Controlled Query Evaluation over OWL 2 RL Ontologies[⋆]

Bernardo Cuenca Grau[1], Evgeny Kharlamov[1], Egor V. Kostylev[2], and Dmitriy Zheleznyakov[1]

[1] Department of Computer Science, University of Oxford
[2] School of Informatics, University of Edinburgh

Abstract. We study confidentiality enforcement in ontology-based information systems where ontologies are expressed in OWL 2 RL, a profile of OWL 2 that is becoming increasingly popular in Semantic Web applications. We formalise a natural adaptation of the Controlled Query Evaluation (CQE) framework to ontologies. Our goal is to provide CQE algorithms that (i) ensure confidentiality of sensitive information; (ii) are efficiently implementable by means of RDF triple store technologies; and (iii) ensure maximality of the answers returned by the system to user queries (thus restricting access to information as little as possible). We formally show that these requirements are in conflict and cannot be satisfied without imposing restrictions on ontologies. We propose a fragment of OWL 2 RL for which all three requirements can be satisfied. For the identified fragment, we design a CQE algorithm that has the same computational complexity as standard query answering and can be implemented by relying on state-of-the-art triple stores.

1 Introduction

Preserving confidentiality of information (i.e., ensuring that sensitive data is only accessible to authorised users) is a critical requirement for the design of information systems. In recent years, Semantic Web technologies have become widespread in many application domains. There is consequently a pressing need for suitable confidentiality enforcement infrastructure in ontology-based information systems which rely on RDF as a data model, SPARQL as a query language, and OWL 2 as a language for describing background knowledge.

In traditional database management systems, confidentiality is enforced by means of mandatory and discretionary access control mechanisms, where access to data items such as tuples, entire relational tables, or database views is granted only to certain (groups of) users. Such access control mechanisms are, however, problematic for ontology-based information systems: explicitly represented RDF

[⋆] This work was partially supported by the EU project Optique (FP7-IP-318338), EPSRC project Score!, ERC FP7 grant Webdam (n. 226513), and UK EPSRC project SOCIAM (grant EP/J017728/1). Bernardo Cuenca Grau is also supported by a Royal Society University Research Fellowship.

data is assumed to be incomplete and hence new, implicit, data can be derived from the axioms in the ontology via logical reasoning. By granting access to a certain set of RDF triples, system administrators are de facto disclosing a much larger set of implicit triples, some of which a user might not be allowed to know. In contrast, traditional databases are complete, and hence system data is always explicit; as a result, managing access rights is conceptually a simpler problem.

Controlled Query Evaluation (CQE) [1,2,3,4,5,6] is an approach to confidentiality enforcement where system administrators specify in a declarative way the information that cannot be disclosed to users (neither directly nor indirectly via results of previous queries) by means of a *confidentiality policy*. When given a user query, a *censor* checks whether returning the answer would lead to a violation of the corresponding policy and thus to a disclosure of confidential information to unauthorised users; in that case, the censor returns a distorted answer.

CQE in databases is a long standing research area [1,2,4,6,3]; existing work, however, focuses mostly on complete relational databases. CQE for incomplete databases, which are more closely related to ontologies, remains relatively unexplored and research has so far been limited to foundational aspects [5].

In this paper, we are interested in ensuring confidentiality in ontology-based information systems where the relevant ontologies are expressed in the OWL 2 RL profile [7]—a fragment of OWL 2 for which query answering is known to be theoretically tractable in the size of both ontology and data, and efficiently implementable by means of rule-based triple store technologies. OWL 2 RL has become increasingly popular, and state-of-the-art RL reasoners such as OWLim [8] and Oracle's RDF Semantic Graph [9] provide robust and scalable support for SPARQL queries over OWL 2 RL ontologies and RDF data.

Motivated by the CQE paradigm, we study confidentiality enforcement in the scenario described next. We assume that the information in the system consists of background knowledge formalised as an OWL 2 RL ontology, and dataset formalised as a set of unary and binary facts. The ontology is assumed to be fully known to all users (a worst-case situation for confidentiality enforcement), whereas data is assumed to be hidden. Interaction with the system is restricted to a query interface, which allows users to formulate arbitrary conjunctive queries (which constitute the core of SPARQL). A confidentiality policy is represented as a set of facts logically entailed by the ontology and dataset. Given a user query, the system returns a subset of the certain answers to this query over the ontology and dataset determined by the censor. Thus, we adopt the basic case of the CQE paradigm where the censor only filters out problematic answers.

In this scenario, there is a tradeoff between confidentiality and accessibility of information: a censor that returns the empty answer for each query makes the system secure, but also equally useless. Thus, we are interested in *optimal* censors, i.e., those that return maximal sets of answers to queries which still preserve the required confidentiality. Also, CQE can be computationally expensive and to the best of our knowledge practicable algorithms are yet to be developed. We are consequently interested in censors that can be *efficiently implemented*, ideally by relying on the same technology used for query answering in OWL 2 RL.

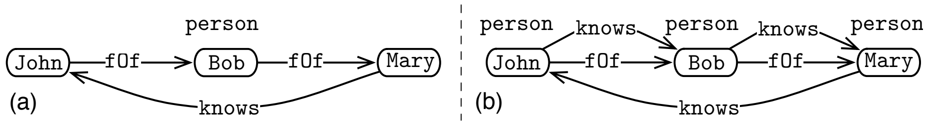

Fig. 1. Dataset $\mathcal{D}_{ex}$ (a), and the same dataset extended with implicit information (b).

The contributions of this paper are as follows. In Section 3 we take existing work on CQE for incomplete databases as a starting point, and present a CQE framework that takes into account the specific features of standard ontology and query languages. In Section 4 we propose the class of *view-definable* censors, which can be implemented by delegating the censor's main computational workload to an OWL 2 RL query answering engine. Roughly speaking, the behaviour of such a censor is determined by what we call a *view*: a dataset that "encodes" the information in the system relevant to the censor's output for any user query. We next explore in Section 5 the formal limitations of our approach and show that censors that are both optimal and view-definable may not exist; furthermore, even if such a censor exists, the corresponding view might be exponentially larger than the system's dataset, thus making efficient implementations difficult. In Section 6, we identify a fragment of OWL 2 RL for which these limitations can be circumvented. Our fragment is able to capture non-trivial extensions of RDF-Schema and is thus relevant for many Semantic Web applications. We consequently provide a practical CQE evaluation algorithm that guarantees both optimality and efficiency if the system's ontology belongs to our fragment. Finally, in Section 7 we observe that there are cases where different optimal view-definable censors exist, and where there is no good reason for choosing one over the others; hence, we study how to deal with such cases.

2 Preliminaries

We assume that all our definitions are parameterised by a first-order signature Σ consisting only of constants, unary predicates, and binary predicates. We also assume first-order logic with equality over Σ, and denote with $\approx$ the special binary equality predicate and $\bot$ the special nullary false predicate. A *dataset* is a finite set of ground (equality-free) atoms over Σ.

Example 1. Consider the following dataset $\mathcal{D}_{ex}$, where the predicate `fOf` represents the "friend of" relation:

$$\mathtt{person}(\mathtt{Bob}), \qquad \mathtt{knows}(\mathtt{Mary},\mathtt{John}), \qquad \mathtt{fOf}(\mathtt{John},\mathtt{Bob}), \qquad \mathtt{fOf}(\mathtt{Bob},\mathtt{Mary}).$$

A graphical representation of $\mathcal{D}_{ex}$ is given in Figure 1(a). ■

Definition 2 (Rule, ontology). *A* rule r *is a first-order sentence of the form* $\forall \boldsymbol{x} \forall \boldsymbol{z}.\, \varphi(\boldsymbol{x},\boldsymbol{z}) \rightarrow \psi(\boldsymbol{x})$*, where* $\boldsymbol{x}$ *and* $\boldsymbol{z}$ *are tuples of variables,* $\varphi(\boldsymbol{x},\boldsymbol{z})$ *is a conjunction of atoms not mentioning* $\approx$ *or* $\bot$*, and* $\psi(\boldsymbol{x})$ *is a single atom. The conjunction*

$\varphi(\boldsymbol{x}, \boldsymbol{z})$ is the body *of r and the atom $\psi(\boldsymbol{x})$ is the* head *of r; quantifiers are often omitted for simplicity. An* ontology *is a finite set of rules.*

We assume first-order semantics of formulae such as rules, and use $\models$ in the standard way as the logical consequence relation.

Example 3. Consider the ontology $\mathcal{O}_{ex}$ consisting of the following rules:

$$\texttt{knows}(x, y) \to \texttt{person}(x);\ \texttt{knows}(x, y) \to \texttt{person}(y);\ \texttt{fOf}(x, y) \to \texttt{knows}(x, y).$$

Intuitively, $\mathcal{O}_{ex}$ says that only people can participate in the relation `knows`, and if two people are friends (i.e., they participate in the `fOf` relation), then they know each other. In Figure 1(b), we depict the dataset $\mathcal{D}_{ex}$ from Example 1 extended with ground atoms that are logically entailed by $\mathcal{O}_{ex} \cup \mathcal{D}_{ex}$. For example, $\mathcal{O}_{ex} \cup \mathcal{D}_{ex} \models \texttt{person}(\texttt{John})$ and $\mathcal{O}_{ex} \cup \mathcal{D}_{ex} \models \texttt{knows}(\texttt{John}, \texttt{Bob})$. ■

OWL 2 RL was designed as “a syntactic subset of OWL 2 which is amenable to implementation using rule-based technologies” [7]. In particular, each OWL 2 RL knowledge base can be normalised as a set of rules. We make two simplifying assumptions w.r.t. the normative specification of OWL 2 RL. First, we ignore datatypes for simplicity; second, we assume that no constants occur in rules. The latter assumption ensures a clean separation between schema knowledge and data. The following definition characterises a class of rules that is sufficient to capture normative OWL 2 RL under our basic assumptions.

Definition 4 (RL ontology). *An ontology $\mathcal{O}$ is an* RL ontology *if it can be partitioned as $\mathcal{O} = \mathcal{O}' \uplus \mathcal{O}''$ such that the following properties hold.*

1. *Each rule r from $\mathcal{O}'$ is constant-free; furthermore, the variables in r consist of a single* root variable *x and a set of* branch variables *$\boldsymbol{y}$ such that:*
   - *(a) each binary atom in the body of r must mention x once and only once;*
   - *(b) each branch variable y occurs in exactly one binary atom in the body;*
   - *(c) if the head atom is binary then it is of the form $y \approx y'$, for some y, y' from $\boldsymbol{y}$, and the binary atoms in the body, mentioning y and y', use the same predicate symbol and have y and y' on the same position.*
2. *Each rule in $\mathcal{O}''$ is of one of the following forms:*
   - *(a) $R(x, y) \to S(x, y)$; or*
   - *(c) $R(x, y) \to S(y, x)$; or*
   - *(b) $R(x, y) \wedge S(y, z) \to T(x, z)$; or*
   - *(d) $R(x, y) \wedge S(x, y) \to \bot$.*

The OWL 2 RL specification requires certain *global restrictions* to hold in order to ensure that OWL 2 RL is a syntactic fragment of OWL 2 DL (e.g., transitive properties cannot occur in cardinality constraints) [7,10]. Such restrictions are not reflected in Definition 4 since they are immaterial to our results.

Example 5. The ontology $\mathcal{O}_{ex}$ is an RL ontology. Moreover, for $\mathcal{O}_{ex}$, $\mathcal{O}'_{ex}$ consists of the first two rules, while $\mathcal{O}''_{ex}$ consists of the last rule only. ■

Definition 6 (Conjunctive query). *A* conjunctive query $Q(\boldsymbol{x})$ *is a first-order formula of the form* $\exists \boldsymbol{y}.\, \varphi(\boldsymbol{x}, \boldsymbol{y})$*, where* $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *are tuples of variables and* $\varphi(\boldsymbol{x}, \boldsymbol{y})$ *is a conjunction of atoms.*

We will write Q instead of $Q(\boldsymbol{x})$ when $\boldsymbol{x}$ is irrelevant or clear from the context.

Definition 7 (Query answering). *Let* $\mathcal{O}$ *be an ontology,* $\mathcal{D}$ *be a dataset, and let* $Q(\boldsymbol{x})$ *be a conjunctive query. A tuple* $\boldsymbol{t}$ *of constants is a* certain answer *to* $Q(\boldsymbol{x})$ *w.r.t.* $\mathcal{O}$ *and* $\mathcal{D}$ *if* $\mathcal{O} \cup \mathcal{D} \models Q(\boldsymbol{t})$*. The set of all certain answers to a conjunctive query* $Q(\boldsymbol{x})$ *w.r.t.* $\mathcal{O}$ *and* $\mathcal{D}$ *is denoted by* $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$*.*

Example 8. Consider the following queries:

$$Q_1(x) = \mathtt{person}(x); \qquad Q_2(x) = \exists y, z.\, \mathtt{fOf}(x, y) \land \mathtt{fOf}(y, z) \land \mathtt{knows}(z, x).$$

Clearly, $\mathsf{cert}(Q_1, \mathcal{O}_{ex}, \mathcal{D}_{ex}) = \{\mathtt{John}, \mathtt{Bob}, \mathtt{Mary}\}$ since all constants in $\mathcal{D}_{ex}$ are entailed to be persons (c.f., Figure 1(b)). Also, $\mathsf{cert}(Q_2, \mathcal{O}_{ex}, \mathcal{D}_{ex}) = \{\mathtt{John}\}$ since `John` is a friend of `Bob`, `Bob` is a friend of `Mary`, and `Mary` knows `John`.

3 Controlled Query Evaluation for Ontologies

Our approach to confidentiality enforcement in ontology-based information systems was inspired by the CQE paradigm for incomplete databases first proposed by Biskup and Weibert [5], which we briefly describe next.

A CQE system in [5] stores a database and a *policy*, which are both defined as sets of propositional sentences. The policy is under the control of the system administrators, and its goal is to declaratively specify the information that is to be kept secret. Both database and policy are hidden from users, and interaction with the system is limited to a query interface. Given a (propositional) user query the system does not directly return the correct answer; instead, a *censor* decides whether the answer needs to be modified according to the policy.

Let $q_1, \ldots, q_n$ be any finite sequence of such user queries, let $v_1, \ldots, v_n$ be the answers (truth values) to these queries returned by the censor for the system's database $\mathbf{D}$, and let γ be an arbitrary sentence in the policy. The *confidentiality* of γ is compromised if the truth values $v_1, \ldots, v_n$ fully determine the truth value of γ over $\mathbf{D}$. In other words, to preserve confidentiality of γ there must exist some other database $\mathbf{D}'$ such that the censor evaluates the queries $q_1, \ldots, q_n$ to the same values $v_1, \ldots, v_n$ over $\mathbf{D}'$, but γ does not hold in $\mathbf{D}'$. This means that $\mathbf{D}$ and $\mathbf{D}'$ are indistinguishable w.r.t. the user queries and hence the user cannot decide whether γ holds or not. Hereby, the censor must preserve confidentiality of all the sensitive data in the policy—that is, it should make sure that users cannot derive any sentence in the policy by posing any finite set of queries.

Our framework focuses on ontology-based information systems and hence deviates from [5] in order to better reflect the specific features of standard ontology and query languages. In the remainder of this section, we formally describe the elements of our approach.

3.1 Policies and CQE-instances

We start with some assumptions made in our framework. Following [5], we assume that both dataset and policy are hidden and that users can pose arbitrary (conjunctive) queries to a query interface. We will assume, however, that the system's ontology is fully known to all users. The rationale behind this assumption is twofold. On the one hand, information represented in ontologies is typically common knowledge, and it is dangerous to enforce confidentiality by relying on users' unawareness of rather straightforward constraints; on the other hand, public availability of the system's background knowledge is key to improving access to information: familiarity with the rules in the ontology can be invaluable for users to formulate accurate queries. Furthermore, in our setting, information is under the control of system developers and domain experts; thus, inconsistencies have been resolved before the query interface is made available to users. We consequently assume that query answering is always performed over a satisfiable ontology and dataset. This assumption makes query results meaningful to users. Finally, we assume that a policy is represented by a set of ground atoms that are logically entailed by the ontology and dataset in the system.

To sum up, the relevant content of the CQE system for the purpose of confidentiality enforcement is formalised in the following definition.

Definition 9 (Policy, CQE-instance). *Let $\mathcal{O}$ be an ontology and let $\mathcal{D}$ be a dataset such that $\mathcal{O} \cup \mathcal{D}$ is satisfiable. A* policy *$\mathcal{P}$ for $\mathcal{O}$ and $\mathcal{D}$ is a dataset such that $\mathcal{O} \cup \mathcal{D} \models \mathcal{P}$, and a* CQE-instance *is the triple $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$.*

Example 10. The dataset $\mathcal{P}_{ex} = \{\mathtt{knows}(\mathtt{Mary}, \mathtt{John})\}$ is a policy for our running example ontology $\mathcal{O}_{ex}$ and dataset $\mathcal{D}_{ex}$ since $\mathcal{O}_{ex} \cup \mathcal{D}_{ex} \models \mathcal{P}_{ex}$. The triple $\mathbf{I}_{ex} = (\mathcal{O}_{ex}, \mathcal{D}_{ex}, \mathcal{P}_{ex})$ thus constitutes a CQE-instance. ■

3.2 Censors and Confidentiality Preservation

In [5], Biskup and Weibert consider censors that can distort query answers in various ways. We adopt a pragmatic approach where censors are required to return a sound, but possibly incomplete set of certain answers. Thus, the goal of such a censor is limited to filtering out answers which may compromise the policy. In contrast to [5], our censors never return unsound answers, or reject queries.

Definition 11 (Censor). *A* censor cens *for a CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ is a function which maps each conjunctive query Q to a subset of $\mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$.*

To align with the semantics of OWL 2, we adopt a notion of confidentiality preservation that is formulated directly in terms of first-order models and entailment. Specifically, with a censor cens for a CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$, we associate the following (possibly infinite) set of first-order sentences:

$$\mathcal{F}(\mathsf{cens}) = \{Q(\boldsymbol{t}) \mid \boldsymbol{t} \in \mathsf{cens}(Q), Q(\boldsymbol{x}) \text{ is a conjunctive query}\}.$$

The set $\mathcal{F}(\mathsf{cens})$ intuitively represents all the information that a user can potentially gain by interacting with the query interface. Since a user can ask only a finite (yet unbounded) number of arbitrary queries, the information that the user can gather from the system can be captured by a finite subset of $\mathcal{F}(\mathsf{cens})$. Confidentiality preservation then amounts to ensuring that no finite subset of $\mathcal{F}(\mathsf{cens})$ can logically entail an atom in the policy when coupled with $\mathcal{O}$.

Definition 12 (Confidentiality preservation). *Let* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *be a CQE-instance. A censor* cens *for* $\mathbf{I}$ *is* confidentiality preserving *if for each ground atom* α *in the policy* $\mathcal{P}$ *and each finite subset* $\mathcal{F}$ *of* $\mathcal{F}(\mathsf{cens})$ *it holds that* $\mathcal{O} \cup \mathcal{F} \not\models \alpha$.

Definition 12 is consistent with the notion of confidentiality preservation in [5]. Indeed, if a censor cens is confidentiality preserving for $\mathbf{I}$, then a user cannot entail any confidential information from $\mathcal{P}$ regardless of what queries they pose. Thus, for each atom α in the policy $\mathcal{P}$ and each finite subset $\mathcal{F}$ of $\mathcal{F}(\mathsf{cens})$ there is a model of $\mathcal{O} \cup \mathcal{F}$ in which α does not hold. Moreover, since $\mathcal{O}$ is an OWL 2 RL ontology and $\mathcal{F}$ is a finite set of positive existential first-order sentences, there always exists a finite model M, which can be seen as a database instance in the sense of Biskup and Weibert. Since $M \models \mathcal{F}$, the model M cannot be distinguished from $\mathcal{D}$ using the query answers returned by the censor. In contrast, $\mathcal{D}$ and M differ w.r.t. the atom α in the policy, which implies that the user cannot decide whether α holds or not in $\mathcal{D}$ based on returned query answers alone.

3.3 Information Access vs. Confidentiality Tradeoff

There is a tradeoff between confidentiality preservation and accessibility of information. On the one hand, a censor for a CQE-instance that returns the empty set of answers for each query is clearly confidentiality preserving, but it also does not provide any useful information; on the other hand, a censor that returns all the certain answers to each query maximises information accessibility, but may not be confidentiality preserving. Hence, we are interested in *optimal* censors, which distort the answer only if necessary for enforcing confidentiality.

Definition 13 (Optimality). *Let* $\mathbf{I}$ *be a CQE-instance, and let* cens *be a confidentiality preserving censor for* $\mathbf{I}$. *The censor* cens *is* optimal *if no other censor* $\mathsf{cens}' \neq \mathsf{cens}$ *for* $\mathbf{I}$ *exists such that (i)* cens' *is confidentiality preserving; and (ii)* $\mathsf{cens}(Q) \subseteq \mathsf{cens}'(Q)$ *holds for each conjunctive query* Q.

As we will discuss later on, there can be several optimal censors for a given CQE-instance. In general, however, there is no good reason for choosing one over the others, so we will design an algorithm which constructs all of them; however, we will also study situations were a unique optimal censor is guaranteed to exist.

We conclude this section with a useful characterisation of the optimality of cens in terms of its associated theory $\mathcal{F}(\mathsf{cens})$.

Proposition 14. *Let* cens *be a confidentiality preserving censor for a CQE-instance* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$. *Then,* cens *is optimal iff for each conjunctive query* $Q(\boldsymbol{x})$ *and each tuple* $\boldsymbol{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ *the fact that* $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \cup \{Q(\boldsymbol{t})\} \not\models \alpha$ *holds for each* $\alpha \in \mathcal{P}$ *implies that* $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \models Q(\boldsymbol{t})$.

4 View-based Controlled Query Evaluation

As already discussed, the main task of the censor in a typical CQE system is to receive answers to user queries as computed by the query answering engine, and to decide, according to the policy, which answers are safe to return to the user and which ones must be distorted. Such a censor is usually conceived as a separate component, which is implemented on top of the query answering engine. Unsurprisingly, implementing the censor of a CQE system becomes a major challenge. The censor's task can be computationally very expensive, and the cost of the censor's evaluation adds to the cost of query answering. Furthermore, implementing the censor may require dedicated algorithms, and, in particular, the highly optimised infrastructure available for query answering might not be reusable. Hence, performance of a CQE system can be significantly affected by the confidentiality enforcement component, and, as a result, the system might not be practically feasible in performance-critical situations.

To address these challenges, we develop a novel approach that deviates from the mainstream separation of query answering engine and censor as different components of a CQE system. More specifically, we propose to exploit the available OWL 2 RL triple store infrastructure as much as possible, by delegating the censor's main computational workload to the query answering engine.

Our key idea is to associate to each CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ a new dataset, which we call a *view*. Such a view $\mathcal{V}$ determines a censor $\mathsf{cens}_{\mathcal{V}}$ in the sense that, for each input user query Q, the censor's output $\mathsf{cens}_{\mathcal{V}}(Q)$ is uniquely and trivially extractable from the set $\mathsf{cert}(Q, \mathcal{O}, \mathcal{V})$ of all certain answers to Q w.r.t. the ontology $\mathcal{O}$ and the view $\mathcal{V}$. Since the view $\mathcal{V}$ is associated only with $\mathbf{I}$ and is query-indenpendent, it also does not need to be recomputed until the underlying dataset $\mathcal{D}$ is updated. In this way, the main workload of the censor in a typical user session boils down to the computation of certain answers, which can be fully delegated to the query answering engine.

Obviously, if we want the censor $\mathsf{cens}_{\mathcal{V}}$ to enjoy the properties we are after, the view $\mathcal{V}$ must be constructed with care. In order for $\mathsf{cens}_{\mathcal{V}}$ to be indeed a censor, $\mathcal{V}$ must not lead to spurious query answers. Furthermore, in order for $\mathsf{cens}_{\mathcal{V}}$ to be confidentiality preserving, $\mathcal{V}$ and $\mathcal{O}$ should not entail any atom in $\mathcal{P}$. Finally, in order for $\mathsf{cens}_{\mathcal{V}}$ to be optimal, $\mathcal{V}$ must "encode" as much information from $\mathcal{D}$ as possible. We next illustrate these ideas with an example.

Example 15. We construct a view $\mathcal{V}_{ex}$ for our example CQE-instance $\mathbf{I}_{ex} = (\mathcal{O}_{ex}, \mathcal{D}_{ex}, \mathcal{P}_{ex})$. Since the policy $\mathcal{P}_{ex}$ contains the atom $\alpha = \mathtt{knows}(\mathtt{Mary}, \mathtt{John})$, we cannot include α in $\mathcal{V}_{ex}$. An obvious possibility would be to define $\mathcal{V}_{ex}$ as $\mathcal{D}_{ex} \setminus \{\alpha\}$, and then $\mathsf{cens}_{\mathcal{V}_{ex}}$ as the function that returns $\mathsf{cert}(Q, \mathcal{O}_{ex}, \mathcal{V}_{ex})$ for each conjunctive query Q. Clearly, $\mathsf{cens}_{\mathcal{V}_{ex}}$ is a censor for $\mathbf{I}_{ex}$, since $\mathsf{cert}(Q, \mathcal{O}_{ex}, \mathcal{V}_{ex}) \subseteq \mathsf{cert}(Q, \mathcal{O}_{ex}, \mathcal{D}_{ex})$, i.e., $\mathsf{cens}_{\mathcal{V}_{ex}}$ returns only sound answers. Furthermore, $\mathsf{cens}_{\mathcal{V}_{ex}}$ is confidentiality preserving: since $\mathcal{O}_{ex} \cup \mathcal{V}_{ex} \not\models \alpha$ and $\mathcal{O}_{ex} \cup \mathcal{V}_{ex} \models \mathcal{F}(\mathsf{cens}_{\mathcal{V}_{ex}})$, it is clear that $\mathcal{O}_{ex} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}_{ex}}) \not\models \alpha$, as required by Definition 12. The censor $\mathsf{cens}_{\mathcal{V}_{ex}}$ is, however, not optimal. To see this, consider the query $Q(x) = \exists y, z.\, \mathtt{fOf}(x, y) \wedge \mathtt{knows}(y, z)$ asking for everyone who is a friend of someone who in turn knows

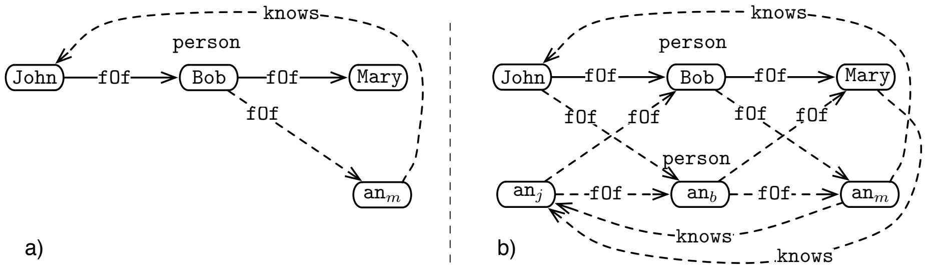

Fig. 2. View $\mathcal{V}_{ex}$ (a), and view defining an optimal censor for $\mathbf{I}_{ex} = (\mathcal{O}_{ex}, \mathcal{D}_{ex}, \mathcal{P}_{ex})$ (b); dashed arrows represent binary atoms that do not occur in $\mathcal{D}_{ex}$.

somebody else. We have $\mathsf{cert}(Q, \mathcal{O}_{ex}, \mathcal{D}_{ex}) = \{\mathtt{John}, \mathtt{Bob}\}$, but $\mathsf{cert}(Q, \mathcal{O}_{ex}, \mathcal{V}_{ex}) = \{\mathtt{John}\}$; nevertheless, answering this query correctly is "harmless" in the sense that $\mathcal{O}_{ex} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}_{ex}}) \cup \{Q(\mathtt{Bob})\} \not\models \alpha$ (c.f. Proposition 14).

Clearly, we cannot add α back into $\mathcal{V}_{ex}$ without compromising the policy, and hence our only choice is to "encode" the missing information in $\mathcal{V}_{ex}$ by some other means. A possibility is to extend the domain of $\mathcal{V}_{ex}$ with an "anonymised copy" $\mathtt{an}_m$ of $\mathtt{Mary}$, and extend $\mathcal{V}_{ex}$ with the atoms $\mathtt{fOf}(\mathtt{Bob}, \mathtt{an}_m)$ and $\mathtt{knows}(\mathtt{an}_m, \mathtt{John})$ (see Figure 2(a)). As a result, we obtain $\mathsf{cert}(Q, \mathcal{O}_{ex}, \mathcal{V}_{ex}) = \{\mathtt{John}, \mathtt{Bob}\}$ and $\mathcal{O}_{ex} \cup \mathcal{V}_{ex} \not\models \alpha$ as we wanted. There is, however, an undesired effect to this modification: for queries such as $Q'(x) = \exists y.\, \mathtt{knows}(x, y)$ we would obtain the fresh constant $\mathtt{an}_m$ as a spurious answer. The obvious fix is to filter out such answers syntactically, and only return those answers in $\mathsf{cert}(Q, \mathcal{O}_{ex}, \mathcal{V}_{ex})$ that mention only constants from the domain of $\mathcal{D}_{ex}$. Although such extended view is still not optimal, we can reiterate this procedure until we achieve optimality. As we will see, the view depicted in Figure 2(b) defines an optimal censor for $\mathbf{I}_{ex}$. ■

We are ready to define the notion of a view $\mathcal{V}$ and its corresponding censor.

Definition 16 (View). *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ be a CQE-instance. A* view *for $\mathbf{I}$ is a dataset $\mathcal{V}$ which satisfies the following properties:*

(i) $\mathcal{O} \cup \mathcal{V} \not\models \alpha$ *for each* $\alpha \in \mathcal{P}$*; and*
(ii) $\boldsymbol{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{V})$ *implies* $\boldsymbol{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ *for each conjunctive query Q and each tuple $\boldsymbol{t}$ of constants from $\mathcal{D}$.*

Definition 17 (View-based censor). *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ be a CQE-instance and let $\mathcal{V}$ be a view for $\mathbf{I}$. The* censor based on $\mathcal{V}$ *(or* view-based censor *when $\mathcal{V}$ is clear) is the function $\mathsf{cens}_{\mathcal{V}}$ mapping a conjunctive query Q to the set of tuples*

$$\{\boldsymbol{t} \mid \boldsymbol{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{V}), \boldsymbol{t} \textit{ has constants only from } \mathcal{D}\}.$$

By Property *(ii)* in Definition 16, each view-based censor is indeed a censor. Proposition 18 establishes that view-based censors are confidentiality preserving.

Proposition 18. *Let $\mathbf{I}$ be a CQE-instance and let $\mathcal{V}$ be a view for $\mathbf{I}$. Then $\mathsf{cens}_{\mathcal{V}}$ is a confidentiality preserving censor for $\mathbf{I}$.*

5 Limitations of View-based Censors

Before investigating the design of efficient view-based CQE algorithms, we first explore the theoretical limitations of our approach. In this section we answer the following important questions about an arbitrary CQE-instance **I**.

1. Is an optimal view-based censor for **I** guaranteed to exist?
2. How large can be the smallest view defining an optimal censor for **I**?

We answer the first question negatively: the presence of equality in the ontology can preclude the existence of an optimal view-based censor, in the sense that the "view" corresponding to such a censor would be necessarily infinite. As a result, there are CQE-instances for which all view-based censors are not optimal.

Concerning the second question, we show that even if the ontology does not contain equalities, the smallest view associated to an optimal censor can be at least exponentially larger than the given instance. This is a crucial limitation in practice, since such a censor would need to compute certain answers over an exponentially large dataset, with the obvious negative effect on performance.

In Section 6 we will restrict the form of ontology rules to guarantee the existence of optimal censors based on small views.

5.1 Non-existence of Optimal View-based Censors

We say that a censor cens for a CQE-instance **I** is *view-definable* if there exists a view $\mathcal{V}$ for **I** such that $\mathsf{cens} = \mathsf{cens}_{\mathcal{V}}$. The following theorem establishes that for some CQE-instances view-definability and optimality of a censor are in conflict.

Theorem 19. *There exists a CQE-instance* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *with an* RL *ontology* $\mathcal{O}$*, for which no censor exists that is both view-definable and optimal.*

The intuition behind the proof is given by means of the following example.

Example 20. Consider the CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$, where $\mathcal{P} = \{\mathtt{emp}(\mathtt{John})\}$, $\mathcal{D} = \{\mathtt{manages}(\mathtt{John}, \mathtt{John})\}$, and $\mathcal{O}$ is defined as follows:

$$\mathcal{O} = \{\mathtt{manages}(x_1, y) \wedge \mathtt{manages}(x_2, y) \to x_1 \approx x_2;\ \mathtt{manages}(x, y) \to \mathtt{emp}(y)\}.$$

The rules in $\mathcal{O}$ say that a person can only be managed by a single manager and that everyone who is managed is an employee. Consider also the following (infinite) sequence of conjunctive queries (for $k \geq 1$):

$$Q_k(x) = \exists x_1, \ldots, x_k.\, \mathtt{manages}(x, x_1) \wedge \ldots \wedge \mathtt{manages}(x_{k-1}, x_k).$$

Clearly, $\mathtt{John}$ is the only certain answer to each Q_k w.r.t. $\mathcal{O}$ and $\mathcal{D}$. Furthermore, answering each of these queries correctly is "harmless" for the confidentiality of the policy, and hence each optimal censor for **I** must answer these queries correctly. Imagine that such an optimal censor is based on some view $\mathcal{V}$; in order for $\mathtt{John}$ to be returned as an answer to a given Q_k, $\mathcal{V}$ must contain atoms

$\texttt{manages}(\texttt{John}, a_1), \ldots, \texttt{manages}(a_{k-1}, a_k)$. Since k is unbounded and the view must be finite, some of the individuals a_i must be equal; but then, the fact that manages is axiomatised in $\mathcal{O}$ as inverse-functional causes the "management chain" in $\mathcal{V}$ to "collapse" into a cycle involving John. This compromises the policy since $\mathcal{O} \cup \mathcal{V}$ implies that John is managed by someone, and hence is an employee. ■

5.2 Exponential Size of Views Defining Optimal Censors

Consider now the situation where an optimal view-based censor exists for a given instance. The following theorem says that the smallest view associated to any such optimal censor might necessarily be of size at least exponential in the size of the given instance. In what follows, $|\mathcal{O}|$ and $|\mathcal{D}|$ denote the number of atoms in the rules of the ontology $\mathcal{O}$ and in the dataset $\mathcal{D}$, respectively.

Theorem 21. *There exists a sequence of CQE-instances $\mathbf{I}_n = (\mathcal{O}_n, \mathcal{D}_n, \mathcal{P})$ for $n \geq 1$ such that $\mathcal{O}_n$ is an equality-free* RL *ontology, $|\mathcal{O}_n| \in O(n)$, $|\mathcal{D}_n| \in O(n)$, and each view $\mathcal{V}$ for $\mathbf{I}_n$ with $\mathsf{cens}_{\mathcal{V}}$ optimal is such that $|\mathcal{V}| \in \Omega(2^n)$.*

Again, we explain the main ideas of the proof by means of an example.

Example 22. Next we give the second element $\mathbf{I}_2 = (\mathcal{O}_2, \mathcal{D}_2, \mathcal{P})$ of the sequence $\mathbf{I}_n$ of CQE-instances. Let $\mathcal{P} = \{\texttt{executive}(\texttt{John})\}$ and

$$\mathcal{O}_2 = \{A^i_1(x) \wedge A^i_2(x) \wedge \texttt{managedBy}(x, y) \rightarrow \texttt{executive}(y) \mid 1 \leq i \leq 2\},$$
$$\mathcal{D}_2 = \{\texttt{managedBy}(\texttt{Bob}, \texttt{John})\} \cup \{A^i_j(\texttt{Bob}) \mid 1 \leq i, j \leq 2\}.$$

The ontology $\mathcal{O}_2$ has two rules with four atoms in each, and the dataset $\mathcal{D}_2$ has $2 \times 2 + 1$ atoms. Consider the following four queries, which ask for those people who manage someone satisfying a given subset of requirements:

$$Q_{j_1,j_2}(y) = \exists x.\, A^1_{j_1}(x) \wedge A^2_{j_2}(x) \wedge \texttt{managedBy}(x, y); \quad 1 \leq j_1, j_2 \leq 2.$$

Clearly, John is the only certain answer to each of these queries w.r.t. $\mathcal{O}_2$ and $\mathcal{D}_2$. Answering these queries correctly does not compromise the policy, because none of the pairs of A^i_j from the queries occur together in the body of any rule in $\mathcal{O}_2$.

So, in order to be optimal, a censor $\mathsf{cens}_{\mathcal{V}}$ must answer all these queries correctly and, for this, the view $\mathcal{V}$ must contain four "witnessing" constants a_{j_1,j_2} for each $1 \leq j_1, j_2 \leq 2$, such that $A^1_{j_1}(a_{j_1,j_2})$, $A^2_{j_2}(a_{j_1,j_2})$, and $\texttt{managedBy}(a_{j_1,j_2}, \texttt{John})$ are in $\mathcal{V}$. Furthermore, any pair of these constants cannot be identified into a single one, since otherwise the user would be able to derive the policy atom.

Similarly, for any other $n \geq 1$, the domain of the view has to contain 2^n different constants $a_{j_1,\ldots,j_n}$ (each witnessing a different query $Q_{j_1,\ldots,j_n}$). ■

This example exploits that RL ontologies allow rules in which unary atoms refer to branch variables. Alternatively, a similar example can be constructed by using rules of the form *2(b)* in Definition 4 (also known as *role chain rules*).

6 Efficient View-based Controlled Query Evaluation

Theorems 19 and 21 show that ensuring optimality comes at the expense of practicality. The proofs of these theorems, however, rely on very specific OWL 2 RL constructs: Theorem 19 critically depends on equality, whereas Theorem 21 requires a rule that mentions a unary atom involving a branch variable (or, alternatively, a rule of the form *2(b)* in Definition 4).

We next present a fragment RL^- of OWL 2 RL for which the limitations from Section 5 can be circumvented. We show that for any CQE-instance involving an RL^- ontology it is possible to construct an optimal view in polynomial time (in the size of $\mathbf{I}$). We start with the definition of RL^-.

Definition 23 (RL^- ontology). *An* RL^- ontology *is an RL ontology $\mathcal{O} = \mathcal{O}' \uplus \mathcal{O}''$ satisfying the following restrictions.*

1. *Each rule r in $\mathcal{O}'$ is equality-free. Furthermore, each unary atom in r (both in the head and body) mentions only the root variable of r.*
2. *There is no rule of the form 2(b) from Definition 4 in $\mathcal{O}''$.*

In particular, this definition ensures that positive rules in $\mathcal{O}'$ are of the form

$$\bigwedge_i A_i(x) \wedge \bigwedge_j R_j(x, y_j) \wedge \bigwedge_k S_k(y_k, x) \to B(x).$$

The ontologies in Examples 20 and 22 are not RL^- ontologies. Nevertheless, RL^- is powerful enough to capture the rules corresponding to RDFS, including subclass and subproperty axioms (i.e., rules of the form $A(x) \to B(x)$ and $R(x, y) \to S(x, y)$) as well as property domain and range axioms (i.e., rules $R(x, y) \to A(x)$ and $R(x, y) \to A(y)$). Additionally, RL^- goes well beyond RDFS and can capture other useful kinds of OWL 2 RL rules.

Example 24. Our example ontology $\mathcal{O}_{ex}$ is expressible as RDFS rules and hence also in RL^-. The following rules are RL^-, but with no correspondence in RDFS:

$$\mathtt{fOf}(x, y) \to \mathtt{fOf}(y, x); \tag{1}$$

$$\mathtt{emp}(x) \wedge \mathtt{manages}(x, y) \to \mathtt{manager}(x); \tag{2}$$

$$\mathtt{student}(x) \wedge \mathtt{onpayroll}(x) \to \mathtt{PHDStudent}(x). \tag{3}$$

Rule (1) axiomatises `fOf` as symmetric. Rule (2) says that employees managing others are managers. Rule (3) says that paid students must be doing a PhD. ■

We next present an algorithm $\mathsf{ComputeView}$ (c.f. Algorithm 1) that takes as input a CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ such that $\mathcal{O}$ is an RL^- ontology and returns a view $\mathcal{V}$ for $\mathbf{I}$ such that $\mathsf{cens}_{\mathcal{V}}$ is guaranteed to be an optimal censor. This algorithm works in polynomial time and, hence, computes $\mathcal{V}$ of polynomial size.

The algorithm $\mathsf{ComputeView}$ starts by creating an anonymised copy of each constant occurring in $\mathcal{D}$ (Line 2), and replicates in $\mathcal{D}_{\mathsf{an}}$ all atoms (unary and binary) of $\mathcal{D}$ on these new constants (Lines 3-4). Moreover, if $R(a, b)$ is in $\mathcal{D}$, then

Algorithm 1: ComputeView

```
   INPUT   : A CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ with an RL$^{-}$ ontology $\mathcal{O}$
   OUTPUT  : A view $\mathcal{V}$ for $\mathbf{I}$
1  $\mathcal{D}_{\mathsf{an}} := \emptyset$;
2  for each constant $a$ occurring in $\mathcal{D}$ do create a fresh constant $an_a$; ;
3  for each $A(a) \in \mathcal{D}$ do $\mathcal{D}_{\mathsf{an}} := \mathcal{D}_{\mathsf{an}} \cup \{A(an_a)\}$; ;
4  for each $R(a,b) \in \mathcal{D}$ do $\mathcal{D}_{\mathsf{an}} := \mathcal{D}_{\mathsf{an}} \cup \{R(an_a, an_b), R(a, an_b),\ R(an_a, b)\}$; ;
5  $\mathcal{D}_{\mathsf{sat}} := \mathcal{D} \cup \mathcal{D}_{\mathsf{an}}$;
6  for each atom $\beta$ s.t. $\mathcal{O} \cup \mathcal{D} \cup \mathcal{D}_{\mathsf{an}} \models \beta$ do $\mathcal{D}_{\mathsf{sat}} := \mathcal{D}_{\mathsf{sat}} \cup \{\beta\}$; ;
7  $\mathcal{V} := \emptyset$;
8  repeat
9      Choose $\beta \in \mathcal{D}_{\mathsf{sat}}$ such that $\mathcal{O} \cup \mathcal{V} \cup \{\beta\} \not\models \alpha$ for each $\alpha \in \mathcal{P}$;
10     $\mathcal{V} := \mathcal{V} \cup \{\beta\}$
11 until no such $\beta \in \mathcal{D}_{\mathsf{sat}}$ can be chosen;
12 return $\mathcal{V}$.
```

the algorithm also relates by R in $\mathcal{D}_{\mathsf{an}}$ the constant a to the anonymised copy an_b of b, and the anonymised copy an_a of a to b (Line 4). Then, the algorithm saturates the resulting dataset $\mathcal{D}_{\mathsf{sat}} = \mathcal{D} \cup \mathcal{D}_{\mathsf{an}}$ with all facts entailed by $\mathcal{O} \cup \mathcal{D}_{\mathsf{sat}}$ (Line 6). Finally, it enforces the policy $\mathcal{P}$ by computing a maximal subset $\mathcal{V}$ of the saturated dataset that respects $\mathcal{P}$ (Lines 8-11), which is finally returned as the output. Note, that different choices in Line 9 might lead to different outputs, i.e., algorithm ComputeView is non-deterministic. Indeed, for a given instance $\mathbf{I}$, there may be several view-based censors that are optimal, and each run of the algorithm ComputeView computes one of them; however, as we will see later on, any view leading to an optimal censor is computed by some run of the algorithm.

Example 25. When receiving our running example CQE-instance $\mathbf{I}_{ex}$ as input, the algorithm ComputeView computes the view given in Figure 2(b). In this case, the algorithm's output is independent from the choices made in Line 9, i.e., all possible runs of the algorithm lead to the same result.

To see how different runs of the algorithm might lead to different outputs, consider a CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ where $\mathcal{O} = \{A(x) \wedge B(x) \rightarrow C(x)\}$, $\mathcal{D} = \{A(a), B(a)\}$, and $\mathcal{P} = \{C(a)\}$. There are two possible runs of ComputeView on $\mathbf{I}$, which lead to two different views $\{A(a), A(an_a), B(an_a), C(an_a)\}$ and $\{B(a), A(an_a), B(an_a), C(an_a)\}$, respectively. ■

The following theorem establishes correctness and complexity of the algorithm.

Theorem 26. *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ be a valid input of the algorithm* ComputeView*.*

(i) If $\mathcal{V}$ is the result of a run of ComputeView *on $\mathbf{I}$ then* $\mathsf{cens}_{\mathcal{V}}$ *is an optimal censor for $\mathbf{I}$.*

(ii) For each optimal censor cens *for $\mathbf{I}$, there exists a run of* ComputeView *on $\mathbf{I}$ that computes $\mathcal{V}$ such that* $\mathsf{cens} = \mathsf{cens}_{\mathcal{V}}$*.*

(iii) The algorithm can be implemented to run in time polynomial in $|\mathcal{D}| + |\mathcal{O}|$.

There is a couple of remarks about the proof of Theorem 26 that are worth to be made here. Polynomial complexity is a direct consequence of two facts: first, the size of $\mathcal{D}_{\mathsf{sat}}$ is polynomial in the size of $\mathcal{D}$ and $\mathcal{O}$; second, checking whether a ground atom can be entailed from an OWL 2 RL ontology and a dataset can be done in polynomial time in both the size of the ontology and dataset [7]. Optimality of $\mathsf{cens}_{\mathcal{V}}$ relies on the fact that $\mathcal{D}_{sat}$ captures all the possible matchings of an input query Q over the least Herbrand model of $\mathcal{O} \cup \mathcal{D}$; for this to be the case, the restrictions on rules imposed by RL^- are the key.

7 Uniqueness of Optimal View-based Censors

The fact that the output of ComputeView is not uniquely determined can be problematic. For example, a CQE system that relies on this algorithm needs to ensure that the same view is used in different user sessions, as well as for different users for which equivalent policies apply. One could, of course, determinise the output of the algorithm using application-dependent heuristics; however, there may not be a good reason for choosing one possible view over the others.

Our first proposal is to impose further restrictions to the ontology language. Example 25 suggests that existence of multiple views is related to the presence of conjunction in the bodies of rules, which suggests the restriction given next.

Definition 27 (Linear RL$^-$ ontology). *An* RL^- *ontology* $\mathcal{O}$ *is* linear *if every rule in* $\mathcal{O}$ *contains exactly one atom in the body.*

Each RDFS ontology, such as $\mathcal{O}_{ex}$ in our running example, is also a linear RL^- ontology; hence, linearity might not be too strict a restriction for many Semantic Web applications. Note also that linear RL^- is a fragment of OWL 2 QL, in the sense that every linear RL^- rule can be transformed into an equivalent OWL 2 QL axiom. In contrast, non-linear RL^- is not captured by OWL 2 QL since it allows for conjunction in the body of rules. The following theorem shows that restricting ourselves to linear ontologies has the desired effect.

Theorem 28. *If* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *is a CQE-instance with* $\mathcal{O}$ *a linear* RL^- *ontology; then, each run of* ComputeView *on* $\mathbf{I}$ *yields the same result.*

Corollary 29. *Let* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *be a CQE-instance with* $\mathcal{O}$ *a linear* RL^- *ontology and let* $\mathcal{V}$ *be the result of a run of* ComputeView *on* $\mathbf{I}$*; then,* $\mathsf{cens}_{\mathcal{V}}$ *is the only optimal censor for* $\mathbf{I}$.

For applications where linearity is too strict, we can give up optimality in favour of uniqueness by taking the “intersection” of all optimal views.

Definition 30. *Let* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *be a CQE-instance with* $\mathcal{O}$ *an* RL^- *ontology. The* WIDTIO-view[3] *for* $\mathbf{I}$ *is the view defined as the following set of atoms:*

$$\{\textit{ground atom } \beta \mid \mathcal{O} \cup \mathcal{V} \models \beta \textit{ for each view } \mathcal{V} \textit{ for } \mathbf{I} \textit{ with } \mathsf{cens}_{\mathcal{V}} \textit{ optimal}\}.$$

[3] This solution is related to the well-known “When In Doubt Through It Out” (WIDTIO) approach [11] to knowledge base update.

The (non-optimal) censor based on the WIDTIO-view implements a "cautious" approach to confidentiality enforcement since it disregards all atoms that could possibly participate in the disclosure of an atom in $\mathcal{P}$. The following theorem shows, however, that this solution might be computationally expensive.

Theorem 31. *Deciding whether a ground atom β belongs to the WIDTIO-view for a CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ with $\mathcal{O}$ an RL^- ontology is* coNP-complete.

8 Related Work

The CQE paradigm was first proposed by Sichermann et al. [6], and was later extended by Biskup, Bonatti, Kraus and Subrahmanian [3,2,4,1]. CQE in the context of incomplete databases was studied by Biskup and Weibert [5]. These foundational works on CQE assume that both the information in the system and user queries are represented in propositional logic. Recently, Biskup and Bonatti studied CQE in relational databases where queries contain answer variables [12].

The formal study of data privacy and information hiding has received significant attention within the database community. Miklau and Suciu introduced *perfect privacy* in data exchange [13]. Rizvi et al. studied view-based authorisation mechanisms [14,15], and Deutsch et al. analysed the logical implications to privacy derived from publishing views of a database [16]. Formal data privacy frameworks have been proposed by Kifer et al. [17] and Evfimievski et. al. [18].

Privacy and information hiding in the context of ontologies has been investigated only recently. Information hiding at the schema level has been studied in [19,20]. Data privacy was studied for $\mathcal{EL}$ ontologies by Tao et al. [21]. Bao et al. introduced the notion of a privacy-preserving reasoner [22] and Stouppa et al. proposed a framework for data privacy in the context of $\mathcal{ALC}$ ontologies [23]. Finally, Calvanese et al. [24] proposed techniques for ontology access authorisation based on Zhang and Mendelzon's database authorisation views paradigm [15].

9 Conclusion and Future Work

We have proposed novel techniques for enforcing confidentiality in information systems that rely on RDF, SPARQL, and OWL 2 RL for representing data, queries, and ontologies, respectively. Our techniques ensure an optimal tradeoff between confidentiality and accessibility of information; furthermore, they can be efficiently implemented by relying on existing highly optimised OWL 2 RL triple stores. Our next step is to implement and test our algorithms, and we are also planning to consider the problem of view maintenance for applications that require very frequent changes to the data, such as data streaming applications. Finally, we will study how our results could be extended to the case where the ontology is expressed in either the QL, or the EL profile of OWL 2.

References

1. Biskup, J., Bonatti, P.A.: Controlled Query Evaluation for Enforcing Confidentiality in Complete Information Systems. Int. J. Inf. Sec. **3**(1) (2004) 14–27
2. Biskup, J., Bonatti, P.A.: Lying Versus Refusal for Known Potential Secrets. Data Knowl. Eng. **38**(2) (2001) 199–222
3. Bonatti, P.A., Kraus, S., Subrahmanian, V.S.: Foundations of Secure Deductive Databases. IEEE Trans. Knowl. Data Eng. **7**(3) (1995) 406–422
4. Biskup, J.: For Unknown Secrecies Refusal Is Better than Lying. Data Knowl. Eng. **33**(1) (2000) 1–23
5. Biskup, J., Weibert, T.: Keeping Secrets in Incomplete Databases. Int. J. Inf. Sec. **7**(3) (2008) 199–217
6. Sicherman, G.L., de Jonge, W., van de Riet, R.P.: Answering Queries Without Revealing Secrets. ACM Trans. Database Syst. **8**(1) (1983) 41–59
7. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles (2nd Edition) (2012) W3C Recommendation.
8. Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., Velkov, R.: OWLim: A Family of Scalable Semantic Repositories. Semantic Web J. **2**(1) (2011) 33–42
9. Wu, Z., Eadon, G., Das, S., Chong, E.I., Kolovski, V., Annamalai, M., Srinivasan, J.: Implementing an Inference Engine for RDFS/OWL Constructs and User-Defined Rules in Oracle. In: ICDE. (2008) 1239–1248
10. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (2012) W3C Recommendation.
11. Eiter, T., Gottlob, G.: On the Complexity of Propositional Knowledge Base Revision, Updates, and Counterfactuals. In: PODS. (1992) 261–273
12. Biskup, J., Bonatti, P.: Controlled Query Evaluation with Open Queries for a Decidable Relational Submodel. Ann. Math. and Artif. Intell. **50**(1-2) (2007) 39–77
13. Miklau, G., Suciu, D.: A Formal Analysis of Information Disclosure in Data Exchange. J. Comput. Syst. Sci. **73**(3) (2007) 507–534
14. Rizvi, S., Mendelzon, A.O., Sudarshan, S., Roy, P.: Extending Query Rewriting Techniques for Fine-Grained Access Control. In: SIGMOD, ACM (2004) 551–562
15. Zhang, Z., Mendelzon, A.O.: Authorization Views and Conditional Query Containment. In: ICDT. (2005) 259–273
16. Deutsch, A., Papakonstantinou, Y.: Privacy in Database Publishing. In: ICDT. (2005) 230–245
17. Kifer, D., Machanavajjhala, A.: A Rigorous and Customizable Framework for Privacy. In: PODS. (2012) 77–88
18. Evfimievski, A.V., Fagin, R., Woodruff, D.P.: Epistemic Privacy. In: PODS, ACM (2008) 171–180
19. Konev, B., Walther, D., Wolter, F.: Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In: IJCAI. (2009) 830–835
20. Cuenca Grau, B., Motik, B.: Reasoning over Ontologies with Hidden Content: The Import-by-Query Approach. J. Artif. Intell. Res. (JAIR) **45** (2012) 197–255
21. Tao, J., Slutzki, G., Honavar, V.: Secrecy-Preserving Query Answering for Instance Checking in $\mathcal{EL}$. In: RR. (2010) 195–203
22. Bao, J., Slutzki, G., Honavar, V.: Privacy-Preserving Reasoning on the Semantic Web. In: WI, IEEE Computer Society (2007) 791–797
23. Stouppa, P., Studer, T.: A Formal Model of Data Privacy. In: PSI. (2007)
24. Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: View-based Query Answering over Description Logic Ontologies. In: KR, AAAI Press (2008)

Appendix: Proofs

Proofs for Section 3.3

Proposition 14. *Let* cens *be a confidentiality preserving censor for a CQE-instance* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$*. Then,* cens *is optimal iff for each conjunctive query* $Q(\boldsymbol{x})$ *and each tuple* $\boldsymbol{t} \in \mathsf{cert}(Q, \mathcal{O}, \mathcal{D})$ *the fact that* $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \cup \{Q(\boldsymbol{t})\} \not\models \alpha$ *holds for each* $\alpha \in \mathcal{P}$ *implies that* $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \models Q(\boldsymbol{t})$*.*

Proof. We show the contrapositive of each direction. Assume that cens is not optimal; then, a censor $\mathsf{cens}' \neq \mathsf{cens}$ for $\mathbf{I}$ exists such that *(i)* cens' is confidentiality preserving and *(ii)* $\mathsf{cens}(Q) \subseteq \mathsf{cens}'(Q)$ for each conjunctive query Q. Condition *(ii)* implies that $\mathcal{F}(\mathsf{cens}) \subseteq \mathcal{F}(\mathsf{cens}')$; additionally, since $\mathsf{cens}' \neq \mathsf{cens}$ we can also conclude the existence of a conjunctive query Q and a tuple of constants $\boldsymbol{t}$ such that $\boldsymbol{t} \in \mathsf{cens}'(Q)$ and $\boldsymbol{t} \notin \mathsf{cens}(Q)$; as a result, $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \not\models Q(\boldsymbol{t})$, while $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}') \models Q(\boldsymbol{t})$, and therefore $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}') \models \mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \cup \{Q(\boldsymbol{t})\}$. From Condition (i), we conclude that $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}') \not\models \alpha$ for each $\alpha \in \mathcal{P}$, which then implies $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \not\models \alpha$; since $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \not\models Q(\boldsymbol{t})$ the condition in the proposition does not hold, as required.

Assume now that the condition in the proposition does not hold. Then, there exist a conjunctive query Q' and a tuple of constants $\boldsymbol{t} \in \mathsf{cert}(Q', \mathcal{O}, \mathcal{D})$ such that $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \cup \{Q'(\boldsymbol{t})\} \not\models \alpha$ for each $\alpha \in \mathcal{P}$, but $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \not\models Q'(\boldsymbol{t})$. Consider the following censor cens' for $\mathbf{I}$:

$$\mathsf{cens}'(Q) = \begin{cases} \mathsf{cens}(Q) & \text{if } Q \neq Q', \\ \mathsf{cens}(Q) \cup \{\boldsymbol{t}\} & \text{if } Q = Q'. \end{cases}$$

Clearly, it holds that $\mathsf{cens}(Q) \subseteq \mathsf{cens}'(Q)$ for each conjunctive query Q. Furthermore, since cens is confidentiality preserving, we can also conclude the same about cens'. Indeed, cens and cens' coincide for all queries except Q'; furthermore, $\mathcal{F}(\mathsf{cens}') = \mathcal{F}(\mathsf{cens}) \cup \{Q'(\boldsymbol{t})\}$ and $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \cup \{Q'(\boldsymbol{t})\} \not\models \alpha$ for each $\alpha \in \mathcal{P}$. As a result, cens is not optimal, as required. □

Proofs for Section 4

Proposition 18. *Let* $\mathbf{I}$ *be a CQE-instance and let* $\mathcal{V}$ *be a view for* $\mathbf{I}$*. Then* $\mathsf{cens}_{\mathcal{V}}$ *is a confidentiality preserving censor for* $\mathbf{I}$*.*

Proof. Consider an arbitrary $\alpha \in \mathcal{P}$; by Property *(i)* in Definition 16 we have $\mathcal{O} \cup \mathcal{V} \not\models \alpha$; furthermore, it is clear that $\mathcal{O} \cup \mathcal{V} \models \mathcal{F}(\mathsf{cens}_{\mathcal{V}})$ by the definition of $\mathcal{F}(\mathsf{cens}_{\mathcal{V}})$; but then, $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}}) \not\models \alpha$, which implies that $\mathsf{cens}_{\mathcal{V}}$ is confidentiality preserving for $\mathbf{I}$. □

Proofs for Section 5.1

Theorem 19. *There exists a CQE-instance* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *with an* RL *ontology* $\mathcal{O}$*, for which no censor exists that is both view-definable and optimal.*

Proof. Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ be the CQE-instance defined as follows:

$$\begin{aligned}
&\mathcal{O} = \{R(x_1, y) \land R(x_2, y) \to x_1 \approx x_2;\ R(x, y) \to A(y)\},\\
&\mathcal{D} = \{R(a, a)\},\\
&\mathcal{P} = \{A(a)\}.
\end{aligned}$$

Consider also the following (infinite) sequence of conjunctive queries (for $k \geq 1$):

$$Q_k(x) = \exists x_1, \ldots, x_k.\, R(x, x_1) \land \ldots \land R(x_{k-1}, x_k).$$

Clearly, a is the only certain answer to each $Q_k(x)$ w.r.t. $\mathcal{O}$ and $\mathcal{D}$. Next, we show the following properties:

(P1) if an interpretation M is such that $M \models \mathcal{O} \cup \{Q_k(a) | k \geq 1\}$ and $M \not\models A(a)$, then M has an infinite domain;
(P2) if cens is optimal for $\mathbf{I}$, then $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \models \{Q_k(a) | k \geq 1\}$.

To prove Property (P1), assume for the sake of contradiction that a finite model M of $\mathcal{O} \cup \{Q_k(a) | k \geq 1\}$ exists such that $M \not\models A(a)$. Let m be the size of the interpretation domain of M. Since $M \models Q_m(a)$, there must exist elements $a_0, a_1, \ldots, a_{m-1}, a_m$ in the domain of M such that $a_0 = a$, and $(a_0, a_1), \ldots, (a_{m-1}, a_m) \in R^M$.[4] Since the size of the domain is just m, there must exist objects a_i and a_j for $0 \leq i < j \leq m$ such that $a_i = a_j$ and as a result $(a_{j-1}, a_i) \in R^M$ and $(a_{i-1}, a_i) \in R^M$. Since $\mathcal{O}$ axiomatises R to be an inverse-functional relation, we have $a_{i-1} = a_{j-1}$ as well. By the same reasoning, we have that $a_{i-2} = a_{j-2}, \ldots, a_0 = a_{j-i}$. Hence, $(a_{j-i-1}, a_0) \in R^M$ and $a \in A^M$, which contradicts our assumption that $M \not\models A(a)$.

To prove Property (P2), consider an arbitrary confidentiality preserving censor cens for $\mathbf{I}$. It suffices to show that the censor cens' defined such that $\mathcal{F}(\mathsf{cens}') = \mathcal{F}(\mathsf{cens}) \cup \{Q_k(a) | k \geq 1\}$ is also a confidentiality preserving censor for $\mathbf{I}$. Since cens is assumed to be confidentiality preserving, it holds that $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}) \not\models A(a)$; thus, there exists an interpretation M such that $M \models \mathcal{O} \cup \mathcal{F}(\mathsf{cens})$, but $M \not\models A(a)$. Let M' be the (infinite) interpretation over elements of M and fresh elements a_k, $k \geq 1$, defined as follows:

$$\begin{aligned}
R^{M'} &= R^M \cup \{(a, a_1)\} \cup \{(a_k, a_{k+1}) | k \geq 1\},\\
A^{M'} &= A^M \cup \{a_k | k \geq 1\}.
\end{aligned}$$

Clearly, $M' \models \mathcal{O}$ because M does and every new domain element belongs to $A^{M'}$. Also, $M' \models \mathcal{F}(\mathsf{cens})$ because M does and M' simply extends M with new information (remember that $\mathcal{F}(\mathsf{cens})$ consists of existentially quantified positive formulas). Also, $M' \models \{Q_k | k \geq 1\}$ and $M' \not\models A(a)$; thus, $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}') \not\models A(a)$, as required.

Next, we use properties (P1) and (P2) to show the claim of the theorem. Assume for the sake of contradiction that there exists a censor $\mathsf{cens}_{\mathcal{V}}$ based on

[4] Here and below in similar cases R^M denotes the interpretation of R in M.

some view $\mathcal{V}$ for $\mathbf{I}$ that is optimal. Since $\mathsf{cens}_{\mathcal{V}}$ is optimal, Property (P2) implies that $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}}) \models \{Q_k(a) | k \geq 1\}$; since $\mathsf{cens}_{\mathcal{V}}$ is based on $\mathcal{V}$, we also have $\mathcal{O} \cup \mathcal{V} \models \{Q_k(a) | k \geq 1\}$. Clearly, $\mathcal{O} \cup \mathcal{V}$ is satisfiable (otherwise $\mathcal{O} \cup \mathcal{V} \models A(a)$ and the censor would not be confidentiality preserving), $\mathcal{O} \cup \mathcal{V}$ must have a least Herbrand model H, which is finite. Since $\mathcal{O} \cup \mathcal{V} \models \{Q_k(a) | k \geq 1\}$ and $H \models \mathcal{O} \cup \mathcal{V}$ we must also have that $H \models \mathcal{O} \cup \{Q_k | k \geq 1\}$. However, $H \not\models A(a)$ since cens is confidentiality preserving, which together with the fact that H is finite contradicts Property (P1). □

Proofs for Section 5.2

Theorem 21. *There exists a sequence of CQE-instances $\mathbf{I}_n = (\mathcal{O}_n, \mathcal{D}_n, \mathcal{P})$ for $n \geq 1$ such that $\mathcal{O}_n$ is an equality-free* RL *ontology, $|\mathcal{O}_n| \in O(n)$, $|\mathcal{D}_n| \in O(n)$, and each view $\mathcal{V}$ for $\mathbf{I}_n$ with $\mathsf{cens}_{\mathcal{V}}$ optimal is such that $|\mathcal{V}| \in \Omega(2^n)$.*

Proof. We first provide a proof that use rules in RL ontologies which use unary atoms with branch variables, and then describe how to adopt it to the case of ontologies using rules of the form 2(b) from Definition 4.

For each number $n \geq 1$, let $\mathbf{I}_n = (\mathcal{O}_n, \mathcal{D}_n, \mathcal{P})$ be the CQE-instance defined as follows:

$$\begin{aligned}
\mathcal{O}_n &= \{A_1^i(x) \wedge A_2^i(x) \wedge R(x,y) \rightarrow B(y) \mid 1 \leq i \leq n\},\\
\mathcal{D}_n &= \{R(a,b)\} \cup \{A_j^i(a) \mid 1 \leq i \leq n, 1 \leq j \leq 2\},\\
\mathcal{P} &= \{B(b)\}.
\end{aligned}$$

Clearly, it holds that $|\mathcal{D}_n| \in O(n)$ and $|\mathcal{O}_n| \in O(n)$. Consider the following queries for each $n \geq 1$:

$$Q^n_{j_1,\ldots,j_n}(y) = \exists x.\, A^1_{j_1}(x) \wedge \ldots \wedge A^n_{j_n}(x) \wedge R(x,y), \text{ such that } 1 \leq j_1, ..., j_n \leq 2.$$

Clearly, we have that b is the only certain answer to $Q^n_{j_1,\ldots,j_n}(y)$ w.r.t. $\mathcal{O}_n$ and $\mathcal{D}_n$, regardless of the values of n and $j_1, \ldots, j_n$.

Take an arbitrary n, and consider a view $\mathcal{V}$ for $\mathbf{I}_n$. The following property holds:

$$\mathcal{O} \cup \mathcal{V} \cup \{Q^n_{j_1,\ldots,j_n}(b) \mid 1 \leq j_1, ..., j_n \leq 2\} \not\models B(b).$$

As a result, if $\mathsf{cens}_{\mathcal{V}}$ is optimal, it must be the case that $\mathcal{O} \cup \mathcal{V} \models \{Q^n_{j_1,\ldots,j_n}(b) \mid 1 \leq j_1, ..., j_n \leq 2\}$. Hence, for each $1 \leq j_1, ..., j_n \leq 2$ the view $\mathcal{V}$ contains a witnessing element $a_{j_1,\ldots,j_n}$, such that $A^i_{j_i}(a_{j_1,\ldots,j_n})$, $1 \leq i \leq n$, and $R(a_{j_1,\ldots,j_n}, b)$ hold in $\mathcal{V}$. However, all these $a_{j_1,\ldots,j_n}$ are pairwise different, since otherwise $A^i_1(a_{j_1,\ldots,j_n}), A^i_2(a_{j_1,\ldots,j_n})$, and $R(a_{j_1,\ldots,j_n}, b)$ hold in $\mathcal{V}$ for some number i, and it would mean that $\mathcal{O} \cup \mathcal{V} \models B(b)$ (which is a contradiction tho the fact that $\mathsf{cens}_{\mathcal{V}}$ is confidentiality preserving). Since, there are 2^n different elements $a_{j_1,\ldots,j_n}$ in $\mathcal{D}_n$ and the choice of $\mathcal{V}$ was arbitrary we finally obtain $|\mathcal{V}| \in \Omega(2^n)$. This ends a first proof of the theorem.

The theorem could also have been proved in an analogous way by considering the following family $\mathbf{I}_n = (\mathcal{O}_n, \mathcal{D}_n, \mathcal{P})$ of CQE-instances instead:

$$\begin{array}{lcr} \mathcal{O}_n & = & \{S_1^i(x_1,x_2) \wedge S_2^i(x_2,x_3) \to R(x_1,x_3) \mid 0 \leq i \leq n+1\}, \\ \mathcal{D}_n & = & \{S_1^i(a,b), S_2^i(b,c) \mid 0 \leq i \leq n+1\}, \\ \mathcal{P} & = & \{R(a,c)\}. \end{array}$$

In this case, we can consider the following queries, where $1 \leq j_1, ..., j_n \leq 2$:

$$Q^n_{j_1,\dots,j_n}(x,z) = \\ \exists y.\, S_1^0(x_1,x_2) \wedge S^1_{j_1}(x_{j_1}, x_{j_1+1}) \wedge \ldots \wedge S^n_{j_n}(x_{j_n}, x_{j_n+1}) \wedge S_2^{n+1}(x_2,x_3).$$

The answer (a,c) must me returned for each of these queries by each optimal view-based censor. Using similar arguments as before, we can again prove an exponential lower bound in the number of “witnessing” constants in the view. □

Proofs for Section 6

Theorem 26. *Let $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ be a valid input of the algorithm* ComputeView*.*

(i) If $\mathcal{V}$ is the result of a run of ComputeView *on* $\mathbf{I}$ *then* $\mathsf{cens}_\mathcal{V}$ *is an optimal censor for* $\mathbf{I}$*.*

(ii) For each optimal censor cens *for* $\mathbf{I}$*, there exists a run of* ComputeView *on* $\mathbf{I}$ *that computes $\mathcal{V}$ such that* $\mathsf{cens} = \mathsf{cens}_\mathcal{V}$*.*

(iii) The algorithm can be implemented to run in time polynomial in $|\mathcal{D}| + |\mathcal{O}|$.

Proof. To prove Statement *(i)* we first show that $\mathcal{V}$ is indeed a view for $\mathbf{I}$, and then the optimality of $\mathsf{cens}_\mathcal{V}$.

By the construction in Lines 8-12 of the algorithm, Requirement *(i)* from Definition 16 of a view holds for $\mathcal{V}$. Also, since $\mathcal{V} \subseteq \mathcal{D}_{\mathsf{sat}}$, where $\mathcal{D}_{\mathsf{sat}}$ is the dataset constructed after Step 6 of the run, for any conjunctive query Q and any tuple $\boldsymbol{t}$ of constants from $\mathcal{D}$ such that $\boldsymbol{t} \in \mathsf{cert}(Q,\mathcal{O},\mathcal{V})$, it holds that $\boldsymbol{t} \in \mathsf{cert}(Q,\mathcal{O},\mathcal{D}_{\mathsf{sat}})$. Since $\boldsymbol{t}$ contains only the constants from $\mathcal{D}$, from the construction we have that $\mathsf{cert}(Q,\mathcal{O},\mathcal{D})$ and $\mathsf{cert}(Q,\mathcal{O},\mathcal{D}_{\mathsf{sat}})$ contain the same tuples of constants from $\mathcal{D}$, which implies, in particular, that $\boldsymbol{t} \in \mathsf{cert}(Q,\mathcal{O},\mathcal{D})$. The last means that Requirement *(ii)* also holds. So, $\mathcal{V}$ is indeed a view for $\mathbf{I}$.

By the Proposition 18, the censor $\mathsf{cens}_\mathcal{V}$ is confidentiality preserving. So, by Proposition 14, to show the optimality of $\mathsf{cens}_\mathcal{V}$ it is enough to prove that for each conjunctive query Q and each tuple $\boldsymbol{t} \in \mathsf{cert}(Q,\mathcal{O},\mathcal{D})$, the following holds.

$$\text{If } \mathcal{O} \cup \mathcal{F}(\mathsf{cens}_\mathcal{V}) \cup \{Q(\boldsymbol{t})\} \not\models \alpha \text{ for every } \alpha \in \mathcal{P}, \text{ then } \mathcal{O} \cup \mathcal{F}(\mathsf{cens}_\mathcal{V}) \models Q(\boldsymbol{t}). \ (\dagger)$$

Assume, for the sake of contradiction, that there exists $Q(\boldsymbol{x}) = \exists \boldsymbol{y}.\, \varphi(\boldsymbol{x},\boldsymbol{y})$ and $\boldsymbol{t} \in \mathsf{cert}(Q,\mathcal{O},\mathcal{D})$ such that (†) does not hold. Since $\boldsymbol{t} \in \mathsf{cert}(Q,\mathcal{O},\mathcal{D})$, there exists a mapping h' from $\boldsymbol{y}$ to the domain of $\mathcal{D}$, such that $\mathcal{O} \cup \mathcal{D} \models \varphi(\boldsymbol{t}, h'(\boldsymbol{y}))$. Let h be a mapping from $\boldsymbol{y}$ to the domain of $\mathcal{D}_{\mathsf{sat}}$ (where again $\mathcal{D}_{\mathsf{sat}}$ is the dataset constructed after Step 6 of the run), such that $h(y) = an_a$ iff $h'(y) = a$ for every constant a and variable y. Clearly, $\mathcal{D}_{\mathsf{sat}} \models \varphi(\boldsymbol{t}, h(\boldsymbol{y}))$. However, $\mathcal{O} \cup \mathcal{V} \not\models \varphi(\boldsymbol{t}, h(\boldsymbol{y}))$, since $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}}) \not\models Q(\boldsymbol{t})$. Hence, there exists a ground atom $\beta \in \varphi(\boldsymbol{t}, h(\boldsymbol{y}))$ such that $\mathcal{O} \cup \mathcal{V} \not\models \beta$, and in particular, $\beta \notin \mathcal{V}$. Though, since Step 11 is passed, there exists a ground atom $\alpha \in \mathcal{P}$ such that $\mathcal{O} \cup \mathcal{V} \cup \{\beta\} \models \alpha$.

Consider first the case when α is a binary atom. Since $\mathcal{O}$ is an RL$^-$ ontology and contains only rules of the forms *2(a)* and *2(c)* from Definition 4 to deduce binary atoms, there exists a single binary atom γ in $\mathcal{V} \cup \{\beta\}$ over the same constants as α (from $\mathcal{D}$), such that $\mathcal{O} \cup \{\gamma\} \models \alpha$. If $\gamma \in \mathcal{V}$, then $\mathcal{O} \cup \mathcal{V} \models \alpha$, i.e., in particular, $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}}) \cup \{Q(\boldsymbol{t})\} \models \alpha$; which, however, contradicts to the original assumption that (†) does not hold. If $\gamma = \beta$, then by the construction of β it holds that $Q(\boldsymbol{t}) = \beta$, since this is the only way to obtain β which use only constants from $\mathcal{D}$; however, then again we have the contradicting $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}}) \cup \{Q(\boldsymbol{t})\} \models \alpha$. So, α cannot be a binary atom.

The case when α is a unary atom can be considered similarly to the binary case. In particular, the fact that $\mathcal{O}$ is an RL$^-$ ontology and hence contains only rules which do not mention branch variables in unary atoms, guarantees, that there exist a set of atoms $\gamma_1, \dots, \gamma_k$ in $\mathcal{V} \cup \{\beta\}$ using the same constant as α (from $\mathcal{D}$), such that $\mathcal{O} \cup \{\gamma\} \models \alpha$. Regardless of whether all γ_i are in $\mathcal{V}$ or there is one equal to β, we can deduce a contradiction $\mathcal{O} \cup \mathcal{F}(\mathsf{cens}_{\mathcal{V}}) \cup \{Q(\boldsymbol{t})\} \models \alpha$ with (†).

This ends the proof of Statement *(i)* of the theorem.

To prove Statement *(ii)*, consider an arbitrary censor cens which is optimal for $\mathbf{I}$. We need to show that there exists a run of $\mathsf{ComputeView}$ with the input $\mathbf{I}$ and an output $\mathcal{V}$ such that $\mathsf{cens} = \mathsf{cens}_{\mathcal{V}}$.

For each tuple $\boldsymbol{t}$ of elements of the domain of $\mathcal{D}$ and each conjunctive query $Q(\boldsymbol{x}) = \exists \boldsymbol{y}.\, \varphi(\boldsymbol{x}, \boldsymbol{y})$ such that $Q(\boldsymbol{t}) \in \mathcal{F}(\mathsf{cens})$, consider all the homomorphisms h from $\varphi(\boldsymbol{x}, \boldsymbol{y})$ to $\mathcal{D}_{\mathsf{sat}}$, where $\mathcal{D}_{\mathsf{sat}}$ is again the dataset after the step of the run, such that $h(\boldsymbol{x}) = \boldsymbol{t}$. Denote $\mathcal{V}$ the dataset which consists of all the images of all homomorphisms h for all conjunctive queries Q and tuples $\boldsymbol{t}$ such that $Q(\boldsymbol{t}) \in \mathcal{F}(\mathsf{cens})$.

Assume, for the sake of contradiction, that there exists an atom α in $\mathcal{P}$ such that $\mathcal{O} \cup \mathcal{V} \models \alpha$. Consider the case when α is a unary atom $A(a)$ (the binary case can be handled similarly). Since cens is a confidentiality preserving censor for $\mathbf{I}$, the atom $A(a)$ does not belong to $\mathcal{V}$. Hence, there exists a rule $\forall x \forall \boldsymbol{z}.\, \varphi'(x, \boldsymbol{z}) \to A(x)$ in $\mathcal{O}$ such that $\mathcal{O} \cup \mathcal{V} \models \exists \boldsymbol{z}.\, \varphi'(a, \boldsymbol{z})$. Since $\mathcal{O}$ is an RL$^-$ ontology, $\varphi'(a, \boldsymbol{z})$ is a conjunction of unary atoms $B(a)$, binary atoms $R(a, z)$, and binary atoms $S(z, a)$ such that each z from $\boldsymbol{z}$ appears in only one binary atom. Again, consider only the binary atoms of the form $S(z, a)$, and the other cases are similar. It holds that $\mathcal{O} \cup \mathcal{V} \models \exists z.\, S(z, a)$. The only possibility is that the sentence $\exists z.\, S(z, a)$ belongs to $\mathcal{F}(\mathsf{cens})$. Summarising over all atoms of $\varphi'(a, \boldsymbol{z})$, we have that $\mathcal{F}(\mathsf{cens}) \models \exists \boldsymbol{z}.\, \varphi'(a, \boldsymbol{z})$. Since $\mathcal{F}(\mathsf{cens})$ is closed under $\mathcal{O}$-entailment, the atom $A(a)$ belongs to $\mathcal{F}(\mathsf{cens})$. This contradicts to the fact that cens is a confidentiality preserving censor for $\mathbf{I}$.

From the fact above it follows that $\mathsf{cens}_{\mathcal{V}}$ is a confidentiality preserving censor for $\mathbf{I}$. From the construction for every conjunctive query Q, it holds that

$\mathsf{cens}(Q) \subseteq \mathsf{cens}_{\mathcal{V}}(Q)$. However, if this inclusion is strict or $\mathcal{V}$ is not maximal, then cens is not optimal.

So, we conclude that $\mathsf{cens}(Q) = \mathsf{cens}_{\mathcal{V}}(Q)$ for $\mathcal{V}$ which is possible to obtain as an output of the algorithm $\mathsf{ComputeView}$ with the input $\mathbf{I}$.

The only thing which is left to prove is that $\mathsf{ComputeView}$ can be implemented to run in time polynomial in $|\mathcal{D}| + |\mathcal{O}|$. It follows directly from the following observations.

- Creating constants an_a (Line 2) takes time linear in the number of constants in $\mathcal{D}$.
- Adding new facts to $\mathcal{D}_{\mathsf{an}}$ (Lines 3-4) takes time linear in $|\mathcal{D}|$. Note that the size of $\mathcal{D}_{\mathsf{an}}$ is linear in the size of $\mathcal{D}$.
- The saturation of $\mathcal{D}_{\mathsf{sat}}$ (Line 6) takes time polynomial in $|\mathcal{D}| + |\mathcal{O}|$. Indeed, the number of atoms β over the signature of $\mathcal{D} \cup \mathcal{D}_{\mathsf{an}}$ is quadratic in the size of $\mathcal{D} \cup \mathcal{D}_{\mathsf{an}}$; the check whether $\mathcal{O} \cup \mathcal{D} \cup \mathcal{D}_{\mathsf{an}} \models \beta$ can be performed in time polynomial in the size of $\mathcal{O} \cup \mathcal{D} \cup \mathcal{D}_{\mathsf{an}}$, and consequently in $|\mathcal{D}| + |\mathcal{O}|$.
- Building $\mathcal{V}$ (Lines 8-11) is polynomial in $|\mathcal{D}| + |\mathcal{O}|$. Indeed, number of checks whether $\mathcal{O} \cup \mathcal{V} \cup \{\beta\} \not\models \alpha$ (Line 9) is bounded by the number of facts in $\mathcal{D}_{\mathsf{sat}}$, which is in the worst case quadratic in $|\mathcal{D}| + |\mathcal{O}|$. The check itself also can be performed in time polynomial in $|\mathcal{D}| + |\mathcal{O}|$. □

Proofs for Section 7

Theorem 28. *If* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *is a CQE-instance with* $\mathcal{O}$ *a linear* RL^- *ontology; then, each run of* $\mathsf{ComputeView}$ *on* $\mathbf{I}$ *yields the same result.*

Proof. Recall that $\mathcal{O}$ is a linear RL^- ontology, that is, each rule in $\mathcal{O}$ contains exactly one atom in its body. Thus, if $\mathcal{O} \cup \mathcal{D}' \models \alpha$, then there exists a single atom β in $\mathcal{D}'$ such that $\mathcal{O} \cup \{\beta\} \models \alpha$, where $\mathcal{D}'$ is a dataset and α a ground atom.

Let $\mathcal{D}_{\mathsf{sat}}$ be the dataset constructed after Step 6 of the run of the algorithm. We define the dataset $\mathcal{D}_{\mathsf{min}}$ as follows:

$$\mathcal{D}_{\mathsf{min}} = \{\beta \mid \beta \in \mathcal{D}_{\mathsf{sat}} \text{ and } \mathcal{O} \cup \{\beta\} \models \alpha \text{ for some } \alpha \in \mathcal{P}\}.$$

Clearly, $\mathcal{V} = \mathcal{D}_{\mathsf{sat}} \setminus \mathcal{D}_{\mathsf{min}}$ is a view for $\mathbf{I}$; moreover, there is a run of $\mathsf{ComputeView}$ on $\mathbf{I}$ such that $\mathcal{V}$ is returned as output. Assume that the result of $\mathsf{ComputeView}$ on $\mathbf{I}$ is not unique and there is another possible output $\mathcal{V}' \neq \mathcal{V}$. Clearly, $\mathcal{V}' \cap \mathcal{D}_{\mathsf{min}} \neq \emptyset$; therefore, $\mathcal{O} \cup \mathcal{V}' \models \alpha$ for some $\alpha \in \mathcal{P}$, which contradicts the assumption that $\mathcal{V}'$ is a possible output of $\mathsf{ComputeView}$ on $\mathbf{I}$. □

Corollary 29. *Let* $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ *be a CQE-instance with* $\mathcal{O}$ *a linear* RL^- *ontology and let* $\mathcal{V}$ *be the result of a run of* $\mathsf{ComputeView}$ *on* $\mathbf{I}$*; then,* $\mathsf{cens}_{\mathcal{V}}$ *is the only optimal censor for* $\mathbf{I}$.

Proof. By Theorem 28, each run of $\mathsf{ComputeView}$ computes the same result and hence all runs of $\mathsf{ComputeView}$ yield the same result $\mathcal{V}$. But then, the claim in the corollary follows directly from Statement *(ii)* in Theorem 26. □

Theorem 31. *Deciding whether a ground atom β belongs to the WIDTIO-view for a CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ with $\mathcal{O}$ an* RL^{-} *ontology is* coNP-complete.

Proof. Let $\mathcal{V}_{\mathrm{W}}$ be the WIDTIO-view for $\mathbf{I}$. To show coNP-membership of our decision problem, we prove that its complement is in NP. For this, to check whether a ground atom β does not belong to $\mathcal{V}_{\mathrm{W}}$, we can use the following guess-and-check algorithm.

1. Guess a dataset $\mathcal{D}'$ mentioning only predicates and constants from $\mathcal{D}$.
2. Check that the following conditions hold:
   (a) $\mathcal{O} \cup \mathcal{D} \models \mathcal{D}'$;
   (b) $\mathcal{O} \cup \mathcal{D}' \not\models \beta$; and
   (c) for each ground atom γ such that $\mathcal{O} \cup \mathcal{D} \models \gamma$ and $\mathcal{O} \cup \mathcal{D}' \not\models \gamma$, we have $\mathcal{O} \cup \mathcal{D}' \cup \{\gamma\} \models \alpha$ for some ground atom $\alpha \in \mathcal{P}$.

Since $\mathcal{O}$ is an OWL 2 RL ontology, $\mathcal{D}'$ can be at most of polynomial size in $|\mathcal{O}| + |\mathcal{D}|$, and conditions 2(a), 2(b) and 2(c) can be checked in polynomial time.

Next, we show the coNP-hardness of our problem by reducing its complement to the 3SAT problem. Let ψ be an arbitrary propositional formula

$$\bigwedge_{i=1}^{n} (L_{i1} \vee L_{i2} \vee L_{i3}),$$

where each L_{ij} is a propositional literal (i.e., either a propositional variable or its negation). Let $\{x_1, \ldots, x_m\}$ are all the propositional variables occurring in ψ.

Next, we construct a CQE-instance $\mathbf{I} = (\mathcal{O}, \mathcal{D}, \mathcal{P})$ that we can use to determine the satisfiability of ψ. The ontology $\mathcal{O} = \mathcal{O}_{assign} \cup \mathcal{O}_{\psi}$ is defined as follows:

$$\begin{aligned} \mathcal{O}_{assign} &= \bigcup_{k=1}^{m} \{A_k^T(x) \wedge A_k^F(x) \to B(x)\}, \text{ and} \\ \mathcal{O}_{\psi} &= \bigcup_{i=1}^{n} \bigcup_{j=1}^{3} \{C_{i-1}(x) \wedge A_{ij}(x) \to C_i(x)\}, \text{ where} \\ A_{ij} &= \begin{cases} A_k^T, & \text{if } L_{ij} = x_k \text{ for some } k \in \{1, \ldots, m\}, \\ A_k^F, & \text{if } L_{ij} = \neg x_k \text{ for some } k \in \{1, \ldots, m\}. \end{cases} \end{aligned}$$

Furthermore, the dataset $\mathcal{D}$ is $\{C_0(a)\} \cup \bigcup_{k=1}^{m} \{A_k^T(a), A_k^F(a)\}$ and the policy is given by $\mathcal{P} = \{B(a), C_n(a)\}$.

It is easy to see that translation of ψ into $\mathbf{I}$ is polynomial in the size of ψ. Next we show the following property:

$$C_0(a) \notin \mathcal{V}_{\mathrm{W}} \text{ if and only if there is an assignment } \nu \text{ s.t. } \nu(\psi) = \mathbf{True}.$$

Specifically, we show now that *(1)* if there is an assignment ν such that $\nu(\psi) = \mathbf{True}$, then $C_0(a) \notin \mathcal{V}_{\mathrm{W}}$; and *(2)* if for each assignment ν, it holds that $\nu(\psi) = \mathbf{False}$, then $C_0(a) \in \mathcal{V}_{\mathrm{W}}$.

(1) Assume that there is an assignment ν such that $\nu(\psi) = \mathbf{True}$. We are going to construct a view $\mathcal{V}$ such that $\mathcal{O} \cup \mathcal{V} \not\models C_0(a)$ and $\mathsf{cens}_{\mathcal{V}}$ is optimal for $\mathbf{I}$. The existence of this $\mathcal{V}$ yields that $C_0(a) \notin \mathcal{V}_{\mathrm{W}}$ due to Definition 30.

By Theorem 26, there is a run of $\mathsf{ComputeView}$ on $\mathbf{I}$ resulting into $\mathcal{V}$. We are going to show that $\mathcal{V}$ is as follows:

$$\mathcal{V} = \mathcal{D}_{\mathsf{sat}} \setminus \left(\bigcup_{\nu(x_k)=\mathbf{True}} \{A_k^F(a)\} \cup \bigcup_{\nu(x_k)=\mathbf{False}} \{A_k^T(a)\} \right) \setminus \bigcup_{i=0}^{n} \{C_i(a)\} \setminus \{B(a)\},$$

$$\text{where } \mathcal{D}_{\mathsf{sat}} = \bigcup_{x \in \{a, an_a\}} \left(\bigcup_{k=1}^{m} \{A_k^T(x), A_k^F(x)\} \cup \bigcup_{i=0}^{n} \{C_i(x)\} \right).$$

($\mathcal{D}_{\mathsf{sat}}$ is the dataset constructed after Step 6 of the run).

To prove Statement **(1)**, it suffices to show that $\mathcal{V}$ is a maximal subset of $\mathcal{D}_{\mathsf{sat}}$ such that $\mathcal{O} \cup \mathcal{V} \not\models \alpha$ for each $\alpha \in \mathcal{P}$. First, we show that $\mathcal{O} \cup \mathcal{V} \not\models \alpha$ for each $\alpha \in \mathcal{P}$.

$\alpha = B(a)$. Due to the construction of $\mathcal{V}$, it holds that $\mathcal{O} \cup \mathcal{V} \not\models B(a)$.

$\alpha = C_n(a)$. Assume by contrast that $\mathcal{O} \cup \mathcal{V} \models C_n(a)$. Since $C_n(a) \notin \mathcal{V}$ due to the construction of $\mathcal{V}$ and the rules $C_{n-1}(x) \wedge A_{nj}(x) \to C_n(x)$ are in $\mathcal{O}$ for $j \in \{1, 2, 3\}$, we conclude that $\mathcal{O} \cup \mathcal{V} \models C_{n-1}(a)$ and $\mathcal{O} \cup \mathcal{V} \models A_{nj}(a)$ for some $j \in \{1, 2, 3\}$. Observe that $C_{n-1}(a) \notin \mathcal{V}$, so we conclude that $\mathcal{O} \cup \mathcal{V} \models C_{n-2}(a)$ applying the same arguments as above. Continuing this way, one can show that $\mathcal{O} \cup \mathcal{V} \models C_0(a)$; since there is no rule with its head of the form $C_0(x)$, it holds that $C_0(a) \in \mathcal{V}$, which contradicts the construction of $\mathcal{V}$. Thus, $\mathcal{O} \cup \mathcal{V} \not\models C_n(a)$.

It remains to show the maximality of $\mathcal{V}$, i.e., that for every atom $\beta \in \mathcal{D}_{\mathsf{sat}} \setminus \mathcal{V}$, it holds that $\mathcal{O} \cup \mathcal{V} \cup \{\beta\} \models \alpha$ for some $\alpha \in \mathcal{P}$. If $\beta \in \mathcal{P}$, then this holds trivially. Assume that $\beta \notin \mathcal{P}$; we have the following options:

$\beta = A_k^T(a)$ (resp., $\beta = A_k^F(a)$) for some $k \in \{1, \ldots, m\}$. Clearly, by construction of $\mathcal{V}$, we have that $\mathcal{O} \cup \mathcal{V} \cup \{\beta\} \models B(a)$, which proves the maximality of $\mathcal{V}$.

$\beta = C_{i-1}(a)$ for some $i \in \{1, \ldots, n\}$; then $\mathcal{O} \cup \mathcal{V} \cup \{C_{i-1}(a)\} \models C_i(a)$. Indeed, the rules $C_{i-1}(x) \wedge A_{ij}(x) \to C_i(x)$ are in $\mathcal{O}$ for $j \in \{1, 2, 3\}$, and $A_{ij}(a) \in \mathcal{V}$ for some $j \in \{1, 2, 3\}$. The latter membership follows from the following observation. By assumption, $\nu(\psi) = \mathbf{True}$, that is $\nu(L_{i1} \vee L_{i2} \vee L_{i3}) = \mathbf{True}$, which means that $\nu(L_{ij}) = \mathbf{True}$ for some $j \in \{1, 2, 3\}$. If $L_{ij} = x_k$ for some $k \in \{1, \ldots, m\}$, then $A_{ij} = A_k^T$ and $A_k^T(a) \in \mathcal{V}$ by construction of $\mathcal{V}$. If $L_{ij} = \neg x_k$, then $A_{ij} = A_k^F$ and $A_k^F(a) \in \mathcal{V}$.
Continuing this way, one can easily show that $\mathcal{O} \cup \mathcal{V} \cup \{C_{i-1}(a)\} \models C_n(a)$, which proves the maximality of $\mathcal{V}$.

This concludes the proof of Statement **(1)**.

(2) Assume that there is no assignment ν such that $\nu(\psi) = \mathbf{True}$. We are going to show that, for each view $\mathcal{V}$ such that $\mathsf{cens}_{\mathcal{V}}$ is optimal, it holds that $\mathcal{O} \cup \mathcal{V} \models C_0(a)$.

Let $\mathcal{V}'$ be such a view; by Theorem 26, there is a run of $\mathsf{ComputeView}$ on $\mathbf{I}$ resulting into $\mathcal{V}'$. Observe that

$$\text{at most one of } A_k^T(a) \text{ and } A_k^F(a) \text{ belongs to } \mathcal{V}' \tag{4}$$

due to the fact that $\mathcal{O} \cup \{A_k^T(a), A_k^F(a)\} \models B(a)$ and $\mathcal{O} \cup \mathcal{V}' \not\models B(a)$ by construction of $\mathcal{V}'$. We define an assignment ν' as follows:

$$\nu'(x_k) = \begin{cases} \textbf{True}, \text{ if } A_k^T \in \mathcal{V}', \\ \textbf{False}, \text{otherwise.} \end{cases}$$

Assume by contrast that $C_0(a) \notin \mathcal{V}'$; then it holds that $\mathcal{O} \cup \mathcal{V}' \cup \{C_0(a)\} \models \alpha$ for some $\alpha \in \mathcal{P}$. Obviously, this α can equal only $C_n(a)$. This means that there is a sequence of rules from $\mathcal{O}$

$$\begin{gathered} C_0(x) \wedge A_{1j_1}(x) \models C_1(x), \\ \dots \\ C_{n-1}(x) \wedge A_{nj_n}(x) \models C_n(x) \end{gathered}$$

with each $j_i \in \{1, 2, 3\}$ such that each $A_{ij_i}(a) \in \mathcal{V}'$. By the assumption of Statement **(2)**, $\nu'(\psi) = \textbf{False}$, that is, for some $i \in \{1, \dots, n\}$, it holds that $\nu'(L_{i1}) = \nu'(L_{i2}) = \nu'(L_{i3}) = \textbf{False}$. From the definitions of ν' and A_{ij} and Observation (4), it is easy to see that $A_{i1}(a)$, $A_{i2}(a)$, and $A_{i3}(a)$ are not in $\mathcal{V}'$, which contradicts the assumption that $\mathcal{O} \cup \mathcal{V}' \cup \{C_0(a)\} \models C_n(a)$. The obtained contradiction proves that $C_0(a) \in \mathcal{V}'$ and therefore $C_0(a) \in \mathcal{V}_{\mathrm{W}}$. This concludes the proof of Statement **(2)** and consequently of the theorem. □