H}_{\mathcal{O}',\mathcal{V}'}$; thus, $\mathcal{O}' \cup \mathcal{V} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. But then, since P is tree-shaped and $\mathcal{O}'$ is a rewriting of $\mathcal{O}$ we have $\mathcal{O} \cup \mathcal{V} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ (see [Stefanoni *et al.*, 2013]), as required.

Now we concentrate on the optimality of the view. Assume by contradiction that $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$ is not optimal, then, by Proposition 30, there exists a BCQ Q such that *(i)* $\mathcal{O} \cup \mathcal{D} \models Q$; *(ii)* $\mathcal{O} \cup \mathcal{V} \not\models Q$; and *(iii)* $\mathcal{O} \cup \mathcal{V} \cup \{Q\} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. Since $\mathcal{O} \cup \mathcal{D} \models Q$ and $\mathcal{O}' \models \mathcal{O}$ we have *(iv)* $\mathcal{O}' \cup \mathcal{D} \models Q$. Furthermore, condition *(iii)* implies that $\mathcal{O} \cup \mathcal{V} \cup [Q] \not\models P(\vec{s})$ and since P is tree-shaped and $\mathcal{O}'$ is a rewriting of $\mathcal{O}$ we have $\mathcal{O}' \cup \mathcal{V} \cup [Q] \not\models P(\vec{s})$, which by the fact that $\mathcal{V} \models \mathcal{V}'$ then also implies that *(v)* $\mathcal{O}' \cup \mathcal{V}' \cup \{Q\} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. But then, *(iv)* and *(v)* and the fact that $\mathcal{V}'$ is optimal for $\mathbf{I}'$ we must have $\mathcal{O}' \cup \mathcal{V}' \models Q$. Since $\mathcal{V} = \mathcal{H}_{\mathcal{O}',\mathcal{V}'}$ we have $\mathcal{V} \models Q$, which contradicts *(ii)*. □

A.3 Proofs for Section 6

For the sake of ease in the proofs for theorems and propositions of this section we will consider only the class of BCQs with constants. Clearly, any results obtained for this class will also hold for the class of all CQs. Before proceeding to the main proofs, we introduce few definitions and lemmas.

Let $\mathcal{O}$ be a Datalog ontology and $\mathcal{D}$ a dataset; let $\mathbb{Q}'$ be a possibly infinite set of queries such that $\mathcal{O} \cup \mathcal{D} \models Q$ for each $Q \in \mathbb{Q}'$. Then a censor $\mathsf{cens}_{\mathbb{Q}'}$ is defined as follows:

$$\mathsf{cens}_{\mathbb{Q}'}(Q) = \mathtt{True} \quad \text{iff} \quad \mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) = \mathtt{True} \text{ and } [Q] \not\models Q' \text{ for each } Q' \in \mathbb{Q}'.$$

Lemma 32. *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ be a CQE instance; let Υ be a pseudo-obstruction based on a subset $\mathbb{S}$ of $\mathbb{Q}$. Then,* $\mathsf{cens}_\Upsilon = \mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$.

Proof. Let Q be a CQ such that $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) = \mathtt{True}$.

Assume that $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}(Q) = \mathtt{False}$; this yields that $[Q] \models Q'$ for some $Q' \in \mathbb{Q} \setminus \mathbb{S}$. Then there exists $Q'' \in \Upsilon$ such that $Q' \models Q''$ and thus $[Q] \models Q''$, i.e., $\mathsf{cens}_\Upsilon(Q) = \mathtt{False}$.

Assume that $\mathsf{cens}_\Upsilon(Q) = \mathtt{False}$; this yields that $[Q] \models Q''$ for some $Q'' \in \Upsilon$. Note that $Q'' \in \mathbb{Q} \setminus \mathbb{S}$ since $\Upsilon \subseteq \mathbb{Q} \setminus \mathbb{S}$ and thus $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}(Q) = \mathtt{False}$. □

The lemma above allows us to speak of obstruction censors in terms of either Υ or $\mathbb{Q} \setminus \mathbb{S}$, whatever way is more convenient to show the required results. We are going to show now that a censor cens is optimal for a given CQE instance $\mathbf{I}$ iff there exists a maximal subset $\mathbb{S}$ of $\mathbb{Q}$ such that $\mathsf{cens} = \mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$. But first we need the following notion of a normalised proof.

Definition 33. *Let $\mathcal{O}$ be a Datalog ontology, $\mathcal{D}$ a dataset, and G_0 a goal. A proof π of length n of G_0 in $\mathcal{O} \cup \mathcal{D}$ is* normalised *if there is $k \leq n$ such that $r_i \in \mathcal{O}$ for each $i < k$ and $r_j \in \mathcal{D}$ for each $j \geq k$. Moreover, the number k is called the* frontier *of π, denoted* $\mathsf{fr}(\pi)$.

Intuitively, a normalised proof π works as follows: first we rewrite the initial query G_0 over the ontology $\mathcal{O}$ until we obtain the query $G_{\mathsf{fr}(\pi)-1}$ that can be mapped into $\mathcal{D}$, and then we perform such a mapping applying (r_i, θ_i) with $i \geq \mathsf{fr}(\pi)$. Observe that for every G_i with $i < \mathsf{fr}(\pi)$ it holds that $\mathcal{O} \cup G_i \models G_0$.

We exploit the following known result about SLD resolution over Datalog ontologies.

Lemma 34. *Let $\mathcal{O}$ be a Datalog ontology, let $\mathcal{D}$ be a dataset, and let G_0 be a goal such that $\mathcal{O} \cup \mathcal{D} \models G_0$. Then there exists a normalised SLD proof π of G_0 in $\mathcal{O} \cup \mathcal{D}$.*

Lemma 35. *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ be a CQE instance with $\mathcal{O}$ a Datalog ontology and* cens *a censor for $\mathcal{O}$ and $\mathcal{D}$. Then* cens *is optimal for $\mathbf{I}$ iff there exists a maximal subset $\mathbb{S}$ of $\mathbb{Q}$ such that (i) $\mathcal{O} \cup \mathbb{S} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ and (ii)* $\mathsf{cens} = \mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$.

Proof. We start with the "only if"-direction. Let us assume that such maximal subset $\mathbb{S}$ exists. We show that $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$ is optimal.

First, we show that $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$ is confidentiality preserving. Assume the contrary; then, there is a (finite) subset $\mathbb{F}$ of $\mathsf{Th}_{\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}}$ such that $\mathcal{O} \cup \mathbb{F} \models P(\vec{s})$ for some $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. This yields the existence of proof π of $P(\vec{s})$ in $\mathcal{O} \cup [\mathbb{F}]$, where $[\mathbb{F}] = \bigcup_{Q \in \mathbb{F}} [Q]$. Due to Lemma 34, we can assume that π is normalised with frontier $k+1$. Let G_k be the goal right before frontier in π. Since π is normalised, then G_k is proved by using only facts from $[\mathbb{F}]$. So, we can write G_k as $G_k = B_1 \wedge \ldots \wedge B_m$, where each B_j is the conjunction of all atoms that are proved using facts only from a particular $[Q_j]$. Obviously, the order in which these B_j are proved is irrelevant, so let us assume that all B_j have been proved except for B_i; since, the different B_j can share variables, the remaining goal to prove may not be just B_i, but rather $B_i\theta_i$, with θ_i some substitution. We make the following observations:

1. $B_i\theta_i$ does not mention any constants not in $\mathcal{O} \cup \mathcal{D}$. Indeed, for any distinct queries Q_k, Q_j in $\mathbb{F}$ we have that $[Q_k]$ and $[Q_j]$ only share constants from $\mathcal{O} \cup \mathcal{D}$ $[Q_k]$; thus, if $B_i\theta_i$ contains some constant coming from $[Q_j]$ with $j \neq i$, it would not be possible to prove $B_i\theta_i$ using only facts from $[Q_i]$.
2. There exists a proof of $P(\vec{s})$ in $\mathcal{O} \cup \mathcal{D}$ such that $B_i\theta_i$ occurs as a subgoal. We construct such proof as follows. First, we can "reach" goal G_k because it only requires rules from $\mathcal{O}$. Note also that each B_j follows from $\mathcal{O} \cup \mathcal{D}$, so we can continue the proof by showing all B_j except for B_i. Then, we can do it in such a way we reach precisely $B_i\theta_i$ as a subgoal.
3. $Q_i \models \exists^* B_i\theta_i$ since $B_i\theta_i$ is provable from $[Q_i]$.

Observation 2 means that $B_i\theta_i \in \mathbb{Q}$ for all $1 \leq i \leq m$. Furthermore, since the censor answers $\mathtt{True}$ for each Q_i we have that $B_i\theta_i \in \mathbb{S}$. But then, $\mathcal{O} \cup \mathbb{S} \models P(\vec{s})$, which is a contradiction.

Now we show the optimality of $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$. Clearly, a censor cens for $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ is optimal if and only if for each CQ $Q(\vec{x})$ and each $\vec{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ the fact that $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \cup \{Q(\vec{t})\} \not\models P(\vec{s})$ holds for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ implies that $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \models Q(\vec{t})$. Due to this, $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$ is optimal if and only if for each Q such that $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) = \mathtt{True}$ and $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}} \cup \{Q\} \not\models P(\vec{s})$, it holds that $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}} \models Q$. Assume to the contrary that there exists a CQ Q such that

$\mathsf{cert}(Q, \mathcal{O}, \mathcal{D}) = \mathtt{True}$ and $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}} \cup \{Q\} \not\models P(\vec{s})$, but $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}} \not\models Q$. The latter means that $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}(Q) = \mathtt{False}$, that is, $[Q] \models Q'$, for some $Q' \in \mathbb{Q} \setminus \mathbb{S}$. Recall that for any $Q \in \mathbb{Q} \setminus \mathbb{S}$ it holds that $\mathcal{O} \cup \mathbb{S} \cup \{Q\} \models P(\vec{s})$ due to maximality of $\mathbb{S}$. Observe that $\mathbb{S} \subseteq \mathsf{Th}_{\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}}$; this yields $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}} \cup \{Q\} \models P(\vec{s})$, which contradicts the initial assumption and concludes the "only if"-direction.

Now we consider the "if"-direction. Let us now assume that cens is optimal, and let $\mathbb{Q}' = \{Q \mid \mathsf{cens}(Q) = \mathtt{False}\}$. Consider the following subset $\mathbb{S}$ of $\mathbb{Q}$: $\mathbb{S} = \mathbb{Q} \setminus \mathbb{Q}'$. To prove the "if"-direction, it suffices to prove the following two conditions: *(i)* $\mathbb{S}$ is a maximal subset of $\mathbb{Q}$ such that $\mathcal{O} \cup \mathbb{S} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ and *(ii)* $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}} = \mathsf{cens}$.

To show *(i)*, assume that $\mathcal{O} \cup \mathbb{S} \cup \{Q\} \models P(\vec{s})$ for some $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ and some $Q \in \mathbb{Q}$. Clearly, since by construction $\mathbb{S} \subseteq \mathsf{Th}_{\mathsf{cens}}$, it holds that $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \cup \{Q\} \models P(\vec{s})$, and therefore $\mathsf{cens}(Q) = \mathtt{False}$, i.e. $Q \in \mathbb{Q}'$, which implies *(i)*.

To show *(ii)*, let us pick an arbitrary Q such that $\mathcal{O} \cup \mathcal{D} \models Q$ but $\mathsf{cens}(Q) = \mathtt{False}$ and hence $Q \in \mathbb{Q}'$. Since cens is optimal, we have that $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \cup \{Q\} \models P(\vec{s})$ for some $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$, so let $\mathbb{F}$ be any minimal subset of $\mathsf{Th}_{\mathsf{cens}}$ such that $\mathcal{O} \cup \mathbb{F} \cup \{Q\} \models P(\vec{s})$. Following the same arguments as we used in the "only if" direction we have that there exists $G \in \mathbb{Q} \setminus \mathbb{S}$ such that $Q \models \exists^* G$; since $\exists^* G$ is part of the obstruction, then $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}(Q) = \mathtt{False}$. Finally, assume that $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}(Q) = \mathtt{False}$; then, $Q \models \exists^* G$ for some $G \in \mathbb{Q} \setminus \mathbb{S}$. Since $\mathbb{Q} \setminus \mathbb{S} \subseteq \mathbb{Q}$, we have that $\mathsf{cens}(Q) = \mathtt{False}$, as required. □

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.*

Proof. Let us prove Statement 1. Assume that Υ is a finite pseudo-obstruction. By Lemma 32, we have that $\mathsf{cens}_\Upsilon = \mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$. By the "only if" statement in Lemma 35, we have that $\mathsf{cens}_{\mathbb{Q}\setminus\mathbb{S}}$ is optimal. But then, since Υ is finite, then U is an obstruction.

Next, we show Statement 2. Assume by contradiction that each pseudo-obstruction is infinite, but there is an optimal censor based on an obstruction U. Since $\mathsf{ocens}_{\mathbf{I}}^U$ is an optimal censor, then the "if" direction of Lemma 35 tells us that there exists a pseudo-obstruction Υ such that $\mathsf{ocens}_{\mathbf{I}}^U = \mathsf{cens}_\Upsilon$. We can show that then there exists a finite pseudo-obstruction which contradicts the assumption above. Pick any CQ Q from U; then, clearly, $\mathsf{ocens}_{\mathbf{I}}^U(Q) = \mathtt{False}$ and hence $\mathsf{cens}_\Upsilon(Q) = \mathtt{False}$. The latter implies that there exists $Q' \in \Upsilon$ such that $Q \models Q'$. Let us now construct $U' = \bigvee_{Q \in U} Q'$, which is finite and also a "subset" of Υ. To obtain a contradiction, it thus suffices to show now that $\mathsf{ocens}_{\mathbf{I}}^{U'} = \mathsf{cens}_\Upsilon$. Indeed, for each CQ Q such that $\mathsf{cert}(Q, \mathcal{D}, \mathcal{O}) = \mathtt{True}$ (recall that $\mathsf{ocens}_{\mathbf{I}}^U = \mathsf{cens}_\Upsilon$):

- Assume that $\mathsf{ocens}_{\mathbf{I}}^U(Q) = \mathtt{False}$; then there is Q' in U such that $[Q] \models Q'$, which yields $[Q] \models Q''$ with Q'' from U', and therefore $\mathsf{ocens}_{\mathbf{I}}^{U'}(Q) = \mathtt{False}$.
- Assume that $\mathsf{ocens}_{\mathbf{I}}^{U'}(Q) = \mathtt{False}$; then $[Q] \models Q''$ for some Q'' in U', and consequently, since $Q'' \in \mathbb{Q} \setminus \mathbb{S}$, we conclude that $\mathsf{cens}_\Upsilon(Q) = \mathtt{False}$.

The obtained contradiction concludes the proof. □

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.

Proof. To show the first statement, consider $\mathbf{I}_1 = (\mathcal{O}_1, \mathcal{D}_1, P_1)$, where $\mathcal{D}_1 = \{R(a,a), A(a)\}$, $P_1 = A(a)$, and the guarded RL (and EL) ontology $\mathcal{O}_1 = \{R(x,y) \wedge A(y) \to A(x)\}$. Since this CQE instance is guarded and tree-shaped, by Theorem 14 we can devise an optimal view. No optimal obstruction, however, exists, which is shown in Example 20.

To show the second statement, consider CQE instance $\mathbf{I}_2 = (\mathcal{O}_2, \mathcal{D}_2, P_2)$, with $\mathcal{D}_2 = \{R(a,a)\}$, $P_2 = A(a)$, and $\mathcal{O}_2 = \{R(x_1, y) \wedge R(x_2, y) \to x_1 \approx x_2, R(x,y) \to A(y)\}$. From [Cuenca Grau *et al.*, 2013] we know that no optimal view exists for this instance, and the proof can be easily extended to our framework (note that our notion of a censor $\mathsf{vcens}_{\mathbf{I}}^{\mathcal{V}}$ based on a view $\mathcal{V}$ differs from the one in [Cuenca Grau *et al.*, 2013]) extends also to the case where views are not required to be sound. However, $U = A(a) \vee \exists x.\, R(x,a)$ is an optimal obstruction, since there is only one proof of $A(a)$ with subgoal $R(x,a)$. □

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 $\mathsf{ocens}_{\mathbf{I}}^U$ *is the unique optimal censor for* $\mathbf{I}$*.*

Proof. Optimality and uniqueness follows from Theorem 22 and the facts that *(i)* the set S is exactly $\mathbb{Q}$ *(ii)* the only maximal subset $\mathbb{S}$ of $\mathbb{Q}$ such that $\mathcal{O} \cup \mathbb{S}$ does not entail any $P(\vec{s})$ is the empty set. To prove the former fact, first observe that any goal that can appear in any SLD proof in $\mathcal{O} \cup \mathcal{D}$ is isomorphic to one of the nodes of the proof-graph of $\mathcal{O} \cup \mathcal{D}$; then Fact (i) follows directly from the construction of the proof-graph. Fact (ii) follows from the observation that each SLD proof in case of linear $\mathcal{O}$ is normalised, and therefore for each $Q \in S$ it holds that $\mathcal{O} \cup Q \models P(\vec{s})$ for some $\vec{s} \in \mathsf{cert}(\mathcal{P}, \mathcal{O}, \mathcal{D})$.

Finally, polynomiality follows from the fact that in linear Datalog the size of the proof-graph is at most cubic in $|\mathcal{O} \cup \mathcal{D}|$. □

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

Proof. Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$ be a CQE instance with $\mathcal{O}$ in QL. Let cens' be the optimal censor for $\mathbf{I}' = (\Xi_\sigma(\mathcal{O}), \mathcal{D}, P)$, where σ is a set of constants of $\mathbf{I}$ and $\Xi_\sigma(\mathcal{O})$ is a linear Datalog ontology. By Theorem 25, $\mathsf{cens}' = \mathsf{ocens}^U_{\mathbf{I}'}$ for the UCQ U as defined in the theorem. Let $\mathsf{cens} = \mathsf{ocens}^U_{\mathbf{I}}$. We are going to show that cens is an optimal censor for $\mathbf{I}$.

Confidentiality preservation. Assume that cens is not confidentiality preserving for $\mathbf{I}$, that is, $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \models P(\vec{s})$ for some $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. This means that there exist $Q_1, \ldots, Q_n \in \mathsf{Th}_{\mathsf{cens}}$ such that $\mathcal{O} \cup \{Q_1, \ldots, Q_n\} \models P(\vec{s})$; clearly, $\mathcal{O} \cup \mathcal{D} \models Q_i$ for each $i \in \{1, \ldots, n\}$. By Proposition 18, $\Xi_\sigma(\mathcal{O}) \models \mathcal{O}$ and consequently $\Xi_\sigma(\mathcal{O}) \cup \mathcal{D} \models Q_i$ for each $i \in \{1, \ldots, n\}$. Since cens' is confidentiality preserving for $\mathbf{I}'$, we conclude that $\{Q_1, \ldots, Q_n\} \not\subseteq \mathsf{Th}_{\mathsf{cens}'}$, so there is $j \in \{1, \ldots, n\}$ such that $\mathsf{cens}'(Q_j) = \mathtt{False}$; i.e., $[Q_i] \models U$. The last entailment implies that $\mathsf{cens}(Q_j) = \mathtt{False}$, i.e., $Q_j \notin \mathsf{Th}_{\mathsf{cens}}$, which yields a contradiction and thus cens is confidentiality preserving for $\mathbf{I}$.

Optimality. Assume, for the sake of getting a contradiction, that cens is not optimal for $\mathbf{I}$, that is, there exists Q such that *(i)* $\mathcal{O} \cup \mathcal{D} \models Q$, *(ii)* $Q \notin \mathsf{Th}_{\mathsf{cens}}$, and *(iii)* $\mathcal{O} \cup \mathsf{Th}_{\mathsf{cens}} \cup \{Q\} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$. This yields $[Q] \models u$ for some disjunct u in U and consequently $\mathsf{cens}'(Q) = \mathtt{False}$. Note that for each disjunct u in U, it holds that $\Xi_\sigma(\mathcal{O}) \cup \{u\} \models P(\vec{s})$ for some $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$; thus $\Xi_\sigma(\mathcal{O}) \cup \{Q\} \models P(\vec{s})$. There are the following cases depending on the form of u.

- If u is of the form $A(a)$ or $R(a, b)$ with $a, b \in \sigma$, then $\mathcal{O} \cup \{u\} \models P(\vec{s})$ since, due to Proposition 18, $\Xi_\sigma(\mathcal{O})$ is a σ-rewriting of $\mathcal{O}$; thus, $\mathcal{O} \cup \{Q\} \models P(\vec{s})$ which yields a contradiction with *(iii)*.
- If u is of the form $\exists y.R(a, y)$ with $a \in \sigma$, then let $\mathcal{O}_{\min}$ be a minimal subset of $\Xi_\sigma(\mathcal{O})$ such that $\mathcal{O}_{\min} \cup \{u\} \models P(\vec{s})$. Due to the assumption, it holds $\mathcal{O} \cup \{u\} \not\models P(\vec{s})$; thus, $\mathcal{O}_{\min} \not\subseteq \mathcal{O}$ and therefore $\mathcal{O}_{\min}$ includes one of the rules introduced by Ξ. That is, $\mathcal{O}_{\min}$ contains (some of) the following rules that come from the Skolemisation $\Xi_\sigma(r)$ of some rule $r = A(x) \to \exists y.[S(x, y) \wedge B(y)]$ of Type (3) in $\mathcal{O}$:

  $$A(x) \to P_S(x, c_{A,S}), \quad P_S(x, y) \to S(x, y), \quad \text{and } P_S(x, y) \to B(y). \tag{13}$$

  Consider a proof $\pi = G_0 \to \ldots \to G_n$ of $P(\vec{s})$ in $\Xi_\sigma(\mathcal{O}) \cup [\exists y.R(a, y)]$, where $G_0 = P(\vec{s})$. Clearly, G_i can be obtained from G_{i-1} by applying a rule from $\mathcal{O}_{\min}$ for each $i = 1, \ldots, n-1$, and $G_{n-1} = R(a, x')$ for some x' since the last step of the proof is applying the only rule from $[\exists y.R(a, y)]$. Let G_k be the first goal in π obtained from G_{k-1} by applying a rule from Equation (13); clearly, $\mathcal{O} \cup \{\exists^* G_{k-1}\} \models \exists^* G_0$. We have the following cases.

  - Assume that we apply the third rule from Equality (13) to $G_{k-1} = B(b)$ for some constant b (note that a goal $B(x)$ with x a Skolem constant cannot appear by applying QL rules except for Type (3)). Then $G_k = P_S(x, b)$, and the only rule that has P_S in its head is the first one from Equality (13); however, this rule cannot be applied to G_k since we cannot unify b and $c_{A,S}$. Thus, this case is invalid.
  - Assume that we apply the second rule from Equality (13) to $G_{k-1} = S(b, d)$ for some constants b and d. This case is always invalid due to the same reason as the previous one.
  - Assume that we apply the third rule from Equality (13) to $G_{k-1} = S(b, x)$ for some constant b and Skolem constant x. Then, $G_k = P_S(b, x)$ and G_{k+1} is obtained from G_k by applying the first rule from Equation (13); that is, $G_{k+1} = A(b)$. But then we have that $A(x) \to \exists y.[S(x, y) \wedge B(y)] \in \mathcal{O}$ and consequently $\mathcal{O} \cup \{A(b)\} \models \exists^* G_{k-1}$. W.l.o.g. we can assume that starting from G_{k+1} rules only from $\mathcal{O}$ are used, which means that $\mathcal{O} \cup [\exists y.R(s, y)] \models A(b)$.
  - No other case is possible.

  Thus $O \cup \{u\} \models P(\vec{s})$ which contradicts *(iii)*.

Thus, cens is optimal for $\mathbf{I}$, which concludes the proof. □

B Appendix (Algorithms)

Algorithm 1: Compute an optimal view for a guarded tree-shaped CQE instance

```
INPUT : a guarded CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, P)$
OUTPUT: a dataset $\mathcal{V}$
$\mathcal{O}_E := \mathcal{O} \cup \bigcup_{\text{binary } R \text{ in } \mathcal{O}} \{R(x,y) \rightarrow \delta_R(x),\ R(x,y) \rightarrow \rho_R(y)\}$;
$\mathcal{H} :=$ the minimal Herbrand model for $\mathcal{O}_E$ and $\mathcal{D}$;
$\mathcal{V} :=$ a maximal subset of unary atoms from $\mathcal{H}$ s.t. $\mathsf{cert}(P, \mathcal{O}_E, \mathcal{V}) = \emptyset$;
for each constant $a$ from $\mathcal{H}$ do $\mathcal{V} :=$ AddUnPredicates($a$);
for each $R(a,b) \in \mathcal{H}$ such that $R$ is not $\approx$ do $\mathcal{V} :=$ AddBinPredicates($R(a,b)$);
return $\mathcal{V}$;
```

Algorithm 2: Sub-routines for Algorithm 1

Sub-routine AddUnPredicates

```
INPUT : a constant $a$
OUTPUT: a dataset $\mathcal{V}'$
$\mathcal{V}' := \mathcal{V}$;
$C := \{A \mid A(a) \in \mathcal{H}\}$;
$\sigma_a := \{a\}$;
for each subset $Sub$ of $C$ closed under $\mathcal{O}_E$ do
    create a globally fresh copy $a_{Sub}$ of $a$;
    if $\mathcal{O}_E \cup \mathcal{V}' \cup \{A(a_{Sub}) \mid A \in Sub\} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$ then
        $\mathcal{V}' := \mathcal{V}' \cup \{A(a_{Sub}) \mid A \in Sub\}$;
        $\sigma_a := \sigma_a \cup \{a_{Sub}\}$;
return $\mathcal{V}'$;
```

Sub-routine AddBinPredicates

```
INPUT : a binary atom $R(a,b)$
OUTPUT: a dataset $\mathcal{V}'$
$\mathcal{V}' := \mathcal{V}$;
for each pair $a^* \in \sigma_a$ and $b^* \in \sigma_b$ do
    if CheckRole($R(a^*, b^*), \mathcal{V}'$) then $\mathcal{V}' := \mathcal{V}' \cup \{R(a^*, b^*)\}$;
return $\mathcal{V}'$;
```

Sub-routine CheckRole

```
INPUT : a binary atom $R(a,b)$, a dataset $\mathcal{V}'$
OUTPUT: True or False
if $\mathcal{O}_E \cup \mathcal{V}' \cup \{R(a,b)\} \not\models P(\vec{s})$ for each $\vec{s} \in \mathsf{cert}(P, \mathcal{O}, \mathcal{D})$
and $\mathcal{O}_E \cup \mathcal{V}' \cup \{R(a,b)\} \models C(c)$ implies $C(c) \in \mathcal{V}$ for any unary predicate $C$ then
    return True;
else return False;
```

C Appendix (Reduction)

In this section, we show the reduction of the problem of uniform boundedness for binary Datalog to the problem of existence of optimal obstructions for Datalog CQE instances (see Section 6).

Let $\mathcal{O}$ be a binary Datalog ontology over a signature Σ (observe that w.l.o.g. we can assume that $\mathcal{O}$ is connected). Then, $\mathcal{O}$ is *uniformly bounded* if there is a constant N such that for every dataset $\mathcal{D}$ over Σ and for every ground atom $P(\vec{t})$, if the atom has a proof from $\mathcal{O}$ and $\mathcal{D}$, then it has a proof not longer than N. It is well known that each relation $P(\vec{x})$ defined by $\mathcal{O}$ is equivalent to an infinite union of CQs $\bigvee_{i=1}^{\infty} \varphi_i^P(\vec{x})$. Note that each $\varphi_i^P(\vec{x})$ is a result of applying some sequence of rules from $\mathcal{O}$ to $P(\vec{x})$. Moreover,

(P1) a Datalog ontology is uniformly bounded if and only if there exists a number N such that each $P(\vec{x})$ is equivalent to $\bigvee_{i=1}^{N} \varphi_i^P(\vec{x})$.

Now we are ready to provide the required reduction. Let $\mathcal{O}$ be a binary Datalog ontology. We are going to construct a CQE instance $\mathbf{I} = (\mathcal{O}', \mathcal{D}, P)$ which admits an optimal obstruction if and only if $\mathcal{O}$ is uniformly bounded. The ontology $\mathcal{O}'$ of $\mathbf{I}$ is defined as

$$\begin{aligned}&\mathcal{O} \cup \{R_1^A(a,x) \wedge A(x) \to P \mid A \text{ is unary and } A \in \Sigma\}\\ &\quad \cup \{R_1^S(a,x_1) \wedge R_2^S(a,x_2) \wedge S(x_1,x_2) \to P \mid S \text{ is binary and } S \in \Sigma\},\end{aligned}$$

where all R_1^A and R_i^S and P are fresh predicates. The dataset $\mathcal{D}$ is equal to

$$\{A(a), S(a,a) \mid A \text{ is a unary and } S \text{ is a binary predicates from } \Sigma'\} \cup \{P\},$$

where Σ' is Σ extended with fresh predicates R_i^Q. Observe that this dataset admits any possible proof of P.

Let $\mathbb{Q} \setminus \mathbb{S}$ are built as in Definition 21. It is easy to see that $\mathbb{Q} \setminus \mathbb{S}$ contains the queries ψ_i^A and ψ_i^S of the form $\exists x.R^A(a,x) \wedge \varphi_i^A(x)$ and $\exists x_1 \exists x_2 . R_1^S(a,x_1) \wedge R_2^S(a,x_2) \wedge \varphi_i^S(x_1,x_2)$, respectively, for each $A, S \in \Sigma$ as each of them with the help of $\mathcal{O}'$ compromises the policy.

Assume that $\mathcal{O}$ is not uniformly bounded; then, due to Property (P1), there is some $Q \in \Sigma$ such that for any number N we have that $\bigvee_{i=1}^{N} \varphi_i^Q(\vec{x}) \not\equiv \bigvee_{i=1}^{\infty} \varphi_i^Q(\vec{x})$. That is, it is not the case that for each φ_i^Q there exists φ_j^Q with $j \leq N$ such that there is a homomorphism from $\varphi_j^Q(\vec{x})$ to $\varphi_i^Q(\vec{x})$ (note that here distinguished variables are mapped into themselves). This immediately yields that it is not the case that for each number N and for each ψ_i^Q there exists ψ_j^Q with $j \leq N$ such that there is a homomorphism from ψ_j^Q to ψ_i^Q (note that, although here we do not have distinguished variables, we still have that the variables of ψ_j^Q that correspond to distinguished variables of $\varphi_j^Q(\vec{x})$ are mapped to the variables of ψ_i^Q that correspond to distinguished variables of $\varphi_i^Q(\vec{x})$ since they are "marked" by predicates R_i^Q). Moreover, for every predicate T different from Q, it holds that for any i and any j there is no homomorphism from ψ_i^Q to ψ_j^T since the former one mentions the predicate R_1^Q and the latter one R_1^T. Hence, there is no finite pseudo-obstruction for $\mathbf{I}$ and therefore, due to Theorem 22, no optimal obstruction censor for $\mathbf{I}$ exists.

Assume that $\mathcal{O}$ is uniformly bounded and N is a number such that for any dataset, if a fact can proved from $\mathcal{O}$ and the dataset, then there is a proof of this fact not longer than N. Let $\mathbb{T}$ be a subset of $\mathbb{Q} \setminus \mathbb{S}$ consisting of those Boolean CQs $\exists^* G$, where G is a sub-goal in some proof of P in $\mathcal{O}' \cup \mathcal{D}$ of length not longer than $N+3$. We claim that the UCQ $U = \bigvee_{\varphi \in \mathbb{T}} \varphi$ is an optimal obstruction for $\mathbf{I}$. Assume that there exists a Boolean CQ $\psi = \exists^* G_0$ from $\mathbb{Q} \setminus \mathbb{S}$ with G_0 a sub-goal coming from some proof of length greater than $N+3$. This means that $\mathcal{O}' \cup \mathbb{S} \cup \{\psi\} \models P$. Than there exists a proof π of P from $\mathcal{O}' \cup \mathcal{A}$, where $\mathcal{A} = [\psi] \cup \bigcup_{\varphi \in \mathbb{S}} [\varphi]$, of length no longer than $N+3$ (1 step to apply one of the rules $R_1^S(a,x_1) \wedge R_2^S(a,x_2) \wedge S(x_1,x_2) \to P$ from $\mathcal{O}'$, N steps to proof $S(x_1,x_2)$ using rules from $\mathcal{O} \cup \mathcal{A}$, and 2 additional steps to proof $R_1^S(a,b_1) \wedge R_2^S(a,b_2)$ using facts from $\mathcal{A}$ for some elements b_1 and b_2). W.l.o.g., we can assume that this proof is normalised. Recall that all the rules that are applied after the frontier are from $\mathcal{A}$. We can assume w.l.o.g. that rules from $[\psi]$ are applied only at the very end of the proof. Clearly, the goal G right before we start to apply the rules from $[\psi]$ is such that *(i)* $\exists^* G \in \mathbb{T}$ and *(ii)* there is a homomorphism from $\exists^* G$ to ψ. These properties imply that $\mathbb{T}$ is a pseudo-obstruction and, since it is finite, by Theorem 22 we have that an optimal obstruction censor for $\mathbf{I}$ exists.