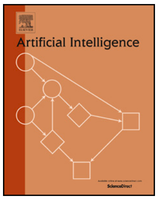
Artificial Intelligence 348 (2025) 104402

Contents lists available at ScienceDirect

Artificial Intelligence

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



Enhancing cooperativity in controlled query evaluation over ontologies

Piero Bonatti [a], Gianluca Cima [b], Domenico Lembo [b], Francesco Magliocca [a], Lorenzo Marconi [b], Riccardo Rosati [b], Luigi Sauro [a], Domenico Fabio Savo [c],*

[a] *Università degli Studi di Napoli Federico II, Italy*
[b] *Sapienza Università di Roma, Italy*
[c] *Università degli Studi di Bergamo, Italy*

ARTICLE INFO

Keywords:
Description logics
Ontologies
Confidentiality preservation
Query answering
Data complexity

ABSTRACT

Controlled Query Evaluation (CQE) is a methodology designed to maintain confidentiality by either rejecting specific queries or adjusting responses to safeguard sensitive information. In this investigation, our focus centers on CQE within Description Logic ontologies, aiming to ensure that queries are answered truthfully as long as possible before resorting to deceptive responses, a cooperativity property which is called the “longest honeymoon”. Our work introduces new semantics for CQE, denoted as MC-CQE, which enjoys the longest honeymoon property and outperforms previous methodologies in terms of cooperativity.

We study the complexity of query answering in this new framework for ontologies expressed in the Description Logic DL-Lite$_{\mathcal{R}}$. Specifically, we establish data complexity results under different maximally cooperative semantics and for different classes of queries. Our results identify both tractable and intractable cases. In particular, we show that the evaluation of Boolean unions of conjunctive queries is the same under all the above semantics and its data complexity is in AC^0. This result makes query answering amenable to SQL query rewriting. However, this favorable property does not extend to open queries, even with a restricted query language limited to conjunctions of atoms. While, in general, answering open queries in the MC-CQE framework is intractable, we identify a sub-family of semantics under which answering full conjunctive queries is tractable.

1. Introduction

Information systems often have to handle sensitive knowledge, such as medical records, personal details or financial data, which, if exposed without some protection, could compromise individuals’ privacy and confidentiality [1,2]. The objective of confidentiality-preserving query answering is to prevent unauthorized access to this information while maintaining a cooperative approach by providing honest answers to queries whenever possible. Controlled Query Evaluation (CQE) addresses this issue by refusing to respond to certain queries or altering answers when necessary to protect sensitive data. Initially studied for databases [3–5], CQE was then

* Corresponding author.
E-mail addresses: pab@unina.it (P. Bonatti), cima@diag.uniroma1.it (G. Cima), lembo@diag.uniroma1.it (D. Lembo), francesco.magliocca@unina.it (F. Magliocca), marconi@diag.uniroma1.it (L. Marconi), rosati@diag.uniroma1.it (R. Rosati), luigi.sauro@unina.it (L. Sauro), domenicofabio.savo@unibg.it (D.F. Savo).

https://doi.org/10.1016/j.artint.2025.104402
Received 6 March 2024; Received in revised form 4 August 2025; Accepted 5 August 2025

Available online 12 August 2025

investigated in the context of ontologies and Semantic Web applications [1,6,7], which have experienced a remarkable surge in popularity due to their ability to represent and interlink a wide range of information sources, encompassing both public organizations and private individuals. CQE over ontologies turned out to be particularly challenging, considering that the usage of ontologies enables the deduction of implicit information from explicit data, which further escalates the risk of information leakage. This scenario is the one considered in the present paper.

Clearly, the effectiveness of any confidentiality-preserving system in preventing information leakage depends greatly on the attacker's background knowledge and capabilities. Following [8], in this paper, we assume that a potential attacker is aware of the intensional component of the ontology (a.k.a. TBox), the set of potential secrets, and the algorithm used to protect confidentiality. In other words, the only piece of information unknown to the attacker is the extensional data (a.k.a. ABox) and its logical consequences. These assumptions allow us to encompass a number of practical scenarios. Indeed, TBoxes are typically public ontologies utilized within specific classes of applications. Even if this is not the case, the attacker may still possess some familiarity with the application domain and partially reconstruct the employed TBox. Similarly, a malicious user might be aware of the type of information that the system aims to keep confidential. For instance, one can expect that the government would not want to disclose the occupation of an employee acting as a spy. Lastly, the assumption that the attacker knows how sensitive data are preserved is an analogue of Kerckhoffs's principle for cryptoanalysis, the attacker could be an insider who knows or even designed the confidentiality-preserving system.

Under such assumptions, effective CQE is usually enforced by means of a data protection *policy*, i.e., a set of logic formulas that must be satisfied to ensure confidentiality, and by employing *censors*. These censors work by hiding information, while satisfying the policy, in a manner that makes it impossible to discern the actual knowledge from an alternative version devoid of any secrets. Various censors employ different techniques to obscure answers, making them generally not comparable with each other.

Several existing works propose static CQE methods, where a censor is pre-constructed or approximated, determining in advance which queries should be truthfully answered [1,6,7,9,10]. However, some of these approaches lack full cooperativeness, as they fail to consider the users' interests when selecting the secure view of the data.

In contrast, inspired by the work of Biskup and Bonatti [11], this paper introduces a family of *Maximally Cooperative CQE* (MC-CQE) semantics that dynamically decide whether to provide truthful answers or employ deception based on the stream of user queries. Essentially, at each step a MC-CQE semantics aims at maximizing the chances of answering the next query honestly by committing only to the current answer, as opposed to adhering a priori to a single censor. At the same time, MC-CQE semantics are *secure* under the assumptions mentioned above.

In more detail, MC-CQE semantics capture two essential desiderata: (i) indistinguishability-based confidentiality, and (ii) the so-called "longest honeymoon" property. Indistinguishability-based confidentiality [1,8,9] ensures security by mandating that the answers to a query are always obtainable by applying the censoring mechanism to the same query evaluated over an ontology without confidential information. This last ontology is thus indistinguishable from the original one for the users, preventing them from determining whether the underlying data contain sensitive information or not. Instead, the longest honeymoon [11] property states that, given a sequence of queries, the system returns the longest possible sequence of honest answers before resorting to deception. Several arguments support this property. Firstly, in the absence of specific knowledge about users' intentions, the order in which queries are posed is assumed to reflect their relative importance. Secondly, since we cannot anticipate the nature or number of future queries, responding truthfully to the current query, if feasible, represents the most cooperative strategy at this time. Furthermore, we will prove that our approach is optimal in a more classical sense: the set of queries truthfully answered is always maximal under set containment.

We compare our approach with two other semantics from the literature: skeptical reasoning [7] and IGA [12]. Our analysis reveals that skeptical reasoning fails to consistently guarantee indistinguishability-based privacy preservation. In contrast, our MC-CQE method demonstrates superior cooperativeness, comprehensively capturing all the answers that the compared methods provide.

More generally, we show that MC-CQE semantics cannot be replicated by a static CQE approach through data-independent modifications of the TBox and the formulas representing the data protection policy. Consequently, specific techniques are necessary to implement our approach.

From a computational perspective, we investigate the data complexity of query answering in MC-CQE when working with ontologies expressed in DL-Lite$_{\mathcal{R}}$, which is the Description Logic fragment that underlies the OWL 2 QL profile [13]. In this context, our query language includes the union of conjunctive queries, and data-protection policies are defined using denial formulas, as described in [9,10,14].

In case of Boolean queries (those without free variables), our approach results in being *first-order rewritable*, implying that queries can be answered in AC^0 in data complexity. When the framework is lifted to open queries, a query may have multiple maximally cooperative answers; thus, we introduce preference orderings to make a selection over these answers. Unfortunately, in such generalized setting, membership in AC^0 does not hold, even when we restrict the query language to only include conjunctions of atoms (full CQs). More broadly, we show that the data complexity of answering open conjunctive queries is $\Delta_2^p[O(\log n)]$-hard for every MC-CQE semantics. Nevertheless, if we narrow our focus to a specific subset of MC-CQE defined by lexicographic preference orderings over answers, the data complexity of open query answering becomes PTIME-complete for full CQs and Δ_2^p-complete for conjunctive queries.

The rest of the paper is organized as follows. In Section 2, we give some preliminary notions on Description Logics and Union of Conjunctive Queries. In Section 3, we introduce the MC-CQE semantics and compare it with skeptical reasoning and IGA. In Section 4, we focus on Boolean Unions of Conjunctive Queries (BUCQs) and prove that all possible MC-CQE semantics reduce to a single one, which we term dynCQE. Next, in Section 5, we provide a specialized query rewriting algorithm, illustrating that dynCQE query processing of BUCQs is *first-order rewritable* when ontologies are expressed in DL-Lite$_{\mathcal{R}}$. In Section 6, we extend dynCQE to the case of UCQs that may contain free variables. We particularly focus on lexicographic preferences and prove that answering a sequence of UCQs can be reduced to dynCQE over the sequence of instantiations of the input queries. For the non-Boolean case, we also investigate in Section 7 and in Section 8 the data complexity of answering open queries under generic MC-CQE semantics, and analyze in Section 9 the same problem under lexicographic preferences. Finally, the paper concludes with Section 10 on related work and Section 11 providing final remarks that outline the significance of the MC-CQE framework and its potential for practical implementations.

This paper is an extended version of [14] and generalizes the results that appeared in its previous version in several directions. In particular, the entire treatment on open queries is a new contribution, being [14] only focused on dynamic CQE for Boolean queries.

2. Preliminaries

We assume that the reader is familiar with the basics of (function-free) first-order (FO) logic. To denote FO formulas we will use the Greek letters ϕ and ψ. For the sake of readability, an FO formula ϕ may sometimes be denoted as $\phi(\vec{x})$, where $\vec{x}$ is the sequence of the free variables occurring in ϕ. An FO theory $\mathfrak{T}$ is a finite set of first-order sentences (i.e., closed FO formulas), and we say that $\mathfrak{T}$ is consistent if it has at least one model (i.e., there exists an interpretation satisfying all the sentences in $\mathfrak{T}$), inconsistent otherwise. We will also use the terms *query* and *Boolean query* as synonyms of FO formula and closed FO formula, respectively.

For the technical treatment, we also resort to Description Logics (DLs), which are fragments of FO logic underpinning the OWL 2 standard [15]. We introduce here the notions needed in this work and refer the reader to [16,17] for further details. The languages of our interest are built from an alphabet Γ that consists of countably infinite and mutually disjoint sets of unary predicates Γ_C (a.k.a. *atomic concepts*), binary predicates Γ_R (a.k.a. *atomic roles*), constants Γ_I (a.k.a. *individual names*), and variables Γ_V. An atom is a formula of the form $A(t)$ or $P(t_1, t_2)$, where $A \in \Gamma_C$ is an atomic concept, $P \in \Gamma_R$ is an atomic role, and each term t, t_1 and t_2 is either a variable from Γ_V or a constant from Γ_I. A formula is *ground* if all its terms are constants. In particular, ground atoms are also called *facts*.

In this paper, a DL ontology $\mathcal{O}$ is an FO theory constituted by a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, where $\mathcal{A}$ contains the ground atoms of $\mathcal{O}$ describing the extensional knowledge of $\mathcal{O}$, whereas $\mathcal{T}$ describes the intensional knowledge of $\mathcal{O}$. A DL ontology $\mathcal{O}$ *entails* an FO sentence ϕ, denoted $\mathcal{O} \models \phi$, if ϕ is true in every model of $\mathcal{O}$. Hereinafter, we denote with $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ the set of ground atoms entailed by $\mathcal{T} \cup \mathcal{A}$. Also, we say that an FO sentence ϕ *evaluates to true in* an ABox $\mathcal{A}$ if the evaluation of ϕ in the least Herbrand model of $\mathcal{A}$ is true [18], otherwise ϕ *evaluates to false in* $\mathcal{A}$.

Given a sequence of variables $\vec{x} = \langle x_1, \dots, x_k \rangle$, a substitution for $\vec{x}$ is a total mapping from the variables in $\vec{x}$ to terms in $\Gamma_I \cup \Gamma_V$. The notation $\{x_1 \leftarrow t_1, \dots, x_k \leftarrow t_k\}$, where $x_i \neq x_j$ for every $i, j \in [1, k]$, refers to a substitution σ mapping each variable x_i to the term t_i, for $i \in [1, k]$. The substitution σ is called ground if all terms t_i are constants in Γ_I. Applying σ to an FO formula ϕ returns a formula $\sigma(\phi)$ which is obtained by replacing in ϕ each free occurrence of x_i with t_i, for $i \in [1, k]$. Sometimes, it will be convenient to use the empty substitution ϵ, i.e., the substitution that does not map any variable. Obviously, $\epsilon(\phi) = \phi$ for any FO formula ϕ.

Given an FO query $\phi(\vec{x})$, an *answer* to ϕ is a ground substitution σ for $\vec{x}$ and $\mathfrak{S}_\phi$ denotes the set of all the answers to ϕ. An answer $\sigma \in \mathfrak{S}_\phi$ is a *certain answer* with respect to an ontology $\mathcal{O}$ iff the range of σ is contained in the signature of $\mathcal{O}$ and $\mathcal{O} \models \sigma(\phi)$. We denote with $cert(\mathcal{O}, \phi)$ the set of certain answers to ϕ with respect to $\mathcal{O}$. Note that $cert(\mathcal{O}, \phi)$ is always finite and it applies to both open and closed formulas. In particular, for a sentence ϕ, we have that $cert(\mathcal{O}, \phi) = \emptyset$ if $\mathcal{O} \not\models \phi$, and that $cert(\mathcal{O}, \phi) = \{\epsilon\}$ (i.e., the only certain answer is the empty substitution) otherwise.

While our definitions in the framework apply to every DL ontology, our complexity results focus on ontologies expressed in DL-Lite$_{\mathcal{R}}$ [19], which is the logical counterpart of OWL 2 QL [13]. In this DL, a role R is an atomic role P or its inverse P^-, whereas a (basic) concept B takes the form A, $\exists P$, or $\exists P^-$, where A is an atomic concept, whereas $\exists P$ and $\exists P^-$ denote the domain and the range of a role P, respectively. A DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ is a set of *positive inclusions* of the form $B_1 \sqsubseteq B_2$ or $R_1 \sqsubseteq R_2$, and *negative inclusions* of the form $B_1 \sqsubseteq \neg B_2$ or $R_1 \sqsubseteq \neg R_2$. Each positive and negative inclusion of a DL-Lite$_{\mathcal{R}}$ TBox can be equivalently written using the FO syntax, as shown by Table 1.

We also concentrate on specific classes of FO formulas as query language. A *conjunctive query* (CQ) is an FO query of the form $q(\vec{x}) = \exists \vec{y}(conj(\vec{x}, \vec{y}))$, where $conj(\vec{x}, \vec{y})$ is a conjunction $\alpha_1 \wedge \ldots \wedge \alpha_n$ of atoms where $\vec{x}$ and $\vec{y}$ indicate all the variables occurring in it. A *union of conjunctive queries* (UCQ) is a disjunction $q_1(\vec{x}) \vee \ldots \vee q_n(\vec{x})$ of CQs. Closed CQs are called *Boolean conjunctive queries* (BCQs) and closed UCQs are called Boolean unions of conjunctive queries (BUCQs). CQs with no existentially quantified variables are called *full CQs* and, similarly, UCQs with no existentially quantified variables are called *full UCQs*. Sometimes we write $q' \in q$ to indicate that the CQ q' is one of the CQs of the UCQ q. Note that a ground atom can be seen as a BCQ with no variables, and that a BCQ is a BUCQ with only one disjunct.

Given a BUCQ q and an ABox $\mathcal{A}$, 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 BUCQ q, a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, an *image of q in $\mathcal{A}$ with respect to $\mathcal{T}$* is a minimal subset $\mathcal{A}'$ of $\mathcal{A}$ such that $\mathcal{T} \cup \mathcal{A}' \models q$.

Given a CQ q, we denote by $Atoms(q)$ the set of atoms appearing in q, and call the cardinality of $Atoms(q)$ the *length* of q. Given two UCQs $q_1 = q_1^1 \vee \ldots \vee q_1^n$ and $q_2 = q_2^1 \vee \ldots \vee q_2^m$, we denote by $q_1 \wedge q_2$ the UCQ

$$\begin{array}{c}(q_1^1 \wedge q_2^1) \vee \ldots \vee (q_1^1 \wedge q_2^m) \vee \\ \vdots \\ (q_1^n \wedge q_2^1) \vee \ldots \vee (q_1^n \wedge q_2^m)\,.\end{array}$$

Table 1

For both $i = 1$ and $i = 2$, if R_i is an atomic role P (i.e., $R_i = P$), then $R_i(x_1, x_2) = P(x_1, x_2)$; otherwise (i.e., $R_i = P^-$ is the inverse of an atomic role), then $R_i(x_1, x_2) = P(x_2, x_1)$. Furthermore, for both $i = 1$ and $i = 2$, if B_i is an atomic concept A (i.e., $B_i = A$), then $B_i(x) = A(x)$; if B_i is the domain of an atomic role P (i.e., $B_i = \exists P$), then $B_i(x) = \exists y\, P(x, y)$; if B_i is the range of an atomic role P (i.e., $B_i = \exists P^-$), then $B_i(x) = \exists y\, P(y, x)$.

DL-Lite$_\mathcal{R}$ inclusion assertion	FO translation
$B_1 \sqsubseteq \neg B_2$	$\forall x\, (B_1(x) \rightarrow B_2(x))$
$B_1 \sqsubseteq \neg B_2$	$\forall x\, (B_1(x) \rightarrow \neg B_2(x))$
$R_1 \sqsubseteq R_2$	$\forall x_1, x_2\, (R_1(x_1, x_2) \rightarrow R_2(x_1, x_2))$
$R_1 \sqsubseteq \neg R_2$	$\forall x_1, x_2\, (R_1(x_1, x_2) \rightarrow \neg R_2(x_1, x_2))$

Table 2

Complexity classes used in the paper.

Δ_2^p	The class of decision problems decidable in polynomial time by a deterministic Turing machine using an oracle for a problem in NP.
$\Delta_2^p[O(\log n)]$	The class of decision problems decidable by a deterministic Turing machine in polynomial time and using only a logarithmic (in the size of the input) number of queries to an oracle for a problem in NP.
Σ_2^p	The class of decision problems decidable in polynomial time by a non-deterministic Turing machine using an oracle for a problem in NP.
AC^0	The class of decision problems decidable by a uniform family of circuits of constant depth and polynomial size, with unlimited fan-in AND gates and OR gates.

We recall that computing the set of certain answers to a UCQ in DL-Lite$_\mathcal{R}$ is FO rewritable, i.e., for every DL-Lite$_\mathcal{R}$ TBox $\mathcal{T}$ and UCQ 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 $cert(\mathcal{T} \cup \mathcal{A}, q) = \{\sigma \mid \sigma(q_r)$ evaluates to true in $\mathcal{A}\}$, for each ABox $\mathcal{A}$ such that $\mathcal{T} \cup \mathcal{A}$ is consistent. Intuitively, FO rewritability means that, for each ABox $\mathcal{A}$, the certain answers to q with respect to $\mathcal{T} \cup \mathcal{A}$ can be obtained by evaluating q_r over $\mathcal{A}$ seen as a database.

Let us now formalize the form of policy considered in this paper. A *policy* $\mathcal{P}$ is a (finite) set of *denials*, i.e., sentences of the form $q \rightarrow \bot$, where q is a BCQ. An interpretation *satisfies* a denial $q \rightarrow \bot$ if it does not satisfy the BCQ q. We denote by $q(\mathcal{P})$ the BUCQ $\bigvee_{q \rightarrow \bot \in \mathcal{P}} q$. An interpretation satisfies $\mathcal{P}$ if it does not satisfy $q(\mathcal{P})$.

As anticipated in the introduction, our complexity results refer to data complexity, i.e., the complexity computed with respect to the size of the ABox only [20]. We expect the reader to be familiar with the basic complexity classes PTIME, NP, and coNP [21]. Additionally, we will refer to the complexity classes described in Table 2. Finally, we recall that the problems that are shown to be FO rewritable fall into the complexity class AC^0, with respect to data complexity.

3. Controlled query evaluation framework

3.1. MC-CQE semantics

We now introduce our formal framework for CQE. All definitions and properties given in this section apply to any DL language.

A *CQE specification* is a pair $\langle \mathcal{T}, \mathcal{P} \rangle$, where $\mathcal{T}$ is a TBox and $\mathcal{P}$ is a policy, such that $\mathcal{T} \cup \mathcal{P}$ is consistent. A *CQE instance* is a triple $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$, where $\langle \mathcal{T}, \mathcal{P} \rangle$ is a CQE specification, and $\mathcal{A}$ is an ABox such that $\mathcal{T} \cup \mathcal{A}$ is consistent. The intended meaning of a CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$ is that, for any possible CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$, every BCQ q occurring in a denial contained in $\mathcal{P}$ must always result as non-entailed by $\mathcal{T} \cup \mathcal{A}$, even though it might actually be. Given a DL language $\mathcal{L}$, an $\mathcal{L}$ CQE specification is a CQE specification whose TBox is expressed in $\mathcal{L}$. The same notation extends to CQE instances.

Specifically, we assume to have a single user asking a sequence of queries to the CQE system.[1] Such queries are evaluated one after the other, and each such evaluation provides the user with new information. The semantics for query answering adopted by a CQE system must ensure that, even by collecting such information, it is impossible for the user to discover data protected by the policy (i.e., to infer a query mentioned in the policy). The distinguishing feature of our framework is that the query answering semantics that we propose takes specifically into account which information has been acquired by the user thanks to the queries she asked previously. In other terms, unlike other proposals, we do not assume that a user asking a query may have asked in the past any query and in any order, but take a trace of what she really asked. As we will see in formal terms, this choice allows our framework to increase the data that can be communicated to a user, provided that secrets are not divulged.

To this aim, we introduce the notion of protection state, which captures the history of queries asked by the user over a CQE instance.

[1] Multiple users may be assimilated into one single user asking a sequence of queries that incorporates the queries asked by each user.

Definition 1 *(State).* Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance. A *protection state of* $\mathcal{E}$ (or simply state of $\mathcal{E}$) is a pair $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, where $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ (with $n \geq 1$) is a sequence of queries.

We are now ready to provide the notion of CQE semantics, which represents a minimal requirement for safeguarding private information.

Definition 2 *(CQE semantics).* A *CQE semantics* is a function **cqe** that associates to each protection state $\mathcal{S} = \langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \langle q_1, \dots, q_n \rangle \rangle$ a sequence of answer sets $\mathbf{cqe}(\mathcal{S}) = \langle R_1, \dots, R_n \rangle$ such that:

(*i*) $R_i \subseteq \mathit{cert}(\mathcal{T} \cup \mathcal{A}, q_i)$, for each $i \in [1, n]$, and
(*ii*) $\mathcal{T} \cup \mathcal{P} \cup \bigcup_{i=1}^{n} \{\sigma(q_i) \mid \sigma \in R_i\}$ is consistent.

For each $i \in [1, n]$, we shall denote R_i, with $\mathbf{cqe}(\mathcal{S}, q_i)$.

Intuitively, a CQE semantics provides an answer to a query q_i in $\mathcal{Q}$ only if that answer is a certain answer to q_i with respect to the ontology $\mathcal{T} \cup \mathcal{A}$, i.e., without considering the protection policy (Condition (*i*)). Moreover, the overall set of answers provided by the system to the queries in $\mathcal{Q}$ must not violate, along with the TBox, the policy (Condition (*ii*)).

As anticipated in the introduction, we adopt the following attack model: *we assume that a potential attacker is aware of the CQE specification* $\langle \mathcal{T}, \mathcal{P} \rangle$ *as well as the actual CQE semantics* **cqe** *which is employed to conceal sensitive information. The only piece of information unknown to the attacker is the ABox* $\mathcal{A}$.

As the following example shows, not all CQE semantics can protect private data from malicious users effectively.

Example 1. Consider a CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$, where the TBox $\mathcal{T}$ is empty, i.e., $\mathcal{T} = \emptyset$, and the policy $\mathcal{P}$ is intended to hide the secret $C(o)$, i.e., $\mathcal{P} = \{C(o) \rightarrow \bot\}$. Moreover, consider the sequence $\mathcal{Q} = \langle q_1, q_2 \rangle$ of queries, where $q_1 = \exists x\, C(x)$ and $q_2 = C(x)$. Consider now the CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$, where $\mathcal{A} = \{C(o)\}$, and a CQE semantics **cqe** based on the principle of maximizing the answers that can be provided to the user as long as they satisfy Definition 2. Based on this principle, since $\mathcal{T} \cup \mathcal{P} \cup \{\exists x\, C(x)\}$ is consistent as required by Definition 2, **cqe** returns $\{\epsilon\}$ (i.e., it answers true) to the query q_1. From this answer, a user knows that $\exists x\, C(x)$ holds. Conversely, the only certain answer in $\mathit{cert}(\mathcal{T} \cup \mathcal{A}, q_2)$ is $\{x \leftarrow o\}$, which, however, reveals the secret $C(o)$, hence **cqe** returns $\emptyset$ to the query q_2.

From the assumptions that we already mentioned in the introduction, an attacker has perfect knowledge of both the CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$ and the rules governing the behavior of the underlying CQE semantics. Thus, knowing that the TBox is empty and that the **cqe** lies only when forced in order to keep a secret, the attacker can tell, on the basis of the answer to query q_2, that $C(o)$ is true. The secret is thus revealed.

Let us now consider a slightly different case in which we have the new CQE instance $\mathcal{E}' = \langle \mathcal{T}, \mathcal{P}, \mathcal{A}' \rangle$, where $\mathcal{T}$ and $\mathcal{P}$ are as above, while $\mathcal{A}'$ is the new ABox $\mathcal{A}' = \{C(o), C(o')\}$. Moreover, let us consider again the same CQE semantics **cqe** and sequence $\mathcal{Q}$ of queries. Applying **cqe** on the state $\mathcal{S}' = \langle \mathcal{E}', \mathcal{Q} \rangle$ leads us to the following answers:

$$\mathbf{cqe}(\mathcal{S}', q_1) = \{\epsilon\} \subseteq \mathit{cert}(\mathcal{T} \cup \mathcal{A}', q_1);$$

$$\mathbf{cqe}(\mathcal{S}', q_2) = \{\{x \leftarrow o'\}\} \subseteq \mathit{cert}(\mathcal{T} \cup \mathcal{A}', q_2).$$

In this case, since the answer to query q_1 is justified by the answer to query q_2, an attacker cannot tell whether the CQE semantics is hiding information. In fact, in the eyes of the attacker, the ABox $\mathcal{A}'$, which contains the secret $C(o)$, is indistinguishable from the ABox $\mathcal{A}'' = \{C(o')\}$, which does not contain any secret. □

Roughly speaking, the example above shows that in those cases where a CQE semantics succeeds in confusing an attacker by behaving as if it is acting on data that does not contain secrets, it turns out to be more secure. This property goes by the name of *indistinguishability* [22]. Clearly, in order for a CQE semantics to fully satisfy the indistinguishability property, it must be able to perform the simulation for every possible ABox underhand. Therefore, returning to the example, it is clear that the semantics **cqe** does not enjoy the property of indistinguishability since for the case of the ABox $\mathcal{A}$ there exists no secret-free ABox on which **cqe** would behave in the same way.

Towards a characterization of a CQE semantics that does not suffer from the problem exhibited by the previous example, we introduce the notion of censor as defined in [10, Definition 1]. Intuitively, a censor specifies which consequences of an ontology can be disclosed without violating the policy.

Definition 3 *(Censor).* Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, a *censor* for $\mathcal{E}$ is a subset C of $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ such that $\mathcal{T} \cup \mathcal{P} \cup C$ is consistent.

Example 2. Consider the following CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$:

$$\mathcal{T} = \{\mathsf{minor} \sqsubseteq \mathsf{person},\ \mathsf{hasCR} \sqsubseteq \mathsf{person}\};$$

$$\mathcal{P} = \{\exists x\, (\mathsf{minor}(x) \wedge \mathsf{hasCR}(x)) \rightarrow \bot\};$$

$\mathcal{A} = \{\mathsf{minor}(bob), \mathsf{hasCR}(bob)\}$.

In words, the TBox $\mathcal{T}$ states that both those who are minors and those who have criminal records are persons, while the policy $\mathcal{P}$ is intended to hide the existence of minors who have criminal records. Moreover, $\mathcal{A}$ states that *bob* is a minor having criminal records.

The following are censors for $\mathcal{E}$ according to Definition 3:

$C_1 = \{\mathsf{minor}(bob), \mathsf{person}(bob)\}$;

$C_2 = \{\mathsf{hasCR}(bob), \mathsf{person}(bob)\}$;

$C_3 = \{\mathsf{person}(bob)\}$;

$C_4 = \emptyset$. □

As demonstrated in the preceding example, a CQE instance $\mathcal{E}$ may give rise to multiple censors. Hereafter, *Cens*$(\mathcal{E})$ denotes the set of all censors for $\mathcal{E}$. We observe that the empty set is a censor for any CQE instance $\mathcal{E}$, and thus we always have that *Cens*$(\mathcal{E}) \neq \emptyset$.

Censors for a given CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ enjoy some notable properties: (*i*) they are ABoxes that do not contain any secrets according to the policy $\mathcal{P}$, and (*ii*) they exclusively contain information that is entailed by $\mathcal{T} \cup \mathcal{A}$. Therefore, as long as a CQE semantics mimics the behavior of a censor C (making C and $\mathcal{A}$ indistinguishable) we can be assured of obtaining correct answers while simultaneously thwarting attempts by malicious users to deduce a secret. Building upon these observations, we use the concept of censors as the foundation for formulating our notion of secure CQE semantics, rooted in the indistinguishability property.

Definition 4 *(Secure CQE semantics).* A CQE semantics **cqe** is *secure* if for each protection state $\mathcal{S} = \langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \mathcal{Q} \rangle$, with $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$, there exists a censor $C \in Cens(\mathcal{E})$ such that $\mathbf{cqe}(\mathcal{S}, q_i) = \mathbf{cqe}(\langle \langle \mathcal{T}, \mathcal{P}, C \rangle, \mathcal{Q} \rangle, q_i)$, for all $i \in [1, n]$.

It is worth noting that a CQE semantics that consistently yields no answers (i.e., $\mathbf{cqe}(\mathcal{S}, q_i) = \emptyset$ for any possible state $\mathcal{S} = \langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \mathcal{Q} \rangle$) trivially satisfies Definition 4. To ensure the practicality of CQE semantics, it is then necessary to establish a notion of *cooperativeness*. In practical applications, queries are typically submitted sequentially, and the system is unaware of which queries, if any, will follow at each step. In this scenario, it is reasonable to adopt the so-called *longest honeymoon approach*: "one must keep telling the whole truth as long as it is possible".

The following notion of maximally cooperative censor formally captures our idea of longest honeymoon.

Definition 5 *(Maximally cooperative censor).* Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance and let $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ be a sequence of queries over $\mathcal{E}$, with $n \geq 1$. If C_1, C_2 are two censors for $\mathcal{E}$, we shall say that C_2 is *more cooperative* than C_1 with respect to $\mathcal{Q}$ if there exists a natural number $m \in [0, n-1]$ such that:

(*i*) $cert(\mathcal{T} \cup C_1, q_i) = cert(\mathcal{T} \cup C_2, q_i)$ for each $i \in [1, m]$;
(*ii*) $cert(\mathcal{T} \cup C_1, q_{m+1}) \subset cert(\mathcal{T} \cup C_2, q_{m+1})$.

Finally, we shall say that a censor C for $\mathcal{E}$ is *maximally cooperative* with respect to $\mathcal{Q}$ if there is no other censor for $\mathcal{E}$ which is more cooperative than C with respect to $\mathcal{Q}$.

Example 3. Consider the CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ and the censors C_1 and C_2 for $\mathcal{E}$ of Example 2. Then, consider the sequence $\mathcal{Q} = \langle q_1, q_2, q_3 \rangle$ of queries, where:

$q_1 = \mathsf{person}(x)$;

$q_2 = \mathsf{minor}(bob)$;

$q_3 = \mathsf{hasCR}(bob)$.

We have that both $cert(\mathcal{T} \cup C_1, q_1)$ and $cert(\mathcal{T} \cup C_2, q_1)$ are equal to $\{\{x \leftarrow bob\}\}$. Moreover, $cert(\mathcal{T} \cup C_1, q_2)$ is equal to $\{\epsilon\}$, i.e., true, whereas $cert(\mathcal{T} \cup C_1, q_3) = \emptyset$, i.e., false. Conversely, $cert(\mathcal{T} \cup C_2, q_2)$ returns false and $cert(\mathcal{T} \cup C_2, q_3)$ returns true.

Then, according to Definition 5, C_1 is *maximally cooperative* with respect to $\mathcal{Q}$ while C_2 is not. However, it is straightforward to see that, if we consider $\mathcal{Q}' = \langle q_1, q_3, q_2 \rangle$, the situation is just the opposite. □

We are now ready to provide our definition of maximally cooperative CQE semantics.

Definition 6 *(MC-CQE semantics).* A CQE semantics **cqe** is *maximally cooperative (MC)* if, for each protection state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, with $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ and $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$, there exists a maximally cooperative censor C for $\mathcal{E}$ with respect to $\mathcal{Q}$ such that $\mathbf{cqe}(\mathcal{S}, q_i) = cert(\mathcal{T} \cup C, q_i)$, for all $i \in [1, n]$.

Clearly, if a censor C is maximally cooperative for $\mathcal{E}$ with respect to a sequence of queries $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$, then, for each censor C' that contains C and $i \in [1, n]$, we have by monotonicity that $cert(\mathcal{T} \cup C, q_i) = cert(\mathcal{T} \cup C', q_i)$. Consequently, we will assume without loss of generality that every censor C in Definition 6 is *optimal*, i.e., maximal w.r.t. set inclusion. Hereafter, we denote with *OptCens*($\mathcal{E}$) the set of optimal censors, given a CQE instance $\mathcal{E}$.

Definition 6 would be ineffective without the capacity to safeguard secrets; however, it is noteworthy that MC-CQE semantics do, in fact, ensure security, as defined in Definition 4. Preliminarily, we show that if an ABox is harmless (i.e., $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$ is consistent), then no MC-CQE semantics unnecessarily conceals a certain answer.

Proposition 1. *Let* ***cqe*** *be an MC-CQE semantics. Then, for each CQE instance* $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ *such that* $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$ *is consistent and each sequence of queries* $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$, $\mathbf{cqe}(\langle \mathcal{E}, \mathcal{Q} \rangle, q_i) = \mathit{cert}(\mathcal{T} \cup \mathcal{A}, q_i)$, *for all* $i \in [1, n]$.

Proof. By Definition 6 we know that there exists a maximally cooperative censor C w.r.t. $\mathcal{Q}$ such that $\mathbf{cqe}(\langle \mathcal{E}, \mathcal{Q} \rangle, q_i) = \mathit{cert}(\mathcal{T} \cup C, q_i)$, for all $i \in [1, n]$. Then the thesis directly derives from the fact that $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ is the only optimal censor and, by monotonicity, $\mathit{cert}(\mathcal{T} \cup \mathsf{cl}_{\mathcal{T}}(\mathcal{A}), q_i) = \mathit{cert}(\mathcal{T} \cup C, q_i)$. Consequently, it follows that $\mathbf{cqe}(\langle \mathcal{E}, \mathcal{Q} \rangle, q_i) = \mathit{cert}(\mathcal{T} \cup \mathsf{cl}_{\mathcal{T}}(\mathcal{A}), q_i) = \mathit{cert}(\mathcal{T} \cup \mathcal{A}, q_i)$. ■

Then, the fact that every MC-CQE semantics ensures security directly derives from the observation that, by Definition 3, $\mathcal{T} \cup \mathcal{P} \cup C$ is consistent for any censor C of $\mathcal{E}$.

Corollary 1. *Let* ***cqe*** *be an MC-CQE semantics, then* ***cqe*** *is a secure CQE semantics.*

Finally, we say that a CQE semantics **cqe** is *static* when it can be designed as follows: given a CQE instance $\mathcal{E}$, select an optimal censor C for $\mathcal{E}$. Subsequently, for any query sequence $\mathcal{Q}$, the semantics returns $\mathbf{cqe}(\mathcal{S}, q_i) = \mathit{cert}(\mathcal{T} \cup C, q_i)$ for all $i \in [1, n]$. While a static CQE semantics is inherently secure by construction, it may not be maximally cooperative in general.

Example 4. Consider two static CQE semantics $\mathbf{cqe}_{c1}$ and $\mathbf{cqe}_{c2}$ and assume that on the CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}\ \mathcal{A} \rangle$ of Example 2

$$\mathbf{cqe}_{c1}(\mathcal{S}, q_i) = \mathit{cert}(\mathcal{T} \cup C_1, q_i) \text{ for all } i \in [1, 3]$$

$$\mathbf{cqe}_{c2}(\mathcal{S}, q_i) = \mathit{cert}(\mathcal{T} \cup C_2, q_i) \text{ for all } i \in [1, 3]$$

Then, $\mathbf{cqe}_{c1}$ behaves as C_1 which we know from Example 3 to be maximally cooperative w.r.t. $\mathcal{Q}$ but not w.r.t. $\mathcal{Q}'$. Conversely, $\mathbf{cqe}_{c2}$ behaves as C_2 which is maximally cooperative w.r.t. $\mathcal{Q}'$ but it is not w.r.t. $\mathcal{Q}$. Therefore, according to Definition 6, neither $\mathbf{cqe}_{c1}$ nor $\mathbf{cqe}_{c2}$ are MC-CQE semantics. □

3.2. Decision problem associated with the framework

Given the general framework presented so far, we are now ready to define the recognition problem associated with the query answering problem under MC-CQE semantics. Formally, we define REC[$\mathcal{L}$, **cqe**, $\mathcal{L}'$] as the following decision problem, which is parametric with respect to a DL language $\mathcal{L}$, an MC-CQE semantics **cqe**, and a query language $\mathcal{L}'$:

Input: (*i*) A state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, where $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ is an $\mathcal{L}$ CQE instance and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ is a sequence of queries formulated in $\mathcal{L}'$, and (*ii*) $\sigma \in \mathfrak{G}_{q_n}$
Question: Does $\sigma \in \mathbf{cqe}(\mathcal{S}, q_n)$?

Obviously, if $\mathcal{L}'$ is a query language involving only Boolean queries, then the problem makes sense only when σ is set to ϵ.

As the ABox $\mathcal{A}$ of the CQE instance $\mathcal{E}$ is typically significantly larger than the other components, we are interested in the *data complexity* [20] version of the above problem, which is the complexity where only the ABox $\mathcal{A}$ is regarded as the input while all the other components are assumed to be fixed.

In the following sections, we analyze the data complexity of the above decision problem, focusing on the following query languages $\mathcal{L}'$: **BUCQ**, **FullCQ**, **CQ**, **FullUCQ**, and **UCQ**. Furthermore, we point out that all the upper bound results we will provide in this paper are with respect to the DL language $\mathcal{L}$ = DL-Lite$_{\mathcal{R}}$, while all the lower bound results are shown already for empty TBoxes. Thus, to simplify the presentation, from now we simply write REC[**cqe**, $\mathcal{L}'$] to implicitly refer to REC[DL-Lite$_{\mathcal{R}}$, **cqe**, $\mathcal{L}'$].

3.3. Comparison with other CQE semantics

We conclude this section by providing a comparison of MC-CQE semantics with other relevant CQE semantics from the literature. As in Example 4, a first strategy may consist in arbitrarily choosing an optimal censor among the ones for a CQE instance $\mathcal{E}$ [23,9,24]. However, as the following proposition shows, this way may lead to going against the longest-honeymoon approach.

Proposition 2. *Let* $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ *be a CQE instance such that* $|\mathit{OptCens}(\mathcal{E})| > 1$ *and let* $C \in \mathit{OptCens}(\mathcal{E})$. *Then, there exists a sequence* $\mathcal{Q}$ *for which* C *is not maximally cooperative with respect to* $\mathcal{Q}$.

Proof. By assumption, $\mathit{OptCens}(\mathcal{E})$ contains at least two optimal censors. Then, other than C, there must exist another optimal censor C' for $\mathcal{E}$. By definition, it follows that there is at least one ground atom $\alpha \in C'$ that is not contained in C. Now, let $\mathcal{Q} = \langle q \rangle$, where $q = \alpha$. Since $\mathit{cert}(\mathcal{T} \cup C, q) \subset \mathit{cert}(\mathcal{T} \cup C', q)$, then C' is more cooperative than C w.r.t. $\mathcal{Q}$, and therefore C is not maximally cooperative w.r.t. $\mathcal{Q}$. ■

Other two notable CQE semantics proposed in the literature are:

- *skeptical reasoning* [9,7], where the set of certain answers to a query q with respect to a CQE instance $\mathcal{E}$, denoted by $\mathit{cert}_{\mathsf{SK}}(\mathcal{E}, q)$, is $\bigcap_{C \in \mathit{OptCens}(\mathcal{E})} \mathit{cert}(\mathcal{T} \cup C, q)$;
- its approximation, called IGA semantics [12], under which the set of certain answers to a query q with respect to a CQE instance $\mathcal{E}$, denoted by $\mathit{cert}_{\mathsf{IGA}}(\mathcal{E}, q)$, is $\mathit{cert}(\mathcal{T} \cup C_{\mathsf{IGA}}, q)$, where $C_{\mathsf{IGA}} = \bigcap_{C \in \mathit{OptCens}(\mathcal{E})} C$.

The following example shows that skeptical reasoning is not a *secure* CQE semantics (see Definition 4).

Example 5. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance where $\mathcal{T} = \emptyset$ whereas $\mathcal{P}$ and $\mathcal{A}$ are defined as follows:

$$\mathcal{P} = \{\exists x\,(A(x) \wedge B(x)) \rightarrow \bot\};$$

$$\mathcal{A} = \{A(a), B(a)\}.$$

The optimal censors are $C_1 = \{A(a)\}$ and $C_2 = \{B(a)\}$. Then, consider the sequence $\mathcal{Q} = \langle q_1, q_2, q_3 \rangle$ of queries, where:

$$q_1 = A(a) \vee B(a);$$

$$q_2 = A(a);$$

$$q_3 = B(a).$$

We have that:

- The answer to the query q_1 is true:

$$\mathit{cert}_{\mathsf{SK}}(\mathcal{E}, q_1) = \mathit{cert}(\mathcal{T} \cup C_1, q_1) \cap \mathit{cert}(\mathcal{T} \cup C_2, q_1) = \{\epsilon\} \cap \{\epsilon\} = \{\epsilon\}.$$

  Therefore, the user knows that all censors satisfy $A(a) \vee B(a)$.
- The answer to the query q_2 is false:

$$\mathit{cert}_{\mathsf{SK}}(\mathcal{E}, q_2) = \mathit{cert}(\mathcal{T} \cup C_1, q_2) \cap \mathit{cert}(\mathcal{T} \cup C_2, q_2) = \{\epsilon\} \cap \emptyset = \emptyset.$$

  Consequently, the user knows that there is at least one censor in which $A(a)$ does not hold, we call it Clue_1.
- The answer to the query q_3 is also false:

$$\mathit{cert}_{\mathsf{SK}}(\mathcal{E}, q_3) = \mathit{cert}(\mathcal{T} \cup C_1, q_3) \cap \mathit{cert}(\mathcal{T} \cup C_2, q_3) = \emptyset \cap \{\epsilon\} = \emptyset.$$

  Therefore, the user knows that there is at least one censor in which $B(a)$ does not hold, we call it Clue_2.

So, thanks to the answer to q_1 we know that in all censors $A(a) \vee B(a)$ must hold. In Clue_1, there exists a censor C where $A(a)$ does not hold, but $A(a) \vee B(a)$ must hold. Therefore, $B(a)$ holds in C. Vice versa, according to Clue_2, there exists a censor C' such that $B(a)$ does not hold, but $A(a) \vee B(a)$ must hold. Therefore, $A(a)$ holds in C'. Since censors contain only facts that are entailed by $\mathcal{T} \cup \mathcal{A}$, the secret $\exists x\,(A(x) \wedge B(x))$ is unveiled.

Observe that, according to Definition 6, C_1 is the only maximally cooperative censor for $\mathcal{E}$. Therefore, under MC-CQE semantics, the first two queries evaluate to true, whereas the third query evaluates to false. □

The following proposition shows that both above CQE semantics turn out to be always a sound approximation of any MC-CQE semantics.

Proposition 3. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, let $\mathcal{Q}$ be a non-empty sequence of queries, let $q \in \mathcal{Q}$, and let* **cqe** *be an MC-CQE semantics. Then, $\mathit{cert}_{\mathsf{IGA}}(\mathcal{E}, q) \subseteq \mathit{cert}_{\mathsf{SK}}(\mathcal{E}, q) \subseteq \textbf{cqe}(\langle \mathcal{E}, \mathcal{Q} \rangle, q)$. The inclusions in the other direction do not necessarily hold.*

Proof. Suppose that $\sigma \in \mathit{cert}_{\mathsf{IGA}}(\mathcal{E}, q)$, i.e., $\mathcal{T} \cup C_{\mathsf{IGA}} \models \sigma(q)$. By [12, Proposition 1], we already know that $\mathcal{T} \cup C \models \sigma(q)$ for every $C \in \mathit{OptCens}(\mathcal{E})$, which means that $\sigma \in \mathit{cert}_{\mathsf{SK}}(\mathcal{E}, q)$. Now, let C the censor for $\mathcal{E}$ that is maximally cooperative with respect to $\mathcal{Q}$, i.e., such that $\mathit{cert}(\mathcal{T} \cup C, q_i) = \textbf{cqe}(\mathcal{S}, q_i)$ for all $i \in [1, n]$ (cf. Definition 6). Since $C \in \mathit{OptCens}(\mathcal{E})$, then we trivially have that $\mathit{cert}_{\mathsf{SK}}(\mathcal{E}, q) \subseteq \mathit{cert}(\mathcal{T} \cup C, q_i)$.

As for the converse, consider the CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ and the sequence $\mathcal{Q} = \langle q_1, q_2, q_3 \rangle$ of queries of Example 5. It is straightforward to verify that the censor $C_1 = \{A(a)\}$ is the only maximal cooperative censor for $\mathcal{E}$ with respect to $\mathcal{Q}$. So, an MC-CQE semantics **cqe** is such that $cert(\mathcal{T} \cup C_1, q_i) = \mathbf{cqe}(\mathcal{S}, q_i)$, for all $q_i \in \mathcal{Q}$. So, with $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, we have that: $\mathbf{cqe}(\mathcal{S}, q_1) = cert(\mathcal{T} \cup C_1, q_1) = cert_{\mathsf{SK}}(\mathcal{E}, q_1)$, but we know from Example 5 that $\mathbf{cqe}(\mathcal{S}, q_2) = cert(\mathcal{T} \cup C_1, q_2) = \{\epsilon\}$ whereas $cert_{\mathsf{SK}}(\mathcal{E}, q_2) = \emptyset$, from which we have the thesis.

Finally, note that $cert_{\mathsf{IGA}}(\mathcal{T} \cup C_1, q_1) = \emptyset \subset \{\epsilon\} = cert_{\mathsf{SK}}(\mathcal{T} \cup C_1, q_1)$, i.e., the answers under the IGA semantics are a strict subset of the ones under skeptical reasoning. ■

4. Dynamic CQE for Boolean queries

In this section, we show how the notion of MC-CQE semantics specializes in the case of Boolean queries.

Recall that, given an ontology $\mathcal{O}$ and a Boolean query q, then the entailment of q is expressed in terms of certain answers as follows:

$$cert(\mathcal{O}, q) = \begin{cases} \{\epsilon\} & \text{if } \mathcal{O} \models q \\ \emptyset & \text{otherwise,} \end{cases}$$

where ϵ is the empty substitution [18]. Consequently, the notion of cooperativeness in Definition 5 specializes to: given two censors C_1, C_2 for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$, C_2 is *more cooperative* than C_1 with respect to a sequence of Boolean queries $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ if there exists a natural number $0 \le m < n$ such that:

(*i*) $\mathcal{T} \cup C_1 \models q_i \iff \mathcal{T} \cup C_2 \models q_i$ for each $i \in [1, m]$;
(*ii*) $\mathcal{T} \cup C_2 \models q_{m+1}$ and $\mathcal{T} \cup C_1 \not\models q_{m+1}$.

Similarly, Definition 6 can be reformulated as follows: a function **cqe** is an MC-CQE semantics if, for each protection state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, where $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ is a sequence of Boolean queries, there exists a maximally cooperative censor C for $\mathcal{E}$ with respect to $\mathcal{Q}$ such that $\mathcal{T} \cup C \models q_i$ if and only if $\mathbf{cqe}(\mathcal{S}, q_i) = \{\epsilon\}$, for all $i \in [1, n]$.

We now provide an MC-CQE semantics, named dynCQE, which results in being unique, in the sense that any other MC-CQE semantics is equivalent to it in the case of Boolean queries.

First, we introduce for each Boolean query a preference relation over censors.

Definition 7. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance and q be a Boolean query, then the preference relation $\preceq_q$ is a weak order (i.e., a total and transitive relation) defined as follows: for each $C_1, C_2 \in Cens(\mathcal{E})$,

$$C_1 \preceq_q C_2 \text{ iff } \mathcal{T} \cup C_1 \models q \text{ implies } \mathcal{T} \cup C_2 \models q\,.$$

In particular, C_1 and C_2 are equivalent ($C_1 \sim_q C_2$) just in the case they return the same answer to the query q. Conversely, if $\mathcal{T} \cup C_1 \not\models q$ whereas $\mathcal{T} \cup C_2 \models q$, this means that $\mathcal{T} \cup \mathcal{A} \models q$ and C_2 can safely and correctly answer q. Therefore, C_2 is strictly more preferable than C_1.

Definition 7 identifies a preference relation $\preceq_q$ for each Boolean query q, we use such a family of preferences to dynamically choose maximally cooperative optimal censors given a sequence $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ of Boolean queries. To this end, $\mathcal{Q}_i = \langle q_1, \ldots, q_i \rangle$, with $i \le n$, denotes the subsequence of the first i queries. Analogously, given a CQE instance $\mathcal{E}$, $\mathcal{S}_i$ refers to the state $\langle \mathcal{E}, \mathcal{Q}_i \rangle$. Moreover, for the sake of readability, we also utilize in the following definition a fictitious state $\mathcal{S}_0 = \langle \mathcal{E}, \langle\rangle \rangle$ as base case of induction.

Definition 8. Let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ be a protection state, where $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ is a CQE instance and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ is a sequence of Boolean queries. The set $StCens(\mathcal{S}_i)$ of censors of the state $\mathcal{S}_i$, with $i \in [0, n]$, is inductively defined as follows:

- $StCens(\mathcal{S}_0) = OptCens(\mathcal{E})$;
- $StCens(\mathcal{S}_i) = \max_{\preceq_{q_i}} StCens(\mathcal{S}_{i-1})$.

Informally speaking, each set $StCens(\mathcal{S}_i)$ (with $i \in [1, n]$) in the above definition progressively selects the optimal censors for $\mathcal{E}$ that maximize the preference relation $\preceq_{q_i}$. Note that, being $\preceq_{q_i}$ a weak order, $StCens(\mathcal{S}_i)$ may include multiple censors. However, all these censors are by construction mutually equivalent, in the sense that they provide the same answer to the query q_i.

The following lemma shows that $StCens(\mathcal{S})$ in Definition 8 coincides with Definition 3 in [14] (the conference version of the present paper).

Lemma 1. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, and let $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \ge 1$) be a sequence of Boolean queries. Then, the set $StCens(\mathcal{S}_i)$ of censors of $\mathcal{S}_i$, with $i \in [0, n]$, in Definition 8 can be reformulated as follows:*

- $StCens(\mathcal{S}_0) = OptCens(\mathcal{E})$
- $StCens(\mathcal{S}_i) = \begin{cases} StCens(\mathcal{S}_{i-1}) & \textit{if } \{C \in StCens(\mathcal{S}_{i-1}) \mid \mathcal{T} \cup C \models q_i\} = \emptyset, \\ \{C \in StCens(\mathcal{S}_{i-1}) \mid \mathcal{T} \cup C \models q_i\} & \textit{otherwise.} \end{cases}$

Proof. The thesis directly derives from the fact that a censor C in $\mathit{StCens}(\mathcal{S}_{i-1})$ is maximal w.r.t. $\preceq_{q_i}$ iff $\mathcal{T} \cup C \models q_i$ or for all censors C' in $\mathit{StCens}(\mathcal{S}_{i-1})$, $\mathcal{T} \cup C' \not\models q_i$. ■

The next lemma states additional properties that will be subsequently used.

Lemma 2. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, and let $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 1$) be a sequence of Boolean queries. Then we have that, for all $i \in [1, n]$,*

- *$\mathit{StCens}(\mathcal{S}_{i-1}) \supseteq \mathit{StCens}(\mathcal{S}_i) \supset \emptyset$;*
- *for all $j \in [i, n]$, $C \in \mathit{StCens}(\mathcal{S}_i)$ and $C' \in \mathit{StCens}(\mathcal{S}_j)$, $\mathcal{T} \cup C \models q_i$ iff $\mathcal{T} \cup C' \models q_i$.*

Proof. The first property is a direct consequence of Definition 8.

Based on Lemma 1, we know that either all censors in $\mathit{StCens}(\mathcal{S}_i)$ satisfy q_i or none of them do. As a result, the second property holds when $i = j$, and it also holds when $i < j$ due to the inclusion $\mathit{StCens}(\mathcal{S}_j) \subseteq \mathit{StCens}(\mathcal{S}_i)$. ■

We are now ready to define our dynamic CQE semantics dynCQE as follows.

Definition 9 *(Dynamic CQE – dynCQE)*. Let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ be a protection state, where $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ is a CQE instance and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 1$) a sequence of Boolean queries. We define $\mathsf{dynCQE}(\mathcal{S}, q_i) = \mathit{cert}(\mathcal{T} \cup C, q_i)$, for every $C \in \mathit{StCens}(\mathcal{S})$. Then, $\mathit{EntQ}(\mathcal{S})$ denotes the set of queries q_i in $\mathcal{Q}$ such that $\mathsf{dynCQE}(\mathcal{S}, q_i) = \{\epsilon\}$.

Example 6. Certain pharmaceutical products can accurately identify the specific disease afflicting an individual. For example, medications containing phenytoin or those categorized as anti-seizure drugs are indicative of epilepsy.

Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, where:

$$\mathcal{T} = \{\mathsf{Abc} \sqsubseteq \mathsf{Antiseizure}\};$$

$$\mathcal{P} = \{\exists x, y\, (\mathsf{buy}(x, y) \land \mathsf{Antiseizure}(y)) \rightarrow \bot,$$

$$\exists x, y\, (\mathsf{buy}(x, y) \land \mathsf{contain}(y, \mathsf{phenytoin})) \rightarrow \bot\};$$

$$\mathcal{A} = \{\mathsf{buy}(\mathsf{john}, \mathsf{m}_a), \mathsf{Abc}(\mathsf{m}_a), \mathsf{buy}(\mathsf{alice}, \mathsf{m}_b), \mathsf{contain}(\mathsf{m}_b, \mathsf{phenytoin})\}.$$

In words, the TBox states that Abc is an anti-seizure medication, while the policy conceals the presence of patients suffering from epilepsy.

Let us start by considering an empty sequence of BCQs. By definition, we have that $\mathit{StCens}(\langle \mathcal{E}, \langle\rangle \rangle)$ coincides with the set of the optimal censors for $\mathcal{E}$ (i.e., its $\subseteq$-maximal censors). Note that – in this example – $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ is a finite set of ground atoms, therefore the optimal censors are finitely many; the reader may verify that $\mathit{OptCens}(\mathcal{E})$ consists of the following sets:

$$C_1 = \{\mathsf{buy}(\mathsf{john}, \mathsf{m}_a), \mathsf{buy}(\mathsf{alice}, \mathsf{m}_b)\};$$

$$C_2 = \{\mathsf{buy}(\mathsf{john}, \mathsf{m}_a), \mathsf{contain}(\mathsf{m}_b, \mathsf{phenytoin})\};$$

$$C_3 = \{\mathsf{Abc}(\mathsf{m}_a), \mathsf{Antiseizure}(\mathsf{m}_a), \mathsf{buy}(\mathsf{alice}, \mathsf{m}_b)\};$$

$$C_4 = \{\mathsf{Abc}(\mathsf{m}_a), \mathsf{Antiseizure}(\mathsf{m}_a), \mathsf{contain}(\mathsf{m}_b, \mathsf{phenytoin})\}.$$

Let $q_1 = \mathsf{buy}(\mathsf{john}, \mathsf{m}_a)$ be the first query. The censors C_1 and C_2 agree with answering $\{\epsilon\}$ to this query. All the censors that disagree with such an answer are then removed, obtaining $\mathit{StCens}(\langle \mathcal{E}, \langle q_1 \rangle \rangle) = \{C \in \mathit{StCens}(\langle \mathcal{E}, \langle\rangle \rangle) \mid \mathcal{T} \cup C \models q_1\} = \{C_1, C_2\}$. Then, let $q_2 = \mathsf{Abc}(\mathsf{m}_a)$ be a new query in the sequence. Since neither $\mathcal{T} \cup C_1$ nor $\mathcal{T} \cup C_2$ entail q_2, then $\mathit{StCens}(\langle \mathcal{E}, \langle q_1, q_2 \rangle \rangle) = \mathit{StCens}(\langle \mathcal{E}, \langle q_1 \rangle \rangle)$. Now, consider adding $q_3 = \exists x\, \mathsf{buy}(x, \mathsf{m}_b)$ to the sequence. Since $\mathcal{T} \cup C_1 \models q_3$ while $\mathcal{T} \cup C_2 \not\models q_3$, we have $\mathit{StCens}(\mathcal{S}) = \{C_1\}$, where $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ with $\mathcal{Q} = \langle q_1, q_2, q_3 \rangle$. Clearly, we have that $\mathsf{dynCQE}(\mathcal{S}, q_1) = \mathsf{dynCQE}(\mathcal{S}, q_3) = \{\epsilon\}$ while $\mathsf{dynCQE}(\mathcal{S}, q_2) = \emptyset$. □

We now show that the notion of secure and maximally cooperative semantics reduces to dynCQE when we restrict to Boolean queries. First, the following intermediate result shows that a state of a CQE instance cannot discriminate between two optimal censors if they have answered all the queries posed so far in the same way.

Lemma 3. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 1$) be a sequence of Boolean queries, and C and C' be two optimal censors for $\mathcal{E}$ such that $\mathcal{T} \cup C \models q_i \iff \mathcal{T} \cup C' \models q_i$, for all $i \in [1, n]$. Then, $C \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$ iff $C' \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$.*

Proof. The proof is by induction on the length of $\mathcal{Q}$.

Case $n = 1$. By construction $\mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle) = \max_{\preceq_{q_1}} \mathit{OptCens}(\mathcal{E})$. Then, since $C \sim_{q_1} C'$, by Definition 8 we have the thesis.

Case $n > 1$. By assumption, we have that $\mathcal{T} \cup C \models q_i$ iff $\mathcal{T} \cup C' \models q_i$, for all $i \in [1, n-1]$, and $\mathcal{T} \cup C \models q_n$ iff $\mathcal{T} \cup C' \models q_n$. Then, by the inductive hypothesis, $C \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q}' \rangle)$ iff $C' \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q}' \rangle)$, with $\mathcal{Q}' = \langle q_1, \ldots, q_{n-1} \rangle$. Moreover, since $C \sim_{q_n} C'$, by Definition 8 we have the thesis. ■

Secondly, we prove that for every CQE instance $\mathcal{E}$ and every sequence of Boolean queries $\mathcal{Q}$, the set $\mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$ contains all and only the optimal censors for $\mathcal{E}$ that are maximally cooperative with respect to $\mathcal{Q}$.

Lemma 4. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 1$) be a sequence of Boolean queries. An optimal censor C for $\mathcal{E}$ is maximally cooperative with respect to $\mathcal{Q}$ iff $C \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$.*

Proof. We start by showing that every $C \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$ is maximally cooperative with respect to $\mathcal{Q}$. Assume that, for some $C \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$, there exists a more cooperative censor C'. This means that, for some $m \in [0, n-1]$, (i) $\mathcal{T} \cup C \models q_i \iff \mathcal{T} \cup C' \models q_i$, for each $i \leq m$, and (ii) $\mathcal{T} \cup C \not\models q_{m+1}$ and $\mathcal{T} \cup C' \models q_{m+1}$, i.e., $C \prec_{q_{m+1}} C'$.

By assumption $C \in \mathit{StCens}(\mathcal{S}_n)$, which means by Lemma 2 that C occurs also in $\mathit{StCens}(\mathcal{S}_m)$ and $\mathit{StCens}(\mathcal{S}_{m+1})$. From $C \in \mathit{StCens}(\mathcal{S}_m)$, condition (i), and Lemma 3, we have that C' belongs to $\mathit{StCens}(\mathcal{S}_m)$ too. Then, from (ii) and Definition 8, C is not in $\mathit{StCens}(\mathcal{S}_{m+1})$, a contradiction.

Conversely, assume that an optimal censor C for $\mathcal{E}$ is maximally cooperative w.r.t. $\mathcal{Q}$, then we show that $C \in \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$. For the sake of contradiction, assume that $C \notin \mathit{StCens}(\langle \mathcal{E}, \mathcal{Q} \rangle)$. So, there exists in $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ a query q_i such that $C \in \mathit{StCens}(\mathcal{S}_{i-1}) \setminus \mathit{StCens}(\mathcal{S}_i)$, with $\mathcal{S}_i = \langle \mathcal{E}, \langle q_1, ..., q_i \rangle \rangle$. Hence, $\mathcal{T} \cup C \not\models q_i$ and there exists a censor $C' \in \mathit{StCens}(\mathcal{S}_i)$ such that $\mathcal{T} \cup C' \models q_i$ and $\mathcal{T} \cup C' \models q_j \iff \mathcal{T} \cup C \models q_j$ for every $j \in [1, i-1]$. This means that C' is more cooperative than C w.r.t. $\mathcal{Q}$, a contradiction. ■

Finally, the following proposition shows that, when $\mathcal{Q}$ is a sequence of Boolean queries, any MC-CQE semantics is equivalent to dynCQE.

Proposition 4. *Let **cqe** be an MC-CQE semantics, then for every protection state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, where $\mathcal{E}$ is a CQE instance and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ is a non-empty sequence of Boolean queries, we have that $\mathbf{cqe}(\mathcal{S}, q_i) = \{\epsilon\}$ iff $\mathsf{dynCQE}(\mathcal{S}, q_i) = \{\epsilon\}$ for every $i \in [1, n]$.*

Proof. The thesis derives from the following equivalences: $\mathbf{cqe}(\mathcal{S}, q_i) = \{\epsilon\}$ iff there exists a maximally cooperative censor C for $\mathcal{E}$ w.r.t. $\mathcal{Q}$ such that $\mathcal{T} \cup C \models q_i$ (by definition) iff $C \in \mathit{StCens}(\mathcal{S})$ (by Lemma 4). Then, since all optimal censors in $\mathit{StCens}(\mathcal{S})$ agree on q_i, by Definition 9, we have the thesis. ■

5. BUCQ evaluation under dynCQE semantics in DL-Lite$_\mathcal{R}$

In this section, we characterize the precise data complexity of the decision problem REC[dynCQE, **BUCQ**]. Observe that, due to Proposition 4, REC[dynCQE, **BUCQ**] is equivalent to the decision problem REC[**cqe**, **BUCQ**], for any MC-CQE semantics **cqe**.

In our discussion about the complexity of BUCQ entailment, we will make extensive use of the algorithm *PerfectRef* presented in [19], which takes as input a UCQ q and a DL-Lite$_\mathcal{R}$ TBox $\mathcal{T}$ and computes a UCQ q_r that is the perfect reformulation of q with respect to $\mathcal{T}$. This is stated in the following proposition from [19].

Proposition 5. *Let $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ be a consistent DL-Lite$_\mathcal{R}$ ontology, let q be a UCQ, and let σ be an answer to q, i.e., $\sigma \in \mathfrak{S}_q$. Then, $\sigma \in \mathit{cert}(\mathcal{O}, q)$ if and only if $\sigma(\mathit{PerfectRef}(q, \mathcal{T}))$ evaluates to true in $\mathcal{A}$.*

The next proposition follows from the definition of satisfaction of a denial and from Proposition 5. We recall that $q(\mathcal{P})$ is the BUCQ $\bigvee_{q \to \bot \in \mathcal{P}} q$, and we observe that $\mathit{PerfectRef}(q(\mathcal{P}), \mathcal{T})$ is a BUCQ as well.

Proposition 6. *Let $\mathcal{T} \cup \mathcal{A}$ be a consistent DL-Lite$_\mathcal{R}$ ontology and let $\mathcal{P}$ be a policy. Then, $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}$ is a consistent FO theory iff $\mathit{PerfectRef}(q(\mathcal{P}), \mathcal{T})$ evaluates to false in $\mathcal{A}$.*

We are now ready to examine the problem of BUCQ entailment under dynCQE. A first approach to face it might consist in finding a reduction to the stateless CQE approach, for which algorithms are already known. It turns out, however, that the behavior of dynCQE cannot be intensionally simulated by a stateless CQE approach, independent of query history.

We show that, given a DL-Lite$_\mathcal{R}$ CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$ and a sequence of BUCQs $\mathcal{Q}$, in general it is not possible to find a DL-Lite$_\mathcal{R}$ CQE specification $\langle \mathcal{T}', \mathcal{P}' \rangle$ that fully captures the semantics of the secure state $\langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \mathcal{Q} \rangle$ for every ABox $\mathcal{A}$ (i.e., independently from the data).

Proposition 7. *There exist a DL-Lite$_\mathcal{R}$CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$ and a BCQ q such that there exists no DL-Lite$_\mathcal{R}$ CQE specification $\langle \mathcal{T}', \mathcal{P}' \rangle$ such that, for every ABox $\mathcal{A}$, $\mathit{OptCens}(\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle) = \mathit{StCens}(\mathcal{S})$, where $\mathcal{S} = \langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \langle q \rangle \rangle$.*

Proof. Let $\mathcal{T} = \emptyset$, let $\mathcal{P} = \{C(x) \wedge D(x) \rightarrow \bot\}$, and let $q = \exists x\, C(x)$. Suppose that there exist a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}'$ and a policy $\mathcal{P}'$ such that, for every ABox $\mathcal{A}$, $\mathit{OptCens}(\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle) = \mathit{StCens}(\mathcal{S})$, where $\mathcal{S} = \langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \langle q \rangle \rangle$.

Now consider the ABox $\mathcal{A} = \{C(a_1), C(a_2), D(a_1), D(a_2)\}$, where a_1, a_2 are individual names that do not appear in $\mathcal{P}'$. The optimal censors for $\langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ are $C_1 = \{C(a_1), C(a_2)\}$, $C_2 = \{C(a_1), D(a_2)\}$, $C_3 = \{D(a_1), C(a_2)\}$, $C_4 = \{D(a_1), D(a_2)\}$. Among such optimal censors, only C_4 does not satisfy q. Therefore, $\mathit{StCens}(\mathcal{S}) = \{C_1, C_2, C_3\}$. Since by hypothesis $\mathit{StCens}(\mathcal{S}) = \mathit{OptCens}(\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle)$, it follows that $\mathcal{T}' \cup \mathcal{P}' \cup C_4$ is inconsistent and $\mathcal{T}' \cup \mathcal{P}' \cup C_3$ is consistent. Consequently, by Proposition 6, $\mathit{PerfectRef}(q(\mathcal{P}'), \mathcal{T}')$ evaluates to true in C_4 and evaluates to false in C_3.

On the other hand, it is immediate to see that, for every BUCQ q that does not mention individual names in $\mathcal{A}$, q evaluates to true in C_4 only if q evaluates to true in C_3. Consequently, since by construction $\mathcal{A}$ does not include any individual name occurring in $\mathcal{P}'$, $\mathit{PerfectRef}(q(\mathcal{P}'), \mathcal{T}')$ evaluates to true in C_4 only if $\mathit{PerfectRef}(q(\mathcal{P}'), \mathcal{T}')$ evaluates to true in C_3. A contradiction. ■

Moreover, we prove that, given a DL-Lite$_{\mathcal{R}}$ CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$ and a sequence of BUCQs $\mathcal{Q}$, in general it is not possible to find a DL-Lite$_{\mathcal{R}}$ CQE specification $\langle \mathcal{T}', \mathcal{P}' \rangle$ that, for every ABox $\mathcal{A}$, fully captures the dynCQE semantics over the secure state $\langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \mathcal{Q} \rangle$ by skeptically reasoning over the optimal censors for $\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle$.

Proposition 8. *There exist a DL-Lite$_{\mathcal{R}}$ CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$ and a BCQ q such that there exists no DL-Lite$_{\mathcal{R}}$ CQE specification $\langle \mathcal{T}', \mathcal{P}' \rangle$ such that, for every ABox $\mathcal{A}$ and for every BUCQ q', $\mathcal{T}' \cup C \models q'$ for every $C \in \mathit{OptCens}(\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle)$ iff $\mathsf{dynCQE}(\mathcal{S}, q') = \{\epsilon\}$, where $\mathcal{S} = \langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \langle q, q' \rangle \rangle$.*

Proof. We consider $\mathcal{T}$, $\mathcal{P}$, and q as in the proof of Proposition 7. Suppose there exist a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}'$ and a policy $\mathcal{P}'$ such that, for every ABox $\mathcal{A}$ and for every BUCQ q', $\mathcal{T} \cup C \models q'$ for every $C \in \mathit{OptCens}(\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle)$ iff $\mathcal{T} \cup C \models q'$ for every $C \in \mathit{StCens}(\mathcal{S})$. Now let $\mathcal{A}$, C_1, C_2, C_3, and C_4 be as in the proof of Proposition 7. First, observe that the ground BUCQ corresponding to the set of optimal censors $\mathit{StCens}(\mathcal{S})$ must be entailed by every $C \in \mathit{OptCens}(\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle)$, while no BUCQ more specific than this should be entailed. Consequently, there must exist a bijection between $\mathit{StCens}(\mathcal{S})$ and the projection on the predicates C and D of $\mathit{OptCens}(\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle)$. This implies that there must exist an optimal censor C_3' of $\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle$ whose projection on the predicates C and D coincides with C_3. Moreover, $\mathcal{T}' \cup \mathcal{P}' \cup C_4$ must be inconsistent (otherwise, there would be an optimal censor for $\langle \mathcal{T}', \mathcal{P}', \mathcal{A} \rangle$ containing C_4, contradicting the existence of the above bijection). But again, as explained in the proof of Proposition 7, if $\mathcal{T}' \cup \mathcal{P}' \cup C_4$ is inconsistent, then $\mathcal{T}' \cup \mathcal{P}' \cup C_3$ (and hence $\mathcal{T}' \cup \mathcal{P}' \cup C_3'$) is also inconsistent because $\mathcal{P}'$ does not mention the individual names occurring in $\mathcal{A}$. Consequently, the above bijection cannot exist, thus proving the thesis. ■

In the remainder of this section, we study the complexity of REC[dynCQE, **BUCQ**]. We start by showing a fundamental property of dynCQE query entailment, which holds for all DLs.

Proposition 9. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ be a sequence of BUCQs, and let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$. For every $i \in [1, n]$, we have that $q_i \in \mathit{EntQ}(\mathcal{S})$ iff there exists a censor C for $\mathcal{E}$ such that*

$$\mathcal{T} \cup C \models \Big(\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{i-1})} q \Big) \wedge q_i$$

where we set $\mathit{EntQ}(\mathcal{S}_0) = \emptyset$.

Proof. ($\Leftarrow$:) Suppose there exists a censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models (\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{i-1})} q) \wedge q_i$. Then it follows immediately that there exists an optimal censor C' for $\mathcal{E}$ such that $C' \supseteq C$, consequently $\mathcal{T} \cup C' \models (\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{i-1})} q) \wedge q_i$. Hence, by Lemma 1 we have that $C' \in \mathit{StCens}(\langle \mathcal{E}, \langle q_1, \ldots, q_i \rangle \rangle)$, which implies $q_i \in \mathit{EntQ}(\mathcal{S})$.

($\Rightarrow$:) Suppose $q_i \in \mathit{EntQ}(\mathcal{S})$. Now, let C' be an optimal censor for $\mathcal{E}$ such that $C' \in \mathit{StCens}(\mathcal{S})$. We have that $\mathcal{T} \cup C' \models q$ for every $q \in \mathit{EntQ}(\mathcal{S})$, and since $q_i \in \mathit{EntQ}(\mathcal{S})$ and $\mathit{EntQ}(\mathcal{S}_{i-1}) \subseteq \mathit{EntQ}(\mathcal{S})$, it follows that $\mathcal{T} \cup C' \models (\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{i-1})} q) \wedge q_i$, thus proving the thesis. ■

We are now ready to prove the next result.

Proposition 10. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ CQE instance and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 0$) be a sequence of BUCQs. For every $i \in [1, n]$, we have that $q_i \in \mathit{EntQ}(\mathcal{S})$ iff there exists an image I of $\mathit{PerfectRef}((\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{i-1})} q) \wedge q_i, \mathcal{T})$ in $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ such that $\mathit{PerfectRef}(q(\mathcal{P}), \mathcal{T})$ evaluates to false in I.*

Proof. First note that, from the definition of censor for a CQE instance and the definition of image of a BUCQ q it straightforwardly follows that there exists a censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models q$ iff there exists an image I of q in $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ with respect to $\mathcal{T}$ such that $\mathcal{T} \cup \mathcal{P} \cup I$ is consistent.

By Proposition 5, Proposition 6 and from what above, it follows that there exists a censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models q$ iff there exists an image I of $\mathit{PerfectRef}(q, \mathcal{T})$ in $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ such that $\mathit{PerfectRef}(q(\mathcal{P}), \mathcal{T})$ evaluates to false in I. This property, along with Proposition 9, immediately implies the thesis. ■

Now observe that: (i) $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ can be computed in PTIME in data complexity; (ii) every image of a BUCQ q has a size that is not larger than the length of the longest BCQ in q; (iii) such a maximum length is a constant in data complexity; (iv) all the conditions in the proposition can be verified in PTIME in data complexity [19]. This implies that REC[dynCQE, **BUCQ**] is in PTIME in data complexity.

In the following, we provide a tighter upper bound, showing that this entailment problem is in AC^0 in data complexity. We do so by proving that the problem is FO rewritable. In fact, given a DL-Lite$_{\mathcal{R}}$ CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$ and a sequence $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ of BUCQs, every query q_i in $\mathcal{Q}$ can be translated into an FO sentence q_i' such that, for all ABoxes $\mathcal{A}$, q_i' evaluates to true in $\mathcal{A}$ iff $q_i \in \mathit{EntQ}(\mathcal{S})$, where $\mathcal{S} = \langle \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle, \mathcal{Q} \rangle$.

To this purpose, we will find an FO query that depends on the intensional part of the state, i.e., the TBox, the policy, and the sequence of queries, and such that its evaluation on the ABox is true if and only if the condition expressed in Proposition 10 holds (Proposition 13). Towards this result, we will make the following two intermediate steps:

- first (Proposition 11), given a query q on a DL-Lite$_{\mathcal{R}}$ CQE specification $\langle \mathcal{T}, \mathcal{P} \rangle$, we will find a query whose evaluation on $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ corresponds to checking the existence of an optimal censor C for the CQE instance $\langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ such that $\mathcal{T} \cup C \models q$;
- then (Corollary 2), we will find an FO query such that its evaluation on $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ is true if and only if the condition expressed in Proposition 10 holds.

Given two BCQs q and q', a *mapping of q' into q* is a function $h : \mathit{Atoms}(q') \to \mathit{Atoms}(q)$ such that there exists a substitution σ_h such that (*i*) $\sigma_h(\alpha) = \sigma_h(h(\alpha))$ for every atom $\alpha \in \mathit{Atoms}(q')$; (*ii*) σ_h replaces variables occurring either in q' or in q with either variables of q or constants. Hereafter, we assume that σ_h is the most general substitution such that (*i*) and (*ii*) hold, and we denote by $\mathit{Map}(q', q)$ the set of all mappings of q' into q.

Furthermore, we denote by $\sigma_h[q]$ the restriction of σ_h that replaces only the variables of q. For example, if $q = \exists x, y, z\,(C(x) \land R(y,z))$ and $q' = \exists x'\,(C(x') \land R(x', a))$ (where a is a constant and all other terms are variables), then $\sigma_h = \{x' \leftarrow x, y \leftarrow x, z \leftarrow a\}$ and $\sigma_h[q] = \{y \leftarrow x, z \leftarrow a\}$.

Given two BCQs q and q', we denote by $\mathit{Unify}(q', q)$ the formula:

$$\bigvee_{h \in \mathit{Map}(q',q)} \Big(\bigwedge_{x \leftarrow t\, \in \sigma_h[q]} x = t \Big)$$

Definition 10 *(BraveRef)*. Given a BUCQ q, a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$ and a policy $\mathcal{P}$, we define $\mathit{BraveRef}(q, \mathcal{T}, \mathcal{P})$ as the FO sentence:

$$\bigvee_{q_r \in \mathit{PerfectRef}(q,\mathcal{T})} \exists \vec{x}_r \Big(\mathit{conj}_r(\vec{x}_r) \land \Big(\bigwedge_{q_d \in \mathit{PerfectRef}(q(\mathcal{P}),\mathcal{T})} \neg \mathit{Unify}(q_d, q_r) \Big) \Big)$$

(where we assume $q_r = \exists \vec{x}_r\,(\mathit{conj}_r(\vec{x}_r))$).

Example 7. Consider the DL-Lite$_{\mathcal{R}}$ CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ and the query sequence $\mathcal{Q} = \langle q_1, q_2, q_3 \rangle$ of Example 6. We show the output of *BraveRef* for the BCQs $q' = q_1 \land q_2$ and $q'' = q_1 \land q_3$, which will be useful later on.

We first point out that, for the given TBox $\mathcal{T}$, *PerfectRef* has no impact on q' or q'' (i.e., $\mathit{PerfectRef}(q', \mathcal{T}) = q'$ and $\mathit{PerfectRef}(q'', \mathcal{T}) = q''$), implying that their rewriting formulas will have only one disjunct. Moreover, we have that:

$$\begin{aligned} \mathit{PerfectRef}(q(\mathcal{P}), \mathcal{T}) = & \exists x, y\,(\mathsf{buy}(x,y) \land \mathsf{Antiseizure}(y)) \lor \\ & \exists x, y\,(\mathsf{buy}(x,y) \land \mathsf{contain}(y, \mathsf{phenytoin})) \lor \\ & \exists x, y\,(\mathsf{buy}(x,y) \land \mathsf{Abc}(y)). \end{aligned}$$

Now, according to the definition of *BraveRef*, we have to consider $\mathit{Unify}(q_d, q')$, where q_d is any query in $\mathit{PerfectRef}(q(\mathcal{P}), \mathcal{T})$. For this, we consider all mappings h in $\mathit{Map}(q_d, q')$, and the only resulting unifier σ_h is $\{x \leftarrow \mathsf{john}, y \leftarrow \mathsf{m}_a\}$. However, the restriction $\sigma_h[q']$ is empty since q_1 and q_2 are ground, so $\mathit{Unify}(q_d, q')$ is an empty conjunction, i.e., *true*. Hence:

$$\mathit{BraveRef}(q', \mathcal{T}, \mathcal{P}) = \mathsf{buy}(\mathsf{john}, \mathsf{m}_a) \land \mathsf{Abc}(\mathsf{m}_a) \land \neg \mathit{true}.$$

Obviously, the above BCQ evaluates to false on any given ABox. Note that there does not exist any optimal censor C for $\mathcal{E}$ (actually, for any CQE instance $\langle \mathcal{T}, \mathcal{P}, \mathcal{A}' \rangle$) such that $\mathcal{T} \cup C \models q'$.

On the other hand, q'' alone does not unify with any body of any (rewritten) denial. We hence have that:

$$\mathit{BraveRef}(q'', \mathcal{T}, \mathcal{P}) = q'' = \exists x\,(\mathsf{buy}(\mathsf{john}, \mathsf{m}_a) \land \mathsf{buy}(x, \mathsf{m}_b)).$$

One can see that this last BCQ evaluates to true in $\mathcal{A}$. As we are going to show, it is not a coincidence that there exists an optimal censor for $\mathcal{E}$ (in particular, the censor C_1) which, together with $\mathcal{T}$, entails q''. □

Example 7 illustrates how *BraveRef* can be used to verify the existence of an optimal censor for a given CQE instance that entails a specific BUCQ. To formally prove that this holds in general, we first demonstrate the following property of *Unify*.

Lemma 5. *Let $q = \exists \vec{x}\, conj(\vec{x})$ be a BCQ, q' be a BUCQ, and $\mathcal{A}$ be an ABox. There exists an image I of q in $\mathcal{A}$ such that q' evaluates to false in I iff the following sentence evaluates to true in $\mathcal{A}$:*

$$\exists \vec{x} \left(conj(\vec{x}) \wedge \bigwedge_{q_d \in q'} \neg Unify(q_d, q) \right).$$

Proof. We show that, if $q' = \exists \vec{y}\, conj'(\vec{y})$ is a BCQ, then there exists an image I of q in $\mathcal{A}$ such that q' evaluates to false in I iff $\exists \vec{x}\,(conj(\vec{x}) \wedge \neg Unify(q', q))$ evaluates to true in $\mathcal{A}$. The extension to BUCQs is a direct consequence. Moreover, we assume w.l.o.g. that $\vec{x}$ and $\vec{y}$ are disjoint.

($\Rightarrow$:) Let I be an image of q in $\mathcal{A}$ such that q' evaluates to false in I, then there exists an answer substitution σ_I of the variables in $\vec{x}$ such that $I = Atoms(\sigma_I(q))$, which also implies that $\sigma_I(conj(\vec{x}))$ evaluates to true in $\mathcal{A}$. It remains to show that $\sigma_I(Unify(q', q))$ evaluates to false in $\mathcal{A}$. By contradiction, assume h to be a mapping from q' to q such that $\sigma_I(\bigwedge_{x \leftarrow t \in \sigma_h[q]} x = t)$ evaluates to true in $\mathcal{A}$. This implies that σ_I is a specialization of $\sigma_h[q]$, i.e., $\sigma_I \circ \sigma_h[q] = \sigma_I$. Then, for each atom $\beta \in Atoms(q')$, we have by construction that

$$[\sigma_I \circ \sigma_h](\beta) = \sigma_I(\sigma_h(\beta)) = \sigma_I(\sigma_h(h(\beta))) = [\sigma_I \circ \sigma_h[q]](h(\beta)) = \sigma_I(h(\beta)).$$

Since $h(\beta)$ is an atom in q and I is equal to $Atoms(\sigma_I(q))$, $[\sigma_I \circ \sigma_h](\beta)$ occurs in I, for each $\beta \in Atoms(q')$. But this means that q' evaluates to true in I, against the hypothesis.

($\Leftarrow$:) Assume that $\exists \vec{x} \left(conj(\vec{x}) \wedge \neg Unify(q', q) \right)$ evaluates to true in $\mathcal{A}$. This means that there exists an answer σ to q such that (i) $Atoms(\sigma(q)) \subseteq \mathcal{A}$, and (ii) $\sigma(Unify(q', q))$ evaluates to false in $\mathcal{A}$. Let $I = Atoms(\sigma(q))$. From (i) we know that I is an image of q in $\mathcal{A}$. Then, assume by contradiction that q' evaluates to true in I. This means that, for some answer σ' to q', $Atoms(\sigma'(q')) \subseteq I$. From $Atoms(\sigma'(q')) \subseteq I$ we know that there exists a mapping h of q' into q such that, for every $\beta \in Atoms(q')$, $\sigma'(\beta) = \sigma(h(\beta))$. Therefore, the corresponding MGU σ_h generalizes both σ and σ' when restricted, respectively, to the variables $\vec{x}$ and $\vec{y}$. In other words, there exists a ground substitution σ'' such that:

- $\sigma(x) = [\sigma'' \circ \sigma_h](x)$ for every x occurring in $\vec{x}$;
- $\sigma'(y) = [\sigma'' \circ \sigma_h](y)$ for every y occurring in $\vec{y}$;

However, this is possible only if, for every $\{x \leftarrow t\} \subseteq \sigma_h[q]$, it holds that $\sigma(x) = \sigma''(\sigma_h(x)) = \sigma''(t) = \sigma''(\sigma_h(t)) = \sigma(t)$,[2] i.e., the sentence $\sigma(x = t)$ is valid. This implies that $\sigma(Unify(q', q))$ contains at least one valid disjunct and, consequently, that it evaluates to true in $\mathcal{A}$, thus contradicting point (ii). ■

We use the above lemma to establish the fundamental property of the above query reformulation function *BraveRef*.

Proposition 11. *Let $\langle \mathcal{T}, \mathcal{P} \rangle$ be a DL-Lite$_\mathcal{R}$ CQE specification and let q be a BUCQ. For every ABox $\mathcal{A}$, there exists an optimal censor C for $\langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ such that $\mathcal{T} \cup C \models q$ iff $BraveRef(q, \mathcal{T}, \mathcal{P})$ evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$.*

Proof. Let us call $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$. First, from Proposition 5 it follows that there exists an optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models q$ iff there exists an optimal censor C for $\mathcal{E}$ such that $PerfectRef(q, \mathcal{T})$ evaluates to true in C.

Now, since $C \subseteq \mathsf{cl}_\mathcal{T}(\mathcal{A})$, and since every subset $\mathcal{A}'$ of $\mathsf{cl}_\mathcal{T}(\mathcal{A})$ such that $\mathcal{T} \cup \mathcal{P} \cup \mathcal{A}'$ is consistent is contained in some optimal censor for $\mathcal{E}$, it follows that there exists an optimal censor C for $\mathcal{E}$ such that $PerfectRef(q, \mathcal{T})$ evaluates to true in C iff there exists an image I of $PerfectRef(q, \mathcal{T})$ in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$ such that $\mathcal{T} \cup \mathcal{P} \cup I$ is consistent. Now, by Proposition 6, $\mathcal{T} \cup \mathcal{P} \cup I$ is consistent iff $PerfectRef(q(\mathcal{P}), \mathcal{T})$ evaluates to false in I. Consequently, Lemma 5 implies the thesis. ■

Then, we use *BraveRef* to define the new query reformulation function *StateRef* as follows.

Definition 11 *(StateRef)*. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ CQE instance, $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 0$) be a sequence of BUCQs, let i be such that $i \in [1, n]$, and let $I \subseteq \{1, \ldots, i-1\}$: I represents a set of indexes of the queries that precede query q_i in $\mathcal{Q}$ and that are assumed to be true in the state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$. We define $StateRef(\mathcal{S}, i, I)$ as the FO sentence:

$$\left(\bigwedge_{\substack{j \in [1, i-1] \\ \wedge\, j \notin I}} \neg BraveRef\Big(\big(\bigwedge_{\ell \in I \,\wedge\, \ell < j} q_\ell \big) \wedge q_j, \mathcal{T}, \mathcal{P} \Big) \right) \wedge BraveRef\Big(\big(\bigwedge_{\ell \in I} q_\ell \big) \wedge q_i, \mathcal{T}, \mathcal{P} \Big)$$

Example 8. Consider the DL-Lite$_\mathcal{R}$ CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ and the query sequence $\mathcal{Q} = \langle q_1, q_2, q_3 \rangle$ introduced in Example 6, and recall the reformulations illustrated in Example 7. By setting $i = 3$ and $I = \{1\}$, we have:

[2] In particular, $\sigma''(t) = \sigma''(\sigma_h(t))$ holds because $\sigma_h(t) = t$, since the MGU σ_h is idempotent.

$$\begin{aligned} \mathit{StateRef}(\mathcal{S}, i, I) =& \neg \mathit{BraveRef}(q_1 \wedge q_2, \mathcal{T}, \mathcal{P}) \wedge \mathit{BraveRef}(q_1 \wedge q_3, \mathcal{T}, \mathcal{P}) \\ =& \neg(\mathsf{buy}(\mathsf{john}, \mathsf{m}_a) \wedge \mathsf{Abc}(\mathsf{m}_a) \wedge \neg \mathit{true}) \wedge \\ & \exists x\, (\mathsf{buy}(\mathsf{john}, \mathsf{m}_a) \wedge \mathsf{buy}(x, \mathsf{m}_b)). \quad \square \end{aligned}$$

The function *StateRef* is characterized by the following lemma.

Lemma 6. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ CQE instance, let $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ be a sequence of BUCQs, let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, let i be such that $i \in [1, n]$, and let $I \subseteq \{1, \dots, i-1\}$. Then, the FO sentence $\mathit{StateRef}(\mathcal{S}, i, I)$ evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$ iff the following two conditions hold:*

- *$q_i \in \mathit{EntQ}(\mathcal{S})$ and*
- *for each $j \in [1, i-1]$, $q_j \in \mathit{EntQ}(\mathcal{S})$ iff $j \in I$*

Proof. ($\Leftarrow$:) First, suppose that $\mathit{StateRef}(\mathcal{S}, i, I)$ evaluates to false in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$. We have two cases:

(*i*) The sentence $\mathit{BraveRef}((\bigwedge_{\ell \in I} q_\ell) \wedge q_i, \mathcal{T}, \mathcal{P})$ (i.e., the last conjunct of $\mathit{StateRef}(\mathcal{S}, i, I)$) evaluates to false in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$. Then, Proposition 11 implies that there exists no optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models (\bigwedge_{\ell \in I} q_\ell) \wedge q_i$, which implies that either $q_i \notin \mathit{EntQ}(\mathcal{S})$ or there exists $j \in I$ such that $q_j \notin \mathit{EntQ}(\mathcal{S})$, thus proving the thesis.

(*ii*) The sentence $\mathit{BraveRef}((\bigwedge_{\ell \in I} q_\ell) \wedge q_i, \mathcal{T}, \mathcal{P})$ evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$ and, for some j such that $j \in [1, i-1]$ and $j \notin I$, the conjunct $\neg \mathit{BraveRef}((\bigwedge_{\ell \in I \wedge \ell < j} q_\ell) \wedge q_j, \mathcal{T}, \mathcal{P}))$ evaluates to false in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$. Let us assume that j is the least index such that the above property holds. Then, Proposition 11 implies that there exists an optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models (\bigwedge_{\ell \in I \wedge \ell < j} q_\ell) \wedge q_j$, which implies that $q_j \in \mathit{EntQ}(\mathcal{S})$, thus proving the thesis.

($\Rightarrow$:) Suppose that $\mathit{StateRef}(\mathcal{S}, i, I)$ evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$. Then, also its last conjunct $\mathit{BraveRef}((\bigwedge_{\ell \in I} q_\ell) \wedge q_i, \mathcal{T}, \mathcal{P})$ does. Therefore, by Proposition 11 there exists an optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models (\bigwedge_{\ell \in I} q_\ell) \wedge q_i$, i.e., such that $\mathcal{T} \cup C \models q_\ell$ for every $\ell \in I \cup \{i\}$.

We now prove by induction that every $j \in [1, i-1]$ is such that $q_j \in \mathit{EntQ}(\mathcal{S})$ iff $j \in I$.

- Base case: $j = 1$. We prove that $1 \in I$ iff $q_1 \in \mathit{EntQ}(\mathcal{S})$:

  (*i*) Suppose that $1 \in I$. Then, for what above, there exists an optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models q_1$. Thus, we have that $q_1 \in \mathit{EntQ}(\mathcal{E}, \langle q_1 \rangle)$, which by Lemma 2 implies that $q_1 \in \mathit{EntQ}(\mathcal{S})$.

  (*ii*) Suppose that $1 \notin I$. In this case, $\neg \mathit{BraveRef}(q_1, \mathcal{T}, \mathcal{P})$ is one of the conjuncts of $\mathit{StateRef}(\mathcal{S}, i, I)$. Since the latter evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$, also $\neg \mathit{BraveRef}(q_1, \mathcal{T}, \mathcal{P})$ does. Hence, $\mathit{BraveRef}(q_1, \mathcal{T}, \mathcal{P})$ evaluates to false in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$, which by Proposition 11 implies that there exists no optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models q_1$. Thus, it holds that $q_1 \notin \mathit{EntQ}(\mathcal{S})$.

- Inductive case: $j \leq i-1$ and, for each $\ell \in [1, j-1]$, $\ell \in I$ iff $q_\ell \in \mathit{EntQ}(\mathcal{S})$. We prove that $j \in I$ iff $q_j \in \mathit{EntQ}(\mathcal{S})$. First observe that, since $\mathit{EntQ}(\mathcal{S}_k) \subseteq \mathit{EntQ}(\mathcal{S})$ for every $k < i$, then every $q_\ell \in \mathit{EntQ}(\mathcal{S})$ such that $\ell \in [1, j-1]$ belongs to $q_\ell \in \mathit{EntQ}(\mathcal{S}_{j-1})$ too. Then, we can assume w.l.o.g. that, for each $\ell \in [1, j-1]$, $\ell \in I$ iff $q_\ell \in \mathit{EntQ}(\mathcal{S}_{j-1})$. Now, two cases are possible:

  (*i*) $j \in I$. As claimed above, there exists an optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models q_\ell$ for every $\ell \in I \cup \{i\}$ and, in particular, for every $\ell \leq j$ belonging to I. Observe that $\mathcal{T} \cup C \models q_\ell$ holds for $\ell = j$ and, by the above assumption, for every $q_\ell \in \mathit{EntQ}(\mathcal{S}_{j-1})$. Consequently, there exists an optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models (\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{j-1})} q) \wedge q_j$, which implies that $C \in \mathit{StCens}(\mathcal{S}_j)$, therefore $q_j \in \mathit{EntQ}(\mathcal{S}_j)$, and hence $q_j \in \mathit{EntQ}(\mathcal{S})$.

  (*ii*) $j \notin I$. In this case, since $\mathit{StateRef}(\mathcal{S}, i, I)$ evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$, also its conjunct $\neg \mathit{BraveRef}((\bigwedge_{\ell \in I \wedge \ell < j} q_\ell) \wedge q_j, \mathcal{T}, \mathcal{P})$ evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$. Then, by the above assumption, we have that $\mathit{BraveRef}((\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{j-1})} q) \wedge q_j, \mathcal{T}, \mathcal{P})$ evaluates to false in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$. By Proposition 11, this implies that there exists no optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models (\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{j-1})} q) \wedge q_j$, which implies that $q_j \notin \mathit{EntQ}(\mathcal{S})$.

Then, since for each $j \in [1, i-1]$, $q_j \in \mathit{EntQ}(\mathcal{S})$ iff $j \in I$, and since $\mathit{BraveRef}((\bigwedge_{j \in I} q_j) \wedge q_i, \mathcal{T}, \mathcal{P})$ evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$, Proposition 11 implies that there exists an optimal censor C for $\mathcal{E}$ such that $\mathcal{T} \cup C \models (\bigwedge_{q \in \mathit{EntQ}(\mathcal{S}_{i-1})} q) \wedge q_i$, which implies that $C \in \mathit{StCens}(\mathcal{S}_i)$, therefore $q_i \in \mathit{EntQ}(\mathcal{S}_i)$, and hence $q_i \in \mathit{EntQ}(\mathcal{S})$, thus proving the thesis. ■

A direct consequence of Lemma 6 is that the function *StateRef* allows for reducing dynCQE query entailment to evaluating an FO query.

Corollary 2. *Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ CQE instance, $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ (with $n \geq 0$) be a sequence of BUCQs and $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$. For every $i \in [1, n]$ we have that $\mathsf{dynCQE}(\mathcal{S}, q_i) = \{\epsilon\}$ iff the following FO sentence evaluates to true in $\mathsf{cl}_\mathcal{T}(\mathcal{A})$:*

$$\bigvee_{I \in \wp(\{1, \dots, i-1\})} \mathit{StateRef}(\mathcal{S}, i, I),$$

where $\wp(\{1,\ldots,i-1\})$ *denotes the powerset of* $\{1,\ldots,i-1\}$.

The last two results show the FO rewritability of the problems studied on $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$. We now modify the respective reformulations to evaluate them directly on the ABox $\mathcal{A}$ and thus produce "genuine" FO rewritability results.

In what follows, we will make use of the algorithm *AtomRewr* provided in [9], which we now briefly describe. Given an FO sentence ϕ and a DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$, $\mathit{AtomRewr}(\phi,\mathcal{T})$ computes the FO sentence obtained from ϕ by replacing every atom $\alpha = p(\vec{x})$ (where $\vec{x}$ are all the variables occurring in α) with the disjunction of atoms corresponding to the perfect rewriting of the non-Boolean atomic query $q_\alpha = \{\vec{x} \mid p(\vec{x})\}$ with respect to $\mathcal{T}$.

For our purposes, we recall the key property of *AtomRewr* provided in [9].

Proposition 12. *For every FO sentence ϕ, DL-Lite$_{\mathcal{R}}$ TBox $\mathcal{T}$, and ABox $\mathcal{A}$, ϕ evaluates to true in $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ iff $\mathit{AtomRewr}(\phi,\mathcal{T})$ evaluates to true in $\mathcal{A}$.*

Now, Proposition 12 and Corollary 2 immediately imply the next property.

Proposition 13. *Let $\mathcal{E} = \langle \mathcal{T},\mathcal{P},\mathcal{A}\rangle$ be a DL-Lite$_{\mathcal{R}}$ CQE instance, $\mathcal{Q} = \langle q_1,\ldots,q_n\rangle$ be a sequence of BUCQs. For every $i \in [1,n]$, we have that $q_i \in \mathit{EntQ}(\mathcal{S})$ iff the following FO sentence evaluates to true in $\mathcal{A}$:*

$$\mathit{AtomRewr}\Big(\bigvee_{I\in\wp(\{1,\ldots,i-1\})} \mathit{StateRef}(\mathcal{S},i,I),\mathcal{T}\Big).$$

Example 9. Let $\mathcal{E}$ and $\mathcal{Q} = \langle q_1,q_2,q_3\rangle$ be as in Example 6. According to Proposition 13, the query $q_3 = \exists x\,\mathsf{buy}(x,\mathsf{m}_b)$ belongs to $\mathit{EntQ}(\langle\mathcal{E},\mathcal{Q}\rangle)$ if and only if the FO sentence below evaluates to true in $\mathcal{A}$ (f_I denotes the sub-formula considering the guess I of the indexes of the queries that precede the query q_3):

$$\begin{array}{r|l}
 & \mathit{AtomRewr}(\bigvee_{I\in\wp(\{1,2\})} \mathit{StateRef}(\langle\mathcal{E},\mathcal{Q}\rangle,i,I),\mathcal{T}) = \\
f_{I=\emptyset} & \neg\mathit{BraveRef}(q_1,\mathcal{T},\mathcal{P}) \wedge \neg\mathit{BraveRef}(q_2,\mathcal{T},\mathcal{P}) \wedge \mathit{BraveRef}(q_3,\mathcal{T},\mathcal{P}) \vee \\
f_{I=\{1\}} & \neg\mathit{BraveRef}(q_1 \wedge q_2,\mathcal{T},\mathcal{P}) \wedge \mathit{BraveRef}(q_1 \wedge q_3,\mathcal{T},\mathcal{P}) \vee \\
f_{I=\{2\}} & \neg\mathit{BraveRef}(q_1,\mathcal{T},\mathcal{P}) \wedge \mathit{BraveRef}(q_2 \wedge q_3,\mathcal{T},\mathcal{P}) \vee \\
f_{I=\{1,2\}} & \mathit{BraveRef}(q_1 \wedge q_2 \wedge q_3,\mathcal{T},\mathcal{P}) = \\
f_{I=\emptyset} & \neg\mathsf{buy}(\mathsf{john},\mathsf{m}_a) \wedge \neg\mathsf{Abc}(\mathsf{m}_a) \wedge \exists x\,\mathsf{buy}(x,\mathsf{m}_b) \vee \\
f_{I=\{1\}} & \neg(\mathsf{buy}(\mathsf{john},\mathsf{m}_a) \wedge \mathsf{Abc}(\mathsf{m}_a) \wedge \neg\mathit{true}) \wedge \exists x\,(\mathsf{buy}(\mathsf{john},\mathsf{m}_a) \wedge \mathsf{buy}(x,\mathsf{m}_b)) \vee \\
f_{I=\{2\}} & \neg(\mathsf{buy}(\mathsf{john},\mathsf{m}_a)) \wedge (\exists x\,(\mathsf{Abc}(\mathsf{m}_a) \wedge \mathsf{buy}(x,\mathsf{m}_b))) \vee \\
f_{I=\{1,2\}} & \exists x\,(\mathsf{buy}(\mathsf{john},\mathsf{m}_a) \wedge \mathsf{Abc}(\mathsf{m}_a) \wedge \mathsf{buy}(x,\mathsf{m}_b)) \wedge \neg\mathit{true}
\end{array}$$

which, indeed, evaluates to true in $\mathcal{A}$ thanks to $f_{I=\{1\}}$. □

The following result is a direct consequence of Proposition 13.

Theorem 1. *$\mathit{REC}[\mathsf{dynCQE},\boldsymbol{\mathit{BUCQ}}]$ is FO rewritable, and therefore in AC^0 in data complexity.*

6. Dynamic CQE for open UCQs

We now extend dynamic CQE to accommodate queries with free variables. Proposition 4 proved that all MC-CQE semantics *collapse* to dynCQE when dealing with Boolean queries. In particular, a Boolean query always divides censors into two distinct sets (those that satisfy the query and those that do not), consequently, maximal cooperativity induces a single preference that favors censors in the former set (see Definition 7).

Extending dynCQE to UCQs essentially means extending the notion of preference over censors. However, an open query q can yield multiple sets of certain answers. Then, any such set that qualifies as a secure answer and is not contained in any other secure answer (since providing more information is inherently more cooperative) represents a legitimate preference. This leads to a family of possible extensions of dynCQE.[3]

In this regard, we initially consider a wide range of preferences that involve the comparison of finite sets of certain answers as a whole. Subsequently, we narrow our focus to preferences derived from lexicographic orders induced by single answers. As we will show, this specific class of preferences enables us to reduce the problem of evaluating an open query to the Boolean case.

[3] Consider, for example, two drugs that are commonly prescribed individually, but whose combination provides strong evidence of a specific disease we intend to keep confidential. In such a case, disclosing that a patient is taking both drugs would compromise privacy. However, in the interest of cooperativeness, it may be acceptable to reveal only one. Both choices are legitimate and should be made based on the specific context. This scenario will be formalized in greater detail in Example 11.

6.1. General preferences

For each UCQ q, let $\preceq_q$ be a total order on $\mathbf{Fin}(\mathfrak{G}_q)$ (the set of all *finite* subsets of $\mathfrak{G}_q$), which extends set inclusion. In particular, requiring each $\preceq_q$ to be a total order means that, given any finite collection of $\mathbf{Fin}(\mathfrak{G}_q)$, there is always a single preferred one.

We use the total order $\preceq_q$ to select the preferred censors among the available ones to answer q. To this end, we define the following weak order between censors.

Definition 12 *(Preference between censors).* Let $\mathcal{C}_1, \mathcal{C}_2 \in \mathit{Cens}(\mathcal{E})$ and let q be a UCQ, we define:

$$\mathcal{C}_1 \preceq_q \mathcal{C}_2 \text{ iff } \mathit{cert}(\mathcal{T} \cup \mathcal{C}_1, q) \preceq_q \mathit{cert}(\mathcal{T} \cup \mathcal{C}_2, q).$$

As usual, in what follows, we use $\sim_q$ and $\prec_q$ to denote the symmetric and asymmetric parts of $\preceq_q$, respectively.

By the fact that $\preceq_q$ is a total order and extends set inclusion, it is straightforward to see that $\preceq_q$ refines the notion of maximal cooperativity, specifically:

$$\mathit{cert}(\mathcal{T} \cup \mathcal{C}_1, q) \subseteq \mathit{cert}(\mathcal{T} \cup \mathcal{C}_2, q) \text{ implies } \mathcal{C}_1 \preceq_q \mathcal{C}_2, \tag{1}$$

$$\mathit{cert}(\mathcal{T} \cup \mathcal{C}_1, q) \subset \mathit{cert}(\mathcal{T} \cup \mathcal{C}_2, q) \text{ implies } \mathcal{C}_1 \prec_q \mathcal{C}_2, \tag{2}$$

$$\mathit{cert}(\mathcal{T} \cup \mathcal{C}_1, q) = \mathit{cert}(\mathcal{T} \cup \mathcal{C}_2, q) \text{ iff } \mathcal{C}_1 \sim_q \mathcal{C}_2. \tag{3}$$

Hereafter, we assume to fix a set of total orders $\{\preceq_q\}_{q \in \mathbf{UCQ}}$, which induces a family $\preceq = \{\preceq_q\}_{q \in \mathbf{UCQ}}$ according to Definition 12. We can now use this family of preferences between censors to extend Definition 8 to the case of open UCQs as follows.

Definition 13. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, and let $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 1$) be a sequence of UCQs. The set $\mathit{StCens}[\preceq](\mathcal{S}_i)$ of censors of $\mathcal{S}_i$, with $i \in [1, n]$, is inductively defined as follows:

- $\mathit{StCens}[\preceq](\mathcal{S}_0) = \mathit{OptCens}(\mathcal{E})$;
- $\mathit{StCens}[\preceq](\mathcal{S}_i) = \max\limits_{\preceq_{q_i}} \mathit{StCens}[\preceq](\mathcal{S}_{i-1})$.

Analogously to the Boolean case, being each $\preceq_{q_i}$ a weak order, the set $\mathit{StCens}[\preceq](\mathcal{S}_i)$ may include multiple censors. However, by construction, all these censors provide the same set of certain answers for the query q_i, which is indeed the set of certain answers that is maximal with respect to the fixed total order $\preceq_{q_i}$. As a special case, if none of the optimal censors $\mathcal{C}$ in $\mathit{StCens}[\preceq](\mathcal{S}_{i-1})$ satisfies (together with $\mathcal{T}$) a query q_i (i.e., $\mathit{cert}(\mathcal{T} \cup \mathcal{C}, q_i) = \emptyset$, for all $\mathcal{C} \in \mathit{StCens}[\preceq](\mathcal{S}_{i-1})$), then no censor prevails and hence $\mathit{StCens}[\preceq](\mathcal{S}_i) = \mathit{StCens}[\preceq](\mathcal{S}_{i-1})$. From this observation, it is straightforward to see that Lemma 2 can be generalized to the case of open queries, for any family $\preceq$ of preferences over censors.

Then, we define a CQE semantics as follows:

Definition 14 *(Dynamic CQE for **UCQ** – dynCQE[$\preceq$]).* Let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ be a protection state, where $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ is a CQE instance and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (with $n \geq 1$) a sequence of UCQs. We define $\mathsf{dynCQE}[\preceq](\mathcal{S}, q_i) = \mathit{cert}(\mathcal{T} \cup \mathcal{C}, q_i)$, where $\mathcal{C}$ is any censor in $\mathit{StCens}[\preceq](\mathcal{S})$.

Moreover, $\mathit{EntQ}[\preceq](\mathcal{S})$ denotes the set of instantiated queries entailed by $\mathcal{S}$:

$$\mathit{EntQ}[\preceq](\mathcal{S}) = \bigcup_{i=1}^{n} \{\sigma(q_i) \mid \sigma \in \mathsf{dynCQE}[\preceq](\mathcal{S}, q_i)\}.$$

Example 10. A European e-gov facility aims to conceal the identity of underage people who serves a sentence of imprisonment.

Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, where:

$$\mathcal{T} = \emptyset;$$

$$\mathcal{P} = \{\exists x\, (\mathsf{EUcitizen}(x) \wedge \mathsf{Underage}(x) \wedge \mathsf{Convict}(x)) \rightarrow \bot\};$$

$$\mathcal{A} = \{\mathsf{EUcitizen(john)}, \mathsf{Underage(john)}, \mathsf{Convict(john)}\}.$$

Let us start by considering an empty sequence of UCQs. By definition, we have that $\mathit{StCens}(\langle \mathcal{E}, \langle\rangle \rangle)$ coincides with the set of optimal censors for $\mathcal{E}$:

$$\mathcal{C}_1 = \{\mathsf{EUcitizen(john)}, \mathsf{Underage(john)}\};$$

$$\mathcal{C}_2 = \{\mathsf{EUcitizen(john)}, \mathsf{Convict(john)}\};$$

$$\mathcal{C}_3 = \{\mathsf{Underage(john)}, \mathsf{Convict(john)}\}.$$

Let $q_1 = \mathsf{EUcitizen}(x)$ be the first query. We have that $cert(\mathcal{C}_1, q_1) = cert(\mathcal{C}_2, q_1) = \{\{x \leftarrow \mathsf{john}\}\}$ whereas $cert(\mathcal{C}_3, q_1) = \emptyset$. Note that any $\preceq_{q_1}$ which extends set inclusion induces a preference over censors such that $\mathcal{C}_3 \prec_{q_1} \mathcal{C}_1 \sim_{q_1} \mathcal{C}_2$. As a result, $StCens(\langle \mathcal{E}, \langle q_1 \rangle\rangle)$ consists of the censors $\mathcal{C}_1$ and $\mathcal{C}_2$ and returns the fact that John is an EU citizen.

Then, let $q_2 = \mathsf{Underage}(x) \wedge \mathsf{Convict}(x)$. Neither $\mathcal{C}_1$ nor $\mathcal{C}_2$ can answer this query, i.e., $cert(\mathcal{C}_1, q_1) = cert(\mathcal{C}_2, q_1) = \emptyset$. Moreover, by (3), $\mathcal{C}_1$ and $\mathcal{C}_2$ are both $\preceq_{q_2}$-maximal; this implies that $StCens(\langle \mathcal{E}, \langle q_1, q_2 \rangle\rangle) = StCens(\langle \mathcal{E}, \langle q_1 \rangle\rangle)$. □

Let us show that $\mathsf{dynCQE}[\preceq]$ is an MC-CQE semantics.

Theorem 2. *For each protection state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ with $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$, there exists a censor $\mathcal{C} \in Cens(\mathcal{E})$ such that $\mathcal{C}$ is maximally cooperative with respect to $\mathcal{Q}$ and $\mathsf{dynCQE}[\preceq](\mathcal{S}, q_i) = cert(\mathcal{T} \cup \mathcal{C}, q_i)$, for all $i \in [1, n]$.*

Proof. Let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ be a protection state with $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ and let $\mathcal{C}$ be any optimal censor for $\mathcal{E}$ that belongs to $StCens[\preceq](\mathcal{S})$. By definition of $\mathsf{dynCQE}[\preceq]$, it directly follows that for all $i \in [1, n]$, $\mathsf{dynCQE}[\preceq](\mathcal{S}, q_i) = cert(\mathcal{T} \cup \mathcal{C}, q_i)$. It remains to prove that $\mathcal{C}$ is maximally cooperative with respect to $\mathcal{Q}$. Assume, by contradiction, that there exists a censor $\mathcal{C}' \in Cens(\mathcal{E})$ which is more cooperative than $\mathcal{C}$ with respect to $\mathcal{Q}$, i.e., there exists a natural number m such that:

(*i*) $cert(\mathcal{T} \cup \mathcal{C}, q_i) = cert(\mathcal{T} \cup \mathcal{C}', q_i)$, for all $i \in [1, m]$;
(*ii*) $cert(\mathcal{T} \cup \mathcal{C}, q_{m+1}) \subset cert(\mathcal{T} \cup \mathcal{C}', q_{m+1})$.

It is straightforward to see that Lemma 3 can be extended to $\mathsf{dynCQE}[\preceq]$ over UCQs (the proof is essentially the same), therefore Condition (*i*) implies that for all $i \in [1, m]$, $\mathcal{C} \in StCens[\preceq](\mathcal{S}_i)$ iff $\mathcal{C}' \in StCens[\preceq](\mathcal{S}_i)$. Then, given that $\mathcal{C} \in StCens[\preceq](\mathcal{S})$, we have that $\mathcal{C} \in StCens[\preceq](\mathcal{S}_m)$ and $\mathcal{C}' \in StCens[\preceq](\mathcal{S}_m)$ too. Now, by assumption, $\preceq_{q_{m+1}}$ extends set inclusion, therefore Condition (*ii*) implies that $\mathcal{C} \prec_{q_{m+1}} \mathcal{C}'$. By construction this means that $\mathcal{C} \notin StCens[\preceq](\mathcal{S}_{m+1})$, which contradicts that $\mathcal{C} \in StCens[\preceq](\mathcal{S})$. ■

A direct consequence of Theorem 2 and Proposition 4 is that any $\mathsf{dynCQE}[\preceq]$ behaves as dynCQE over Boolean queries.

Corollary 3. *Let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, where $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ is a sequence of BUCQs. For each family $\preceq$ of preferences over censors,* $\mathsf{dynCQE}[\preceq](\mathcal{S}, q_i) = \mathsf{dynCQE}(\mathcal{S}, q_i)$*, for all $i \in [1, n]$.*

6.2. Lexicographic preferences

A way to obtain a preference relation is to consider a total order on the single answers of a given UCQ q and then use such an order to compare finite sets of answers lexicographically. In what follows, we show that every preference relation defined in this way is a total order over $\mathbf{Fin}(\mathfrak{G}_q)$, which extends set inclusion. Moreover, we shall also show that with such preference relations the problem of evaluating a sequence of open queries can be reduced to evaluating a sequence of Boolean queries.

Definition 15. Let $\trianglelefteq_q$ be a total order over $\mathfrak{G}_q$. For each $R, S \in \mathbf{Fin}(\mathfrak{G}_q)$, we consider the symmetric difference of R and S: $D(R, S) = (R \cup S) \setminus (R \cap S)$ and we say that $R \preceq_q S$ iff $R = S$ or $\max_{\trianglelefteq_q} D(R, S) \in S$.

Proposition 14. *Let q be a UCQ, let $\trianglelefteq_q$ be a total order on $\mathfrak{G}_q$, then the binary relation $\preceq_q$ on $\mathbf{Fin}(\mathfrak{G}_q)$ obtained from Definition 15 is a total order which extends set inclusion.*

Proof. We start by observing that, for each $R, S \in \mathbf{Fin}(\mathfrak{G}_q)$ the set $D(R, S)$ has the following properties: $R = S$ iff $D(R, S) = \emptyset$, $D(R, S) = D(S, R)$ and moreover, $D(R, S) \subseteq R \cup S$, therefore $D(R, S)$ is finite. This in turn means that since $\trianglelefteq_q$ is a total order, if $D(R, S) \neq \emptyset$, then it has a maximum. Now, let us prove that $\preceq_q$ is a total order over $\mathbf{Fin}(\mathfrak{G}_q)$:

- Reflexivity: let $R \in \mathbf{Fin}(\mathfrak{G}_q)$, then by definition it directly follows that $R \preceq_q R$ holds.
- Antisymmetry: Let $R, S \in \mathbf{Fin}(\mathfrak{G}_q)$ such that $R \preceq_q S$ and $S \preceq_q R$. If $R \neq S$, then by our previous observation, $\max_{\trianglelefteq_q} D(R, S) \in R$ and $\max_{\trianglelefteq_q} D(S, R) \in S$, but $D(R, S) = D(S, R)$ and so this means that $\max_{\trianglelefteq_q} D(R, S) \in R \cap S$. But by definition of $D(R, S)$, none of its elements can be contained both in R and S, so we have a contradiction. Hence, $R = S$.
- Transitivity: Let $R, S, T \in \mathbf{Fin}(\mathfrak{G}_q)$ such that $R \preceq_q S$ and $S \preceq_q T$. If $R = S$ or $S = T$ or $R = T$, then, we directly get $R \preceq_q T$. So, we can assume that R, S and T are pairwise distinct. Then, there exist $\sigma_1 = \max_{\trianglelefteq_q} D(R, S)$ and $\sigma_2 = \max_{\trianglelefteq_q} D(S, T)$ such that $\sigma_1 \in S \setminus R$ and $\sigma_2 \in T \setminus S$. Clearly since $\sigma_1 \in S$ and $\sigma_2 \notin S$, $\sigma_1 \neq \sigma_2$. Now, let $\sigma = \max_{\trianglelefteq_q} \{\sigma_1, \sigma_2\}$ and let us prove that $\sigma \in T \setminus R$. We distinguish two cases:
  - if $\sigma = \sigma_1$, then $\sigma_2 \lhd_q \sigma_1$ and so $\sigma_1 \notin D(S, T)$. Since $\sigma_1 \in S$, then $\sigma_1 \in T$ as well. Hence $\sigma = \sigma_1 \in T. \setminus R$.
  - if $\sigma = \sigma_2$, then $\sigma_1 \lhd_q \sigma_2$ and so $\sigma_2 \notin D(R, S)$. Since $\sigma_2 \notin S$, $\sigma_2 \notin R$ as well. Hence $\sigma = \sigma_2 \in T \setminus R$.

  Finally, we prove that for each $\tau \in D(R, T)$, if $\sigma \lhd_q \tau$, then $\tau \in T \setminus R$. Assume by contradiction that there exists $\tau \in R \setminus T$ such that $\sigma \lhd_q \tau$, then $\sigma_1 \lhd_q \tau$ and $\sigma_2 \lhd_q \tau$. This means that $\tau \notin D(R, S)$ and $\tau \notin D(S, T)$. Now, since $\tau \in R$ and $\tau \in D(R, S)$, we have that $\tau \in S$ too. At the same time, since $\tau \notin T$ and $\tau \notin D(S,T)$, we have that $\tau \notin S$, a contradiction. This proves that $\max_{\trianglelefteq_q} D(R,T) \in T$.

- Totality: Let $R, S \in \mathbf{Fin}(\mathfrak{S}_q)$, we want to prove that either $R \leq_q S$ or $S \leq_q R$, but, by Definition 15, this is equivalent to proving that either $R = S$ or $D(R,S)$ has maximum w.r.t. $\trianglelefteq_q$. But from our observations above, we have that if $R \neq S$, $D(R,S)$ has maximum w.r.t. $\trianglelefteq$.

Finally, we prove that $\leq_q$ extends set inclusion. Let $R, S \in \mathbf{Fin}(\mathfrak{S}_q)$ such that $R \subseteq S$. By definition, $D(R,S) \subseteq R \cup S$, but $R \subseteq S$, so $D(R,S) \subseteq S$. Hence, if $R \neq S$, $D(R,S)$ has a maximum that is necessarily contained in S and so $R \leq_q S$. ■

Example 11. Interferon and ribavirin are both antiviral medications that are used to treat hepatitis C. Interferon is a protein that helps the body's immune system fight infection. Ribavirin is a synthetic drug that inhibits the replication of viruses. Taking both interferon and ribavirin was a strong indicator that a patient is affected by hepatitis C.

Then, consider the CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$, where:

$$\mathcal{T} = \emptyset;$$

$$\mathcal{P} = \{\exists x\,(\mathsf{take}(x, \mathsf{ribavirin}) \wedge \mathsf{take}(x, \mathsf{interferon})) \rightarrow \bot\};$$

$$\mathcal{A} = \{\mathsf{take}(\mathsf{john}, \mathsf{ribavirin}), \mathsf{take}(\mathsf{john}, \mathsf{interferon})\}$$

The optimal censors for $\mathcal{E}$ are $C_1 = \{\mathsf{take}(\mathsf{john}, \mathsf{ribavirin})\}$ and $C_2 = \{\mathsf{take}(\mathsf{john}, \mathsf{interferon})\}$. Now, given the query $q = \mathsf{take}(\mathsf{john}, y)$, C_1 returns a single answer $\sigma_1 = \{y \leftarrow \mathsf{ribavirin}\}$, whereas C_2 returns $\sigma_2 = \{y \leftarrow \mathsf{interferon}\}$. Since interferon is more commonly used than ribavirin, we can prefer to disclose a generic drug with respect to a more specific one; we can express such preference by imposing $\sigma_1 \triangleleft_q \sigma_2$, which implies that $C_1 \prec_q C_2$, as required. □

The following example shows that lexicographic preferences can be uniformly defined for any query, starting from an ordering over single individual names.

Example 12. Assume that there is a total order $\preceq$ over the individual names of the alphabet. We now define, for each UCQ q, a total order $\trianglelefteq_q$ over $\mathfrak{S}_q$ as the lexicographic order induced by $\preceq$. Let $q(x_1, \dots, x_n)$ be an open n-ary UCQ (with $n > 0$) and let $\sigma_1, \sigma_2 \in \mathfrak{S}_q$, we say that $\sigma_1 \trianglelefteq_q \sigma_2$ iff $\sigma_1 = \sigma_2$ or there exists $i \in [0, n-1]$ such that:

- $\sigma_1(x_j) = \sigma_2(x_j)$, for each $j \in [0, i]$;
- $\sigma_1(x_{i+1}) \prec \sigma_2(x_{i+1})$. □

Hereafter, with a slight abuse of notation, we shortly write $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ to refer to a $\mathsf{dynCQE}[\preceq]$ semantics where $\preceq$ results from Definition 15 and a collection $\trianglelefteq$ of $\trianglelefteq_q$ for each UCQ q. Similarly, $\mathit{StCens}[\preceq]$ is replaced by $\mathit{StCens}_{seq}[\trianglelefteq]$ and $\mathit{EntQ}[\preceq]$ is replaced by $\mathit{EntQ}_{seq}[\trianglelefteq]$.

A family of preferences of this kind induces, for each protection state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, a corresponding finite sequence of BUCQs $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q})$ as follows.

Definition 16. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance and let q be a UCQ, we define the sequence of BUCQs generated by $\trianglelefteq$:

$$\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, q) = \langle \sigma_1(q), \dots, \sigma_k(q) \rangle$$

where $\sigma_1, \dots, \sigma_k$ are the elements of $\mathit{cert}(\mathcal{T} \cup \mathcal{A}, q)$ ordered *decreasingly* with respect to $\trianglelefteq_q$. Furthermore, if $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ is a sequence of UCQs, we define

$$\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q}) = \mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, q_1) \circ \cdots \circ \mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, q_n),$$

where $\circ$ denotes the concatenation of tuples.

Using this construction, we now prove that evaluating a sequence of UCQs $\mathcal{Q}$ under a $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ semantics is equivalent to evaluating a sequence of BUCQs.

Proposition 15. *Let $\trianglelefteq = \{\trianglelefteq_q\}_{q \in UCQ}$ be a family of total orders over certain answers. For each protection state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, we have that*

$$\mathit{EntQ}_{seq}[\trianglelefteq](\mathcal{S}) = \mathit{EntQ}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S})),$$

where $\mathsf{BoolState}[\trianglelefteq](\mathcal{S}) = \langle \mathcal{E}, \mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q}) \rangle$.

Proof. First, note that all the BUCQs that occur in $\mathit{EntQ}_{seq}[\trianglelefteq](\mathcal{S})$ or $\mathit{EntQ}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S}))$ also occur in $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q})$.

Assume by contradiction that $\mathit{EntQ}_{seq}[\trianglelefteq](\mathcal{S}) \neq \mathit{EntQ}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S}))$. Then, there exists a BUCQ $\sigma(q_i)$ that is the first BUCQ in $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E},\mathcal{Q})$ contained in $\mathit{EntQ}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S}))$ or $\mathit{EntQ}_{seq}[\trianglelefteq](\mathcal{S})$, but not in both sets. We have two cases:

If $\sigma(q_i) \in \mathit{EntQ}_{seq}[\trianglelefteq](\mathcal{S})$ and $\sigma(q_i) \notin \mathit{EntQ}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S}))$, then let $C \in \mathit{StCens}_{seq}[\trianglelefteq](\mathcal{S})$ and $C' \in \mathit{StCens}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S}))$, we have that:

- $\mathcal{T} \cup C \models q$ iff $\mathcal{T} \cup C' \models q$, for each q that precedes $\sigma(q_i)$ in $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E},\mathcal{Q})$;
- $\mathcal{T} \cup C \models \sigma(q_i)$ but $\mathcal{T} \cup C' \not\models \sigma(q_i)$.

This means that C is more cooperative with respect to $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E},\mathcal{Q})$ than C'; but by Lemma 4, C' is maximally cooperative with respect to $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E},\mathcal{Q})$, so we get a contradiction.

On the other hand, if $\sigma(q_i) \in \mathit{EntQ}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S}))$ and $\sigma(q_i) \notin \mathit{EntQ}_{seq}[\trianglelefteq](\mathcal{S})$, then let $C \in \mathit{StCens}(\mathsf{BoolState}[\trianglelefteq](\mathcal{S}))$ and $C' \in \mathit{StCens}_{seq}[\trianglelefteq](\mathcal{S})$, we have that:

(*i*) for each $j < i$, $\mathit{cert}(\mathcal{T} \cup C, q_j) = \mathit{cert}(\mathcal{T} \cup C', q_j)$;
(*ii*) for each τ such that $\sigma \lhd_{q_i} \tau$, $\mathcal{T} \cup C \models \tau(q_i)$ iff $\mathcal{T} \cup C' \models \tau(q_i)$
(*iii*) $\mathcal{T} \cup C \models \sigma(q_i)$ and $\mathcal{T} \cup C' \not\models \sigma(q_i)$

Condition (*i*) implies that for each $j < i$, $C' \sim_{q_j} C$, while Conditions (*ii*) and Conditions (*iii*), by Definition 15, imply that $C' \prec_{q_i} C$. Therefore, by construction $C' \notin \mathit{StCens}[\preceq](\mathcal{S})$, which is a contradiction. ■

6.3. Not every MC-CQE semantics is an instance of dynCQE[$\preceq$]

We conclude this section by showing that there exists an MC-CQE semantics that cannot be simulated by dynCQE[$\preceq$], for any family $\preceq$ of preferences.

Let $\trianglelefteq$ be a family of total orders as in Section 6.2 and consider a new family of total orders $\trianglelefteq'$ defined as follows: for each UCQ q and for each $\sigma, \tau \in \mathfrak{S}_q$, we set $\sigma \trianglelefteq'_q \tau$ iff $\tau \trianglelefteq_q \sigma$.

Now, given $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$, we define the following function:

$$\mathbf{cqe}(\langle \mathcal{E},\mathcal{Q}\rangle) = \begin{cases} \mathsf{dynCQE}_{seq}[\trianglelefteq'](\langle \mathcal{E},\mathcal{Q}\rangle) & \text{if } |\mathcal{A}| \text{ is even,} \\ \mathsf{dynCQE}_{seq}[\trianglelefteq](\langle \mathcal{E},\mathcal{Q}\rangle) & \text{otherwise} \end{cases}$$

Proposition 16. ***cqe*** *is an MC-CQE semantics.*

Proof. Let $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ be a CQE instance, $\mathcal{Q} = \langle q_1, \dots, q_n \rangle$ a sequence of UCQs and $\mathcal{S}$ the protection state $\langle \mathcal{E},\mathcal{Q}\rangle$; we want to prove that there exists a censor $C \in \mathit{Cens}(\mathcal{E})$ which is maximally cooperative with respect to $\mathcal{Q}$ and such that $\mathbf{cqe}(\mathcal{S}, q_i) = \mathit{cert}(\mathcal{T} \cup C, q_i)$ for all $i \in [1,n]$. By definition of **cqe**, we have two cases: if $\mathcal{E}$ has an even number of ABox assertions, then $\mathbf{cqe}(\mathcal{S}) = \mathsf{dynCQE}_{seq}[\trianglelefteq'](\mathcal{S})$; otherwise, $\mathbf{cqe}(\mathcal{S}) = \mathsf{dynCQE}_{seq}[\trianglelefteq](\mathcal{S})$. In both cases, by Theorem 2, there exists $C \in \mathit{Cens}(\mathcal{E})$ which is maximally cooperative with respect to $\mathcal{Q}$ such that $\mathbf{cqe}(\mathcal{S}, q_i) = \mathit{cert}(\mathcal{T} \cup C, q_i)$ for all $i \in [1,n]$, hence we have the thesis. ■

Finally, we show that the semantics we have just defined is not a semantics of the dynCQE[$\preceq$] family.

Proposition 17. *There exists no family of total orders $\preceq$ such that for each protection state $\langle \mathcal{E},\mathcal{Q}\rangle$,* ***cqe***$(\langle \mathcal{E},\mathcal{Q}\rangle) = \mathsf{dynCQE}[\preceq](\langle \mathcal{E},\mathcal{Q}\rangle)$.

Proof. Consider the CQE instances $\mathcal{E}_1 = \langle \emptyset, \mathcal{P}, \mathcal{A}_1 \rangle$ and $\mathcal{E}_2 = \langle \emptyset, \mathcal{P}, \mathcal{A}_2 \rangle$, where:

$$\mathcal{P} = \{P(a) \wedge P(b) \rightarrow \bot\};$$

$$\mathcal{A}_1 = \{P(a), P(b), Q(c)\};$$

$$\mathcal{A}_2 = \{P(a), P(b)\}.$$

Consider also the query $q(x) = P(x)$ and, without loss of generality, assume that

$$\{x \leftarrow a\} \lhd_q \{x \leftarrow b\}.$$

Then, by construction, we have:

$$\mathbf{cqe}(\mathcal{S}_1, q) = \mathsf{dynCQE}_{seq}[\trianglelefteq](\mathcal{S}_1, q) = \{\{x \leftarrow b\}\},$$

$$\mathbf{cqe}(\mathcal{S}_2, q) = \mathsf{dynCQE}_{seq}[\trianglelefteq'](\mathcal{S}_2, q) = \{\{x \leftarrow a\}\},$$

where $\mathcal{S}_1 = \langle \mathcal{E}_1, \langle q \rangle \rangle$ and $\mathcal{S}_2 = \langle \mathcal{E}_2, \langle q \rangle \rangle$. Therefore, $\mathbf{cqe}(\mathcal{S}_1) \neq \mathbf{cqe}(\mathcal{S}_2)$.

It remains to show that, for any family of total orders $\preceq$, $\mathsf{dynCQE}[\preceq](\mathcal{E}, Q)$ behaves the same on $\mathcal{S}_1$ and $\mathcal{S}_2$. The optimal censors for $\mathcal{E}_1$ are the following:

$$C_1 = \{P(a), Q(c)\}$$

$$C_2 = \{P(b), Q(c)\}$$

while the optimal censors for $\mathcal{E}_2$ are the following:

$$C_1' = \{P(a)\}$$

$$C_2' = \{P(b)\}.$$

Observe that $cert(C_1, q) = cert(C_1', q)$ and $cert(C_2, q) = cert(C_2', q)$, therefore, by Definition 12, $C_1 \preceq_q C_2$ iff $C_1' \preceq_q C_2'$. Subsequently, $\mathsf{dynCQE}[\preceq](\mathcal{S}_1) = \mathsf{dynCQE}[\preceq](\mathcal{S}_2)$. ■

7. Open query evaluation under any MC-CQE semantics

In this section, we provide general data complexity lower bounds for the decision problems REC[**cqe**, **FullCQ**] and REC[**cqe**, **CQ**], which hold for any MC-CQE semantics **cqe** (Definition 6).

We start by proving that we can never recover FO rewritability for open queries, even when the query language is reduced to **FullCQ**. Then, we give a proof of intractability for the case of CQs (and thus even for UCQs). Note that in both cases lower bounds hold even if the TBox is assumed to be empty, and thus the results hold for every DL.

7.1. Data complexity for full CQs

To prove the following result, we need some preliminary notions from descriptive complexity.[4] Given an FO signature Σ, we denote by STRUCT[Σ] the set of finite FO interpretations over Σ. Let $\mathcal{I} \in$ STRUCT[Σ], we denote by $\mathsf{dom}(\mathcal{I})$ $\mathcal{I}$'s domain. As usual, given a predicate symbol P or a constant symbol c in Σ, we denote with $P^{\mathcal{I}}$ and $c^{\mathcal{I}}$ the interpretation of P and c according to $\mathcal{I}$. Let k be a positive natural number, $\varphi(x_1, \dots, x_k)$ a FO formula over the signature Σ and let $a_1, \dots, a_k \in \mathsf{dom}(\mathcal{I})$, we shall use the notation $\mathcal{I} \models \varphi[a_1, \dots, a_k]$ to mean that the tuple $\langle a_1, \dots, a_k\rangle$ satisfies φ in $\mathcal{I}$. Moreover, we denote by $\varphi(\mathcal{I})$ the set of all k-ples $\vec{v}$ of elements of $\mathsf{dom}(\mathcal{I})$ that satisfy φ in $\mathcal{I}$. Let X be a subset of $\mathsf{dom}(\mathcal{I})^k$, we say that X is *FO definable in $\mathcal{I}$ by φ* if $X = \varphi(\mathcal{I})$. Additionally, given an element $\vec{v}$ of $\mathsf{dom}(\mathcal{I})^k$, we say that $\vec{v}$ is *FO definable in $\mathcal{I}$ by φ* if $\{\vec{v}\}$ is FO definable in $\mathcal{I}$ by φ. From now on, as done in [25], we assume that every signature Σ contains the constant symbols $\bar{0}, \bar{1}, max$ and the binary predicates $\leq$ and BIT. Furthermore, for every signature Σ, we only consider FO interpretations $\mathcal{I} \in$ STRUCT[Σ] such that $\mathsf{dom}(\mathcal{I}) = \{0, \dots, n\}$, for some positive natural number n, $\leq^{\mathcal{I}}$ is the canonical total order over natural numbers, the constant symbols $\bar{0}$ and $\bar{1}$ are interpreted as the natural numbers 0 and 1 respectively, and max is interpreted as the natural number n (the maximum according to $\leq$). Moreover, the binary predicate BIT is interpreted as follows: $(a, b) \in BIT^{\mathcal{I}}$ iff the b-th digit (starting from the least significant bit) in the binary representation of a is 1.

Now we consider a notion of reducibility between decision problems over FO interpretations. Let Σ_1, Σ_2 be two signatures and let $D_1 \subseteq$ STRUCT[Σ_1] and $D_2 \subseteq$ STRUCT[Σ_2], a *first-order reduction from D_1 to D_2* is a function R : STRUCT[Σ_1] → STRUCT[Σ_2] such that there exist:

- a FO formula φ^R_{dom} over Σ_1 with k free variables,
- a FO formula φ^R_P over Σ_1 with $k \cdot r$ free variables for each predicate symbol P of arity r in Σ_2,
- a FO formula φ^R_c over Σ_1 with k free variables for each constant symbol c in Σ_2,

such that for each $\mathcal{I} \in$ STRUCT[Σ_1]:

- $\mathcal{I} \in D_1$ iff $R(\mathcal{I}) \in D_2$;
- the domain of $R(\mathcal{I})$ is FO definable in $\mathcal{I}$ by φ^R_{dom}, and so:

$$\mathsf{dom}(R(\mathcal{I})) = \{\vec{v} \in \mathsf{dom}(\mathcal{I})^k \mid \mathcal{I} \models \varphi^R_{\mathsf{dom}}[v_1, \dots, v_k]\}$$

- for each predicate symbol P of arity r in Σ_2, its interpretation $P^{R(\mathcal{I})}$ according to $R(\mathcal{I})$ is defined as follows

$$P^{R(\mathcal{I})} = \{(\vec{v}_1, \dots, \vec{v}_r) \in \mathsf{dom}(R(\mathcal{I}))^r \mid \mathcal{I} \models \varphi^R_P[v_{11}, \dots, v_{1k}, \dots, v_{r1}, \dots, v_{rk}]\}$$

- for each constant symbol c in Σ_2, its interpretation $c^{R(\mathcal{I})}$ according to $R(\mathcal{I})$ is FO definable in $\mathcal{I}$ by φ^R_c.

To prove the next theorem, we shall use the decision problem MAJORITY, which is defined as follows:

[4] For an in-depth discussion, refer to [25].

Input: A binary string B
Question: Is the number of occurrences of the zero bit greater than or equal to the number of occurrences of the one bit in B?

Theorem 3. *There exists no MC-CQE semantics* **cqe** *such that REC[*$\mathbf{cqe}, \mathbf{FullCQ}$*] is in AC*^{0} *in data complexity, even in the case the TBox is empty.*

Proof. We start by proving that for each MC-CQE semantics **cqe**, we can reduce MAJORITY to REC[$\mathbf{cqe}, \mathbf{FullCQ}$] using a fixed policy and a fixed sequence of queries.

Given a binary string B of length n, we denote by $\mathcal{O}_B$ the set of all positions in which the digit 1 occurs in B and by $\mathcal{Z}_B$ the set of all positions in which the digit 0 occurs in B. Then, for each $i \in [1, n]$, we consider an individual i representing the i-th position inside the string. Moreover, we use a role noteq to encode the binary inequality predicate between positions and a role E that will yield at the end of the query processing a (possibly partial) injective function from $\mathcal{O}_B$ to $\mathcal{Z}_B$. Finally, we use the concept $\mathsf{unselected}$ to mark the positions $\mathcal{O}_B$ that are not contained in the domain of E and the ground atom $\mathsf{majority}(1)$ to signal whether the string being encoded in the ABox is an instance of MAJORITY. More specifically, we define an ABox $\mathcal{A}_B$ containing the following assertions:

- $\mathsf{E}(\mathsf{i},\mathsf{j})$, for each position $i \in \mathcal{O}_B$ and $j \in \mathcal{Z}_B$;
- $\mathsf{noteq}(\mathsf{i},\mathsf{j})$ for all positions i, j such that $i \neq j$;
- $\mathsf{unselected}(\mathsf{i})$ for each position $i \in \mathcal{O}_B$;
- $\mathsf{majority}(1)$.

Now, we define the following fixed policy $\mathcal{P}$:

1. $\exists x, y, z\, (\mathsf{E}(x, y) \wedge \mathsf{E}(x, z) \wedge \mathsf{noteq}(y, z)) \rightarrow \bot$
2. $\exists x, y, z\, (\mathsf{E}(x, z) \wedge \mathsf{E}(y, z) \wedge \mathsf{noteq}(x, y)) \rightarrow \bot$
3. $\exists x, y\, (\mathsf{unselected}(x) \wedge \mathsf{E}(x, y)) \rightarrow \bot$
4. $\exists x\, (\mathsf{majority}(1) \wedge \mathsf{unselected}(x)) \rightarrow \bot$

The first denial requires that the role E be a function, whereas the second denial states that E is injective. The third denial enforces that the positions satisfying $\mathsf{unselected}$ are only those not in the domain of E, and finally, the last denial requires that $\mathsf{majority}(1)$ holds iff there is no individual for which $\mathsf{unselected}$ holds.

Then, consider the fixed sequence of queries $\mathcal{Q}$, where:

$$q_1(x, y) = \mathsf{noteq}(x, y)$$

$$q_2(x, y) = \mathsf{E}(x, y)$$

$$q_3(x) = \mathsf{unselected}(x)$$

$$q_4 = \mathsf{majority}(1)$$

Given any MC-CQE semantics **cqe**, let us prove that for each binary string B, $B \in$ MAJORITY iff q_4 is a consequence of the state $\mathcal{S} = \langle\langle\emptyset, \mathcal{A}_B, \mathcal{P}\rangle, \mathcal{Q}\rangle$ under **cqe**.

Assume that B is a binary string such that $B \in$ MAJORITY. Let $\mathcal{C}$ be any maximally cooperative censor for $\mathcal{Q}$. Since query q_1 does not trigger any denial, maximal cooperativity forces $\mathcal{C}$ to contain all the facts regarding noteq. Then, when considering query q_2, because of Denials 1 and 2, the facts regarding E contained in $\mathcal{C}$ will represent a partial injective function f from elements of $\mathcal{O}_B$ to elements of $\mathcal{Z}_B$. Let us prove that f is total. Assume by contradiction that there exists a position $i \in \mathcal{O}_B$ which is not in the domain of f, $dom(f)$. This means that the cardinality of $dom(f)$ is strictly smaller than the cardinality of $\mathcal{O}_B$. By assumption, $B \in$ MAJORITY, hence the cardinality of $\mathcal{O}_B$ is less than or equal to the cardinality of $\mathcal{Z}_B$, therefore we can deduce that the cardinality of $dom(f)$ is strictly smaller than the cardinality of $\mathcal{Z}_B$. This means that there exists a position $j \in \mathcal{Z}_B$ which does not appear in the range of f. So, if i and j are the individuals respectively representing the two positions i and j, $(\mathcal{C} \setminus \{\mathsf{unselected}(\mathsf{i})\}) \cup \{\mathsf{E}(\mathsf{i},\mathsf{j})\}$ is still a censor and is more cooperative than $\mathcal{C}$ with respect to $\mathcal{Q}$. This is absurd. Therefore, f is total and hence $cert(\mathcal{C}, q_3) = \emptyset$ by Denial 3. Then, by maximal cooperativity, we have that $\mathcal{C}$ satisfies $q_4 = \mathsf{majority}(1)$.

Finally, let **cqe** be an MC-CQE semantics, by Definition 6, there exists a maximally cooperative censor $\mathcal{C}$ for $\mathcal{Q}$ such that for each q_i in $\mathcal{Q}$, $\mathbf{cqe}(\langle\langle\emptyset, \mathcal{A}_B, \mathcal{P}\rangle, \mathcal{Q}\rangle, q_i) = cert(\mathcal{C}, q_i)$. By what we proved above, $\mathbf{cqe}(\mathcal{S}, q_4) = cert(\mathcal{C}, q_4) = \{\epsilon\}$.

For the other way round, assume that q_4 is a consequence of the state $\mathcal{S}$ under an MC-CQE semantics **cqe**. By Denial 4 and the fact that **cqe** is secure, we have that $\mathbf{cqe}(\mathcal{S}, q_3)$ must be empty. On the other hand, maximal cooperativity and Denial 3 imply that the answer set $\mathbf{cqe}(\mathcal{S}, q_2)$ represents a *total* injective function from $\mathcal{O}_B$ to $\mathcal{Z}_B$ and therefore, the cardinality of $\mathcal{O}_B$ is less than or equal to the cardinality of $\mathcal{Z}_B$, i.e., $B \in$ MAJORITY.

We can now prove our thesis. First, observe that an ABox can be naturally represented as its least Herbrand model, which is a finite FO interpretation. Conversely, every finite FO interpretation is the least Herbrand model of an ABox in a canonical way. So, we can see REC[$\mathbf{cqe}, \mathbf{FullCQ}$] as a decision problem over a class of FO interpretations. We can do the same with MAJORITY too. A binary string can be represented as an FO interpretation over the signature Σ_B containing a unary predicate O marking which positions in the string contain the bit 1. Now we prove that MAJORITY is first-order reducible to REC[$\mathbf{cqe}, \mathbf{FullCQ}$]. Given a binary string $B \in \mathrm{STRUCT}[\Sigma_B]$, we can define the FO interpretation $\mathcal{I}_B$ over a signature containing the role names E and noteq, and the concept names $\mathsf{unselected}$ and $\mathsf{majority}$ as follows:

- The domain of $\mathcal{I}_B$ is the same as the domain of B, so it is FO definable in B by the following formula:
$$\varphi_{\mathsf{dom}}(i) := (i = i)$$
- E is FO definable in B by the following formula:
$$\varphi_{\mathsf{E}}(i, j) := \neg O(i) \wedge O(j)$$
- noteq is FO definable in B by the following formula:
$$\varphi_{\mathsf{noteq}}(i, j) := \neg(i = j)$$
- $\mathsf{unselected}$ is FO definable in B by the following formula:
$$\varphi_{\mathsf{unselected}}(i) := O(i)$$
- $\mathsf{majority}$ is FO definable in B by the following formula:
$$\varphi_{\mathsf{majority}}(i) := (i = \bar{1})$$

Observe that $\mathcal{I}_B$ is the least Herbrand model of $\mathcal{A}_B$. Therefore, combining this observation with what we have proved above, we have defined a first-order reduction from MAJORITY to REC[**cqe**, **FullCQ**]. To conclude, we use the following property of AC^0 proved in [25]: Let Σ_1, Σ_2 be two FO signatures, let $D_1 \subseteq \mathrm{STRUCT}[\Sigma_1]$ and $D_2 \subseteq \mathrm{STRUCT}[\Sigma_2]$ two decision problems. If D_1 is first-order reducible to D_2 and $D_2 \in \mathrm{AC}^0$, then $D_1 \in \mathrm{AC}^0$ too. Moreover, in [25] it is proved that MAJORITY is not in AC^0, therefore, we deduce that REC[**cqe**, **FullCQ**] is not in AC^0. ■

7.2. Data complexity for CQs and full UCQs

In the following theorem, we will use the problem PARITY(SAT), which is defined as follows:

Input: A sequence of m Boolean formulas $\Phi = \langle \phi_1, \ldots \phi_m \rangle$ in 3-CNF
Question: Is the number of satisfiable formulas occurring in Φ even?

The problem is $\Delta_2^p[O(\log n)]$-complete even if we assume that the formulas of the input sequence do not have any common variable and that the input sequence is such that the unsatisfiability of ϕ_i implies the unsatisfiability of every ϕ_j such that $j > i$ [26].[5] So, we will make such assumptions on Φ.

Theorem 4. *Let* **cqe** *be an MC-CQE semantics, then REC[***cqe**, **CQ***] is $\Delta_2^p[O(\log n)]$-hard in data complexity, even in case the TBox is empty.*

Proof. Since $\Delta_2^p[O(\log n)]$ is closed by complement, the complementary problem of PARITY(SAT), called coPARITY(SAT), is still $\Delta_2^p[O(\log n)]$-complete, so we prove the theorem by reducing coPARITY(SAT) to REC[**cqe**, **CQ**] using a fixed policy and a fixed sequence of queries.

Let $\Phi = \langle \phi_1, \ldots, \phi_m \rangle$ be a sequence of 3-CNF formulas. As we noted above, we can assume that the formulas of Φ don't have any common variable and that the unsatisfiability of ϕ_i implies the unsatisfiability of every ϕ_j such that $j > i$.

Let us define an ABox $\mathcal{A}_\Phi$. First we add to $\mathcal{A}_\Phi$ the facts:

- $\mathsf{NextF(i, j)}$, for each i, j such that $1 \le i < j \le m$ and $j = i + 1$;
- $\mathsf{Even(i)}$, for each even i;

using the concept Even and the role NextF and using for each natural number $i \in [1, m]$ an individual name i.

Then, let the 3-CNF formula $\varphi_j = \bigwedge_{i=1}^{k_j} c_i^j$ be the j-th formula of the input sequence. We encode φ_j in a set of facts as follows: Each clause c_i^j in φ_j is referred to by an individual name c_i^j and each variable x appearing in φ_j corresponds to an individual name x. Then, we use the roles $\mathsf{cvar}_1, \mathsf{cvar}_2, \mathsf{cvar}_3$ to express the variables contained in each clause, and the roles $\mathsf{cneg}_1, \mathsf{cneg}_2, \mathsf{cneg}_3$ to encode where the negations appear in the clause. Furthermore, given a literal l, we denote with $var(l)$ the individual representing the variable occurring in l. Moreover, we use the individual name 1 to refer to the *true* truth value and the individual name 0 to refer to the *false* truth value, and a function $negated(l)$ indicating whether l is negated or not, that is, $negated(l)$ yields the individual 0 if l is positive, and the individual 1 otherwise. Note that $negated(l)$ returns the truth assignment of $var(l)$ that makes l false.

[5] Note that this assumption requires checking only that the index of the last satisfiable formula is even, rather than counting the total number of satisfiable formulas.

For each clause $c_i^j = l_{i,j}^1 \vee l_{i,j}^2 \vee l_{i,j}^3$, we add the following facts to $\mathcal{A}_\Phi$:

- $\mathsf{Seq}(\mathsf{j}, \mathsf{c}_i^j)$
- $\mathsf{cvar}_s(\mathsf{c}_i^j, var(l_{i,j}^s))$
- $\mathsf{cneg}_s(\mathsf{c}_i^j, negated(l_{i,j}^s))$

where $s \in \{1,2,3\}$.

As we observed above, if $\mathsf{cneg}_s(\mathsf{c}_i^j, t)$ has been added to $\mathcal{A}_\Phi$, then t (either 0 or 1) represents the truth value we should assign to $var(l_{i,j}^s)$ to make $l_{i,j}^s$ false.

Then, for each $j \in [1,m]$ and for each variable x occurring in φ_j we include in $\mathcal{A}_\Phi$ the following facts:

- $\mathsf{V}(\mathsf{x}, 1), \mathsf{V}(\mathsf{x}, 0)$
- $\mathsf{Undef}(\mathsf{j}, \mathsf{x}), \mathsf{Sat}(\mathsf{j})$

The first two facts encode the two possible truth assignments for each variable. While the atom $\mathsf{Undef}(\mathsf{j}, \mathsf{x})$ represents the fact that it was not possible to consistently assign a truth value for x in φ_j, vice versa the atom $\mathsf{Sat}(\mathsf{j})$ represents the fact that φ_j is satisfiable.

Lastly, we include the following fact:

- parity-sat(0)

which is used to represent the fact that the input sequence of formulas is not an instance of PARITY(SAT).

Now, we define the following fixed policy $\mathcal{P}$:

1. $\exists v\,(\mathsf{V}(v,1) \wedge \mathsf{V}(v,0)) \rightarrow \bot$
2. $\exists c, v_1, v_2, v_3, t_1, t_2, t_3\,(\bigwedge_{i=1}^{3}(\mathsf{cvar}_i(c, v_i) \wedge \mathsf{cneg}_i(c, t_i) \wedge \mathsf{V}(v_i, t_i))) \rightarrow \bot$
3. $\exists j, v, t\,(\mathsf{Undef}(j, v) \wedge \mathsf{V}(v, t)) \rightarrow \bot$
4. $\exists j, v\,(\mathsf{Undef}(j, v) \wedge \mathsf{Sat}(j)) \rightarrow \bot$
5. $\exists j, j', x\,(\mathsf{Sat}(j) \wedge \mathsf{NextF}(j, j') \wedge \mathsf{Undef}(j', x) \wedge \mathsf{Even}(j) \wedge \text{parity-sat}(0)) \rightarrow \bot$

Informally, the first denial will be used to force a choice between $\mathsf{V}(\mathsf{x}, 1)$ and $\mathsf{V}(\mathsf{x}, 0)$, for each variable individual x, which corresponds to assigning either *true* or *false* to each propositional variable x in the corresponding formula of the sequence (remember that by assumption each variable occurs in at most one formula). The second denial allows us to restrict the attention only to those assignments that do not falsify any of the clauses c in the input formulas. Furthermore, the third denial expresses the fact that if an assignment has been found for a variable of a formula φ_j then it cannot be considered undefined. The fourth denial asserts that a formula φ_j is satisfiable iff it does not contain undefined variables. Finally, the fifth denial will be used to enforce that parity-sat(0) holds only when the last satisfiable formula in the input sequence Φ is odd.

Then, we define the following fixed sequence of CQs $\mathcal{Q}$:

$$q_1(c, v) = \mathsf{cvar}_1(c, v)$$

$$q_2(c, v) = \mathsf{cvar}_2(c, v)$$

$$q_3(c, v) = \mathsf{cvar}_3(c, v)$$

$$q_4(c, t) = \mathsf{cneg}_1(c, t)$$

$$q_5(c, t) = \mathsf{cneg}_2(c, t)$$

$$q_6(c, t) = \mathsf{cneg}_3(c, t)$$

$$q_7(x) = \exists t\, \mathsf{V}(x, t)$$

$$q_8(j, x) = \mathsf{Undef}(j, x)$$

$$q_9(j, j', x) = \mathsf{Sat}(j) \wedge \mathsf{NextF}(j, j') \wedge \mathsf{Undef}(j', x) \wedge \mathsf{Even}(j)$$

$$q_{10} = \text{parity-sat}(0).$$

Now, let $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ be a protection state of the CQE instance $\mathcal{E} = \langle \emptyset, \mathcal{A}_\Phi, \mathcal{P} \rangle$, and let us prove that, for every maximally cooperative censor C for $\mathcal{S}$, $C \models q_{10}$ iff Φ is an instance of coPARITY(SAT). Let C be any maximally cooperative censor for $\mathcal{S}$. Queries $q_1, \ldots, q_6$ do not trigger any denial, and so they can be answered honestly. This forces C to keep the facts of the ABox $\mathcal{A}$ relative to the roles $\mathsf{cvar}_1, \mathsf{cvar}_2, \mathsf{cvar}_3, \mathsf{cneg}_1, \mathsf{cneg}_2, \mathsf{cneg}_3$, thus fixing the structure of the input sequence of formulas Φ. When evaluating query q_7, because of the first two denials, C contains a maximal set of variables for which we can find a truth value which does not make any of the clauses in the input sequence false. Here the assumption that each variable occurs at most in one input formula is crucial, because it implies the absence of any interaction between distinct formulas, so assigning a truth value to as many variables as possible means trying to satisfy as many formulas of the sequence as possible. In other words, for each formula $\varphi_j \in \Phi$, the answer set for q_7 contains all the individuals corresponding to the variables of φ_j iff the latter is satisfiable. Now, due to the form of the third denial, after the execution of query q_8 the censor C contains a fact of the form $\mathsf{Undef}(\mathsf{j}, \mathsf{x})$ only for the variables x that have not received a truth assignment. In other terms, at least one such a fact will belong to C iff ϕ_j is unsatisfiable. Consequently, query $cert(C, q_9) \neq \emptyset$ iff the last satisfiable formula in the sequence Φ is even. Therefore, because of the fifth denial, query q_{10} holds in C (i.e., $cert(C, q_{10}) = \{\epsilon\}$) iff the last satisfiable formula in the sequence Φ is not even.

Finally, let **cqe** be any MC-CQE semantics. By definition, there exists a maximally cooperative censor C for S such that, for each query q' in Q, $\mathbf{cqe}(S, q') = cert(C, q')$. Therefore, by what we proved above we can conclude that $\mathbf{cqe}(S, q_{10}) = \{\epsilon\}$ iff the last satisfiable formula in the sequence Φ is not even (that is, there is an odd number of satisfiable formulas in Φ, given the initial assumption on Φ). ■

Theorem 5. *Let **cqe** be an MC-CQE semantics, then REC[**cqe**, **FullUCQ**] is $\Delta_2^p[O(\log n)]$-hard in data complexity, even in case the TBox is empty.*

Proof. We can re-use the same construction used in Theorem 4: in fact, all the queries used, except for query q_7, are also full UCQs. However, it is easy to verify that query q_7 can be replaced with the full UCQ $q_7'(x) = V(x, 1) \vee V(x, 0)$ without affecting the validity of the proof. ■

8. Open query evaluation under dynCQE[⪯] with tractable preferences

In this section, we show that the complexity of REC[dynCQE[⪯], **CQ**] lies between $\Delta_2^p[O(\log n)]$ and Σ_2^p, under the reasonable assumption that the total preference orderings $\leq_q$ are computable in polynomial time.

The lower bound follows immediately from Theorem 2 and Theorem 5; it does not depend on the complexity of the preference orderings.

Corollary 4. *For all families ⪯ of preferences over censors (Definition 12), REC[*dynCQE[⪯], ***CQ***] *is $\Delta_2^p[O(\log n)]$-hard in data complexity. The same holds for **FullUCQ**.*

Moreover, in the next section, we will prove that with tractable lexicographic orderings, REC[dynCQE[⪯], **FullCQ**] is PTIME-hard.

The upper bound can be proved as follows:

Theorem 6. *For all families ⪯ of polynomial-time computable preferences over censors (Definition 12), REC[*dynCQE[⪯], ***UCQ***] *is in Σ_2^p in data complexity.*

Proof. Let $\langle S, \sigma\rangle$ be any instance of REC[dynCQE[⪯], **UCQ**], where $S = \langle \mathcal{E}, Q\rangle$, $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A}\rangle$, $Q = \langle q_1, \dots, q_n\rangle$, and $\sigma \in \mathfrak{S}_{q_n}$.

Recall that the number of ground atoms that can be built using the symbols in $\mathcal{T}$, $\mathcal{P}$, and $\mathcal{A}$ is polynomial in the size of $\mathcal{A}$, so the same bound applies to the size of $\mathsf{cl}_{\mathcal{T}}(\mathcal{A})$ and all the censors for $\mathcal{E}$.

Claim 1: Checking whether a set of ground atoms C belongs to $OptCens(\mathcal{E})$ is in PTIME.

Indeed, $C \in OptCens(\mathcal{E})$ iff $C \subseteq \mathsf{cl}_{\mathcal{T}}(\mathcal{A})$, $\mathcal{T} \cup C \cup \mathcal{P}$ is consistent, and for all ground atoms α built with the symbols in $\mathcal{E}$, $\mathcal{T} \cup C \cup \{\alpha\} \cup \mathcal{P}$ is inconsistent. Since UCQ answering in DL-Lite$_{\mathcal{R}}$ is in PTIME in data complexity and the above checks involve a polynomial number of UCQs, Claim 1 is proved.

Moreover, recall that – by the construction in Definition 13 – all the censors in $StCens[\preceq](S_i)$ answer $q_1, \dots, q_i$ in the same way (because the orderings $\leq_{q_i}$ on the answer sets are total) and we have:

Claim 2: For every integer $i \in [1, n]$ and every $C \in StCens[\preceq](S_i)$, $StCens[\preceq](S_i)$ equals the set of all $C' \in OptCens(\mathcal{E})$ such that $cert(\mathcal{T} \cup C', q_j) = cert(\mathcal{T} \cup C, q_j)$ for every integer $j \in [1, i]$.

Claim 3: For all $C \in StCens[\preceq](S_{i-1})$ ($i \in [1, n-1]$), checking whether $C \notin StCens[\preceq](S_i)$ is in NP.

Claim 3 can be proved by the following algorithm: guess a set of ground atoms C'; check whether $C' \in OptCens(\mathcal{E})$; check whether $cert(\mathcal{T} \cup C', q_j) = cert(\mathcal{T} \cup C, q_j)$ for all $j \in [1, i]$; check whether $cert(\mathcal{T} \cup C, q_i) <_{q_i} cert(\mathcal{T} \cup C', q_i)$ (in polynomial time, by hypothesis); if all these tests succeed, then accept, otherwise reject. By the hypothesis on C, Claim 2, and the definition of $\prec_{q_i}$, the above algorithm accepts iff there exists $C' \in StCens[\preceq](S_{i-1})$ such that $C \prec_{q_i} C'$, which is equivalent to $C \notin \max_{\preceq_{q_i}} StCens[\preceq](S_{i-1}) = StCens[\preceq](S_i)$. Moreover, by Claim 1 and the tractability of ground atomic queries in DL-Lite$_{\mathcal{R}}$ in data complexity, the algorithm runs in polynomial time. This proves Claim 3.

We are now ready to prove the theorem. REC[dynCQE[⪯], **UCQ**] can be decided in nondeterministic polynomial time using an oracle for NP as follows: First guess a set of ground atoms C; check whether $C \in OptCens(\mathcal{E}) = StCens[\preceq](S_0)$ (in polynomial time, by Claim 1), then for all $i \in [1, n]$ check whether $C \in StCens[\preceq](S_i)$ using the oracle (by Claim 3). Note that $C \in StCens[\preceq](S_n)$ iff all the above tests succeed. The algorithm accepts the input iff the above tests succeed and, moreover, $\sigma \in cert(\mathcal{T} \cup C, q_n)$; this final test takes polynomial time in DL-Lite$_{\mathcal{R}}$. It follows immediately that the above algorithm decides REC[dynCQE[⪯], **UCQ**] in polynomial time, therefore REC[dynCQE[⪯], **UCQ**] is in Σ_2^p. ■

In the following section, we prove tighter complexity bounds by exploiting the restrictions on the structure of preference orderings introduced in Definition 15.

9. Open query evaluation under $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ semantics

Now we focus on the complexity of $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ in the case where, for each UCQ q, $\trianglelefteq_q$ is decidable in logarithmic space.[6]

9.1. *The case when queries are UCQs*

We now provide a complete characterization of the complexity of reasoning over CQs, full UCQs, and UCQs under $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ semantics. In the following theorem, we will use the decision problem LEX-SAT, which is known to be Δ_2^p-complete [27]:

Input: A 3-CNF formula ϕ with propositional variables $x_1, \dots, x_n$
Question: Is ϕ satisfiable and x_n *true* in the lexicographically maximum assignment satisfying ϕ?[7]

Theorem 7. *REC*[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, ***CQ***] *is Δ_2^p-hard in data complexity, even in case the TBox is empty.*

Proof. Since Δ_2^p is closed under complement, the complementary problem of LEX-SAT (coLEX-SAT) is still Δ_2^p-complete, so we prove the theorem by reducing coLEX-SAT to REC[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, **CQ**] using a fixed privacy policy and a fixed sequence of queries. Let φ be a 3-CNF formula using variables $x_1, \dots, x_n$, we construct an ABox $\mathcal{A}_\varphi$ as follows: First, we consider the CQ $q(x) = \mathsf{LexLastV}(x)$ and the total order $\trianglelefteq_q$, then we pick n constants $\mathsf{x}_1, \dots, \mathsf{x}_n$ such that $\mathsf{x}_n \triangleleft_q \dots \triangleleft_q \mathsf{x}_1$. For each $i \in [1, n]$, x_i will represent the variable x_i in φ. Then, we pick two other constants 1 and 0 representing the truth and falsity, respectively, and for each clause c_i in φ, we pick a constant symbol c_i representing it. Furthermore, given a literal l of φ, we denote with $var(l)$ the individual representing the variable occurring in l. Moreover, $negated(l)$ indicates whether l is negated or not, that is, $negated(l)$ yields the individual 0 if l is positive, and the individual 1 otherwise. Note that $negated(l)$ returns the truth assignment of $var(l)$ that makes l false.

We then add the following ground atoms to $\mathcal{A}_\varphi$:

- $\mathsf{Undef}(\mathsf{x}_i)$, $\mathsf{V}(\mathsf{x}_i, 1)$, $\mathsf{V}(\mathsf{x}_i, 0)$, $\mathsf{LexLastV}(\mathsf{x}_i)$ for each $i \in [1, n]$.
- $\mathsf{L}(\mathsf{x}_n)$.
- $\mathsf{lex\text{-}sat}(0)$,

and for each clause $c_i = l_i^1 \vee l_i^2 \vee l_i^3$, we add the following ground atoms:

- $\mathsf{cvar}_1(\mathsf{c}_i, var(l_i^1))$
- $\mathsf{cneg}_1(\mathsf{c}_i, negated(l_i^1))$
- $\mathsf{cvar}_2(\mathsf{c}_i, var(l_i^2))$
- $\mathsf{cneg}_2(\mathsf{c}_i, negated(l_i^2))$
- $\mathsf{cvar}_3(\mathsf{c}_i, var(l_i^3))$
- $\mathsf{cneg}_3(\mathsf{c}_i, negated(l_i^3))$

Now, we define the following fixed policy $\mathcal{P}$:

1. $\exists v\, (\mathsf{V}(v, 1) \wedge \mathsf{V}(v, 0)) \to \bot$
2. $\exists v\, (\mathsf{LexLastV}(v) \wedge \mathsf{V}(v, 0)) \to \bot$
3. $\exists x, v_1, v_2, v_3, t_1, t_2, t_3\, (\bigwedge_{i=1}^{3} (\mathsf{cvar}_i(x, v_i) \wedge \mathsf{cneg}_i(x, t_i) \wedge \mathsf{V}(v_i, t_i))) \to \bot$
4. $\exists v, t\, (\mathsf{V}(v, t) \wedge \mathsf{Undef}(v)) \to \bot$
5. $\exists v_1, v_2\, (\mathsf{Undef}(v_1) \wedge \mathsf{L}(v_2)) \to \bot$
6. $\exists v\, (\mathsf{L}(v) \wedge \mathsf{V}(v, 1) \wedge \mathsf{lex\text{-}sat}(0)) \to \bot$

Informally, the first denial will be used to force to choose either $\mathsf{V}(\mathsf{x}_i, 1)$ or $\mathsf{V}(\mathsf{x}_i, 0)$, for each constant x_i, which corresponds to assigning either *true* or *false* to each propositional variable x_i of ϕ. The second denial enforces that all variables that belong to $\mathsf{LexLastV}$ must not be assigned the truth value 0. The third denial allows us to restrict the attention only to those assignments that satisfy ϕ. Furthermore, the fourth denial will be used to check whether there exists at least a satisfying assignment to ϕ, and the fifth denial will be used to discard $\mathsf{L}(\mathsf{x}_n)$ if ϕ is not satisfiable. Finally, the last denial will be used to enforce that whenever there exists a satisfying truth assignment for ϕ assigning 1 to x_n, then $\mathsf{lex\text{-}sat}(0)$ cannot hold.

Then, we define the following fixed sequence of queries $\mathcal{Q}$.

$$q_1(c, v) = \mathsf{cvar}_1(c, v)$$

[6] An example of such a family is the total order $\trianglelefteq_q$ in Example 12, provided that $\leq$ is decidable in logarithmic space.

[7] A truth assignment v of the propositional variables $x_1, \dots, x_n$ can be represented by a binary number m_v of length n, where the most significant bit is the truth value assigned to x_1, and so on. Then, the lexicographically maximum assignment satisfying ϕ is the assignment v, among those satisfying ϕ, that minimize m_v.

$q_2(c, v) = \mathsf{cneg}_1(c, v)$

$q_3(c, v) = \mathsf{cvar}_2(c, v)$

$q_4(c, v) = \mathsf{cneg}_2(c, v)$

$q_5(c, v) = \mathsf{cvar}_3(c, v)$

$q_6(c, v) = \mathsf{cvar}_3(c, v)$

$q_7(v) = \exists t\, \mathsf{V}(v, t)$

$q_8(v) = \mathsf{Undef}(v)$

$q_9(v) = \mathsf{LexLastV}(v)$

$q_{10}(v) = \mathsf{L}(v) \wedge \mathsf{V}(v, 1)$

$q_{11} = \mathsf{lex\text{-}sat}(0).$

We conclude the proof by showing that, given a 3-CNF formula ϕ with n variables, the last query q_{11} of the query sequence Q is entailed in the CQE instance $(\emptyset, \mathcal{P}, \mathcal{A}_\phi)$ under the semantics $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ if and only if ϕ is unsatisfiable or ϕ is satisfiable but x_n is *false* in the lexicographically maximum assignment satisfying ϕ.

After the first 6 queries, we can restrict the attention only to those censors that do not remove any "clause". Consider now query q_7. At this point, due to the first denial, we are forced to either remove $\mathsf{V}(\mathsf{x}_i, 1)$ or remove $\mathsf{V}(\mathsf{x}_i, 0)$ (or both) for each $i \in [1, n]$. Now there are two cases: either ϕ is satisfiable or ϕ is not satisfiable.

If ϕ is not satisfiable, then every surviving censor has removed both $\mathsf{V}(\mathsf{x}_i, 1)$ and $\mathsf{V}(\mathsf{x}_i, 0)$ for at least one x_i (to not violate the denials for the clauses). This means that, in each surviving censor, $\mathsf{Undef}(\mathsf{x}_i)$ is preserved for at least one x_i, which allows us to return at least one answer for q_8. However, due to the fifth denial, the ground atom $\mathsf{L}(\mathsf{x}_n)$ must be removed, and therefore q_{10} cannot return any answer. This means that the atom $\mathsf{lex\text{-}sat}(0)$ can be preserved and query q_{11} can be answered positively, as required.

If ϕ is satisfiable, then it is easy to see that all the tuples (x_i) for $i \in [1, n]$ will return as answers to query q_7. So, every surviving censor indicates a satisfying assignment to ϕ. It follows that $\mathsf{Undef}(\mathsf{x}_i)$ is not present in any of the surviving censors, allowing us to preserve $\mathsf{L}(\mathsf{x}_i)$. Now, consider query q_9. Because of the second denial, q_9 can hold only for those variables which have been assigned the truth value 1, therefore, since x_i comes before than x_{i+1}, for each $i \in [1, n-1]$, $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ will try to check whether there is a censor that assigns 1 to x_1, then if it's possible to assign 1 to x_2 and so on, therefore among the surviving censors (those that indicate a satisfying assignment for ϕ) it will be selected the one corresponding to the lexicographically maximum assignment to ϕ. Now, if such last satisfying assignment assigns *true* to x_n, then query q_{10} returns as answers the nonempty set $\{\{v \leftarrow \mathsf{x}_n\}\}$ and so, because of the sixth denial, query q_{11} must be answered negatively. On the other hand, if such a last satisfying assignment assigns *false* to x_n, then query q_{10} returns no answer, and so query q_{11} can be answered positively without triggering any denial. This allows us to conclude that q_{11} is answered positively iff the last satisfying assignment assigns *true* to x_n.

Finally, observe that since $\trianglelefteq_q$ is decidable in logarithmic space, it is clear that we can build $\mathcal{A}_\varphi$ in logarithmic space too. Therefore, the reduction can be carried in logarithmic space, and this proves that $\mathrm{REC}[\mathsf{dynCQE}_{seq}[\trianglelefteq], \mathbf{CQ}]$ is Δ_2^p-hard in data complexity. ■

Theorem 8. *REC*$[\mathsf{dynCQE}_{seq}[\trianglelefteq], \boldsymbol{\mathit{FullUCQ}}]$ *is* Δ_2^p*-hard in data complexity, even in case the TBox is empty.*

Proof. We can re-use the same construction used in Theorem 7: in fact, all the queries used, except for query q_7, are also full UCQs. However, it is easy to verify that query q_7 can be replaced with the full UCQ $q_7'(x) = V(x, 1) \vee V(x, 0)$ without affecting the validity of the proof. ■

In the next theorem, we will define an algorithm that decides $\mathrm{REC}[\mathsf{dynCQE}_{seq}[\trianglelefteq], \mathbf{UCQ}]$ and study its data complexity. To do so, consider the following decision problem $\boldsymbol{P}$.

Input: A DL-Lite$_\mathcal{R}$ CQE instance $\langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ and a set S of BUCQs
Question: Does there exist a censor C for $\langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ such that $\mathcal{T} \cup C \models q$ holds for each $q \in S$?

This problem is in NP in data complexity, where only the ABox $\mathcal{A}$ varies. Indeed, the set of ground atoms $\mathsf{cl}_\mathcal{T}(\mathcal{A})$ has polynomial size with respect to $\mathcal{A}$, hence the censor C can be computed by a polynomial number of guesses. Then, checking whether $\mathcal{T} \cup \mathcal{A} \models q$, for each $q \in S$, is in AC^0.

Therefore, Algorithm 1 can be executed by a nondeterministic Turing machine in polynomial time.

Theorem 9. *REC*$[\mathsf{dynCQE}_{seq}[\trianglelefteq], \boldsymbol{\mathit{UCQ}}]$ *is in* Δ_2^p *in data complexity.*

Proof. Let us fix a TBox $\mathcal{T}$, a privacy policy $\mathcal{P}$, a sequence of UCQs $Q = \langle q_1, \ldots, q_n \rangle$ and $\sigma \in \mathfrak{S}_{q_n}$ and let $\mathfrak{O}$ be an oracle for the data complexity version of the problem $\boldsymbol{P}$ defined above. Algorithm 2 defines a procedure using oracle $\mathfrak{O}$ to solve $\mathrm{REC}[\mathsf{dynCQE}_{seq}[\trianglelefteq], \mathbf{UCQ}]$.

Algorithm 1: Existence of a censor satisfying a set of UCQs.

```
Data: A DL-Lite$_\mathcal{R}$ CQE instance $\langle \mathcal{T}, \mathcal{P}, \mathcal{A} \rangle$ and a set $\mathcal{S}$ of BUCQs
guess a subset $C$ of $\mathsf{cl}_\mathcal{T}(\mathcal{A})$;
if $\mathcal{T} \cup C \cup \mathcal{P}$ is consistent then
    return $\mathcal{T} \cup C \models \mathcal{S}$;
else
    return false;
```

Algorithm 2: Algorithm for REC[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, **UCQ**].

```
Data: A state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$, where $\mathcal{E}$ is a DL-Lite$_\mathcal{R}$ CQE instance and $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ is a sequence of UCQs and $\sigma \in \mathfrak{S}_{q_n}$
1 $\langle q'_1, \ldots, q'_k \rangle \leftarrow \mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q})$;
2 $pos \leftarrow \emptyset$;
3 for $j \in [1, k]$ do
4     $res \leftarrow$ call the oracle $\mathfrak{O}$ with input $(\mathcal{E}, pos \cup \{q'_j\})$;
5     if $res$ = "yes" then
6         $pos \leftarrow pos \cup \{q'_j\}$;
7 return $\sigma(q_n) \in pos$;
```

It is clear that this algorithm correctly decides for each ABox $\mathcal{A}$, whether $\sigma(q_n)$ is true under the $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ semantics. So we just have to prove that it runs in polynomial time.

Observe that, since $\mathcal{T}$ and $\mathcal{Q}$ are fixed, for each $i \in [1, n]$, the set $cert(\mathcal{T} \cup \mathcal{A}, q_i)$ can be computed in polynomial time, and so has polynomial size. Moreover, $\trianglelefteq_{q_i}$ can be decided in logarithmic space, so we can sort the elements of each $cert(\mathcal{T} \cup \mathcal{A}, q_i)$ with respect to $\trianglelefteq_{q_i}$ in polynomial time. These observations imply that the sequence $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, q_i)$ can be built in polynomial time and has polynomial length. Consequently, we can obtain $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q})$ by concatenation of the $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, q_i)$, for each $i \in [1, n]$. Hence, line 1 takes polynomial time and produces a polynomial length sequence. This implies that the loop on lines 3–6 is executed a polynomial number of times, and so Algorithm 2 runs in polynomial time. ■

Theorems 7, 8 and 9 imply the following completeness results.

Corollary 5. *REC*[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, ***CQ***]*, REC*[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, ***FullUCQ***] *and REC*[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, ***UCQ***] *are Δ_2^p-complete in data complexity.*

9.2. The case when queries are full CQs

Now we provide a complete characterization of the complexity of reasoning over full CQs under $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ semantics.

We start by recalling some notions on graphs. An ordered graph is a triple $G = (V, E, <)$ where V is the set of vertices, $<$ a total order on V and $E \subseteq \{\{v_1, v_2\} \mid v_1, v_2 \in V\}$ is symmetric binary relation representing the undirected edges of the graph. A clique of G is a subset $C \subseteq V$ such that for each $v_1, v_2 \in C$, $\{v_1, v_2\} \in E$. A clique of G is maximal if it is not strictly contained in any other clique of G. Now, we can extend the total order $<$ on vertices to a lexicographic order on cliques as follows: Let C_1 and C_2 be two cliques of G, we say that C_1 comes lexicographically before C_2 if there exists a vertex $v \in V$ such that $v \in C_1 \setminus C_2$ and, for each $v' \in V$ such that $v' < v$, $v' \in C_1$ iff $v' \in C_2$.

In the next theorem, we use the decision problem LFMC, which is known to be PTIME-complete [28], defined as follows:

Input: An ordered graph $G = (V, E, <)$ and a vertex $v \in V$
Question: Determine whether v belongs to the lexicographically first (with respect to $<$) maximal clique of G.

Theorem 10. *REC*[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, ***FullCQ***] *is* PTIME*-hard in data complexity, even in case the TBox is empty.*

Proof. We prove the theorem by reducing the problem LFMC to REC[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, **FullCQ**] using a fixed privacy policy and a fixed sequence of queries.

Let $G = (V, E, <)$ be an ordered graph and let $v \in V$. Let us define the following ABox $\mathcal{A}_{G,v}$ using the concept Clique to contain the clique we want to find in G, the role notE to represent the absence of an edge between two vertices of the graph, the concept $\mathsf{unselected}$ to encode the fact that we want to check whether the vertex v is in the clique we found and the atom $\mathsf{LFMC}(1)$ to express whether or not v was included in the lexicographically first maximal clique of G. In order to represent the vertices of G, we consider the query $q(x) = \mathsf{Clique}(x)$ and a set of individual names with the same cardinality as V. Then, the mapping between vertices of G and individual constants is increasingly monotonic w.r.t. $\trianglelefteq_q$, that is $w_1 < w_2$ iff the corresponding individual constants w_1 and w_2 are such that $\mathsf{w}_1 \triangleleft_q \mathsf{w}_2$.

More specifically, we define the ABox $\mathcal{A}_{G,v}$ as follows:

- $\mathsf{Clique}(\mathsf{w})$, for each $w \in V$;
- $\mathsf{notE}(\mathsf{w}_1, \mathsf{w}_2)$, for each $w_1, w_2 \in V$ such that $\{w_1, w_2\} \notin E$;

- unselected(v);
- LFMC(1).

Observe that, since $\trianglelefteq_q$ is by assumption decidable using logarithmic space, we can pick the individuals representing vertices of G in logarithmic space and thus build $\mathcal{A}_{G,v}$ in logarithmic space.

Now we define the following fixed policy $\mathcal{P}$.

1. $\exists x, y\, (\mathsf{Clique}(x) \wedge \mathsf{notE}(x, y) \wedge \mathsf{Clique}(y)) \rightarrow \bot$
2. $\exists x\, (\mathsf{unselected}(x) \wedge \mathsf{Clique}(x)) \rightarrow \bot$
3. $\exists x\, (\mathsf{LFMC}(1) \wedge \mathsf{unselected}(x)) \rightarrow \bot$

The first denial expresses the fact that all pairs of vertices in the extension of Clique must be connected by an edge, i.e., Clique represents a clique, the second denial expresses the fact that the concepts unselected and Clique must be disjoint, while the third denial expresses the fact that LFMC(1) can be true only when unselected does not hold for any individual of the knowledge base.

Then we define the following fixed sequence of full CQs $\mathcal{Q}$.

$$q_1(x, y) = \mathsf{notE}(x, y)$$

$$q_2(x) = \mathsf{Clique}(x)$$

$$q_3(x) = \mathsf{unselected}(x)$$

$$q_4 = \mathsf{LFMC}(1).$$

Let G be an ordered graph, v one of its vertices, and let us consider the protection state $\mathcal{S} = \langle\langle \emptyset, \mathcal{P}, \mathcal{A}_{G,v}\rangle, \mathcal{Q}\rangle$. We conclude by proving that LFMC(1) is entailed by $\mathcal{S}$ under the $\mathsf{dynCQE}_{seq}[\trianglelefteq]$ semantics iff (G, v) is an instance of LFMC. First, observe that query q_1 can be answered honestly, and it forces all maximally cooperative censors to retain all facts pertaining to the structure of G. Consequently, due to the first denial and maximal cooperativity, answering query q_2 requires computing a maximal clique of the graph G. According to the semantics of $\mathsf{dynCQE}_{seq}[\trianglelefteq]$, the resulting set is the lexicographically first, based on $\trianglelefteq_{q_2}$. Then, since the mapping between vertices in G and individual constants occurring in $\mathcal{A}_G$ is order-preserving, the answer set for q_2 describes the lexicographically first maximal clique of G as per the $<$ relation. Subsequently, because of denial 2 and by definition of $\mathcal{A}_{G,v}$, answering q_3 involves verifying whether the computed clique includes vertex v. Finally, because of denial 3, q_4 will be true iff v is in the lexicographically first clique of G. ■

We now prove the completeness by showing a PTime algorithm.

Theorem 11. *REC*[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, ***FullCQ***] *is in* PTime *in data complexity.*

Proof. Let us fix a TBox $\mathcal{T}$, a privacy policy $\mathcal{P}$, a sequence of full CQs $\mathcal{Q} = \langle q_1, \dots, q_n\rangle$ and $\sigma \in \mathfrak{S}_{q_n}$. Algorithm 3 solves REC[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, **FullCQ**].

Algorithm 3: Algorithm for REC[$\mathsf{dynCQE}_{seq}[\trianglelefteq]$, **FullCQ**].

```
Data: A state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q}\rangle$, where $\mathcal{E} = \langle \mathcal{T}, \mathcal{P}, \mathcal{A}\rangle$ is a DL-Lite$_\mathcal{R}$ CQE instance and $\mathcal{Q} = \langle q_1, \dots, q_n\rangle$ is a sequence of full CQs and $\sigma \in \mathfrak{S}_{q_n}$
1 $\langle q'_1, \dots, q'_k\rangle \leftarrow \mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q})$;
2 $C \leftarrow \emptyset$;
3 for $j \in [1, k]$ do
4     if $\mathcal{T} \cup C \cup Atoms(q'_j) \cup \mathcal{P}$ is consistent then
5         $C \leftarrow C \cup Atoms(q'_j)$;
6 return $\mathcal{T} \cup C \models \sigma(q_n)$;
```

As discussed in Theorem 9, $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q})$ has polynomial size and can be computed in polynomial time. Then, since $\mathcal{T}$ and $\mathcal{P}$ are fixed, checking the consistency of $\mathcal{T} \cup C \cup Atoms(q'_j) \cup \mathcal{P}$ in Step 4 can be done in polynomial time. Consequently, the algorithm runs in polynomial time. Due to Proposition 15, in order to prove the correctness of the algorithm, it is sufficient to prove that the censor C in Step 6 is maximally cooperative with respect to $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q})$. To do so, we denote by C_i the set C after the i-th iteration of the for loop in Step 3. Assume by contradiction that there exists a censor C' for $\mathcal{E}$ which is more cooperative than C with respect to $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E}, \mathcal{Q}) = \langle q'_1, \dots, q'_k\rangle$. We can assume that C' is optimal. By definition, there exists a natural number m such that:

- For each $i \in [1, m]$, $\mathcal{T} \cup C \models q'_i$ iff $\mathcal{T} \cup C' \models q'_i$;
- $\mathcal{T} \cup C \not\models q'_{m+1}$ and $\mathcal{T} \cup C' \models q'_{m+1}$.

Now, observe that since $\mathcal{C}'$ is an optimal censor and each q'_j, with $j \in [1,k]$, appearing in $\mathsf{BoolInst}[\trianglelefteq](\mathcal{E},\mathcal{Q})$ is a Boolean full CQ, then $\mathcal{T} \cup \mathcal{C}' \models q'_j$ iff $Atoms(q'_j) \subseteq \mathcal{C}'$. On the other hand, by construction, we also have that $\mathcal{T} \cup \mathcal{C} \models q'_j$ iff $Atoms(q'_j) \subseteq \mathcal{C}$. Therefore, it follows that:

- For each $i \in [1,m]$, $Atoms(q'_i) \subseteq \mathcal{C}$ iff $Atoms(q'_i) \subseteq \mathcal{C}'$;
- $Atoms(q'_{m+1}) \nsubseteq \mathcal{C}$;
- $Atoms(q'_{m+1}) \subseteq \mathcal{C}'$.

This situation can only occur because $\mathcal{T} \cup \mathcal{C}_m \cup Atoms(q'_{m+1}) \cup \mathcal{P}$ is inconsistent. On the other hand, we have that $\mathcal{C}_m \subseteq \mathcal{C}'$ and thus $\mathcal{T} \cup \mathcal{C}' \cup \mathcal{P}$ would be inconsistent too. A contradiction. ■

The above theorems imply the following final result.

Corollary 6. *REC*$[\mathsf{dynCQE}_{seq}[\trianglelefteq], \textbf{\textit{FullCQ}}]$ *is* PTIME-*complete in data complexity.*

10. Related work

Controlled Query Evaluation was first investigated in the context of databases in [3,11,23,22].

The framework has then been extended to Description Logics in different ways. In [29] the authors define CQE for ontologies specified in Boolean $\mathcal{ALC}$ and queries that are $\mathcal{ALC}$ formulas. The focus of the work is on the properties that a censor should have to guarantee certain desiderata regarding data confidentiality protection. This investigation has been later generalized to general forms of logic in [30]. In [24,6], the confidentiality of secrets is obtained through *confidentiality preserving* (CP) censors. More specifically, reference [24] considers ontologies in OWL 2 RL, the tractable profile of OWL 2 [13] that is encodable in Datalog, and a policy that is a set of ground atoms. The notion of CP censor defined allows a user to obtain a maximal set of answers to each query without being able to infer the atoms in the policy. Additionally, the paper pinpoints a subset of OWL 2 RL for which an optimal censor can be computed in polynomial time. Continuing their exploration in [6], the same authors delve into ontologies specified in Datalog or in one of the OWL 2 profiles, with the policy expressed as a CQ. Similar to [24], the primary focus of the paper is on the existence and computation of an optimal censor, according to two different censor notions, called view-based and obstruction-based. Various computational complexity results are provided. Interestingly, for OWL 2 QL ontologies, i.e., the OWL 2 counterpart of DL-Lite$_{\mathcal{R}}$ (which is the logic studied in the present paper), an obstruction-based censor in the considered setting can be obtained in polynomial time. The CQE problem for DL ontologies has been then addressed, and somehow revisited, in [7], where the authors study skeptical reasoning in CQE, that is, the problem of computing only the query answers that are returned by *all* censors. This form of query answering is studied for ontologies expressed in two popular lightweight DLs, that is, DL-Lite$_{\mathcal{R}}$ and $\mathcal{EL}_{\bot}$. Algorithms and complexity results are provided for different, incomparable notions of censors, differing from one another for the censor language on which they are based. Intuitively, a censor for a language $\mathcal{L}$ returns a set of sentences in $\mathcal{L}$ that are entailed by the original ontology and that together with the TBox do not infer data protected by the policy. The policy is given as a set of denials, exactly as done in the present paper. Interestingly, a case in which skeptical reasoning in CQE is polynomial in data complexity is identified for ontologies expressed in DL-Lite$_{\mathcal{R}}$. This result has been later refined in [31], where the problem is shown to be FO rewritable and thus in AC^0 in data complexity. We point out that all the above mentioned approaches do not take into account what a user asked in the past, that is, in answering a user query they do not consider a protection state, as we do in the present paper (see Definition 1). As a consequence, to ensure confidentiality, it has to be assumed that a user may have asked whatever query she wanted. Intuitively, this assumption may compromise cooperativeness, as we have defined it in the present paper. In particular, we have formally shown in Section 3 that the skeptical reasoning semantics is always a sound approximation of any MC-CQE semantics (the skeptical reasoning semantics corresponds to the semantics studied in [7] when the censor language is the language of ground atoms constructible on the ontology signature). As for safeness guarantees, it has to be said that the type of censors studied in [24,6,7] do not take into account possible background knowledge owned by an attacker and thus, as it was shown in [8], in the presence of such additional knowledge, security breaches under these censors may occur.

A safer notion of censor, protecting even in the above mentioned situation, is the indistinguishability based (IB) one. The first CQE method for Description Logics based on IB censors was introduced in [1,32]. The confidentiality model proposed in these papers takes into account both object-level and meta-level background knowledge of the attacker and is more robust and general. However, CQ answering and FO rewritability are not addressed in those papers. Moreover, the *secure views* of [1] are constructed from a sequence of queries that covers *all* possible relevant queries, while the properties we investigate in this paper hold for arbitrary (possibly non-exhaustive) sequences of queries submitted by the users.

In [9] IB censors are compared with CP censors that in general do not enjoy the indistinguishability property. Moreover, [9] provides algorithms and complexity results for skeptical reasoning under IB censors, that is, the problem of computing only the query answers that are returned by *all* IB censors, for the case of DL-Lite$_{\mathcal{R}}$ ontologies and policy that is a set of denials. The paper shows that the problem is, in general, intractable. To re-gain tractability, the paper proposes a sound approximation of IB censors for which answering CQs is FO rewritable. We notice that the approximation proposed in [9] corresponds to the IGA semantics studied in [12], and then adopted in [33] for more expressive policies allowing for numerical restrictions in the denial body. We have recalled the IGA semantics in Section 3 and have shown in Proposition 3 that it is, in general, less cooperative than MC-CQE semantics.

Table 3
Data complexity of the examined decision problem.

Query language	General MC-CQE semantics	Tractable preferences	Lexicographic preferences
BUCQ	in AC^0	in AC^0	in AC^0
FullCQ	not in AC^0	PTIME-hard	PTIME-complete
CQ **FullUCQ** **UCQ**	$\Delta_2^p[O(\log n)]$-hard	$\Delta_2^p[O(\log n)]$-hard in Σ_2^p	Δ_2^p-complete

The issue of how to select an optimal censor has been tackled in [10]. The selection criterion is based on explicit preferences over predicates, that are specified together with the CQE instance. This approach, in general, is not maximally cooperative w.r.t. a given state, because the optimal censor is selected statically, in a stateless fashion. Moreover, the given preferences are not always able to select a single optimal censor.

Benedikt et al. [34,35] provide, for ontology-based data integration settings, a systematic complexity analysis of confidentiality preserving query answering based on indistinguishability. They do not address the issue of selecting a secure data disclosure among the available ones, but rather study the problem on whether a data integration system, in which mappings connect a source database to a global integrated one (possibly given in terms of an ontology), is such that no data protected by a policy specified on the sources are indeed disclosed. Intuitively, this means checking whether the mappings of the data integration systems are able to filter out confidential data as specified by the source policy. IB censors in OBDA are also considered in [12], where a practical approach to skeptical reasoning in CQE is presented. Differently from our approach, in [12] censors do not take into account the history of the users' queries.

An alternative approach for preserving privacy in query answering relies on probabilistic frameworks, ensuring that users gain no new insights regardless of their prior beliefs when executing queries. In [36], a prior belief is defined as a probability distribution over tuples, indicating the likelihood of each tuple being present in a given database. The security of a view is determined by the inability of its answers to contribute to Bayesian inference, thereby enhancing the probability of a secret. This concept easily extends to multiple views, enabling the examination of potential collusions among attackers. In a similar vein, Biskup et al. [37] adopt a query refusal approach that guarantees the conditional probability of any secret, given the provided answers, remains below a predefined threshold.

Furthermore, Cuenca Grau et al. [2,38] introduce and investigate an anonymization framework for knowledge graphs based on substituting nodes with blanks. This framework is used in [39] to define a CQE semantics in the case when there is no TBox, the privacy policy is defined in the Description Logic $\mathcal{EL}$ and the only allowed queries are instance queries. Note that, as observed in [8], the approach used is vulnerable when the attacker knows the anonymization algorithm and the policy.

We conclude this section by discussing the relationship between the CQE problem, as considered in this paper, and the Consistent Query Answering (CQA) problem, extensively studied for databases and ontologies [40–43]. Both problems address complementary challenges related to reasoning over inconsistent or restricted knowledge. CQA focuses on deriving semantically coherent answers from inconsistent ontologies by considering all possible *repairs*, that is, maximal subsets of the data that restore consistency with the ontology's logical constraints [43]. CQE, on the other hand, enforces confidentiality policies by dynamically censoring query answers that might reveal sensitive information, while still returning maximal non-sensitive subsets of those answers. Optimal censors in CQE selectively withhold the minimal amount of information necessary to prevent policy violations, thereby reflecting a principle of minimal change similar to the one underlying repairs in CQA. Recent studies have established a formal connection between the two frameworks: confidential data in CQE can be viewed as inconsistencies relative to policy constraints, making CQE analogous to answering queries over models that somehow "repair" policy violations [31]. This connection in some cases enables mutual reductions between the CQE and the CQA problem, especially within the context of Description Logics, where both frameworks rely on reasoning over alternative scenarios (repairs or censored views) to compute query answers (see [31] for a detailed discussion). It is also worth noting that, building on the connection between CQE and CQA, priority-based CQA semantics have been adapted to the CQE setting in cases where priority relations among axioms can be exploited to guide the censoring process [44]. Interestingly, a scenario is identified in which answering conjunctive queries under prioritized ontologies is in AC^0 in data complexity. We point out, however, that the investigation in [31,44] shares with CQA approaches the skeptical nature of reasoning. Instead, in the present paper, we aim at maximizing the information preserved in the semantics in order to increase cooperativeness. This difference sets our approach apart from the previous ones. Therefore, the results presented in this work have not been obtained leveraging the CQE-CQA correspondence, and the preference ordering over censors used in the present paper is fundamentally different in nature from the priority relations considered in [44].

11. Conclusions

In this investigation, our objective is to enhance the cooperative nature of indistinguishability-based CQE within Description Logic ontologies. To this end, we have identified a class of maximally cooperative CQE semantics (MC-CQE), enjoying the "longest honeymoon" property. This property guarantees that, when presented with a sequence of queries, the provision of honest answers is maximized before resorting to deceptive responses.

We have shown that our approach delivers a larger set of answers compared to competing methodologies, such as skeptical reasoning and IGA. We have also proved that our method cannot be simulated by modifying the underlying ontology or privacy policy. Instead, it necessitates specific query answering techniques.

We have conducted an in-depth study on the complexity of query answering in the MC-CQE framework. The results of our analysis are summarized in Table 3. We remark that, as stated by Theorem 3, Theorem 4, Theorem 5, Theorem 7, Theorem 8 and Theorem 10, the lower complexity bounds hold for *every* semantics in the indicated family. Moreover, these bounds remain valid even when the TBox is empty.

This work examines the profitability of the MC-CQE framework compared to other methodologies and provides a detailed analysis of its data complexity with respect to DL-Lite$_{\mathcal{R}}$. However, several design and computational challenges remain, creating opportunities for further research.

From a design perspective, we assume that MC-CQE processes all users' queries in a single sequence. While this approach may appear less cooperative from an individual user's perspective, it ensures security even when users collude to share information. The extent to which this assumption can be relaxed depends on the specific application domain. Moreover, specific realizations of maximally cooperative CQE semantics, such as dynCQE and dynCQE[$\preceq$], maintain the entire query history, which can grow rapidly over time. However, as the history expands, many queries may no longer affect the set of *active* censors and, therefore, could be omitted from the history. Furthermore, the number of active censors may decrease to just one, at which point no additional queries need to be stored. As future work, quickly identifying queries that do not influence future evaluations can serve as a performance optimization technique, which may produce shorter FO rewritings for Boolean queries.

From a theoretical perspective, we aim at exploring the combined complexity of our framework when queries may vary. Additionally, at an abstract level, MC-CQE semantics demonstrate a degree of independence from the underlying logical language. Therefore, more expressive denial languages can be considered, enabling the representation of numerical restrictions or inequality relations [10]. Similarly, MC-CQE semantics can be explored within other Description Logic fragments.

Finally, once the theoretical landscape is sufficiently understood, we intend to implement and experimentally evaluate suitable MC-CQE semantics.

CRediT authorship contribution statement

Piero Bonatti: Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Gianluca Cima:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Domenico Lembo:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Francesco Magliocca:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Lorenzo Marconi:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Riccardo Rosati:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Luigi Sauro:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Domenico Fabio Savo:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

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.

Acknowledgements

This work was partially supported by: projects FAIR (PE0000013) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan 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 NRRP Complementary Investments (project n. PNC0000003); POLAR project funded by MUR - Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2022 (cup G53D23002870006); and the dAIbetes Project funded by the EU's Horizon Europe Research and Innovation Programme (grant agreement No. 101136305).

Data availability

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

References

[1] 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, Springer, 2013, pp. 17–32.

[2] B. Cuenca Grau, E.V. Kostylev, Logical foundations of linked data anonymisation, J. Artif. Intell. Res. 64 (2019) 253–314.

[3] 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.

[4] J. Biskup, For unknown secrecies refusal is better than lying, Data Knowl. Eng. 33 (1) (2000) 1–23.

[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] 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.

[7] 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.

[8] P.A. Bonatti, A false sense of security, Artif. Intell. 310 (2022) 103741.

[9] 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.

[10] 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.

[11] 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.

[12] 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), in: Lecture Notes in Computer Science, vol. 12506, Springer, 2020, pp. 128–146.

[13] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 Web Ontology Language Profiles second edition, Dec. 2012, W3C, Recommendation, World Wide Web Consortium, available at http://www.w3.org/TR/owl2-profiles/.

[14] P. 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, Springer International Publishing, Cham, 2022, pp. 428–444.

[15] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider, S. Rudolph, OWL 2 Web Ontology Language: Primer, second edition, Dec. 2012, W3C, Recommendation, World Wide Web Consortium, available at http://www.w3.org/TR/owl2-primer/.

[16] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[17] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press, 2003.

[18] J.W. Lloyd, Foundations of Logic Programming, second, extended edition, Springer, Berlin, Heidelberg, 1987.

[19] 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. Autom. Reason. 39 (3) (2007) 385–429.

[20] 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.

[21] C.H. Papadimitriou, Computational Complexity, Addison Wesley Publ. Co., 1994.

[22] J. Biskup, T. Weibert, Keeping secrets in incomplete databases, Int. J. Inf. Secur. 7 (3) (2008) 199–217.

[23] 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.

[24] 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, Springer, 2013, pp. 49–65.

[25] N. Immerman, Descriptive Complexity, Springer Verlag, 1998.

[26] K.W. Wagner, More complicated questions about maxima and minima, and some closures of NP, Theor. Comput. Sci. 51 (1) (1987) 53–80.

[27] M.W. Krentel, The complexity of optimization problems, J. Comput. Syst. Sci. 36 (3) (1988) 490–509.

[28] R. Greenlaw, H.J. Hoover, W.L. Ruzzo, Limits to Parallel Computation: P-Completeness Theory, Oxford University Press, Inc., USA, 1995.

[29] T. Studer, J. Werner, Censors for Boolean description logic, Trans. Data Priv. 7 (3) (2014) 223–252.

[30] T. Studer, No-go theorems for data privacy, CoRR, arXiv:2005.13811 [abs], 2020, arXiv:2005.13811.

[31] G. Cima, D. Lembo, R. Rosati, D.F. Savo, Controlled query evaluation in description logics through consistent query answering, Artif. Intell. 334 (2024) 104176.

[32] P.A. Bonatti, L. Sauro, I.M. Petrova, A mechanism for ontology confidentiality, in: Proc. of the 29th Italian Conf. on Computational Logic (CICL), in: CEUR Electronic Workshop Proc., vol. 1195, 2014, pp. 147–161, http://ceur-ws.org/.

[33] G. Cima, D. Lembo, L. Marconi, R. Rosati, D.F. Savo, D. Sinibaldi, Controlled query evaluation over ontologies through policies with numerical restrictions, in: Proc. of the 4th IEEE Int. Conf. on Artificial Intelligence and Knowledge Engineering (AIKE), IEEE, 2021, pp. 33–36.

[34] M. Benedikt, B. Cuenca Grau, E.V. Kostylev, Logical foundations of information disclosure in ontology-based data integration, Artif. Intell. 262 (2018) 52–95.

[35] M. Benedikt, B. Cuenca Grau, E. Kostylev, Source information disclosure in ontology-based data integration, in: Proc. of the 31st AAAI Conf. on Artificial Intelligence (AAAI), vol. 31 (1), Feb. 2017.

[36] G. Miklau, D. Suciu, A formal analysis of information disclosure in data exchange, J. Comput. Syst. Sci. 73 (3) (2007) 507–534.

[37] J. Biskup, P.A. Bonatti, C. Galdi, L. Sauro, Inference-proof data filtering for a probabilistic setting, in: Proc. of the 5th Workshop on Society, Privacy and the Semantic Web - Policy and Technology (PrivOn2017), in: CEUR Electronic Workshop Proc., vol. 1951, 2017, http://ceur-ws.org/.

[38] B. Cuenca Grau, E. Kostylev, Logical foundations of privacy-preserving publishing of linked data, in: Proc. of the 30th AAAI Conf. on Artificial Intelligence (AAAI), vol. 30 (1), Feb. 2016.

[39] F. Baader, F. Kriegel, A. Nuradiansyah, R. Peñaloza, Computing compliant anonymisations of quantified ABoxes wrt $\mathcal{EL}$ policies, in: Proc. of the 19th Int. Semantic Web Conf. (ISWC), Springer, 2020, pp. 3–20.

[40] 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.

[41] L.E. Bertossi, Database Repairing and Consistent Query Answering, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2011.

[42] 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.

[43] 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.

[44] G. Cima, D. Lembo, L. Marconi, R. Rosati, D.F. Savo, Indistinguishability in controlled query evaluation over prioritized description logic ontologies, J. Web Semant. 84 (2025) 100841.