Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Indistinguishability in controlled query evaluation over prioritized description logic ontologies

Gianluca Cima [a], Domenico Lembo [a,*], Lorenzo Marconi [a], Riccardo Rosati [a], Domenico Fabio Savo [b]

[a] Sapienza Università di Roma, Italy
[b] Università degli Studi di Bergamo, Italy

ARTICLE INFO

Keywords:
Description logics
Ontologies
Confidentiality preservation
Query answering
Data complexity

ABSTRACT

In this paper we study *Controlled Query Evaluation (CQE)*, a declarative approach to privacy-preserving query answering over databases, knowledge bases, and ontologies. CQE is based on the notion of *censor*, which defines the answers to each query posed to the data/knowledge base. We investigate both semantic and computational properties of CQE in the context of OWL ontologies, and specifically in the description logic DL-Lite$_\mathcal{R}$, which underpins the OWL 2 QL profile. In our analysis, we focus on semantics of CQE based on censors (called *optimal GA censors*) that enjoy the so-called *indistinguishability* property, analyzing the trade-off between maximizing the amount of data disclosed by query answers and minimizing the computational cost of privacy-preserving query answering. We first study the data complexity of *skeptical entailment* of unions of conjunctive queries under all the optimal GA censors, showing that the computational cost of query answering in this setting is intractable. To overcome this computational issue, we then define a different semantics for CQE centered around the notion of *intersection* of all the optimal GA censors. We show that query answering over OWL 2 QL ontologies under the new intersection-based semantics for CQE enjoys tractability and is *first-order rewritable*, i.e. amenable to be implemented through SQL query rewriting techniques and the use of standard relational database systems; on the other hand, this approach shows limitations in terms of amount of data disclosed. To improve this aspect, we add preferences between ontology predicates to the CQE framework, and identify a semantics under which query answering over OWL 2 QL ontologies maintains the same computational properties of the intersection-based approach without preferences.

1. Introduction

Controlled Query Evaluation (CQE) is a declarative approach to privacy-preserving query answering over databases and knowledge bases [1–6]. Such an approach has recently been studied in the context of Semantic Web languages and ontologies [7–11].

In CQE, a data protection policy is defined through a logical theory, and the information disclosed through query answering must comply with such a policy. The notion of *censor* is the one that formalizes the semantics of the above approach. A censor defines the answers to each query posed to the ontology, in a way such that the data protection policy is not violated while the information disclosed to the user of the ontology is maximized, according to some optimization criterion. Different semantics for CQE exist, giving rise to different notions of censors and different computational costs of query answering (see e.g. [11]). Notably, in most of the semantics for CQE defined in the literature, multiple censors for the same ontology may exist. This aspect may have consequences both from the semantic and the computational viewpoint.

In this paper, we conduct an analysis of the formal and computational properties of different approaches to CQE over OWL and Description Logic ontologies. We focus on semantics for censors that enjoy the so-called *indistinguishability* property. In abstract terms, this property implies that a user can never understand whether she is querying an ontology with or without sensitive information. Indistinguishability gives a high level of confidentiality protection to CQE, thus preventing information leakage [7,12]. Specifically, we prove that the class of *optimal GA censors* (censors defined through sets of *ground atoms*) enjoys the above property, thus our subsequent analysis focuses on such a class of censors. In general, multiple optimal GA censors for the same knowledge base may exist.

[*] Corresponding author.
E-mail addresses: cima@diag.uniroma1.it (G. Cima), lembo@diag.uniroma1.it (D. Lembo), marconi@diag.uniroma1.it (L. Marconi), rosati@diag.uniroma1.it (R. Rosati), domenicofabio.savo@unibg.it (D.F. Savo).

https://doi.org/10.1016/j.websem.2024.100841
Received 23 February 2024; Accepted 18 November 2024
Available online 9 December 2024

We analyze both semantic aspects and computational properties of CQE over Description Logic ontologies under the above indistinguishability-based semantics. We are particularly interested in the trade-off between maximizing the amount of information that is disclosed by query answers while preserving confidentiality and minimizing the computational cost of query answering. In our analysis, we focus on ontologies expressed in the Description Logic DL-Lite$_{\mathcal{R}}$ [13], the Description Logic underpinning the W3C-recommended OWL 2 QL profile [14]: our choice is motivated by the well-known nice computational properties of query answering over such ontologies.

More specifically, we first consider the case when the presence of multiple GA censors is solved by arbitrarily selecting a single optimal GA censor. While we show that this approach enjoys nice computational properties, we argue that this solution is not coherent with the declarativity of the CQE approach.

We then analyze the data complexity of *skeptical entailment* of queries (specifically, unions of conjunctive queries) under all the optimal GA censors. In this case, we avoid the above arbitrary choice of a single censor. On the other hand, it turns out that the computational cost of query answering in this setting is problematic: in particular, the data complexity of query answering over ontologies expressed in the lightweight DL DL-Lite$_{\mathcal{R}}$ is intractable (coNP-complete).

To overcome the above computational issue, we then define a different semantics centered around the notion of *intersection of all the optimal GA censors*, called *IGA censor* in the paper, and study the semantic and computational properties of CQE under such semantics. We show that query answering over DL-Lite$_{\mathcal{R}}$ ontologies under the new intersection-based semantics for CQE has the same data complexity as under the standard semantics, that is, it is in AC^0 in data complexity, and enjoys the so-called *first-order rewritability* property (see e.g. [15]). The last property makes it possible to implement query answering algorithms based on query rewriting in SQL and the use of standard relational database systems.

The above result indicates the possibility of effective realization of practical CQE systems based on the notion of IGA censor. This approach, however, shows limitations with respect to the maximization of the information disclosed through query answering.

In order to improve the above aspect, we add preferences between ontology predicates to the CQE framework, in a way analogous to recent work in the area of *Consistent Query Answering* [16]. The presence of such preferences has the consequence of restricting the set of admissible censors, and thus they allow for augmenting the set of query answers both under the skeptical-entailment semantics and under the intersection-based semantics.

While preferences increase the amount of information disclosed by CQE, they constitute in general an overhead in terms of computational cost of query answering, as already shown e.g. in [16]. However, we are able to identify an intersection-based semantics for CQE with preferences under which answering UCQs over DL-Lite$_{\mathcal{R}}$ ontologies is tractable in data complexity and not harder than in the absence of preferences. In particular, under this semantics for CQE, query answering is in AC^0 in data complexity and enjoys the first-order rewritability property, which indicates that such an approach to CQE is potentially able to increase confidentiality-preserving answers to queries while remaining feasible in practice.

The paper is structured as follows. After introducing some preliminaries in Section 2, we define our framework for CQE over DL ontologies in Section 3. Then, in Section 4 we study query answering under optimal GA censors in the case of CQE specifications expressed in DL-Lite$_{\mathcal{R}}$. In Section 5 we introduce the notion of IGA censor, and in Section 6 we study query answering over the IGA censor, again for DL-Lite$_{\mathcal{R}}$ CQE specifications. In Section 7, we extend the CQE framework with preferences expressed over ontology predicates, and in Section 8 we study query answering in the prioritized CQE framework for DL-Lite$_{\mathcal{R}}$ CQE specifications. Finally, we discuss related work in Section 9, and conclude the paper in Section 10.

This paper is an extended and revised version of [17] and, partially, of [18].

2. Preliminaries

We use standard notions of function-free first-order (FO) logic, and in particular, we consider Description Logics (DLs), which are fragments of FO using only unary and binary predicates, called *atomic concepts* and *atomic roles*, respectively [19]. We assume to have the pairwise disjoint countably infinite sets Σ_{C}, $\Sigma_{\mathcal{R}}$, $\Sigma_{\mathcal{I}}$, and $\Sigma_{\mathcal{V}}$ for atomic concepts, atomic roles, *constants* (a.k.a. *individuals*), and *variables*, respectively.

For a DL $\mathcal{L}_{\mathcal{T}}$, an $\mathcal{L}_{\mathcal{T}}$ TBox $\mathcal{T}$ is a finite set of assertions allowed in $\mathcal{L}_{\mathcal{T}}$, adopting symbols from $\Sigma_C \cup \Sigma_{\mathcal{R}}$ as predicates and symbols from $\Sigma_{\mathcal{I}} \cup \Sigma_{\mathcal{V}}$ as terms. The set of atomic concepts, roles, and individuals mentioned in the assertions of $\mathcal{T}$ constitutes the *signature* of $\mathcal{T}$, denoted by $\Sigma(\mathcal{T})$. Given a TBox $\mathcal{T}$, an ABox $\mathcal{A}$ for $\mathcal{T}$ is a finite set of *ground atoms* of the form $A(a)$ and $P(a,b)$, where A and P are an atomic concept and an atomic role, respectively, occurring in the signature of $\mathcal{T}$ and $a, b \in \Sigma_{\mathcal{I}}$. In what follows, when a TBox $\mathcal{T}$ is given, whenever we refer to an ABox $\mathcal{A}$, we implicitly assume that $\mathcal{A}$ is for $\mathcal{T}$ and, unless otherwise specified, that $\mathcal{T} \cup \mathcal{A}$ is a consistent FO theory.

For a DL $\mathcal{L}_{\mathcal{T}}$, an $\mathcal{L}_{\mathcal{T}}$ ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ is constituted by an $\mathcal{L}_{\mathcal{T}}$ TBox $\mathcal{T}$ and by an ABox $\mathcal{A}$. The semantics of an ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ is given in terms of *FO models* (or, simply, *models*) in the standard way [19]. Given a model $\mathcal{M}$ and an FO sentence ϕ, we say that $\mathsf{eval}(\mathcal{M}, \phi)$ is true if the standard evaluation of ϕ in $\mathcal{M}$ returns true [20], and say that $\mathsf{eval}(\mathcal{M}, \phi)$ is false otherwise. Given an ontology $\mathcal{O}$ and an FO sentence ϕ, we say that $\mathcal{O}$ *entails* ϕ, denoted by $\mathcal{O} \models \phi$, if $\mathsf{eval}(\mathcal{M}, \phi)$ is true for every model $\mathcal{M}$ of $\mathcal{O}$. If this is not the case, $\mathcal{O}$ does not entail ϕ, denoted by $\mathcal{O} \not\models \phi$. To simplify the presentation, given a set of ground atoms $\mathcal{A}$ and an FO sentence ϕ, with a slight abuse of notation, we will write $\mathsf{eval}(\mathcal{A}, \phi)$ to actually denote $\mathsf{eval}(H(\mathcal{A}), \phi)$, where $H(\mathcal{A})$ is the Herbrand model of $\mathcal{A}$ [21].

In this paper, we are particularly interested in DL-Lite$_{\mathcal{R}}$ ontologies, where DL-Lite$_{\mathcal{R}}$ is the member of the DL-Lite family [13] underpinning OWL 2 QL [22], i.e. the OWL 2 profile specifically designed for efficient query answering. A DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ consists in a finite set of assertions of the following form[1]:

$B_1 \sqsubseteq B_2$ (denoting positive concept inclusion)
$R_1 \sqsubseteq R_2$ (denoting positive role inclusion)
$B_1 \sqsubseteq \neg B_2$ (denoting negative concept inclusion, a.k.a. concept disjointness)
$R_1 \sqsubseteq \neg R_2$ (denoting negative role inclusion, a.k.a. role disjointness)

where:

- B_1 and B_2 are *basic concepts*, i.e. expressions of the form A, with $A \in \Sigma_C$, $\exists P$, with $P \in \Sigma_{\mathcal{R}}$, or $\exists P^-$. The expressions $\exists P$ and $\exists P^-$ are called unqualified existential restrictions, which denote the set of objects occurring as the first or second argument of P, respectively.
- R_1 and R_2 are *basic roles*, i.e. expressions of the form P or P^- (called the inverse of P).

As usual in query answering over DL ontologies, we focus on the language of conjunctive queries. A *Boolean conjunctive query (BCQ)* q is an FO sentence of the form $\exists \vec{x}.\phi(\vec{x})$, where $\vec{x}$ are variables in $\Sigma_{\mathcal{V}}$, and $\phi(\vec{x})$ is a finite, non-empty conjunction of atoms of the form $\alpha(\vec{t})$, where $\alpha \in \Sigma_C \cup \Sigma_{\mathcal{R}}$, and each term in $\vec{t}$ is either a constant in $\Sigma_{\mathcal{I}}$ or a variable in $\vec{x}$. In a BCQ q, each variable in $\vec{x}$ appears in at least one atom of $\phi(\vec{x})$. For a BCQ q and an atom $\alpha(\vec{t})$, we write $\alpha(\vec{t}) \in q$ to denote the fact that $\alpha(\vec{t})$ occurs in q. A BCQ with inequalities is an FO sentence of the form $\exists \vec{x}.\phi(\vec{x}) \wedge ineqls(\vec{x})$, where $\exists \vec{x}.\phi(\vec{x})$ is a BCQ and $ineqls(\vec{x})$ is a conjunction of inequality atoms of the form $t_1 \neq t_2$, with t_1 and t_2 either variables

[1] For DL-Lite$_{\mathcal{R}}$ assertions we adopt the well-known variable-free DL syntax [19].

in $\vec{x}$ or constants in $\Sigma_{\mathcal{I}}$. A *union of BCQs (BUCQ)* Q is an FO sentence of the form $\bigvee_{i=1}^{n} q_i$, where q_i is a BCQ for each $i = 1, \ldots, n$. With a little abuse of notation, we will sometimes treat a union of BCQs as a set of BCQs.

Given a BCQ q (possibly with inequalities) and an ABox $\mathcal{A}$, we say that an *image of q in $\mathcal{A}$* is a minimal subset $\mathcal{A}'$ of $\mathcal{A}$ such that q evaluates to true in $\mathcal{A}'$. Furthermore, given a BCQ q, a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, we say that an *image of q in $\mathcal{A}$ with respect to $\mathcal{T}$* is a minimal subset $\mathcal{A}'$ of $\mathcal{A}$ such that $\mathcal{T} \cup \mathcal{A}' \models q$.

A *denial assertion* (or simply a denial) is an FO sentence of the form $\forall \vec{x}.\phi(\vec{x}) \rightarrow \bot$, such that $\exists \vec{x}.\phi(\vec{x})$ is a BCQ. Given one such denial δ and an ontology $\mathcal{O}$, note that $\mathcal{O} \cup \{\delta\}$ is consistent if $\mathcal{O} \not\models \exists \vec{x}.\phi(\vec{x})$, and that $\mathcal{O} \cup \{\delta\}$ is inconsistent otherwise.

In the following, with **FO**, **CQ**, and **GA** we denote the languages of function-free FO sentences, BCQs, and ground atoms, respectively, all specified over the alphabets $\Sigma_{\mathcal{C}}, \Sigma_{\mathcal{R}}, \Sigma_{\mathcal{I}}$, and $\Sigma_{\mathcal{V}}$. Note that $\mathbf{GA} \subseteq \mathbf{CQ} \subseteq \mathbf{FO}$. Given an ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ and a language $\mathcal{L} \subseteq \mathbf{FO}$, with $\mathcal{L}(\mathcal{O})$ we refer to the subset of $\mathcal{L}$ containing all those sentences constructible using the atomic concepts and atomic roles in the signature of $\mathcal{T}$ as predicates, and the constants occurring in $\mathcal{T} \cup \mathcal{A}$ and the variables in $\Sigma_{\mathcal{V}}$ as terms. Given a language $\mathcal{L} \subseteq \mathbf{FO}$, a TBox $\mathcal{T}$, and an ABox $\mathcal{A}$, we denote by $\mathsf{cl}_{\mathcal{L}}^{\mathcal{T}}(\mathcal{A})$ the set of those sentences in $\mathcal{L}(\mathcal{T} \cup \mathcal{A})$ that are entailed by $\mathcal{T} \cup \mathcal{A}$, i.e. $\mathsf{cl}_{\mathcal{L}}^{\mathcal{T}}(\mathcal{A}) = \{\phi \mid \phi \in \mathcal{L}(\mathcal{T} \cup \mathcal{A}) \text{ and } \mathcal{T} \cup \mathcal{A} \models \phi\}$.

We recall that, for every DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ and BUCQ Q, it is possible to effectively compute an FO query q_r, called the *perfect reformulation of Q with respect to $\mathcal{T}$*, such that, for each ABox $\mathcal{A}$, we have $\mathcal{T} \cup \mathcal{A} \models Q$ if and only if $\mathsf{eval}(\mathcal{A}, q_r)$ is true [13]. This yields the well-known result that answering BUCQs over DL-Lite$_{\mathcal{R}}$ ontologies is *FO-rewritable*, and then the underlying decision problem is in AC^0 in the size of the ABox, i.e. in the so-called *data complexity* [23]. In this paper, we will use the algorithm $\mathsf{PerfectRef}$ given in [13], which takes as input a BUCQ Q and a TBox $\mathcal{T}$ and uses the positive inclusions in $\mathcal{T}$ as rewriting rules to compute the perfect reformulation of Q with respect to $\mathcal{T}$. $\mathsf{PerfectRef}$ executes two main steps, applied repeatedly to each BCQ $q \in Q$ until a fixpoint is reached: step (i) uses positive inclusions as rewriting rules applied from right to left, to rewrite query atoms one by one, each time producing a new BCQ to be added to the final rewriting q_r; step (ii) unifies the atoms in the query to enable further executions of step (i). We point out that the reformulation q_r returned by $\mathsf{PerfectRef}$ is in turn a BUCQ. The following proposition firstly appeared in [13].

Proposition 1. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{R}}$ TBox and Q be a BUCQ. For every ABox $\mathcal{A}$, we have that $\mathcal{T} \cup \mathcal{A} \models Q$ if and only if $\mathsf{eval}(\mathcal{A}, \mathsf{PerfectRef}(Q, \mathcal{T}))$ is true.*

For the sake of presentation, we will limit our technical treatment to languages containing only closed formulas, but our results also hold for open formulas. In particular, the results on entailment of BUCQ (see Sections 6 and 8) can be extended to arbitrary (i.e. non-Boolean) UCQs in the standard way.[2] All the complexity results in this paper concern data complexity.

3. Framework for CQE in DLs

We now define the framework for CQE over DL ontologies. First of all, given a TBox $\mathcal{T}$, a *policy* $\mathcal{P}$ for $\mathcal{T}$ is a set of denials over the signature of $\mathcal{T}$ such that $\mathcal{T} \cup \mathcal{P}$ is a consistent FO theory. We then introduce CQE specifications.

Definition 1 (*CQE Specification*). Let $\mathcal{L}_{\mathcal{T}}$ be a DL. An $\mathcal{L}_{\mathcal{T}}$ *CQE specification* $\mathcal{E}$ is a pair $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, where $\mathcal{T}$ is an $\mathcal{L}_{\mathcal{T}}$ TBox and $\mathcal{P}$ is a policy for $\mathcal{T}$.

In the rest of the paper, we will omit $\mathcal{L}_{\mathcal{T}}$ for definitions and results applying to any DL.

Example 1. Consider the following DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, where:

$$
\begin{aligned}
\mathcal{T} = \{ & \mathsf{SServ} \sqsubseteq \mathsf{GAgency}, \\
& \mathsf{SServ} \sqsubseteq \exists \mathsf{OperatesIn}, \\
& \mathsf{Manages} \sqsubseteq \mathsf{WorksFor}, \\
& \exists \mathsf{BreachedBy}^- \sqsubseteq \exists \mathsf{OperatesIn} \ \} \\
\mathcal{P} = \{ & \forall x, y.\mathsf{SServ}(x) \land \mathsf{WorksFor}(y, x) \rightarrow \bot, \\
& \forall x, y.\mathsf{SServ}(x) \land \mathsf{BreachedBy}(x, y) \rightarrow \bot \ \}
\end{aligned}
$$

In words, the TBox $\mathcal{T}$ sanctions that (i) every secret service (SServ) is a particular kind of government agency ($\mathsf{GAgency}$) and it must operate in ($\exists\mathsf{OperatesIn}$) some countries, (ii) if somebody manages ($\mathsf{Manages}$) an agency, then she/he works for ($\mathsf{WorksFor}$) that agency, and (iii) who breaches an agency ($\exists\mathsf{BreachedBy}^-$) operates in ($\exists\mathsf{OperatesIn}$) some countries. The data protection policy specified by $\mathcal{P}$ hides all the people working for a secret service and all the secret services that have been breached. □

In this paper we make use of the notion of *confidentiality-preserving (CP)* censors, introduced for the first time in [9] and then generalized in [11] through parameterization with respect to a *censor language*.

Informally, given a CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, a *CP censor* in a language $\mathcal{L}_c$ (called censor language) is a function that, taken an ABox $\mathcal{A}$ for $\mathcal{T}$ as input, establishes which are the sentences in $\mathcal{L}_c$ entailed by $\mathcal{T} \cup \mathcal{A}$ that can be disclosed to the user without violating the policy $\mathcal{P}$.

Definition 2 (*CP Censor*). Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification and $\mathcal{L}_c \subseteq \mathbf{FO}$ be a language. A *confidentiality-preserving (CP) censor* in $\mathcal{L}_c$ for $\mathcal{E}$ is a function $\mathsf{cens}(\cdot)$ that, for each ABox $\mathcal{A}$, returns a set $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}_{\mathcal{L}_c}^{\mathcal{T}}(\mathcal{A})$ such that $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A})$ is a consistent FO theory.

Example 2. Recall the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ provided in Example 1. The following functions are CP censors in **GA** for $\mathcal{E}$, i.e. when the censor language $\mathcal{L}_c$ coincides with the language of ground atoms:

- $\mathsf{cens}_1(\cdot)$ such that $\mathsf{cens}_1(\mathcal{A}) = \mathsf{cl}_{\mathbf{GA}}^{\mathcal{T}}(\mathcal{A}) \setminus \{\mathsf{SServ}(c) \mid c \in \Sigma_{\mathcal{I}}$ and $\mathcal{T} \cup \mathcal{A} \models \exists x.(\mathsf{WorksFor}(x, c) \lor \mathsf{BreachedBy}(c, x))\}$ for each ABox $\mathcal{A}$;
- $\mathsf{cens}_2(\cdot)$ such that $\mathsf{cens}_2(\mathcal{A}) = \mathcal{A} \setminus \{\mathsf{SServ}(c) \mid c \in \Sigma_{\mathcal{I}}$ and $\mathsf{SServ}(c) \in \mathcal{A}\}$ for each ABox $\mathcal{A}$. □

It is easy to see that a CP censor always exists,[3] but, as Example 2 shows, there may be many CP censors in a given censor language $\mathcal{L}_c$ for a given CQE specification $\mathcal{E}$, and so it is reasonable to look for those CP censors preserving as much information as possible. Formally, given two CP censors $\mathsf{cens}(\cdot)$ and $\mathsf{cens}'(\cdot)$ in a censor language $\mathcal{L}_c$ for a CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, we say that $\mathsf{cens}'(\cdot)$ is *more informative* than $\mathsf{cens}(\cdot)$ if:

1. $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cens}'(\mathcal{A})$, for every ABox $\mathcal{A}$, and
2. there exists an ABox $\mathcal{A}'$ such that $\mathsf{cens}(\mathcal{A}') \subset \mathsf{cens}'(\mathcal{A}')$.

We are now ready to formalize the notion of optimal CP censors for a CQE specification.

Definition 3 (*Optimal CP Censor*). Given a CQE specification $\mathcal{E}$ and a language $\mathcal{L}_c$, we say that a CP censor $\mathsf{cens}(\cdot)$ in $\mathcal{L}_c$ for $\mathcal{E}$ is *optimal* in $\mathcal{L}_c$ if there does not exist any CP censor $\mathsf{cens}'(\cdot)$ in $\mathcal{L}_c$ for $\mathcal{E}$ such that $\mathsf{cens}'(\cdot)$ is more informative than $\mathsf{cens}(\cdot)$.

[2] We also note that, since DL-Lite$_{\mathcal{R}}$ is insensitive to the adoption of the *unique name assumption* (UNA) for CQ answering [24], all our results hold both with and without UNA.

[3] Given any CQE specification $\mathcal{E}$, note that the function $\mathsf{cens}(\cdot)$ such that $\mathsf{cens}(\mathcal{A}) = \emptyset$ for each ABox $\mathcal{A}$ is trivially a CP censor in **GA** for $\mathcal{E}$.

Example 3. Consider again the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ and the CP censors $\mathsf{cens}_1(\cdot)$ and $\mathsf{cens}_2(\cdot)$ in **GA** for $\mathcal{E}$ of Example 2. Note that $\mathsf{cens}_2(\cdot)$ is not optimal in **GA** because $\mathsf{cens}_1(\cdot)$ is more informative than $\mathsf{cens}_2(\cdot)$. Indeed, consider the following ABox:

$$\begin{aligned}\mathcal{A} = \{\ & \mathsf{WorksFor}(bob, a_1),\ \mathsf{Manages}(ann, a_2),\ \mathsf{SServ}(a_2),\\ & \mathsf{BreachedBy}(a_2, h_2)\ \}.\end{aligned}$$

It is easy to verify that $\mathsf{cens}_2(\mathcal{A}) \subset \mathsf{cens}_1(\mathcal{A})$ (in particular, $\mathsf{GAgency}(a_2) \in \mathsf{cens}_1(\mathcal{A}) \setminus \mathsf{cens}_2(\mathcal{A})$). Moreover, one can see and that $\mathsf{cens}_2(\mathcal{A}') \subseteq \mathsf{cens}_1(\mathcal{A}')$ for every other ABox $\mathcal{A}'$. On the other hand, one can verify that $\mathsf{cens}_1(\cdot)$ is an optimal CP censor in **GA** for $\mathcal{E}$. □

For a given censor language $\mathcal{L}_c$, we are naturally interested in the censors in $\mathcal{L}_c$ that are maximally informative, i.e. that do not filter out non-confidential information. Therefore, from now on we restrict our attention to optimal censors in $\mathcal{L}_c$.

Moreover, as said in the introduction, to increase the robustness of censors, literature on CQE has often looked at censors satisfying a property of *instance indistinguishability* (or, simply, *indistinguishability)* [3,7,12,25]. Intuitively, a censor fulfilling such a property masks confidential information in such a way that a user cannot distinguish an instance actually containing sensitive data (i.e. those protected by the policy) from an instance without such data, in order to increase the incompleteness of the information of a possible attacker.

A CP censor for a CQE specification $\mathcal{E}$ satisfies the indistinguishability property if for every ABox $\mathcal{A}$ there exists an ABox $\mathcal{A}'$ (possibly distinct from $\mathcal{A}$) that does not violate any data protection property and is indistinguishable from $\mathcal{A}$ from the user perspective. In our framework, this is formalized as follows.

Definition 4 (*Indistinguishability*). Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification, $\mathcal{L}_c \subseteq \mathbf{FO}$, and $\mathsf{cens}(\cdot)$ be a CP censor in $\mathcal{L}_c$ for $\mathcal{E}$. We say that $\mathsf{cens}(\cdot)$ *satisfies the indistinguishability property* if, for every ABox $\mathcal{A}$, there exists an ABox $\mathcal{A}'$ (not necessarily distinct from $\mathcal{A}$) such that:

(*i*) $\mathsf{cens}(\mathcal{A}) = \mathsf{cens}(\mathcal{A}')$, and
(*ii*) $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is a consistent FO theory.

Example 4. Consider again the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ and the optimal CP censor $\mathsf{cens}_1(\cdot)$ in **GA** for $\mathcal{E}$ of Example 2. Observe that $\mathsf{cens}_1(\cdot)$ satisfies the indistinguishability property because, given any ABox $\mathcal{A}$, we have that $\mathsf{cens}_1(\mathcal{A})$ itself plays the role of the ABox $\mathcal{A}'$ satisfying the conditions (*i*) and (*ii*) of Definition 4. □

Previous research in CQE has mostly considered two censor languages: **CQ** [9] and **GA** [11,26]. We now analyze these two languages with respect to the above indistinguishability property.

Our first result states that, in general, optimal censors in **CQ**, i.e. when the censor language $\mathcal{L}_c$ coincides with the language of Boolean Conjunctive Queries, do not satisfy the indistinguishability property.

Proposition 2. *There exists a CQE specification $\mathcal{E}$ and an optimal censor* $\mathsf{cens}(\cdot)$ *for $\mathcal{E}$ in* **CQ** *such that* $\mathsf{cens}(\cdot)$ *does not satisfy the indistinguishability property.*

Proof. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ with $\mathcal{T} = \emptyset$, $\mathcal{P} = \{C(a) \rightarrow \bot\}$. It can be easily verified that the unique optimal censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ in **CQ** is such that: (*i*) for every ABox $\mathcal{A}$ that contains $C(a)$, $\mathsf{cens}(\mathcal{A}) = \{q \in \mathbf{CQ} \mid (\mathcal{A} \setminus \{C(a)\}) \cup \{\exists x.C(x)\} \models q\}$; (*ii*) for every ABox $\mathcal{A}$ that does not contain $C(a)$, $\mathsf{cens}(\mathcal{A}) = \{q \in \mathbf{CQ} \mid \mathsf{eval}(\mathcal{A}, q) \text{ is true}\}$. Now, observe that, for the ABox $\mathcal{A} = \{C(a)\}$, there exists no ABox $\mathcal{A}'$ such that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is consistent (i.e. $\mathcal{A}'$ does not contain $C(a)$) and $\mathsf{cens}(\mathcal{A}') = \mathsf{cens}(\mathcal{A})$. Consequently, $\mathsf{cens}(\cdot)$ does not satisfy the indistinguishability property. ■

We now consider the case of **GA** as the censor language, and show that, whatever is the CQE specification $\mathcal{E}$, all optimal CP censors in **GA** for $\mathcal{E}$ enjoy the indistinguishability property. The formal proof makes use of the following two lemmas.

Lemma 1. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification,* $\mathsf{cens}(\cdot)$ *be an optimal CP censor in $\mathcal{L}_c$ for $\mathcal{E}$, and $\mathcal{A}$ be an ABox such that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$ is consistent. We have that* $\mathsf{cens}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathcal{A})$.

Proof. Note that $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathcal{A})$ trivially holds from the fact that $\mathsf{cens}(\cdot)$ is a CP censor in $\mathcal{L}_c$ for $\mathcal{E}$. We now prove that $\mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathcal{A}) \subseteq \mathsf{cens}(\mathcal{A})$ holds as well.

Towards a contradiction, suppose this is not the case, i.e. there exists a $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathcal{A})$ such that $\gamma \notin \mathsf{cens}(\mathcal{A})$. Since $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$ is consistent, and since $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathcal{A})$, we have $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A} \cup \{\gamma\}$ is consistent as well. So, consider the function $\mathsf{cens}'(\cdot)$ such that (*i*) $\mathsf{cens}(\mathcal{A}') = \mathsf{cens}'(\mathcal{A}')$ for each ABox $\mathcal{A}' \neq \mathcal{A}$, and (*ii*) $\mathsf{cens}'(\mathcal{A}) = \mathsf{cens}(\mathcal{A}) \cup \{\gamma\}$. It is straightforward to verify that $\mathsf{cens}'(\cdot)$ is a CP censor in $\mathcal{L}_c$ for $\mathcal{E}$ and that $\mathsf{cens}'(\cdot)$ is more informative than $\mathsf{cens}(\cdot)$. This contradicts the fact that $\mathsf{cens}(\cdot)$ is an optimal CP censor in $\mathcal{L}_c$ for $\mathcal{E}$. ■

In words, the above result shows that, if the considered ABox does not contain confidential data, then every optimal CP censor in $\mathcal{L}_c$ preserves all the information in $\mathcal{A}$ that can be expressed in $\mathcal{L}_c$. In particular, if $\mathcal{L}_c \supseteq \mathcal{A}$, then each optimal CP censor preserves all the information in $\mathcal{A}$.

Lemma 2. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification,* $\mathsf{cens}(\cdot)$ *be an optimal CP censor in $\mathcal{L}_c$ for $\mathcal{E}$, and $\mathcal{A}$ be an ABox. We have that* $\mathsf{cens}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathsf{cens}(\mathcal{A}))$.

Proof. Note that $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathsf{cens}(\mathcal{A}))$ trivially holds. We now prove that $\mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathsf{cens}(\mathcal{A})) \subseteq \mathsf{cens}(\mathcal{A})$ holds as well.

Towards a contradiction, suppose this is not the case, i.e. there exists a $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathsf{cens}(\mathcal{A}))$ such that $\gamma \notin \mathsf{cens}(\mathcal{A})$. Since $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A})$ is consistent because $\mathsf{cens}(\cdot)$ is a CP censor in $\mathcal{L}_c$ for $\mathcal{E}$, and since $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathcal{L}_c}(\mathsf{cens}(\mathcal{A}))$, we have that $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A}) \cup \{\gamma\}$ is consistent as well. So, exactly as done in the proof of Lemma 1, consider the function $\mathsf{cens}'(\cdot)$ such that (*i*) $\mathsf{cens}(\mathcal{A}') = \mathsf{cens}'(\mathcal{A}')$ for each ABox $\mathcal{A}' \neq \mathcal{A}$, and (*ii*) $\mathsf{cens}'(\mathcal{A}) = \mathsf{cens}(\mathcal{A}) \cup \{\gamma\}$. It is straightforward to verify that $\mathsf{cens}'(\cdot)$ is a CP censor in $\mathcal{L}_c$ for $\mathcal{E}$ and that $\mathsf{cens}'(\cdot)$ is more informative than $\mathsf{cens}(\cdot)$. This contradicts the fact that $\mathsf{cens}(\cdot)$ is an optimal CP censor in $\mathcal{L}_c$ for $\mathcal{E}$. ■

Theorem 1. *Let $\mathcal{E}$ be a CQE specification and* $\mathsf{cens}(\cdot)$ *be an optimal CP censor in* **GA** *for $\mathcal{E}$. We have that* $\mathsf{cens}(\cdot)$ *satisfies the indistinguishability property.*

Proof. According to Definition 2, when the censor language is set to **GA**, for every ABox $\mathcal{A}$ we have that $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$, i.e. $\mathsf{cens}(\mathcal{A})$ is an ABox, and also that $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A})$ is consistent. We show that the ABox $\mathcal{A}'$ of Definition 4 is $\mathsf{cens}(\mathcal{A})$ itself, i.e. we show that $\mathsf{cens}(\mathcal{A}) = \mathsf{cens}(\mathsf{cens}(\mathcal{A}))$ for each ABox $\mathcal{A}$.

Consider any ABox $\mathcal{A}$. By Lemma 2, we have that $\mathsf{cens}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}(\mathcal{A}))$. Furthermore, since $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A})$ is consistent, by Lemma 1 we know that $\mathsf{cens}(\mathsf{cens}(\mathcal{A})) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}(\mathcal{A}))$. It follows that $\mathsf{cens}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}(\mathcal{A})) = \mathsf{cens}(\mathsf{cens}(\mathcal{A}))$, as required. ■

As a consequence of Proposition 2 and Theorem 1, in what follows we focus our study on BUCQ answering over DL-Lite$_{\mathcal{R}}$ CQE specifications by considering optimal CP censors in **GA**, that, for sake of conciseness, we will denote by (optimal) GA censors.

Algorithm 1 $\mathsf{OptGACensor}$

```
input: a DL-Lite$_\mathcal{R}$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, an ABox $\mathcal{A}$;
output: an ABox $\mathcal{A}'$;
1) $\mathcal{A}_\mathcal{T} \leftarrow \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$;
2) $\mathcal{A}' \leftarrow \emptyset$;
3) while $\mathcal{A}_\mathcal{T} \neq \emptyset$ do:
4)     let $\alpha$ be the lexicographically first assertion in $\mathcal{A}_\mathcal{T}$;
5)     $\mathcal{A}_\mathcal{T} \leftarrow \mathcal{A}_\mathcal{T} \setminus \{\alpha\}$;
6)     if $\mathcal{T} \cup \mathcal{A}' \cup \{\alpha\} \cup \mathcal{P}$ is consistent then
7)         $\mathcal{A}' \leftarrow \mathcal{A}' \cup \{\alpha\}$;
8) return $\mathcal{A}'$;
```

4. Query answering under optimal GA censors

In this section we study query answering under optimal GA censors over DL-Lite$_\mathcal{R}$ CQE specifications. In particular, we consider entailment of BUCQs.

A possible strategy for addressing this problem is to choose only one GA censor among the optimal ones, and use it to alter the answers to user queries. In the absence of a criterion for determining which optimal censor is the best for users' purposes, the choice is made in an arbitrary way, as done in [9,27].

In the same spirit, we provide the algorithm $\mathsf{OptGACensor}$ (Algorithm 1), which implements a function that, for every DL-Lite$_\mathcal{R}$ CQE specification $\mathcal{E}$ and every ABox $\mathcal{A}$, returns an ABox that indeed coincides with $\mathsf{cens}(\mathcal{A})$ for an optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$.

The algorithm first computes the set $\mathcal{A}_\mathcal{T}$ of ground atoms entailed by $\mathcal{T} \cup \mathcal{A}$. Then, it iteratively picks a ground atom α from $\mathcal{A}_\mathcal{T}$ following a lexicographic order, and adds α to the current ABox $\mathcal{A}'$ if $\mathcal{T} \cup \mathcal{A}' \cup \{\alpha\}$ does not violate the policy $\mathcal{P}$.

It is easy to see that the algorithm runs in polynomial time in the size of $\mathcal{A}$, as stated by the following theorem, which also establishes the correctness of the algorithm.

Theorem 2. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a DL-Lite$_\mathcal{R}$ CQE specification. The following holds:*

- *There exists an optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ such that, for each ABox $\mathcal{A}$, $\mathsf{OptGACensor}(\mathcal{E}, \mathcal{A})$ returns $\mathsf{cens}(\mathcal{A})$;*
- *Given any ABox $\mathcal{A}$, $\mathsf{OptGACensor}(\mathcal{E}, \mathcal{A})$ runs in polynomial time in the size of $\mathcal{A}$.*

Proof. For each ABox $\mathcal{A}$, the set $\mathcal{A}'$ returned by the algorithm contains only assertions in $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$, that is, it contains only assertions in **GA** entailed by $\mathcal{T} \cup \mathcal{A}$. Moreover, step 6 of the algorithm guarantees that $\mathcal{T} \cup \mathcal{A}' \cup \mathcal{P}$ is a consistent FO theory. Hence, according to Definition 2, the algorithm implements a GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$. Verifying that $\mathsf{cens}(\cdot)$ is optimal is also immediate. Indeed, suppose, by way of contradiction, that there exists an ABox $\mathcal{A}$ and a censor $\mathsf{cens}'(\cdot)$ such that $\mathsf{cens}(\mathcal{A}) \subset \mathsf{cens}'(\mathcal{A})$ and $\mathsf{cens}(\mathcal{A}'') \subseteq \mathsf{cens}'(\mathcal{A}'')$ for every other ABox $\mathcal{A}''$. This means that there exists an assertion $\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ such that α in $\mathsf{cens}'(\mathcal{A}) \setminus \mathsf{cens}(\mathcal{A})$, but since α is not in $\mathsf{cens}(\mathcal{A})$ then $\mathcal{T} \cup \mathsf{cens}(\mathcal{A}) \cup \{\alpha\} \cup \mathcal{P}$ has to be inconsistent (step 6 of the algorithm), and so $\mathcal{T} \cup \mathsf{cens}'(\mathcal{A}) \cup \mathcal{P}$ is inconsistent too, which contradicts the fact that $\mathsf{cens}'(\cdot)$ is a GA censor for $\mathcal{E}$.

As for the complexity, note that the algorithm iterates on the set of ABox assertions $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ by choosing an assertion α and, in each iteration, it checks whether $\mathcal{T} \cup \mathcal{A}' \cup \{\alpha\} \cup \mathcal{P}$ is consistent. Clearly, the algorithm terminates since $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ is finite. Moreover, the thesis follows from the following facts: (*i*) given a DL-Lite$_\mathcal{R}$ TBox, a policy $\mathcal{P}$, and an ABox $\mathcal{A}' \cup \{\alpha\}$, checking whether $\mathcal{T} \cup \mathcal{A}' \cup \{\alpha\} \cup \mathcal{P}$ is consistent can be done in AC^0 w.r.t. the size of $\mathcal{A}' \cup \{\alpha\}$ [28]; (*ii*) the set $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ can be computed in polynomial time w.r.t. $|\mathcal{A}|$ and that its size is polynomial w.r.t. $|\mathcal{A}|$ as well. ■

From Theorem 2 it follows that, given a DL-Lite$_\mathcal{R}$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, an ABox $\mathcal{A}$ and a BUCQ q, it is possible to verify in polynomial time in the size of $\mathcal{A}$ whether $\mathcal{T} \cup \mathsf{cens}(\mathcal{A}) \models q$, where $\mathsf{cens}(\cdot)$ is the optimal GA censor for $\mathcal{E}$ implemented by the algorithm (since the size of $\mathsf{cens}(\mathcal{A})$ is polynomially related to the size of $\mathcal{A}$ and that deciding $\mathcal{T} \cup \mathsf{cens}(\mathcal{A}) \models q$ is in AC^0 w.r.t. the size of $\mathsf{cens}(\mathcal{A})$).

Moreover, given a DL-Lite$_\mathcal{R}$ CQE specification $\mathcal{E}$ and an ABox $\mathcal{A}$, it is possible to simulate the behavior of different optimal GA censors for $\mathcal{E}$ (actually, every optimal GA censor for $\mathcal{E}$) on $\mathcal{A}$ by simply modifying the order in which the ABox assertions from the set $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ are selected by the algorithm (see step 4). More formally, it is easy to see that the following holds.

Proposition 3. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a DL-Lite$_\mathcal{R}$ CQE specification. For every optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ and for every ABox $\mathcal{A}$, there exists a lexicographic order on the ground atoms in $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ for which $\mathsf{OptGACensor}(\mathcal{E}, \mathcal{A})$ returns $\mathsf{cens}(\mathcal{A})$.*

Depending on the application at hand, however, the approach of randomly choosing a censor may not always be considered appropriate [9]. For this reason, [11] studies *skeptical entailment* under all the optimal censors, i.e. a Boolean query q has a positive answer over a CQE specification $\mathcal{E}$ and an ABox $\mathcal{A}$ if q is entailed when considering each optimal GA censor for $\mathcal{E}$.

Definition 5. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification, $\mathcal{A}$ be an ABox, and q be a Boolean query. GA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}(\mathcal{A}) \models q$ for each optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$.

Example 5. Recall the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ of Example 4 and the GA censor $\mathsf{cens}_1(\cdot)$ for $\mathcal{E}$ illustrated in Example 2. Furthermore, consider the GA censor $\mathsf{cens}_4(\cdot)$ for $\mathcal{E}$ such that $\mathsf{cens}_4(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}')$ for each ABox $\mathcal{A}$, where $\mathcal{A}'$ is the ABox obtained from $\mathcal{A}$ by removing the ground atoms $\mathsf{Manages}(a, b)$, $\mathsf{WorksFor}(a, b)$, and $\mathsf{BreachedBy}(b, a)$ for all the individuals $a, b \in \Sigma_\mathcal{I}$ such that $\mathsf{SServ}(b) \in \mathcal{A}$. Now, if we consider the ABox $\mathcal{A}$ of Example 3, then it is easy to verify that every optimal GA censor for $\mathcal{E}$ applied to $\mathcal{A}$ returns one of the two following sets of ground atoms:

$$\begin{aligned}\mathsf{cens}_1(\mathcal{A}) = \{&\mathsf{WorksFor}(bob, a_1),\ \mathsf{Manages}(ann, a_2),\ \mathsf{WorksFor}(ann, a_2), ;\\ &\mathsf{BreachedBy}(a_2, h_2),\ \mathsf{GAgency}(a_2)\}\\ \mathsf{cens}_4(\mathcal{A}) = \{&\mathsf{WorksFor}(bob, a_1),\ \mathsf{SServ}(a_2),\ \mathsf{GAgency}(a_2)\}.\end{aligned}$$

For the BCQ $q = \exists x, y.\mathsf{OperatesIn}(x, y)$, we have that GA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is true because both $\mathcal{T} \cup \mathsf{cens}_1(\mathcal{A}) \models q$ and $\mathcal{T} \cup \mathsf{cens}_4(\mathcal{A}) \models q$ hold. □

Despite skeptically reasoning under all optimal GA censors is a natural approach for avoiding the randomness introduced by arbitrarily choosing one (optimal) GA censor over the others, as shown in [11], the GA-Cens-Entailment problem is intractable in data complexity in our considered scenario (in fact, already for BCQs), unless P = NP.

Theorem 3 (*[11]*). *GA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ for DL-Lite$_\mathcal{R}$ CQE specifications $\mathcal{E}$ and BCQs q is coNP-complete in data complexity.*

In the next section, we propose a new semantics for CQE that soundly approximates GA censor semantics we discussed so far and, at the same time, allows us to overcome the above computational issue.

5. IGA censors

Towards the identification of a practical setting, we propose a new semantically well-founded notion of GA censor which, on the one hand, makes, in the case of DL-Lite$_{\mathcal{R}}$, conjunctive query answering tractable (by approximating the skeptical reasoning), and, on the other hand, avoids arbitrary choices by being always unique.

The approximation we propose consists in considering a non-necessarily optimal GA censor corresponding to computing the intersection of all optimal GA censor for a CQE specification CQE.

Definition 6 (*IGA Censor*)**.** Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification. The *Intersection GA (IGA) censor* for $\mathcal{E}$ is the function $\mathsf{cens}_{\mathsf{IGA}}(\cdot)$ such that, for every ABox $\mathcal{A}$:

$$\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \bigcap_{\mathsf{cens}(\cdot) \in \mathsf{optCens}(\mathcal{E})} \mathsf{cens}(\mathcal{A})$$

where $\mathsf{optCens}(\mathcal{E})$ denotes the set of optimal GA censors for $\mathcal{E}$.

Example 6. Recall the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ and the optimal GA censors $\mathsf{cens}_1(\cdot)$ and $\mathsf{cens}_4(\cdot)$ for $\mathcal{E}$ of Example 5. The IGA censor for $\mathcal{E}$ is the function $\mathsf{cens}_{\mathsf{IGA}}(\cdot)$ such that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \mathsf{cens}_1(\mathcal{A}) \cap \mathsf{cens}_4(\mathcal{A})$ for each ABox $\mathcal{A}$, i.e. $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ is the ABox obtained from $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ by removing (*i*) the ground atoms $\mathsf{Manages}(a, b)$, $\mathsf{WorksFor}(a, b)$, and $\mathsf{BreachedBy}(b, a)$ for all individuals $a, b \in \Sigma_{\mathcal{I}}$ such that $\mathsf{SServ}(b) \in \mathcal{A}$ and (*ii*) the ground atoms $\mathsf{SServ}(b)$ such that $\mathsf{Manages}(a, b) \in \mathcal{A}$, or $\mathsf{WorksFor}(a, b) \in \mathcal{A}$ or $\mathsf{BreachedBy}(b, a) \in \mathcal{A}$ for all individuals $a, b \in \Sigma_{\mathcal{I}}$.

As an example, consider the ABox $\mathcal{A}$ of Example 3. We have that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \{\mathsf{WorksFor}(bob, a_1), \mathsf{GAgency}(a_1)\}$. □

As the above example shows, the IGA censor for a CQE specification $\mathcal{E}$ is not guaranteed to be an optimal GA censor. Notably, however, the following proposition shows that, for any CQE specification $\mathcal{E}$, the IGA censor exists and is unique.

Proposition 4. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification. The IGA censor $\mathsf{cens}_{\mathsf{IGA}}(\cdot)$ for $\mathcal{E}$ exists, is unique, and is a GA censor for $\mathcal{E}$.*

Proof. By looking at Definition 6, in order to prove that $\mathsf{cens}_{\mathsf{IGA}}(\cdot)$ exists, it is enough to show the existence of at least an optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$. The uniqueness, then, directly follows from its definition. To see that there exists at least one optimal GA censor for $\mathcal{E}$, observe that (*i*) the function $\mathsf{cens}_0(\cdot)$ such that $\mathsf{cens}_0(\mathcal{A}) = \emptyset$ for each ABox $\mathcal{A}$ is a GA censor for $\mathcal{E}$, and (*ii*) each GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ is such that $\mathsf{cens}(\mathcal{A})$ is a finite set of ground atoms for each ABox $\mathcal{A}$. This latter is guaranteed by the fact that $\mathsf{cens}(\mathcal{A}) \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ by definition, where $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ is a finite set of ground atoms since $\mathcal{A}$ is a finite set of ground atoms as well. So, (*i*) and (*ii*) imply the existence of at least one optimal GA censor for $\mathcal{E}$.

We now prove that $\mathsf{cens}_{\mathsf{IGA}}(\cdot)$ is a GA censor for $\mathcal{E}$. Let $\mathcal{A}$ be any ABox. Note that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) \subseteq \mathsf{cens}(\mathcal{A})$ for any optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$. Since, however, $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A})$ is consistent for any GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$, we have that $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ is consistent as well due to the monotonicity of first-order-logic. ■

We show now that the IGA censor satisfies the indistinguishability property. First, we provide the counterparts of Lemmas 1 and 2 for the IGA censor.

Lemma 3. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification and $\mathcal{A}$ be an ABox such that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$ is consistent. We have that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$.*

Proof. The thesis trivially follows from the definition of IGA censor and by Lemma 1. ■

Lemma 4. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification and $\mathcal{A}$ be an ABox. We have that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}))$.*

Proof. Note that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}(\mathcal{A}))$ trivially holds. We now prove that $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}(\mathcal{A})) \subseteq \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$.

Towards a contradiction, suppose this is not the case, i.e. there exists a $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}))$ such that $\gamma \notin \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$. Since $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}))$, then there exists in $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ a set of facts $\mathcal{F}$ such that $\mathcal{T} \cup \mathcal{F} \models \gamma$. Moreover, since $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) \subseteq \mathsf{cens}(\mathcal{A})$ for each optimal **GA** censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$, we also have that $\mathcal{F} \subseteq \mathsf{cens}(\mathcal{A})$, and so, given the monotonicity of first-order-logic, we have that $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}(\mathcal{A}))$ for each optimal censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$. Therefore, by Lemma 2, we have also that $\gamma \in \mathsf{cens}(\mathcal{A})$, for each optimal **GA** censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$, from which it clearly holds that $\gamma \in \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$. This contradicts the initial assumption that $\gamma \notin \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$. Thus, it must be that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}))$. ■

We are now ready to show that the indistinguishability property holds for the IGA censor.

Proposition 5. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification. The IGA censor $\mathsf{cens}_{\mathsf{IGA}}(\cdot)$ for $\mathcal{E}$ satisfies the indistinguishability property.*

Proof. We prove that the ABox $\mathcal{A}'$ of Definition 4 is $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ itself, i.e. we show that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ is an ABox such that $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ is consistent and $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \mathsf{cens}_{\mathsf{IGA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}))$ for each ABox $\mathcal{A}$. Consider any ABox $\mathcal{A}$. By Lemma 4, we have that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}(\mathcal{A}))$. Furthermore, since $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ is consistent, by Lemma 3 we know that $\mathsf{cens}_{\mathsf{IGA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}))$. It follows that $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) = \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})) = \mathsf{cens}_{\mathsf{IGA}}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}))$, as required. ■

6. Query answering under the IGA censor

In this section, we focus on DL-Lite$_{\mathcal{R}}$ CQE specifications and BUCQs as queries, and we show that query answering based on the IGA censor is reducible to the evaluation of an FO query over the ABox, i.e. it is FO-rewritable and therefore in AC^0 in data complexity.

Below, we provide the general decision problem we focus on.

Definition 7. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification, $\mathcal{A}$ be an ABox, and q be a Boolean query. IGA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) \models q$.

The following proposition, whose proof is straightforward, says that IGA-Cens-Entailment is a sound approximation of the skeptical reasoning approach under GA censors.

Proposition 6. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification, $\mathcal{A}$ be an ABox, and q be a Boolean query. If IGA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is true, then GA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is true.*

The next example shows that, in general, the converse of Proposition 6 does not hold.

Example 7. We refer to the same CQE specification and ABox of Example 5. While we have that GA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is true, observe that IGA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is false because $\mathcal{T} \cup \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) \not\models q$. □

In what follows on this section, we show that IGA-Cens-Entailment is FO-rewritable in our considered scenario, that is, for every DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E}$ and BUCQ q, one can effectively compute an FO query q_r such that, for every ABox $\mathcal{A}$, IGA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ is true if and only if $\mathsf{eval}(\mathcal{A}, q_r)$ is true. We call such a query q_r the *IGA reformulation* of q with respect to $\mathcal{E}$.

The intuition behind our rewriting algorithm is as follows. For a DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, given any ABox $\mathcal{A}$, we want to filter out facts from $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ that together with the TBox $\mathcal{T}$ lead to the violation of some denial in the policy $\mathcal{P}$. At the same time, we want this elimination of facts to be done in a minimal way, according to our definition of the IGA censor. Thus, only "really dangerous" facts have to be dropped from $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$. Let us first formally define such a minimal set of facts violating the policy, which we call *secrets*.

Definition 8. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification and $\mathcal{A}$ be an ABox. A set of ground atoms $\mathcal{S} \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ is a *secret for* $\mathcal{E}$ *and* $\mathcal{A}$ if

(i) $\mathcal{T} \cup \mathcal{P} \cup \mathcal{S}$ is inconsistent, and
(ii) there is no set $\mathcal{S}'$ of ground atoms such that $\mathcal{S}' \subset \mathcal{S}$ and $\mathcal{T} \cup \mathcal{P} \cup \mathcal{S}'$ is inconsistent.

Given a CQE specification $\mathcal{E}$ and an ABox $\mathcal{A}$, we denote by $secrets(\mathcal{E}, \mathcal{A})$ the set of secrets for $\mathcal{E}$ and $\mathcal{A}$.

Example 8. Consider the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ and ABox $\mathcal{A}$ of the previous examples. We have that $secrets(\mathcal{E}, \mathcal{A}) = \{\{\mathsf{SServ}(a_2), \mathsf{Manages}(ann, a_2)\}, \{\mathsf{SServ}(a_2), \mathsf{WorksFor}(ann, a_2)\}, \{\mathsf{SServ}(a_2), \mathsf{BreachedBy}(a_2, h_2)\}\}$. These are indeed the only three minimal subsets $\mathcal{S}$ of $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ such that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{S}$ is inconsistent. □

The next key proposition shows that, for DL-Lite$_{\mathcal{R}}$ CQE specifications $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, given an ABox $\mathcal{A}$, the IGA censor for $\mathcal{E}$ returns all and only those ground atoms in $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ that does not occur in any secret for $\mathcal{E}$ and $\mathcal{A}$.

Proposition 7. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a DL-Lite$_{\mathcal{R}}$ CQE specification, $\mathcal{A}$ be an ABox, and $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$. We have that $\gamma \notin \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ if and only if $\gamma \in \mathcal{S}$ for some $\mathcal{S} \in secrets(\mathcal{E}, \mathcal{A})$.*

Proof. Note that a ground atom $\gamma \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ does not belong to $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ if and only if there exists an optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ such that $\gamma \notin \mathsf{cens}(\mathcal{A})$. So it is sufficient to show that $\gamma \notin \mathsf{cens}(\mathcal{A})$ for some optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ if and only if there exists a secret $\mathcal{S}$ in $secrets(\mathcal{E}, \mathcal{A})$ such that $\gamma \in \mathcal{S}$.

($\Rightarrow$). Suppose, by way of contradiction, that γ does not belong to any secret in $secrets(\mathcal{E}, \mathcal{A})$. It follows that, for any optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$, $\mathcal{T} \cup \mathcal{P} \cup \mathsf{cens}(\mathcal{A}) \cup \{\gamma\}$ is consistent (otherwise, one can see that γ would belong to some secret in $secrets(\mathcal{E}, \mathcal{A})$) and so $\mathsf{cens}(\cdot)$ is not an optimal GA censor for $\mathcal{E}$, from which the contradiction that $\mathsf{cens}(\cdot)$ is an optimal GA censor for $\mathcal{E}$. Indeed, the function $\mathsf{cens}'(\cdot)$ with $\mathsf{cens}'(\mathcal{A}') = \mathsf{cens}(\mathcal{A}')$ for each ABox $\mathcal{A}' \neq \mathcal{A}$ and $\mathsf{cens}'(\mathcal{A}) = \mathsf{cens}(\mathcal{A}) \cup \{\gamma\}$ would be a GA censor for $\mathcal{E}$ that is more informative than $\mathsf{cens}(\cdot)$.

($\Leftarrow$). Suppose that $\gamma \in \mathcal{S}$ for some secret $\mathcal{S} \in secrets(\mathcal{E}, \mathcal{A})$. We now show that there exists an optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ such that $\gamma \notin \mathsf{cens}(\mathcal{A})$. From Definition 8, we can derive that $\mathcal{T} \cup \mathcal{P} \cup (\mathcal{S} \setminus \{\gamma\})$ is consistent. It follows that there exists at least an optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E}$ such that $(\mathcal{S} \setminus \{\gamma\}) \subseteq \mathsf{cens}(\mathcal{A})$. ■

Identifying the ground atoms occurring in some secret is easier if we can reason on each denial in the policy in isolation. Unfortunately, this may not always be possible even in very simple cases. Consider, for instance, the policy $\mathcal{P} = \{\forall x. A(x) \wedge B(x) \rightarrow \bot, \forall x. A(x) \rightarrow \bot\}$. The set $\{A(d), B(d)\}$ of ground atoms occurring in an ABox $\mathcal{A}$ is a minimal set violating the first denial, but is not a secret for $\langle \mathcal{T}, \mathcal{P} \rangle$ and $\mathcal{A}$ since its subset $\{A(d)\}$ is in $secrets(\langle \mathcal{T}, \mathcal{P} \rangle, \mathcal{A})$ (in this example $\mathcal{T} = \emptyset$).

To be able to solve this issue and treat denials separately, we now introduce the notion of *extended denial assertion* (or simply extended denial), which is a formula of the form $\forall \vec{x}. \phi(\vec{x}) \wedge ineqls(\vec{x}) \rightarrow \bot$ such that $\exists \vec{x}. \phi(\vec{x}) \wedge ineqls(\vec{x})$ is a BCQ with inequalities.

Then, an *extended policy* is a finite set of extended denials. Moreover, in the rest of this section, we call *non-extended denial* a denial as defined in Section 3.

Definition 9. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a CQE specification. An extended policy $\mathcal{P}'$ is a *non-redundant representation of* $\mathcal{P}$ *w.r.t.* $\mathcal{T}$ if the following conditions hold:

(i) for every ABox $\mathcal{A}$, we have that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$ is inconsistent if and only if $\mathcal{P}' \cup \mathcal{A}$ is inconsistent;
(ii) for every $\delta \in \mathcal{P}'$, ABox $\mathcal{A}$, and minimal (w.r.t. set inclusion) set of ground atoms $\mathcal{F} \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ such that $\{\delta\} \cup \mathcal{F}$ is inconsistent, we have that $\mathcal{F} \in secrets(\mathcal{E}, \mathcal{A})$.

One might think that, given a CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, computing an extended policy $\mathcal{P}'$ that is a non-redundant representation of $\mathcal{P}$ w.r.t. $\mathcal{T}$ means simply eliminating from $\mathcal{P}$ each denial δ such that $\mathcal{T} \cup (\mathcal{P} \setminus \{\delta\}) \models \delta$. In fact, only eliminating denials that are (fully) logically inferred by other denials (and the TBox) is not sufficient, since some redundancies can occur for specific instantiations of the denials. For example, consider the CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$, where $\mathcal{T} = \emptyset$ and $\mathcal{P} = \{\delta_1, \delta_2\}$ with $\delta_1 = \forall x, y. Q(x, y) \wedge C(y) \rightarrow \bot$ and $\delta_2 = \forall z. Q(z, z) \rightarrow \bot$. Despite the fact that $\mathcal{T} \cup \{\delta_2\} \not\models \delta_1$, for the particular instantiation in which $x = y$, giving rise to denial $\delta_1[\{x = y\}] = \forall x. Q(x, x) \wedge C(x) \rightarrow \bot$, we have that $\mathcal{T} \cup \{\delta_2\} \models \delta_1[\{x = y\}]$. This implies that, for instance, given an ABox $\mathcal{A}$ containing the ABox assertions $Q(a, a)$ and $C(a)$, although $\mathcal{F} = \{Q(a, a), C(a)\}$ is a minimal violation of δ_1, $\mathcal{F}$ is not a secret for $\mathcal{E}$ and $\mathcal{A}$ since $\{Q(a, a)\}$ is already a secret for $\mathcal{E}$ and $\mathcal{A}$. An extended policy that would be a non-redundant representation of $\mathcal{P}$ w.r.t. $\mathcal{T}$ is $\mathcal{P}' = \{\delta_1', \delta_2\}$, where $\delta_1' = \forall x, y. Q(x, y) \wedge C(y) \wedge x \neq y \rightarrow \bot$.

Actually, given a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ and a policy $\mathcal{P}$, in order to compute an extended policy $\mathcal{P}'$ that is a non-redundant representation of $\mathcal{P}$ w.r.t. $\mathcal{T}$, it is possible to use the same technique illustrated in [28], called $\mathsf{MinUnsatQuery}$, to solve a similar issue. We note that $\mathsf{MinUnsatQuery}$ takes as input a set composed of DL-Lite$_{\mathcal{R}}$ TBox axioms and denial assertions. In what follows, given an extended denial $\delta = \forall \vec{x}. \phi(\vec{x}) \wedge ineqls(\vec{x}) \rightarrow \bot$, we denote by q_δ the BCQ with inequalities associated to δ, i.e. $q_\delta = \exists \vec{x}. \phi(\vec{x}) \wedge ineqls(\vec{x})$, and, on the other direction, given a BCQ with inequalities $q = \exists \vec{x}. \phi(\vec{x}) \wedge ineqls(\vec{x})$, we let $\delta_q = \forall \vec{x}. \phi(\vec{x}) \wedge ineqls(\vec{x}) \rightarrow \bot$. Now, given a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ and a policy $\mathcal{P}$, we define the function $\mathsf{MinPolicy}$ as follows: $\mathsf{MinPolicy}(\mathcal{P}, \mathcal{T}) = \{\delta_q \mid q \in \mathsf{MinUnsatQuery}(\mathcal{T}^p \cup \mathcal{P})\}$, where $\mathcal{T}^p$ is the DL-Lite$_{\mathcal{R}}$ TBox obtained from $\mathcal{T}$ by removing all the disjointness axioms occurring in $\mathcal{T}$.

Example 9. Consider the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ of the running example. Then, $\mathsf{MinPolicy}(\mathcal{P}, \mathcal{T})$ returns of the following set of extended denials:

$$\begin{aligned}
\{\ & \forall x, y.\ \mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y, x) \wedge x \neq y \rightarrow \bot, \\
& \forall x, y.\ \mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(y, x) \wedge x \neq y \rightarrow \bot, \\
& \forall x, y.\ \mathsf{SServ}(x) \wedge \mathsf{Manages}(y, x) \wedge x \neq y \rightarrow \bot, \\
& \quad \forall x.\ \mathsf{SServ}(x) \wedge \mathsf{WorksFor}(x, x) \rightarrow \bot, \\
& \quad \forall x.\ \mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(x, x) \rightarrow \bot, \\
& \quad \forall x.\ \mathsf{SServ}(x) \wedge \mathsf{Manages}(x, x) \rightarrow \bot\ \}.
\end{aligned}$$

Anyway, for the sake of conciseness, in the subsequent examples we always refer to the following simplified version of the above set:

$$\begin{aligned}
\{\ & \forall x, y.\ \mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y, x) \rightarrow \bot, \\
& \forall x, y.\ \mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(y, x) \rightarrow \bot, \\
& \forall x, y.\ \mathsf{SServ}(x) \wedge \mathsf{Manages}(y, x) \rightarrow \bot\ \}.
\end{aligned}$$

Note that such a set is still a non-redundant representation of $\mathcal{P}$ w.r.t. $\mathcal{T}$. □

Proposition 8. *Let $\mathcal{T}$ be a DL-Lite$_{\mathcal{R}}$ TBox and $\mathcal{P}$ be a policy. The extended policy returned by $\mathsf{MinPolicy}(\mathcal{P}, \mathcal{T})$ is a non-redundant representation of $\mathcal{P}$ w.r.t. $\mathcal{T}$.*

Proof. Follows immediately from Definition 9 and from Lemma 6 and Lemma 7 of [28]. ■

We are now ready to provide our query rewriting technique. In what follows, without loss of generality, we assume that, given a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ and a policy $\mathcal{P}$, the output of $\mathsf{MinPolicy}(\mathcal{P}, \mathcal{T})$ is an extended policy such that in each extended denial the arguments of the various atoms are always variables different to one another (the presence of the same variable or of constants can be indeed expressed through equalities).

The basic idea is to exploit Proposition 7 for identifying, given a DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ and an ABox $\mathcal{A}$, those atoms that can be involved in the evaluation of a BUCQ in $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$. This is done by discarding those atoms in $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ that occur in some secret for $\mathcal{E}$ and $\mathcal{A}$, that is, as follows from Proposition 8, the atoms that participate in the image of at least one BCQ with inequalities associated to some denial in $\mathsf{MinPolicy}(\mathcal{P}, \mathcal{T})$.

Let α and β be two atoms. We say that β is *compatible with* α if there exists a mapping μ of the variables occurring in β to the terms occurring in α such that $\mu(\beta) = \alpha$. Given an FO sentence ϕ, we denote by $\mu_{\alpha/\beta}(\phi)$ the FO sentence obtained from ϕ by applying the mapping μ. Moreover, given an atom α and an FO sentence ϕ, we denote by $\mathsf{compSet}(\alpha, \phi)$ the set of atoms of ϕ that are compatible with α.

For an atom α and a DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, we define $\mathsf{D}(\alpha, \mathcal{E})$ as follows:

$$\mathsf{D}(\alpha, \mathcal{E}) = \alpha \wedge \left(\bigwedge_{\substack{\forall \delta \in \mathsf{MinPolicy}(\mathcal{P},\mathcal{T}), \\ \forall \beta \in \mathsf{compSet}(\alpha, q_\delta)}} \forall \vec{w}. \left(\neg \mu_{\alpha/\beta}(q_\delta) \right) \right),$$

where $\vec{w}$ contains all the variables in the various $\mu_{\alpha/\beta}(q_\delta)$ that do not occur in α.

Example 10. Consider again the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ of the running example. We have that: $\mathsf{D}(\mathsf{BreachedBy}(x, y), \mathcal{E}) = \mathsf{BreachedBy}(x, y) \wedge \forall w.(\neg(\mathsf{BreachedBy}(x, w) \wedge \mathsf{SServ}(x)))$. A shorter, yet equivalent, version of such a formula is $\mathsf{BreachedBy}(x, y) \wedge \neg \mathsf{SServ}(x)$. □

Given a BUCQ Q and a DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E}$, we define the Boolean FO query $\mathsf{DQ}(Q, \mathcal{E})$ as follows:

$$\mathsf{DQ}(Q, \mathcal{E}) = \bigvee_{q \in Q} \left(\exists \vec{x}_q. \bigwedge_{\alpha \in q} \mathsf{D}(\alpha, \mathcal{E}) \right),$$

where, for every BCQ $q \in Q$, $\vec{x}_q$ denotes the existential variables of q.

Example 11. Consider the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ and the query q described in previous examples. We have that $\mathsf{PerfectRef}(q, \mathcal{T}) = \exists x, y.\mathsf{OperatesIn}(x, y) \vee \exists x, y.\mathsf{BreachedBy}(x, y) \vee \exists x.\mathsf{SServ}(x)$. Moreover, one can verify that:

$$\begin{aligned}\mathsf{DQ}(\mathsf{PerfectRef}(q, \mathcal{T}), \mathcal{E}) = {} & \exists x, y.\mathsf{OperatesIn}(x, y) \vee \\ & \exists x, y.(\mathsf{BreachedBy}(x, y) \wedge \neg \mathsf{SServ}(x)) \vee \\ & \exists x.(\mathsf{SServ}(x) \wedge \forall y.(\neg \mathsf{BreachedBy}(x, y)) \wedge \\ & \quad \forall y.(\neg \mathsf{Manages}(y, x)) \wedge \\ & \quad \forall y.(\neg \mathsf{WorksFor}(y, x))) \quad \square\end{aligned}$$

Given a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ and an FO query ϕ, we define $\mathsf{expand}(\mathcal{T}, \phi)$ as the FO query obtained from ϕ by replacing every atom α occurring in ϕ with its "TBox-expansion" $\mathsf{expand}(\mathcal{T}, \alpha)$, defined as follows:

$$\mathsf{expand}(\mathcal{T}, \alpha) = \begin{cases} \bigvee_{\mathcal{T} \models D \sqsubseteq C} D(t) \vee & \\ \bigvee_{\mathcal{T} \models \exists R \sqsubseteq C} (\exists x.R(t, x)) \vee & \text{if } \alpha = C(t) \\ \bigvee_{\mathcal{T} \models \exists R^- \sqsubseteq C} (\exists x.R(x, t)), & \\ \bigvee_{\mathcal{T} \models S \sqsubseteq R} S(t_1, t_2) \vee & \text{if } \alpha = R(t_1, t_2) \\ \bigvee_{\mathcal{T} \models S^- \sqsubseteq R} S(t_2, t_1), & \end{cases}$$

Finally, for a BUCQ q and a DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, we define:

$$\mathsf{DRew}(q, \mathcal{E}) = \mathsf{expand}(\mathcal{T}, \mathsf{DQ}(\mathsf{PerfectRef}(q, \mathcal{T}), \mathcal{E})).$$

It is easy to see that $\mathsf{DRew}(q, \mathcal{E})$ is a Boolean FO query.

Example 12. Consider the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ and the query q of the running example. One can see that:

$$\begin{aligned}\mathsf{DRew}(q, \mathcal{E}) = {} & \exists x, y.\mathsf{OperatesIn}(x, y) \vee \\ & \exists x, y.(\mathsf{BreachedBy}(x, y) \wedge \neg \mathsf{SServ}(x)) \vee \\ & \exists x.(\mathsf{SServ}(x) \wedge \forall y.(\neg \mathsf{BreachedBy}(x, y)) \\ & \quad \wedge \forall y.(\neg \mathsf{Manages}(y, x)) \wedge \\ & \quad \forall y.(\neg(\mathsf{WorksFor}(y, x) \vee \mathsf{Manages}(y, x)))) \quad \square\end{aligned}$$

We are now ready to prove that, for DL-Lite$_{\mathcal{R}}$ CQE specifications $\mathcal{E}$ and BUCQs q, the decision problem IGA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ can always be solved by checking whether $\mathsf{DRew}(q, \mathcal{E})$ evaluates to true in $\mathcal{A}$. In other terms, we prove that the problem is FO-rewritable. We start with the next lemma, which immediately follows from the definition of $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ and $\mathsf{expand}(\mathcal{T}, \phi)$.

Lemma 5. *Let ϕ be an FO query. Then, $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \phi)$ is true if and only if $\mathsf{eval}(\mathcal{A}, \mathsf{expand}(\mathcal{T}, \phi))$ is true.*

On the basis of Proposition 1 and Lemma 5, we prove FO-rewritability of IGA-Cens-Entailment$(\mathcal{E}, \mathcal{A}, q)$ for DL-Lite$_{\mathcal{R}}$ CQE specifications $\mathcal{E}$ and BUCQs q.

Theorem 4. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ be a DL-Lite$_{\mathcal{R}}$ CQE specification and q be a BUCQ. For every ABox $\mathcal{A}$, we have that $\mathcal{T} \cup \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) \models q$ if and only if $\mathsf{eval}(\mathcal{A}, \mathsf{DRew}(q, \mathcal{E}))$ is true.*

Proof. By exploiting Proposition 1 and Lemma 5, to prove the thesis of the theorem note that it is enough to show the following: given a BUCQ Q, we have that $\mathsf{eval}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}), Q)$ is true if and only if $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DQ}(Q, \mathcal{E}))$ is true.

First, since $\mathsf{eval}(\mathcal{A}', Q) = \mathsf{eval}(\mathcal{A}', q_1) \vee \ldots \vee \mathsf{eval}(\mathcal{A}', q_n)$ holds for any BUCQ $Q = q_1 \vee \ldots \vee q_n$ and ABox $\mathcal{A}'$, it follows from the definition of $\mathsf{DQ}(\cdot, \cdot)$ that it is enough to prove that, for any BCQ q, $\mathsf{eval}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}), q)$ is true if and only if $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DQ}(q, \mathcal{E}))$ is true. Note that $\mathsf{DQ}(q, \mathcal{E}) = \exists \vec{x}_q. \bigwedge_{\alpha \in q} \mathsf{D}(\alpha, \mathcal{E})$ because q is a BCQ.

(⇒). Suppose $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \exists \vec{x}_q. \bigwedge_{\alpha \in q} \mathsf{D}(\alpha, \mathcal{E}))$ is true. By construction of $\mathsf{D}(\cdot, \cdot)$, this means that there is at least an image I of q in $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ such that $\mathsf{eval}(I, \exists \vec{x}_\alpha.\mathsf{D}(\alpha, \mathcal{E}))$ is true for any $\alpha \in q$, where $\vec{x}_\alpha$ are the variables occurring in the atom α. We denote by I_α the image (ground atom) of each atom α in I. Now, Proposition 8 guarantees that there is no $\mathcal{S} \in$ *secrets*$(\mathcal{E}, \mathcal{A})$ such that $I_\alpha \in \mathcal{S}$, otherwise this would easily contradict the fact that $\mathsf{eval}(I, \exists \vec{x}_\alpha.\mathsf{D}(\alpha, \mathcal{E}))$ is true. It follows that $I = \bigcup_{\alpha \in q} I_\alpha$ is such that there is no atom in I that belongs to some secret for $\mathcal{E}$ and $\mathcal{A}$. Obviously, we have that $\mathsf{eval}(I, q)$ is true. Now, by Proposition 7, it directly follows that $I \subseteq \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$, and therefore $\mathsf{eval}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}), q)$ is true as well because q is a BCQ.

(⇐). Suppose now $\mathsf{eval}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}), q)$ is true. Consider each atom $\alpha \in q$. Since, by Proposition 7, we know that all the ground atoms in $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ do not belong to any secrets for $\mathcal{E}$ and $\mathcal{A}$, by Proposition 8 and by construction of $\mathsf{D}(\cdot, \cdot)$, we easily derive that $\mathsf{eval}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}), \exists \vec{x}_\alpha.\mathsf{D}(\alpha, \mathcal{E}))$ is true as well. Thus, we immediately obtain that $\mathsf{eval}(\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}), \exists \vec{x}_q. \bigwedge_{\alpha \in q} \mathsf{D}(\alpha, \mathcal{E}))$ is true, as required. ■

Example 13. Consider the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$, the ABox $\mathcal{A}$, and the query q of the running example, and consider the IGA reformulation $\mathsf{DRew}(q, \mathcal{E})$ of q w.r.t. $\mathcal{E}$ illustrated in Example 12. Note that $\mathsf{eval}(\mathcal{A}, \mathsf{DRew}(q, \mathcal{E}))$ is false. Indeed, as observed in Example 7, we have that $\mathcal{T} \cup \mathsf{cens}_{\mathsf{IGA}}(\mathcal{A}) \not\models q$. □

Provided that $\mathsf{DRew}(\cdot, \cdot)$ first rewrites the input query via $\mathsf{PerfectRef}$, one may wonder whether rewriting atom by atom through the $\mathsf{expand}(\cdot, \cdot)$ function is necessary. We then show a simple example illustrating how performing such a step is indeed essential for ensuring the correctness of our approach. Consider a TBox $\mathcal{T} = \{A \sqsubseteq B\}$, an

ABox $\mathcal{A} = \{A(o)\}$ and a policy $\mathcal{P} = \{\forall x.A(x) \rightarrow \bot\}$. It is easy to see that there is only one optimal GA censor $\mathsf{cens}(\cdot)$ for $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$ which is such that $\mathsf{cens}(\mathcal{A}) = \{B(o)\}$. Thus, also the IGA censor for $\mathcal{E}$ is such that $\mathsf{cens}_{\mathrm{IGA}}(\mathcal{A}) = \{B(o)\}$. Now, consider a BCQ $q = \exists x.B(x)$, which clearly evaluates to true in $\mathsf{cens}_{\mathrm{IGA}}(\mathcal{A})$. Rewriting q without using the $\mathsf{expand}(\cdot,\cdot)$ function would result in the following FO query: $\phi = \exists x.B(x) \vee \exists x.(A(x) \wedge \neg A(x))$, whose evaluation in $\mathcal{A}$ returns false. On the other hand, by adopting the $\mathsf{expand}(\cdot,\cdot)$ function we get the following FO query: $\mathsf{expand}(\mathcal{T}, \phi) = \exists x.(B(x) \vee A(x)) \vee \exists x.(A(x) \wedge \neg A(x))$, whose evaluation in $\mathcal{A}$ is true.

The corollary below follows from Theorem 4 and the fact that evaluating an FO query over an ABox is in AC^0 in the size of the ABox (i.e. in data complexity).

Corollary 1. *IGA-Cens-Entailment*$(\mathcal{E}, \mathcal{A}, q)$ *for DL-Lite*$_\mathcal{R}$ *CQE specifications* $\mathcal{E}$ *and BUCQs* q *is in* AC^0 *in data complexity.*

7. Prioritized CQE framework

Answering queries under the IGA semantics, considered in the previous section, may be too restrictive. This is due to the fact that any ground atom belonging to any secret is not included in the set returned by the IGA censor. In this section, we present an extension of the CQE framework in which one can specify a priority relation over ontology predicates. These priorities can be used to induce a choice between facts belonging to the same secret, in order to reveal some facts that would instead be kept undisclosed, thus improving the throughput of answers to user queries with respect to the IGA semantics.

Given a TBox $\mathcal{T}$, a *priority relation* $\succ$ *over* $\mathcal{T}$ is an acyclic binary relation over the signature of $\mathcal{T}$, i.e. $\succ \subseteq \Sigma(\mathcal{T}) \times \Sigma(\mathcal{T})$.

Definition 10 (*Prioritized CQE Specification*)**.** Let $\mathcal{L}_\mathcal{T}$ be a DL. A *prioritized* $\mathcal{L}_\mathcal{T}$ *CQE specification* $\mathcal{E}_\succ$ is a triple $\langle \mathcal{T}, \mathcal{P}, \succ \rangle$, such that $\langle \mathcal{T}, \mathcal{P} \rangle$ is an $\mathcal{L}_\mathcal{T}$ CQE specification and $\succ$ is a priority relation over $\mathcal{T}$.

Similarly to what done for (non-prioritized) CQE specifications, we will omit $\mathcal{L}_\mathcal{T}$ for definitions and results applying to any DL language.

Example 14. Suppose that, if we know that a person p works for a secret service s (both $\mathsf{WorksFor}(p, s)$ and $\mathsf{SServ}(s)$ hold), we prefer to disclose the fact that p works for s ($\mathsf{WorksFor}(p, s)$) rather than the fact that s is a secret service ($\mathsf{SServ}(s)$). This will be indicated with $(\mathsf{WorksFor}, \mathsf{SServ}) \in \succ$.

In the following examples, we will refer to the prioritized DL-Lite$_\mathcal{R}$ CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, where $\langle \mathcal{T}, \mathcal{P} \rangle$ is the DL-Lite$_\mathcal{R}$ CQE specification $\mathcal{E}$ of Example 1 and $\succ = \{(\mathsf{WorksFor}, \mathsf{SServ}), (\mathsf{Manages}, \mathsf{SServ})\}$. □

The definitions of GA censor, optimal GA censor, IGA censor, GA-Cens-Entailment, and IGA-Cens-Entailment apply also to a prioritized CQE specification (e.g. given one such specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, $\mathsf{cens}(\cdot)$ is a GA censor for $\mathcal{E}_\succ$ if it is a GA censor for the CQE specification $\mathcal{E} = \langle \mathcal{T}, \mathcal{P} \rangle$). We also use for prioritized CQE specifications the same notations introduced for (non-prioritized) CQE specifications, with the same meaning. In particular, we denote by $\mathsf{optCens}(\mathcal{E}_\succ)$ the set of optimal GA censors for a prioritized CQE specification $\mathcal{E}_\succ$.

We now exploit the priority relation to induce a partial order over censors. We consider two optimality notions introduced by [29] in the context of consistent query answering (CQA) over databases, and recently adopted in [16] for repairing inconsistent prioritized DL ontologies.[4] Whereas the priority relations considered in this paper are intentional, i.e. between ontology predicates, priorities considered in [16,29] are between (conflicting) facts. However, intensional priorities straightforwardly induce priorities over facts: given a TBox $\mathcal{T}$, a priority relation $\succ$ over $\mathcal{T}$, an ABox $\mathcal{A}$, and two assertions $S_1(\vec{n})$ and $S_2(\vec{m})$ in $\mathcal{A}$, we have that $S_1(\vec{n}) \succ S_2(\vec{m})$ if $S_1 \succ S_2$.

Below we take the definitions of Pareto- and Globally-optimal repair from [16] and adapt them to our framework.

Definition 11 (*Pareto/Globally-optimal Censor*)**.** Let $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ be a prioritized CQE specification, $\mathcal{A}$ be an ABox, and $\mathsf{cens}(\cdot) \in \mathsf{optCens}(\mathcal{E}_\succ)$. We say that an ABox $\mathcal{A}' \subseteq \mathsf{cl}_{\mathbf{GA}}^{\mathcal{T}}(\mathcal{A})$, such that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is consistent, is:

- a *Pareto improvement* of $\mathsf{cens}(\mathcal{A})$ w.r.t. $\mathcal{E}_\succ$ if there exists an assertion $\gamma' \in \mathcal{A}' \setminus \mathsf{cens}(\mathcal{A})$ such that, for every assertion $\gamma \in \mathsf{cens}(\mathcal{A}) \setminus \mathcal{A}'$, we have that $\gamma' \succ \gamma$ and $\{\gamma, \gamma'\} \subseteq \mathcal{S}$ for some $\mathcal{S} \in \mathit{secrets}(\mathcal{T}, \mathcal{P}, \mathcal{A})$;
- a *Global improvement* of $\mathsf{cens}(\mathcal{A})$ w.r.t. $\mathcal{E}_\succ$ if $\mathcal{A}' \neq \mathsf{cens}(\mathcal{A})$ and, for every assertion $\gamma \in \mathsf{cens}(\mathcal{A}) \setminus \mathcal{A}'$, there exists an assertion $\gamma' \in \mathcal{A}' \setminus \mathsf{cens}(\mathcal{A})$ such that $\gamma' \succ \gamma$ and $\{\gamma, \gamma'\} \subseteq \mathcal{S}$ for some $\mathcal{S} \in \mathit{secrets}(\mathcal{T}, \mathcal{P}, \mathcal{A})$.

Then, $\mathsf{cens}(\cdot)$ is a *Pareto- (resp. Globally-)optimal censor* for $\mathcal{E}_\succ$ if there exists no other GA censor $\mathsf{cens}'(\cdot)$ for $\mathcal{E}_\succ$ such that, for each ABox $\mathcal{A}$, either $\mathsf{cens}'(\mathcal{A}) = \mathsf{cens}(\mathcal{A})$ or $\mathsf{cens}'(\mathcal{A})$ is a Pareto (resp. Global) improvement of $\mathsf{cens}(\mathcal{A})$ w.r.t. $\mathcal{E}_\succ$.

We denote by $\mathsf{PCens}(\mathcal{E}_\succ)$ (resp. $\mathsf{GCens}(\mathcal{E}_\succ)$) the set of all Pareto- (resp. Globally-) optimal censors for $\mathcal{E}_\succ$. It is easy to see that $\mathsf{GCens}(\mathcal{E}_\succ) \subseteq \mathsf{PCens}(\mathcal{E}_\succ) \subseteq \mathsf{optCens}(\mathcal{E}_\succ)$ for every prioritized CQE specification $\mathcal{E}_\succ$, analogous to the containment between Pareto- and Globally-optimal repairs given in [29]. Also, if $\succ$ is empty, then $\mathsf{PCens}(\mathcal{E}_\succ) = \mathsf{GCens}(\mathcal{E}_\succ) = \mathsf{optCens}(\mathcal{E}_\succ)$.

As done for GA censors (see Definition 6), we define intersection-based versions of both Pareto- and Globally-optimal censors. Given a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, we call:

- *Intersection Pareto (IP) censor* for $\mathcal{E}_\succ$ the function $\mathsf{cens}_{\mathrm{IP}}(\cdot)$ such that, for every ABox $\mathcal{A}$:
$$\mathsf{cens}_{\mathrm{IP}}(\mathcal{A}) = \bigcap_{\mathsf{cens}(\cdot) \in \mathsf{PCens}(\mathcal{E}_\succ)} \mathsf{cens}(\mathcal{A})$$
- *Intersection Global (IG) censor* for $\mathcal{E}_\succ$ the function $\mathsf{cens}_{\mathrm{IG}}(\cdot)$ such that, for every ABox $\mathcal{A}$:
$$\mathsf{cens}_{\mathrm{IG}}(\mathcal{A}) = \bigcap_{\mathsf{cens}(\cdot) \in \mathsf{GCens}(\mathcal{E}_\succ)} \mathsf{cens}(\mathcal{A})$$

Obviously, $\mathsf{cens}_{\mathrm{IP}}(\mathcal{A}) \subseteq \mathsf{cens}_{\mathrm{IG}}(\mathcal{A})$ for each ABox $\mathcal{A}$. Also, if $\succ$ is empty, then, since $\mathsf{PCens}(\mathcal{E}_\succ) = \mathsf{GCens}(\mathcal{E}_\succ) = \mathsf{optCens}(\mathcal{E}_\succ)$, we have that $\mathsf{cens}_{\mathrm{IP}}(\cdot) = \mathsf{cens}_{\mathrm{IG}}(\cdot) = \mathsf{cens}_{\mathrm{IGA}}(\cdot)$.

Given a prioritized CQE specification $\mathcal{E}_\succ$, an ABox $\mathcal{A}$, and a Boolean query q, P-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$ (resp. G-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$) is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}(\mathcal{A}) \models q$ for each $\mathsf{cens}(\cdot) \in \mathsf{PCens}(\mathcal{E}_\succ)$ (resp. $\mathsf{cens}(\cdot) \in \mathsf{GCens}(\mathcal{E}_\succ)$), and IP-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$ (resp. IG-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$) is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}_{\mathrm{IP}}(\mathcal{A}) \models q$ (resp. $\mathcal{T} \cup \mathsf{cens}_{\mathrm{IG}}(\mathcal{A}) \models q$), where $\mathsf{cens}_{\mathrm{IP}}(\cdot)$ (resp. $\mathsf{cens}_{\mathrm{IG}}(\cdot)$) is the IP (resp. IG) censor for $\mathcal{E}_\succ$. It is immediate to see that P-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$ implies G-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$, and IP-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$ (resp. IG-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$) implies P-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$ (resp. G-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$).

From the outcomes presented in [16], which already hold for BCQs, it immediately derives what follows.

Proposition 9. *For prioritized DL-Lite*$_\mathcal{R}$ *CQE specifications* $\mathcal{E}_\succ$ *and BCQs* q*, we have that P-Cens-Entailment*$(\mathcal{E}_\succ, \mathcal{A}, q)$ *and IP-Cens-Entailment* $(\mathcal{E}_\succ, \mathcal{A}, q)$ *are coNP-hard in data complexity, whereas G-Cens-Entailment* $(\mathcal{E}_\succ, \mathcal{A}, q)$ *and IG-Cens-Entailment*$(\mathcal{E}_\succ, \mathcal{A}, q)$ *are* Π_2^p*-hard in data complexity.*

[4] CQA [30] is a well-known declarative approach to inconsistency-tolerant reasoning. As discussed in [11], there is a tight connection between CQA and CQE.

The results in Proposition 9 represent a clear obstacle to the use of the above forms of priority-based censors over real-world, large datasets. In the next section, we will see how these censors can be suitably approximated for practical use.

8. FO-rewritable prioritized CQE in DL-Lite$_\mathcal{R}$

In this section, we first give a deterministic notion of priority-based censor (DD censor), which is an adaptation in our framework of the one studied in [16] in the context of CQA, and its parameterized sound approximation called k-DD censor. Then, we show that BUCQ entailment under k-DD censors in DL-Lite$_\mathcal{R}$ is FO rewritable. The full rewriting algorithm is given in the last part of this section.

8.1. DD censors and k censors

Proposition 9 clearly says that under Pareto or Global censors, or their intersection-based versions, entailment of BCQs is inherently non-deterministic. Towards the identification of a tractable approximation, we give below the notion of *deterministically disclosed atoms* (DDA) and *deterministically censored atoms* (DCA). Hereinafter, given a priority relation $\succ$, a fact α, and a set of facts $\mathcal{F}$, we write $\alpha \succ \mathcal{F}$ if there exists a fact $\beta \in \mathcal{F}$ such that $\alpha \succ \beta$. Moreover, given a fact α, we denote by $inSec(\mathcal{T},\mathcal{P},\mathcal{A},\alpha)$ the set of secrets $\mathcal{S} \in secrets(\mathcal{T},\mathcal{P},\mathcal{A})$ such that $\alpha \in \mathcal{S}$.

Definition 12. Given a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ and an ABox $\mathcal{A}$, we denote by $\mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A})$ and $\mathsf{DCA}(\mathcal{E}_\succ,\mathcal{A})$ the inclusion-minimal subsets of $\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ such that:

$$\begin{aligned}\mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A}) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}) \mid \forall \mathcal{S} \in inSec(\mathcal{T},\mathcal{P},\mathcal{A},\alpha)\ \exists \beta \in \mathcal{S}\\ \text{s.t. } \alpha \succ \beta \lor (\alpha \neq \beta \land \beta \in \mathsf{DCA}(\mathcal{E}_\succ,\mathcal{A}))\}\\ \mathsf{DCA}(\mathcal{E}_\succ,\mathcal{A}) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}) \mid \exists \mathcal{S} \in inSec(\mathcal{T},\mathcal{P},\mathcal{A},\alpha)\\ \text{s.t. } \mathcal{S} \setminus \mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A}) = \{\alpha\}\}\end{aligned}$$

In words, an atom $\alpha \in \mathsf{DDA}$ is such that α does not occur in any secret or, either, in each secret in which it occurs there is an atom β such that $\alpha \succ \beta$ or $\beta \in \mathsf{DCA}$. Instead, an atom in DCA is such that there is a secret where it is the only atom not in DDA. It is immediate to verify that $\mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A})$ and $\mathsf{DCA}(\mathcal{E}_\succ,\mathcal{A})$ are unique for a given pair $(\mathcal{E}_\succ,\mathcal{A})$. We are now ready to provide the definition of *Deterministically Disclosing (DD) censor*.

Definition 13 (*DD Censor*). Let $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ be a prioritized CQE specification. The *Deterministically Disclosing (DD) censor* for $\mathcal{E}_\succ$ is the function $\mathsf{cens}_{\mathsf{DD}}(\cdot)$ such that, for every ABox $\mathcal{A}$:

$$\mathsf{cens}_{\mathsf{DD}}(\mathcal{A}) = \mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A}).$$

Example 15. Let $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ and $\mathcal{A}$ be as in Example 14 and Example 3, respectively. The set of deterministically disclosed atoms is $\mathsf{cens}_{\mathsf{DD}}(\mathcal{A}) = \mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A}) = \{\mathsf{WorksFor}(bob,a_1), \mathsf{WorksFor}(ann,a_2), \mathsf{Manages}(ann,a_2), \mathsf{BreachedBy}(a_2,h_2), \mathsf{GAgency}(a_2)\}$. On the other hand, we have that the set of deterministically censored atoms is $\mathsf{DCA}(\mathcal{E}_\succ,\mathcal{A}) = \{\mathsf{SServ}(a_2)\}$. □

The proposition below follows immediately from the definition of DD censor.[5]

Proposition 10. *Let $\mathcal{E}_\emptyset = \langle \mathcal{T},\mathcal{P},\emptyset\rangle$ be a prioritized CQE specification with an empty priority relation. The DD censor for $\mathcal{E}_\emptyset$ coincides with the IGA censor for $\langle \mathcal{T},\mathcal{P}\rangle$.*

It is also easy to verify that the DD censor satisfies the property given in Definition 4.

Proposition 11. *Let $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ be a prioritized CQE specification. The DD censor $\mathsf{cens}_{\mathsf{DD}}(\cdot)$ for $\mathcal{E}_\succ$ satisfies the indistinguishability property.*

The following proposition, whose proof is straightforward, establishes the relationship between DD censors and the previously presented IP and IG censors.

Proposition 12. *Let $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ be a prioritized CQE specification, and let $\mathsf{cens}_{\mathsf{IP}}(\cdot)$ and $\mathsf{cens}_{\mathsf{IG}}(\cdot)$ be the Intersection Pareto and Global censor for $\mathcal{E}_\succ$, respectively. Then, for every ABox $\mathcal{A}$ we have that:*

$$\mathsf{cens}_{\mathsf{DD}}(\mathcal{A}) \subseteq \mathsf{cens}_{\mathsf{IP}}(\mathcal{A}) \subseteq \mathsf{cens}_{\mathsf{IG}}(\mathcal{A}).$$

Query entailment under DD censors is defined as usual.

Definition 14. Let $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ be a prioritized CQE specification, $\mathcal{A}$ be an ABox, and q be a Boolean query. DD-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$ is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}_{\mathsf{DD}}(\mathcal{A}) \models q$.

From Proposition 12, it follows that DD-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$ implies IP-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$ (and consequently IG-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$), for every prioritized CQE specification $\mathcal{E}_\succ$, ABox $\mathcal{A}$, and Boolean query q.

Given a prioritized CQE specification $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ and an ABox $\mathcal{A}$, it is not difficult to see that $\mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A})$ and $\mathsf{DCA}(\mathcal{E}_\succ,\mathcal{A})$ correspond to the least fixpoint of the equations:

$$\begin{aligned}\mathsf{DDA}_{i+1}(\mathcal{E}_\succ,\mathcal{A}) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}) \mid \forall \mathcal{S} \in inSec(\mathcal{T},\mathcal{P},\mathcal{A},\alpha)\ \exists \beta \in \mathcal{S}\\ \text{s.t. } \alpha \succ \beta \lor (\alpha \neq \beta \land \beta\\ \in \mathsf{DCA}_i(\mathcal{E}_\succ,\mathcal{A}))\}\\ \mathsf{DCA}_{i+1}(\mathcal{E}_\succ,\mathcal{A}) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}) \mid \exists \mathcal{S} \in inSec(\mathcal{T},\mathcal{P},\mathcal{A},\alpha)\\ \text{s.t. } \mathcal{S} \setminus \mathsf{DDA}_i(\mathcal{E}_\succ,\mathcal{A}) = \{\alpha\}\}\end{aligned}$$

where $\mathsf{DDA}_0(\mathcal{E}_\succ,\mathcal{A}) = \mathsf{DCA}_0(\mathcal{E}_\succ,\mathcal{A}) = \emptyset$. Notice that, by definition, $\mathsf{DDA}_i(\mathcal{E}_\succ,\mathcal{A}) = \mathsf{DDA}_{i+1}(\mathcal{E}_\succ,\mathcal{A})$ holds for every odd integer i. For a prioritized DL-Lite$_\mathcal{R}$ CQE specifications $\mathcal{E}_\succ$ and an ABox $\mathcal{A}$, computing such fixpoint is in P in the size of $\mathcal{A}$, and from the results in [16] it also follows that, for BCQs q, DD-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$ is P-hard in data complexity.

By fixing a positive integer k, we can define a new censor that we call *k-DD censor*.

Definition 15. Let $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ be a prioritized CQE specification and k be a positive integer. The *k-DD censor* for $\mathcal{E}_\succ$ is the function $\mathsf{cens}_{\mathsf{DD}_k}(\cdot)$ such that, for every ABox $\mathcal{A}$:

$$\mathsf{cens}_{\mathsf{DD}_k}(\mathcal{A}) = \mathsf{DDA}_k(\mathcal{E}_\succ,\mathcal{A}).$$

We next define Boolean query entailment under k-DD censors.

Definition 16. Let $\mathcal{E}_\succ = \langle \mathcal{T},\mathcal{P},\succ\rangle$ be a prioritized CQE specification, k be a positive integer, $\mathcal{A}$ be an ABox, and q be a Boolean query. kDD-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$ is the problem of deciding whether $\mathcal{T} \cup \mathsf{cens}_{\mathsf{DD}_k}(\mathcal{A}) \models q$.

In the rest of this paper, we study the above problem for prioritized DL-Lite$_\mathcal{R}$ CQE specifications and BUCQs as queries.

Since for every prioritized CQE specification $\mathcal{E}_\succ$, positive integer k, and ABox $\mathcal{A}$, $\mathsf{DDA}_k(\mathcal{E}_\succ,\mathcal{A}) \subseteq \mathsf{DDA}(\mathcal{E}_\succ,\mathcal{A})$, the k-DD censor for $\mathcal{E}_\succ$ constitutes a sound approximation of the DD censor for $\mathcal{E}_\succ$, and thus kDD-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$ implies DD-Cens-Entailment$(\mathcal{E}_\succ,\mathcal{A},q)$ for every Boolean query q. Moreover, it is immediate to verify that the k-DD censor preserves the indistinguishability property.

[5] A similar result is provided in [16, Theorem 38] in the context of CQA.

Example 16. Let $\mathcal{E}_{\succ} = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ and $\mathcal{A}$ be, respectively, the CQE specification and the ABox of the running example. For $k = 3$, the k-DD censor can be computed as follows:

$$
\begin{aligned}
&\mathsf{DDA}_0(\mathcal{E}_{\succ}, \mathcal{A}) = \mathsf{DCA}_0(\mathcal{E}_{\succ}, \mathcal{A}) = \emptyset\\
&\mathsf{DDA}_1(\mathcal{E}_{\succ}, \mathcal{A}) = \{\mathsf{WorksFor}(bob, a_1),\ \mathsf{WorksFor}(ann, a_2),\\
&\qquad\qquad\mathsf{Manages}(ann, a_2),\\
&\qquad\qquad\mathsf{GAgency}(a_2)\}\\
&\mathsf{DCA}_1(\mathcal{E}_{\succ}, \mathcal{A}) = \emptyset\\
&\mathsf{DDA}_2(\mathcal{E}_{\succ}, \mathcal{A}) = \mathsf{DDA}_1(\mathcal{E}_{\succ}, \mathcal{A})\\
&\mathsf{DCA}_2(\mathcal{E}_{\succ}, \mathcal{A}) = \{\mathsf{SServ}(ann)\}\\
&\mathsf{DDA}_3(\mathcal{E}_{\succ}, \mathcal{A}) = \mathsf{DDA}_2(\mathcal{E}_{\succ}, \mathcal{A}) \cup \{\mathsf{BreachedBy}(a_2, h_2)\}
\end{aligned}
$$

Note that for $k = 3$ we reach the fixpoint, i.e. $\mathsf{DDA}(\mathcal{E}_{\succ}, \mathcal{A}) = \mathsf{DDA}_3(\mathcal{E}_{\succ}, \mathcal{A})$ and $\mathsf{DCA}(\mathcal{E}_{\succ}, \mathcal{A}) = \mathsf{DCA}_3(\mathcal{E}_{\succ}, \mathcal{A})$. □

In what follows, we show that kDD-Cens-Entailment$(\mathcal{E}_{\succ}, \mathcal{A}, q)$ is FO-rewritable in our considered scenario for any positive integer k, that is, for every prioritized DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E}_{\succ}$ and BUCQ q, one can effectively compute an FO query q_r such that, for every ABox $\mathcal{A}$, kDD-Cens-Entailment$(\mathcal{E}_{\succ}, \mathcal{A}, q)$ is true if and only if $\mathsf{eval}(\mathcal{A}, q_r)$ is true. We call such a query q_r the *k-DD reformulation* of q with respect to $\mathcal{E}_{\succ}$.

8.2. Query rewriting algorithm

We now give our query rewriting technique for solving kDD-Cens-Entailment$(\mathcal{E}_{\succ}, \mathcal{A}, q)$.

In the following, with a slight abuse of notation, for atoms $\alpha = S_1(\vec{x})$ and $\beta = S_2(\vec{y})$ and a priority relation $\succ$, we say that $\alpha \succ \beta$ if $S_1 \succ S_2$. Now, let α be an atom, Q be a set of FO formulas, and $\succ$ be a priority relation. We denote by $\mathsf{notPreferred}(\alpha, Q, \succ)$ the set of all formulas $\phi \in Q$ such that there does not occur in ϕ any atom β such that $\alpha \succ \beta$.

Let ϕ be the BCQ with inequalities $\exists \vec{x}.\alpha \wedge \beta_1 \wedge \ldots \wedge \beta_n \wedge \lambda_1 \wedge \ldots \wedge \lambda_h$, where α and each β_i are predicate atoms and each λ_i is an inequality, and let $\mathcal{E}_{\succ}$ be a prioritized DL-Lite$_{\mathcal{R}}$ CQE specification. We define:

- $\mathsf{allDD}_i(\phi, \alpha, \mathcal{E}_{\succ}) = \exists \vec{y}.\mathsf{DD}_i(\beta_1, \mathcal{E}_{\succ}) \wedge \ldots \wedge \mathsf{DD}_i(\beta_n, \mathcal{E}_{\succ})$, where $\vec{y}$ are the variables in $\vec{x}$ that do not occur in α;
- $\mathsf{oneDC}_i(\phi, \alpha, \mathcal{E}_{\succ}) = \mathsf{DC}_i(\beta_1, \mathcal{E}_{\succ}) \vee \ldots \vee \mathsf{DC}_i(\beta_n, \mathcal{E}_{\succ})$.

By convention, if $i = 0$, then $\mathsf{allDD}_i(\phi, \alpha, \mathcal{E}_{\succ}) = true$ and $\mathsf{oneDC}_i(\phi, \alpha, \mathcal{E}_{\succ}) = false$. Moreover, we denote by $\mathcal{Q}_{\mathcal{P}}$ the set of BCQs with inequalities corresponding to the extended denials returned by $\mathsf{MinPolicy}(\mathcal{P}, \mathcal{T})$, i.e. $\mathcal{Q}_{\mathcal{P}} = \{q_\delta \mid \delta \in \mathsf{MinPolicy}(\mathcal{P}, \mathcal{T})\}$.

For an atom α, a prioritized DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E}_{\succ} = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, and a positive integer i, we define $\mathsf{DD}_i(\alpha, \mathcal{E}_{\succ})$ as follows:

$$
\mathsf{DD}_i(\alpha, \mathcal{E}_{\succ}) =
$$
$$
\alpha \wedge \left(\bigwedge_{\substack{\forall q_\delta \in \mathsf{notPreferred}(\alpha, \mathcal{Q}_{\mathcal{P}}, \succ),\\ \forall \beta \in \mathsf{compSet}(\alpha, q_\delta)}} \forall \vec{w}.\left(\neg \mu_{\alpha/\beta}(q_\delta) \vee \mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha, \mathcal{E}_{\succ})\right) \right),
$$

where $\vec{w}$ contains all the variables in the various $\mu_{\alpha/\beta}(q_\delta)$ that do not occur in α. In case $i = 0$, we impose $\mathsf{DD}_0(\alpha, \mathcal{E}_{\succ}) = false$.

We also define $\mathsf{DC}_i(\alpha, \mathcal{E}_{\succ})$ as follows:

$$
\mathsf{DC}_i(\alpha, \mathcal{E}_{\succ}) = \bigvee_{\substack{\forall q_\delta \in \mathsf{notPreferred}(\alpha, \mathcal{Q}_{\mathcal{P}}, \succ),\\ \forall \beta \in \mathsf{compSet}(\alpha, q_\delta)}} \exists \vec{v}.\mu_{\alpha/\beta}(q_\delta) \wedge \mathsf{allDD}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha, \mathcal{E}_{\succ}),
$$

where $\vec{v}$ contains all the variables in the various $\mu_{\alpha/\beta}(q_\delta)$ that do not occur in α. In case $i = 0$, we impose $\mathsf{DC}_0(\alpha, \mathcal{E}_{\succ}) = false$.

Given a BUCQ Q, a prioritized DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E}_{\succ}$, and a positive integer k, we define the Boolean FO query k-$\mathsf{DDQ}(Q, \mathcal{E}_{\succ})$ as follows:

$$
k\text{-}\mathsf{DDQ}(Q, \mathcal{E}_{\succ}) = \bigvee_{q \in Q} \left(\exists \vec{x}_q . \bigwedge_{\alpha \in q} \mathsf{DD}_k(\alpha, \mathcal{E}_{\succ}) \right),
$$

where, for every BCQ $q \in Q$, $\vec{x}_q$ denotes the existential variables of q.

Finally, for a BUCQ q, a prioritized DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E}_{\succ} = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$, and a positive integer k, we define:

$$
k\text{-}\mathsf{DDRew}(q, \mathcal{E}_{\succ}) = \mathsf{expand}(\mathcal{T}, k\text{-}\mathsf{DDQ}(\mathsf{PerfectRef}(q, \mathcal{T}), \mathcal{E}_{\succ})).
$$

Notice that, for every odd i, $\mathsf{DD}_i(\alpha, \mathcal{E}_{\succ}) = \mathsf{DD}_{i+1}(\alpha, \mathcal{E}_{\succ})$ (by definition), and thus i-$\mathsf{DDRew}(q, \mathcal{E}_{\succ}) = (i{+}1)$-$\mathsf{DDRew}(q, \mathcal{E}_{\succ})$.

Example 17. Consider the prioritized DL-Lite$_{\mathcal{R}}$ CQE specification $\mathcal{E}_{\succ} = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ and BUCQ q of the running example. For computing the k-DD reformulation k-$\mathsf{DDQ}(Q, \mathcal{E}_{\succ})$ for $k = 3$, we first need to compute the formula $\mathsf{DD}_3(\alpha, \mathcal{E}_{\succ})$ for each atom α occurring in $\mathsf{PerfectRef}(q, \mathcal{T})$ (recall that $\mathsf{PerfectRef}(q, \mathcal{T}) = \{\exists x, y.\mathsf{OperatesIn}(x, y), \exists x, y.\mathsf{BreachedBy}(x, y), \exists x.\mathsf{SServ}(x)\}$). This can be done as follows.

$\mathsf{DD}_0(\alpha, \mathcal{E}_{\succ}) = \mathsf{DC}_0(\alpha, \mathcal{E}_{\succ}) = false,$ for each atom α.

$$
\begin{aligned}
&\mathsf{DD}_1(\mathsf{OperatesIn}(x, y), \mathcal{E}_{\succ}) = \mathsf{OperatesIn}(x, y)\\
&\mathsf{DD}_1(\mathsf{GAgency}(x), \mathcal{E}_{\succ}) = \mathsf{GAgency}(x)\\
&\mathsf{DD}_1(\mathsf{WorksFor}(x, y), \mathcal{E}_{\succ}) = \mathsf{WorksFor}(x, y)\\
&\mathsf{DD}_1(\mathsf{Manages}(x, y), \mathcal{E}_{\succ}) = \mathsf{Manages}(x, y)\\
&\mathsf{DD}_1(\mathsf{BreachedBy}(x, y), \mathcal{E}_{\succ}) = \mathsf{BreachedBy}(x, y)\\
&\qquad\qquad \wedge (\neg(\mathsf{BreachedBy}(x, y) \wedge\\
&\qquad\qquad \mathsf{SServ}(x)) \vee \mathsf{DC}_0(\mathsf{SServ}(x), \mathcal{E}_{\succ}))\\
&\qquad\qquad = \mathsf{BreachedBy}(x, y) \wedge \neg\mathsf{SServ}(x)\\
&\mathsf{DD}_1(\mathsf{SServ}(x), \mathcal{E}_{\succ}) = \mathsf{SServ}(x) \wedge\\
&\quad \forall y.(\neg(\mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(x, y)) \vee \mathsf{DC}_0(\mathsf{BreachedBy}(x, y), \mathcal{E}_{\succ})) \wedge\\
&\quad \forall y.(\neg(\mathsf{SServ}(x) \wedge \mathsf{Manages}(y, x)) \vee \mathsf{DC}_0(\mathsf{Manages}(y, x), \mathcal{E}_{\succ})) \wedge\\
&\quad \forall y.(\neg(\mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y, x)) \vee \mathsf{DC}_0(\mathsf{WorksFor}(y, x), \mathcal{E}_{\succ}))\\
&\qquad\qquad = \mathsf{SServ}(x) \wedge \forall y.(\neg\mathsf{BreachedBy}(x, y)) \wedge\\
&\qquad\qquad \forall y.(\neg\mathsf{Manages}(y, x)) \wedge \forall y.(\neg\mathsf{WorksFor}(y, x))
\end{aligned}
$$

$\mathsf{DC}_1(\alpha, \mathcal{E}_{\succ}) = false,$ for each atom α.

$\mathsf{DD}_2(\alpha, \mathcal{E}_{\succ}) = \mathsf{DD}_1(\alpha, \mathcal{E}_{\succ}),$ for each atom α.

$$
\begin{aligned}
&\mathsf{DC}_2(\mathsf{WorksFor}(x, y), \mathcal{E}_{\succ}) = false\\
&\mathsf{DC}_2(\mathsf{Manages}(x, y), \mathcal{E}_{\succ}) = false\\
&\mathsf{DC}_2(\mathsf{BreachedBy}(x, y), \mathcal{E}_{\succ}) = \mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(x, y)\\
&\qquad\qquad \wedge \mathsf{DD}_1(\mathsf{SServ}(x), \mathcal{E}_{\succ})\\
&\qquad\qquad = \mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(x, y)\\
&\qquad\qquad \wedge \mathsf{SServ}(x) \wedge\\
&\qquad\qquad \forall y'.(\neg\mathsf{BreachedBy}(x, y')) \wedge\\
&\qquad\qquad \forall y'.(\neg\mathsf{Manages}(y', x)) \wedge\\
&\qquad\qquad \forall y'.(\neg\mathsf{WorksFor}(y', x)) = false\\
&\mathsf{DC}_2(\mathsf{SServ}(x), \mathcal{E}_{\succ}) =\\
&\qquad \exists y.(\mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(x, y)\\
&\qquad \wedge \mathsf{DD}_1(\mathsf{BreachedBy}(x, y), \mathcal{E}_{\succ})) \vee\\
&\qquad \exists y.(\mathsf{SServ}(x) \wedge \mathsf{Manages}(y, x)\\
&\qquad \wedge \mathsf{DD}_1(\mathsf{Manages}(y, x), \mathcal{E}_{\succ})) \vee\\
&\qquad \exists y.(\mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y, x)\\
&\qquad \wedge \mathsf{DD}_1(\mathsf{WorksFor}(y, x), \mathcal{E}_{\succ}))\\
&\qquad = \exists y.(\mathsf{SServ}(x) \wedge \mathsf{Manages}(y, x))\\
&\qquad \vee \exists y.(\mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y, x))
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{DD}_3(\mathsf{OperatesIn}(x, y), \mathcal{E}_{\succ}) = \mathsf{OperatesIn}(x, y)\\
&\mathsf{DD}_3(\mathsf{BreachedBy}(x, y), \mathcal{E}_{\succ}) = \mathsf{BreachedBy}(x, y) \wedge (\neg(\mathsf{BreachedBy}(x, y) \wedge\\
&\qquad\qquad \mathsf{SServ}(x)) \vee \mathsf{DC}_2(\mathsf{SServ}(x), \mathcal{E}_{\succ}))\\
&\qquad\qquad = \mathsf{BreachedBy}(x, y) \wedge (\neg\mathsf{SServ}(x) \vee \exists y'.(\mathsf{SServ}(x) \wedge\\
&\qquad\qquad \mathsf{Manages}(y', x)) \vee \exists y'.(\mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y', x)))\\
&\mathsf{DD}_3(\mathsf{SServ}(x), \mathcal{E}_{\succ}) = \mathsf{SServ}(x) \wedge\\
&\quad \forall y.(\neg(\mathsf{SServ}(x) \wedge \mathsf{BreachedBy}(x, y)) \vee \mathsf{DC}_2(\mathsf{BreachedBy}(x, y), \mathcal{E}_{\succ})) \wedge\\
&\quad \forall y.(\neg(\mathsf{SServ}(x) \wedge \mathsf{Manages}(y, x)) \vee \mathsf{DC}_2(\mathsf{Manages}(y, x), \mathcal{E}_{\succ})) \wedge\\
&\quad \forall y.(\neg(\mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y, x)) \vee \mathsf{DC}_2(\mathsf{WorksFor}(y, x), \mathcal{E}_{\succ}))\\
&\qquad\qquad = \mathsf{SServ}(x) \wedge \forall y.(\neg\mathsf{BreachedBy}(x, y)) \wedge\\
&\qquad\qquad \forall y.(\neg\mathsf{Manages}(y, x)) \wedge \forall y.(\neg\mathsf{WorksFor}(y, x))
\end{aligned}
$$

Now, called $q_r = \mathsf{PerfectRef}(q, \mathcal{T})$, we have that:

$$\begin{aligned}
k\text{-}\mathsf{DDQ}(q_r,\mathcal{E}_\succ) = \; & \exists x, y. \mathsf{OperatesIn}(x, y) \vee \\
& \exists x. \Big(\mathsf{SServ}(x) \wedge \forall y. (\neg \mathsf{BreachedBy}(x, y)) \wedge \forall y. (\neg \mathsf{Manages}(y, x)) \wedge \\
& \quad \forall y. (\neg \mathsf{WorksFor}(y, x)) \Big) \vee \\
& \exists x, y. \Big(\Big(\neg \mathsf{SServ}(x) \vee \exists y'. (\mathsf{SServ}(x) \wedge \mathsf{Manages}(y', x)) \vee \\
& \qquad \exists y'. (\mathsf{SServ}(x) \wedge \mathsf{WorksFor}(y', x)) \Big) \wedge \mathsf{BreachedBy}(x, y) \Big)
\end{aligned}$$

and finally:

$$\begin{aligned}
k\text{-}\mathsf{DDRew}(q,\mathcal{E}_\succ) = \; & \exists x, y. \mathsf{OperatesIn}(x, y) \vee \\
& \exists x. \Big(\mathsf{SServ}(x) \wedge \forall y. (\neg \mathsf{BreachedBy}(x, y)) \wedge \forall y. (\neg \mathsf{Manages}(y, x)) \wedge \\
& \quad \forall y. (\neg (\mathsf{Manages}(y, x) \vee \mathsf{WorksFor}(y, x))) \Big) \vee \\
& \exists x, y. \Big(\Big(\neg \mathsf{SServ}(x) \vee \exists y'. (\mathsf{SServ}(x) \wedge \mathsf{Manages}(y', x)) \vee \\
& \qquad \exists y'. (\mathsf{SServ}(x) \wedge (\mathsf{Manages}(y', x) \vee \mathsf{WorksFor}(y', x))) \Big) \wedge \\
& \quad \mathsf{BreachedBy}(x, y) \Big) \quad \square
\end{aligned}$$

It is easy to see that k-$\mathsf{DDRew}(q, \mathcal{E}_\succ)$ is a Boolean FO query. The following theorem states that, for prioritized DL-Lite$_\mathcal{R}$ CQE specifications $\mathcal{E}_\succ$ and BUCQs q, kDD-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$ can always be solved by checking whether k-$\mathsf{DDRew}(q, \mathcal{E}_\succ)$ evaluates to true in $\mathcal{A}$, and this holds for every fixed positive integer k. In other terms, the problem is FO-rewritable. As done for Theorem 4, we do this on the basis of Proposition 1 and Lemma 5.

Theorem 5. *Let $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{P}, \succ \rangle$ be a prioritized DL-Lite$_\mathcal{R}$ CQE specification, k be a positive integer, $\mathsf{cens}_{\mathsf{DD}_k}(\cdot)$ be the k-DD censor for $\mathcal{E}_\succ$, and q be a BUCQ. For every ABox $\mathcal{A}$, we have that $\mathcal{T} \cup \mathsf{cens}_{\mathsf{DD}_k}(\mathcal{A}) \models q$ if and only if $\mathsf{eval}(\mathcal{A}, k\text{-}\mathsf{DDRew}(q, \mathcal{E}_\succ))$ is true.*

Proof. By exploiting Proposition 1 and Lemma 5, to prove the thesis of the theorem note that it is enough to show the following: given a BUCQ Q, we have that $\mathsf{eval}(\mathsf{cens}_{\mathsf{DD}_k}(\mathcal{A}), Q) = \mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), k\text{-}\mathsf{DDQ}(Q, \mathcal{E}_\succ))$.

First, we prove inductively the following property: for every integer i such that $0 \leq i \leq k$, and for every ground atom α, we have that $\alpha \in \mathsf{DDA}_i(\mathcal{E}_\succ, \mathcal{A})$ if and only if $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DD}_i(\alpha, \mathcal{E}_\succ))$ is true and $\alpha \in \mathsf{DCA}_i(\mathcal{E}_\succ, \mathcal{A})$ if and only if $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DC}_i(\alpha, \mathcal{E}_\succ))$ is true.

The base case trivially holds since $\mathsf{DDA}_0(\mathcal{E}_\succ, \mathcal{A}) = \mathsf{DCA}_0(\mathcal{E}_\succ, \mathcal{A}) = \emptyset$ and $\mathsf{DD}_0(\alpha, \mathcal{E}_\succ) = \mathsf{DC}_0(\alpha, \mathcal{E}_\succ) = false$.

We now prove the inductive case. We only prove that $\alpha \in \mathsf{DDA}_i(\mathcal{E}_\succ, \mathcal{A})$ if and only if $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DD}_i(\alpha, \mathcal{E}_\succ))$ is true. One can prove in an analogous way that $\alpha \in \mathsf{DCA}_i(\mathcal{E}_\succ, \mathcal{A})$ if and only if $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DC}_i(\alpha, \mathcal{E}_\succ))$.

Suppose that $\alpha \in \mathsf{DDA}_i(\mathcal{E}_\succ, \mathcal{A})$, i.e. $\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ and for each secret $\mathcal{S} \in inSec(\mathcal{T}, \mathcal{P}, \mathcal{A}, \alpha)$ either $\alpha \succ \beta$ for some $\beta \in \mathcal{S}$ or $\beta \in \mathsf{DCA}_{i-1}(\mathcal{E}_\succ, \mathcal{A})$ for some $\beta \in \mathcal{S}$ with $\alpha \neq \beta$. Consider the formula $\mathsf{DD}_i(\alpha, \mathcal{E}_\succ)$, which is of the form $\alpha \wedge \bigwedge \forall \vec{w}.(\neg \mu_{\alpha/\beta}(q_\delta) \vee \mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha), \mathcal{E}_\succ)$. In particular, consider each conjunct $\forall \vec{w}.(\neg \mu_{\alpha/\beta}(q_\delta) \vee \mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha, \mathcal{E}_\succ))$ in $\mathsf{DD}_i(\alpha, \mathcal{E}_\succ)$. By Proposition 8, either $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mu_{\alpha/\beta}(q_\delta))$ is false, and so α does not belong to any secret that violate δ, or there is a secret $\mathcal{S}$ violating δ such that $\alpha \in \mathcal{S}$. In the former case, we are done. In the latter case, by construction of $\mathsf{DD}_i(\alpha, \mathcal{E}_\succ)$ (in particular, the fact that $q_\delta \in \mathsf{notPreferred}(\alpha, \mathcal{Q}_\mathcal{P}, \succ)$), we have that there is no $\beta \in \mathcal{S}$ with $\alpha \succ \beta$. By the assumption that $\alpha \in \mathsf{DDA}_i(\mathcal{E}_\succ, \mathcal{A})$, it follows that $\beta \in \mathsf{DCA}_{i-1}(\mathcal{E}_\succ, \mathcal{A})$ for some $\beta \in \mathcal{S}$ with $\alpha \neq \beta$. But then, due to the inductive hypothesis, we derive that $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DC}_{i-1}(\beta, \mathcal{E}_\succ))$ is true, and therefore $\mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha, \mathcal{E}_\succ)$ is true as well. Thus, also in the latter case we have that $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \forall \vec{w}.(\neg \mu_{\alpha/\beta}(q_\delta) \vee \mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha, \mathcal{E}_\succ)))$ is true, from which we derive that $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DD}_i(\alpha, \mathcal{E}_\succ))$ is true. Now, suppose that $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DD}_i(\alpha, \mathcal{E}_\succ))$ is true, i.e. $\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A})$ and each $\forall \vec{w}.(\neg \mu_{\alpha/\beta}(q_\delta) \vee \mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha, \mathcal{E}_\succ))$ in $\mathsf{DD}_i(\alpha, \mathcal{E}_\succ)$ is such that $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \forall \vec{w}.(\neg \mu_{\alpha/\beta}(q_\delta) \vee \mathsf{oneDC}_{i-1}(\mu_{\alpha/\beta}(q_\delta), \alpha, \mathcal{E}_\succ)))$ is true. From the latter, by Proposition 8, we derive that every secret $\mathcal{S}$ with $\alpha \in \mathcal{S}$ for which there is no atom $\beta \in \mathcal{S}$ with $\alpha \succ \beta$ is such that there exists an atom $\beta \in \mathcal{S}$ with $\mathsf{eval}(\mathsf{cl}^{\mathcal{T}}_{\mathbf{GA}}(\mathcal{A}), \mathsf{DC}_{i-1}(\beta, \mathcal{E}_\succ))$ true. By the inductive hypothesis, we derive that $\beta \in \mathsf{DCA}_{i-1}(\mathcal{E}_\succ, \mathcal{A})$. Since this holds for every possible secret $\mathcal{S}$ with $\alpha \in \mathcal{S}$ and for which there is no atom $\beta \in \mathcal{S}$ with $\alpha \succ \beta$, by definition we have that $\alpha \in \mathsf{DDA}_i(\mathcal{E}_\succ, \mathcal{A})$.

Now, one can prove the thesis of the theorem following similar considerations as done in the proof of Theorem 4 (it is indeed sufficient to replace $\mathsf{D}(\alpha, \mathcal{E})$ with $\mathsf{DD}_k(\alpha, \mathcal{E}_\succ)$ and $\mathsf{cens}_{\mathsf{IGA}}(\mathcal{A})$ with $\mathsf{cens}_{\mathsf{DD}_k}(\mathcal{A})$, and follow the same line of reasoning). ■

Example 18. Let $\mathcal{E}_\succ$, $\mathcal{A}$ and q be as in Example 14, Examples 1 and 5, respectively, and consider the perfect reformulation 3-$\mathsf{DDRew}(q, \mathcal{E}_\succ)$ of Example 17. One can verify that $\mathsf{eval}(\mathcal{A}, 3\text{-}\mathsf{DDRew}(q_r, \mathcal{E}_\succ))$ is true. Indeed, for $k = 3$ we have that $\mathcal{T} \cup \mathsf{cens}_{\mathsf{DD}_k}(\mathcal{A}) \models q$.

It is also worth noting that, for $k = 1$, we have that $\mathsf{eval}(\mathcal{A}, k\text{-}\mathsf{DDRew}(q_r, \mathcal{E}_\succ))$ is false and $\mathcal{T} \cup \mathsf{cens}_{\mathsf{DD}_k}(\mathcal{A}) \not\models q$. □

The corollary below follows from Theorem 5 and the fact that evaluating an FO query over an ABox is in AC^0 in the size of the ABox.

Corollary 2. *Let k be a positive integer. kDD-Cens-Entailment$(\mathcal{E}_\succ, \mathcal{A}, q)$ for prioritized DL-Lite$_\mathcal{R}$ CQE specifications $\mathcal{E}_\succ$ and BUCQs q is in AC^0 in data complexity.*

9. Related work

Similarly as we do in the present paper, some previous works on CQE have proposed techniques to conceal sensitive data in a manner that makes it impossible for a user to discern the actual database or knowledge base from an alternative version without secrets, as, e.g., in [3,5–8]. Using the terminology proposed in this paper, we may say that all such papers study censors that, even if defined in different, often incomparable ways, enjoy the indistinguishability property that we formulate in Definition 4. As pointed out in [7] and more recently in [12], and as we already remarked in previous sections, censors of this form proved to be more robust than censors not enjoying indistinguishability : the latter, indeed, may be subject to attacks based on object-level and/or meta-level background knowledge. This behavior makes such a property desirable in CQE, and is the behavior we guarantee with the notions of censors studied in this paper.

Our investigation, however, differs form previous ones in various aspects. Compared to the initial work on CQE, such as [1,3,5,6], which merit the introduction and study of CQE in the context of databases, our research addresses this problem in the ambit of Description Logic ontologies and Semantic Web applications, similarly as [7–10]. CQE in this scenario is particularly challenging, considered that the usage of ontologies enables the deduction of implicit information from explicit data, which further escalates the risk of information leakage.

In this respect, latter references are closer to our work. In more detail, Ref. [8] defines CQE for ontologies specified in Boolean $\mathcal{ALC}$ and queries that are $\mathcal{ALC}$ formulas. The paper identifies useful and reasonable properties that a censor should have for protecting confidentiality. In [9] the authors study CQE for ontologies in OWL 2 RL, one of the tractable profile of OWL 2 [22], and a policy expressed by a set of ground atoms. One of the main outcomes of this paper is the identification of a subset of OWL 2 RL for which the computation of a censor is tractable. It has to be noted that the policy language considered in the two works above is not able to consider protection rules involving CQs, as we do. Complexity of censor computation is also studied in [10], where the same authors extend the framework of [9] to consider ontologies specified either in Datalog or in any of the OWL 2 profiles, with the policy expressed as a single CQ.

It is worthwhile remarking that all papers we mentioned so far focus on the problem of verifying the existence of a (optimal) censor ensuring a secure view for the data and on how to construct it. This approach is similar in spirit to what we discuss in Section 4, where we provide an algorithm that returns, in polynomial time in data complexity, one GA censor among several possible optimal ones. This censor is then used for query answering, so that answers to queries are altered exclusively according to the selected censor. Whereas on the one hand picking up an optimal censor guarantees confidentiality protection, on the other hand the selection is typically arbitrary, if there are no additional metadata supporting such a choice. To overcome this problem, in the present paper we also consider a different strategy that involves reasoning skeptically with respect to all optimal censors (Definition 5). This alternative approach was already briefly discussed in [9] and then elaborated and deeply investigated in [11]. In more detail, the latter paper generalizes the notion of censor given in [9,10], by introducing the so-called censor language as a parameter. This idea is recalled here in Definition 2. Then, [11] studies entailment of CQs under GA censors and censors in **CQ** (cf. Definition 5) from the computational viewpoint, for ontologies in $\mathcal{EL}_\bot$ and DL-Lite$_\mathcal{R}$, the logics at the basis of the OWL 2 profiles OWL 2 EL and OWL 2 QL, respectively, and for policies specified in terms of CQs, analogously to the present article. Indistinguishability is however not considered (this aspect is also overlooked in [9,10]). One of the aim of the present paper is to clarify the relationship between the notions of censors of [11] and censors that require the existence of an indistinguishable ABox. As shown in Section 3, GA censors enjoy this property, whereas censors in **CQ** do not.

Further studies on information disclosure leveraging an indistinguishability-based notion of policy compliance have then been developed in the context of information integration. In [25], the authors consider the setting of Ontology-Based Data Access (OBDA) and study the problem of determining whether information that is declared confidential at the data sources through a protection policy, as in CQE, can be inferred by a user only on the basis of the data she can access through the OBDA system, assuming that she is knowledgeable about the OBDA specification. The paper studies the computational complexity of this problem, under various assumptions on the forms of the mapping (GAV or LAV [31]) and on the complexity of conjunctive query entailment for the given ontology language. In [32], a similar problem is studied for data integration systems in the presence of constraints specified over the data sources. CQE in OBDA is also considered in [26], but indistinguishability is not explicitly investigated. The latter paper, however, uses GA censors and introduces the notion of IGA censor studied in the present work (Section 6). As discussed in the previous sections, besides enjoying indistinguishability (Proposition 5), IGA censors have a desirable computational behavior, allowing for conjunctive query answering in AC^0 in data complexity (Corollary 1). On the other hand, an IGA censor in general conceals more data than other forms of censors (e.g., GA censors) to preserve confidentiality. Equipping ontologies with preferences between predicates expressing priorities on the way in which data should be censored may help to mitigate this last issue, as we discussed in Sections 7 and 8.

To the best of our knowledge, this is the first paper considering CQE over prioritized ontologies. The priority-based CQE semantics we propose are adapted from the literature on Consistent Query Answering (CQA). CQA is a declarative approach to inconsistency management in databases and knowledge bases constructed around the notion of repair [30,33,34]. In the context of ontologies, a repair is often defined as a maximal subset of the ABox that is consistent with the TBox (even though alternative definitions have been even proposed in the literature). In other terms, repairs are obtained by resolving conflicts in all possible ways. To some extent, repairs in CQA act as censors in CQE, and query answering in the former framework amounts to skeptically reasoning with respect to all possible repairs, similarly as in CQE we reason with respect to all possible censors. An in-depth investigation on the connection between the two frameworks can be found in [11]. In the context of CQA, the use of preferences has been proposed originally in [29] to select a set of preferred repairs, and thus reducing the level of nondeterminism in reasoning. The setting of preferred repairs has then been investigated in several other papers, in the context of both databases (see, e.g., [35,36]) and ontologies (see, e.g., [16,37–39]).

Among the mentioned papers, the work [16] is certainly the closest to our research. Indeed, our DD censor has a correspondence with the grounded extension introduced in [16] through a transformation of the CQA problem into an argumentation framework. Also, our rewritability result corresponds to an analogous finding mentioned in that paper. Besides the differences between the settings studied in the two papers, we remark that priorities considered in [16] are specified between ABox facts, whereas we here assume priorities between ontology predicates, maintaining this aspect at the intensional level, and thus making priorities easier to manage from the modeling viewpoint. Furthermore, our treatment is tailored to CQE, and does not require transformation into a different problem, thus streamlining the technical aspects of the approach. Also, the rewriting algorithm that we provide allows us to easily exploit the idea of [26] for solving CQE over ontologies through the use of off-the-shelf tools for OBDA [40,41].

We conclude this related work section by mentioning the approach presented in [42]. That paper proposes a way to select censors in the CQE framework that is alternative to the specification of priorities described above. The idea is to use the order of queries posed to the ontology to define a dynamic selection criterion over the GA censors. Given its tight dependency on the history of the user queries, this approach is actually not comparable with the CQE framework of the present paper.

10. Conclusions

In this paper, we investigated CQE in Description Logics through the lens of instance indistinguishability. In particular, we studied different types of censors and identified the ones satisfying such a desirable property. We also introduced an intersection-based semantics for BUCQ entailment that soundly approximates skeptical reasoning while enjoying first-order rewritability in the case of DL-Lite$_\mathcal{R}$ ontologies. We then enriched the framework with the possibility of specifying a priority relation over the predicate signature, and presented a well-founded entailment semantics that improves the throughput of query answers while still preserving confidentiality. At the same time, this novel approach retains the indistinguishability property and maintains the favorable computational complexity of the intersection-based scenario.

We note that these results are not only theoretically significant but also have important practical implications. On one hand, it is worth highlighting the declarativeness of the proposed approach: it provides a system designer with powerful and easy mechanisms to express her protection needs, i.e. denials and preferences, requiring little design effort compared to classical ontology modeling activities. On the other hand, we have identified an implementable case involving widely-used ontology and query languages in the Semantic Web, which are well-suited for data-intensive scenarios. This enabled us to successfully apply these techniques in preliminary experiments using benchmarks for ontology-based data integration, leveraging off-the-shelf OBDA engines as described in [18,26].

Our current research is focused on the problem of intensionally (i.e. independently of the ABox) deciding whether, for a given CQE specification, there exists an integer k such that the k-DD censor converges on the DD censor. Interestingly, for such specifications, entailment of BUCQs under the DD censor is first-order rewritable.

As for future work, it would be highly intriguing to conduct a computational analysis involving ontologies specified in alternative DLs. Then, a possible improvement for our framework may consist in enriching the preference mechanism, e.g. adapting to the CQE setting one of the recent approaches to preferred repairs in CQA [37–39].

CRediT authorship contribution statement

Gianluca Cima: Conceptualization, Formal analysis, Methodology, Supervision, Writing – original draft. **Domenico Lembo:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – original draft. **Lorenzo Marconi:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – original draft. **Riccardo Rosati:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – original draft. **Domenico Fabio Savo:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by : projects FAIR (PE0000013) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan (PNRR) funded by the European Union - NextGenerationEU; GLACIATION project funded under the European Union's HE research and innovation programme (grant agreement No 101070141); ANTHEM project funded by the Italian National Plan for PNRR Complementary Investments (prj. n. PNC0000003 - CUP B53C22006700001).

Data availability

No data was used for the research described in the article.

References

[1] G.L. Sicherman, W. de Jonge, R.P. van de Riet, Answering queries without revealing secrets, ACM Trans. Database Syst. 8 (1) (1983) 41–59.
[2] J. Biskup, For unknown secrecies refusal is better than lying, Data Knowl. Eng. 33 (1) (2000) 1–23.
[3] J. Biskup, P.A. Bonatti, Controlled query evaluation for known policies by combining lying and refusal, Ann. Math. Artif. Intell. 40 (1–2) (2004) 37–62.
[4] J. Biskup, P.A. Bonatti, Lying versus refusal for known potential secrets, Data Knowl. Eng. 38 (2) (2001) 199–222.
[5] J. Biskup, P.A. Bonatti, Controlled query evaluation for enforcing confidentiality in complete information systems, Int. J. Inf. Secur. 3 (1) (2004) 14–27.
[6] J. Biskup, T. Weibert, Keeping secrets in incomplete databases, Int. J. Inf. Secur. 7 (3) (2008) 199–217.
[7] P.A. Bonatti, L. Sauro, A confidentiality model for ontologies, in: Proc. of the 12th Int. Semantic Web Conf, ISWC, in: Lecture Notes in Computer Science, vol. 8218, 2013, pp. 17–32.
[8] T. Studer, J. Werner, Censors for boolean description logic, Trans. Data Priv. 7 (3) (2014) 223–252.
[9] B. Cuenca Grau, E. Kharlamov, E.V. Kostylev, D. Zheleznyakov, Controlled query evaluation over OWL 2 RL ontologies, in: Proc. of the 12th Int. Semantic Web Conf, ISWC, in: Lecture Notes in Computer Science, vol. 8218, 2013 pp. 49–65.
[10] B. Cuenca Grau, E. Kharlamov, E.V. Kostylev, D. Zheleznyakov, Controlled query evaluation for Datalog and OWL 2 profile ontologies, in: Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2015, pp. 2883–2889.
[11] D. Lembo, R. Rosati, D.F. Savo, Revisiting controlled query evaluation in description logics, in: Proc. of the 28th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2019, pp. 1786–1792.
[12] P.A. Bonatti, A false sense of security, Artificial Intelligence 310 (103741) (2022).
[13] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, J. Automat. Reason. 39 (3) (2007) 385–429.
[14] B. Motik, A. Fokoue, I. Horrocks, Z. Wu, C. Lutz, B. Cuenca Grau, OWL Web Ontology Language Profiles, W3C Recommendation, World Wide Web Consortium, 2009, available at http://www.w3.org/TR/owl-profiles/.
[15] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Data complexity of query answering in description logics, Artificial Intelligence 195 (2013) 335–360.
[16] M. Bienvenu, C. Bourgaux, Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation, in: Proc. of the 17th Int. Conf. on the Principles of Knowledge Representation and Reasoning, KR, 2020, pp. 141–151.
[17] G. Cima, D. Lembo, R. Rosati, D.F. Savo, Controlled query evaluation in description logics through instance indistinguishability, in: Proc. of the 29th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2020, pp. 1791–1797.
[18] G. Cima, D. Lembo, L. Marconi, R. Rosati, D.F. Savo, Controlled query evaluation over prioritized ontologies with expressive data protection policies, in: Proc. of the 20th Int. Semantic Web Conf, ISWC, in: Lecture Notes in Computer Science, vol. 12922, Springer, 2021, pp. 374–391.
[19] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation and Applications, second ed., Cambridge University Press, 2007.
[20] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison Wesley Publ. Co., 1995.
[21] J.W. Lloyd, Foundations of Logic Programming (Second, Extended Edition), Springer, Berlin, Heidelberg, 1987.
[22] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 Web Ontology Language Profiles (Second Edition), W3C Recommendation, World Wide Web Consortium, 2012, available at http://www.w3.org/TR/owl2-profiles/.
[23] M.Y. Vardi, The complexity of relational query languages, in: Proc. of the 14th ACM SIGACT Symp. on Theory of Computing, STOC, 1982, pp. 137–146.
[24] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-Lite family and relations, J. Artificial Intelligence Res. 36 (2009) 1–69.
[25] M. Benedikt, B. Cuenca Grau, E.V. Kostylev, Logical foundations of information disclosure in ontology-based data integration, Artificial Intelligence 262 (2018) 52–95.
[26] G. Cima, D. Lembo, L. Marconi, R. Rosati, D.F. Savo, Controlled query evaluation in ontology-based data access, in: Proc. of the 19th Int. Semantic Web Conf, ISWC, 2020, pp. 128–146.
[27] J. Biskup, P.A. Bonatti, Controlled query evaluation with open queries for a decidable relational submodel, Ann. Math. Artif. Intell. 50 (1–2) (2007) 39–77.
[28] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D.F. Savo, Inconsistency-tolerant query answering in ontology-based data access, J. Web Semant. 33 (2015) 3–29.
[29] S. Staworko, J. Chomicki, J. Marcinkowski, Prioritized repairing and consistent query answering in relational databases, Ann. Math. Artif. Intell. 64 (2–3) (2012) 209–246.
[30] M. Arenas, L.E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proc. of the 18th ACM SIGMOD SIGACT SIGART Symp. on Principles of Database Systems, PODS, 1999, pp. 68–79.
[31] M. Lenzerini, Data integration: A theoretical perspective, in: Proc. of the 21st ACM SIGMOD SIGACT SIGART Symp. on Principles of Database Systems, PODS, 2002, pp. 233–246.
[32] M. Benedikt, P. Bourhis, L. Jachiet, M. Thomazo, Reasoning about disclosure in data integration in the presence of source constraints, in: Proc. of the 28th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2019, pp. 1551–1557.
[33] M. Bienvenu, C. Bourgaux, Inconsistency-tolerant querying of description logic knowledge bases, in: Reasoning Web. Semantic Technologies for Intelligent Data Access – 12th Int. Summer School Tutorial Lectures, RW, 2016, pp. 156–202.
[34] L.E. Bertossi, Database Repairing and Consistent Query Answering, in: Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2011.
[35] R. Fagin, B. Kimelfeld, P.G. Kolaitis, Dichotomies in the complexity of preferred repairs, in: Proc. of the 34rd ACM SIGMOD SIGACT SIGAI Symp. on Principles of Database Systems, PODS, 2015, pp. 3–15.
[36] B. Kimelfeld, E. Livshits, L. Peterfreund, Counting and enumerating preferred database repairs, Theoret. Comput. Sci. 837 (2020) 115–157.
[37] M. Bienvenu, C. Bourgaux, F. Goasdoué, Querying inconsistent description logic knowledge bases under preferred repair semantics, in: C.E. Brodley, P. Stone (Eds.), Proc. of the 28th AAAI Conf. on Artificial Intelligence, AAAI, AAAI Press, 2014, pp. 996–1002.
[38] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, Artificial Intelligence 312 (2022) 103772.
[39] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of inconsistency-tolerant query answering in datalog+/- under preferred repairs, in: Proc. of the 20th Int. Conf. on the Principles of Knowledge Representation and Reasoning, KR, 2023, pp. 472–481.
[40] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, G. Xiao, Ontop: Answering SPARQL queries over relational databases, Semant. Web J. 8 (3) (2017) 471–487.

[41] C. Civili, M. Console, G. De Giacomo, D. Lembo, M. Lenzerini, L. Lepore, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, V. Santarelli, D.F. Savo, MASTRO STUDIO: managing ontology-based data access applications, Proc. VLDB Endow. 6 (12) (2013) 1314–1317.

[42] P.A. Bonatti, G. Cima, D. Lembo, L. Marconi, R. Rosati, L. Sauro, D.F. Savo, Controlled query evaluation in OWL 2 QL: A longest honeymoon approach, in: Proc. of the 21st Int. Semantic Web Conf, ISWC, in: Lecture Notes in Computer Science, vol. 13489, 2022, pp. 428–444.