The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

Answering Conjunctive Queries with Safe Negation and Inequalities over RDFS Knowledge Bases

Gianluca Cima, Marco Console, Roberto Maria Delfino, Maurizio Lenzerini, Antonella Poggi
Sapienza University of Rome
{cima, console, delfino, lenzerini, poggi}@diag.uniroma1.it

Abstract

Expressing negative conditions is a crucial feature of query languages for knowledge bases (KBs). Answering such queries over ontological KBs, however, is a very challenging task that becomes undecidable even for lightweight Description Logic (DL) ontologies. Such negative results hold even for Conjunctive Queries (CQs) equipped with basic forms of negative conditions such as the so-called "safe" negation or inequality atoms. One ontology language that is seemingly unaffected by these results is (the DL counterpart of) RDFS even if equipped with disjointness axioms. Answering CQs with inequalities over such ontologies is known to be Π_2^p-complete, if the number of inequality atoms is unbounded, and NP-complete if we limit this number to one. Notably, these results leave open the cases of CQs with a fixed number greater than two of inequality atoms. Additionally, such a thorough analysis is missing for CQs with safe negation.

In this paper, we embark in a refined analysis of the combined complexity of answering CQs with inequality atoms and safe negation over RDFS ontologies augmented with disjointness axioms. Firstly, we provide a unified Π_2^p query answering algorithm for the general problem. Secondly, we confirm the generally held conjecture according to which answering CQs with two inequality atoms over such ontologies is already Π_2^p-hard. This result closes an important gap in the current literature and has an impact on the widely influential problem of query containment. Lastly, for CQs with safe negation, we prove a behavior similar to that of CQs with inequality atoms. Specifically, we show that answering CQs with at most one negated atom can be done in NP, while allowing at most two negated atoms is sufficient to obtain Π_2^p-hardness.

1 Introduction

Knowledge Base (KB) systems are AI systems that have access to a KB, i.e., a symbolic representation of knowledge related to specific application domains. AI systems may interact with their KBs to perform several information management tasks. The results thus obtained are then used to either provide information to end-users or to improve the performance of other AI systems. The recent trend of *knowledge graphs* provides perfect examples of KB systems. In particular, the KB of such systems is the knowledge graph itself, and the tasks they perform may vary from answering user-defined queries (Khan 2023) to enhancing the results of generative AI models (Yang et al. 2024; Pan et al. 2024).

Usually, KBs come with an attached *semantics*, i.e., a formally defined meaning that is the basis for all the information management tasks. Such semantics is often obtained by specifying the KB in a formal language, and then using the semantics of this language to interpret the KB. In this context, a prominent family of languages is that of Description Logics (DLs) (Baader et al. 2017). DLs are fragments of first order logic with a dedicated syntax, based on unary and binary predicates, and a semantics defined via first order interpretations and models. A DL KB, often called *ontology*, is simply a logical theory constituted by DL *axioms* typically organized into two components: a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, specifying, respectively, intensional and extensional knowledge about the domain of interest. Due to these characteristics, DL Ontologies have been successfully used to provide semantics to knowledge graph systems (Xiao et al. 2019). In such frameworks, a knowledge graph is represented by the ABox and the TBox of a DL ontology, and the models of such ontology define the semantics of the knowledge graph.

A crucial computational task underlying information management in DL ontologies is *query answering* (Calvanese et al. 2007; Bienvenu and Ortiz 2015). In simple words, query answering amounts to checking whether a query, i.e., an expression defined in a formal language, is satisfied by all the models of the ontology. Some of the query languages that have received most attention in this context are, arguably, those of *Conjunctive Queries* (*CQs*) and *Unions* thereof (*UCQs*), i.e., queries specified by the existential-positive fragment of first-order logic with (UCQs) or without (CQs) disjunction. After decades of studies, we have a fairly complete picture of the problem of answering UCQs over DL ontologies. Notably, such problem remains decidable, although intractable, even for expressive DLs such as $\mathcal{SHIQ}$ (Glimm et al. 2008).

While expressive enough to define several important properties, UCQs are unable to express even the most basic form of negation. For example, a UCQ cannot ask for individuals of a specific class that do not posses a given property (*safe negation*) or whether there are two distinct individuals with specific characteristics (*inequality atoms*). These features are crucial in several context, e.g., to deal with knowledge graphs that represent information coming from multi-

---

ple, heterogeneous sources that need filtering before use.

Unfortunately, allowing the use of negation, even in the simplest forms, affects dramatically the computational characteristics of UCQs. Indeed, as shown in (Gutiérrez-Basulto et al. 2015), answering UCQs with inequality atoms ($\mathrm{UCQ}^{\neq}$) and UCQs with safe negation ($\mathrm{UCQ}^{\neg s}$) is *undecidable* already for lightweight DL languages, such as *DL-Lite*$_{\mathcal{R}}$ and $\mathcal{EL}$, and similar results carry over to CQs.

One DL language that is seemingly unaffected by these negative results is *DL-Lite*$^{\neg}_{\texttt{RDFS}}$, i.e., the DL counterpart of `RDFS` (Cuenca Grau 2004; Rosati 2007) extended with disjointness axioms and interpreted, coherently with `RDFS`, without the Unique Name Assumption (UNA). While less expressive than other lightweight DL languages, *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ simulates all the first-order logic features of `RDFS`, i.e., the semantically enriched version of one of the most common representations for knowledge graphs. Additionally, it can express data quality requirements by disjointness axioms, a crucial feature for systems based on knowledge graphs.

Interestingly, answering $(\mathrm{U})\mathrm{CQ}^{\neq}$s over *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies has been shown to be Π^p_2-complete in combined complexity (Cima, Lenzerini, and Poggi 2020). However, the reduction used there yields a $\mathrm{CQ}^{\neq}$ query with a number k of inequality atoms that depends on the size of the input instance. In other words, the reduction is not valid if k is fixed a priori. The same work shows that, for *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies, answering $\mathrm{UCQ}^{\neq}$ queries with at most one inequality atom in each disjunct, i.e., fixing k to 1, is an NP-complete problem, and thus it loses a significant part of its complexity (under the usual computational complexity assumptions). By looking at these two results together, a natural question to ask is whether there exists $k \geq 2$ for which answering $\mathrm{CQ}^{\neq}$ queries with a fixed number k of inequality atoms over *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies is Π^p_2-complete. In (Cima, Lenzerini, and Poggi 2020), it was stated the following conjecture

Conjecture 1: Answering $\mathrm{CQ}^{\neq_2}$s (CQs with at most two inequality atoms) over *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies is Π^p_2-hard.

In this paper, we embark on a thorough analysis of the problem of answering UCQs with inequality atoms and safe negation ($\mathrm{UCQ}^{\neg s,\neq}$) over *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies providing the following contributions. Firstly, we present an algorithm to answer $\mathrm{UCQ}^{\neg s,\neq}$s, proving a Π^p_2 upper bound in combined complexity and a coNP upper bound in data complexity (i.e. the complexity where only the ABox is regarded as the input). This result extends the one presented in (Cima, Lenzerini, and Poggi 2020) for UCQs with inequality atoms only. Secondly, while a matching lower bound in data complexity was already known for the problem of answering CQ queries with at most two inequality atoms, we provide a matching lower bound in combined complexity in the same scenario, thus verifying Conjecture 1.

Our result has also an impact on the query containment problem for $\mathrm{CQ}^{\neq}$s. In (Klug 1988), this problem has been shown to be in Π^p_2 and conjectured to be Π^p_2-complete. The conjecture has been later confirmed in (van der Meyden 1997), while (Kolaitis, Martin, and Thakur 1998) studied the problem under several syntactic and structural conditions. To the best of our knowledge, however, only (Cima, Lenzerini, and Poggi 2020) analyzed the impact of restricting the number of inequalities that can appear in the queries. Specifically, that paper shows that if the containing query is restricted to have at most one inequality atom, then the problem becomes NP-complete, i.e. has the same complexity as for CQs (Chandra and Merlin 1977). A further contribution of this paper is to conclude this analysis and show that the query containment problem for $\mathrm{CQ}^{\neq}$s remains Π^p_2-hard even if the containing query and the contained query are restricted to have one and two inequality atoms, respectively.

$\mathcal{Q}$	Data	Combined
$\{(\mathsf{U})\mathsf{CQ}^{\neg s,1}, (\mathsf{U})\mathsf{CQ}^{\neq_1}\}$	P-c	NP-c
$\{\mathsf{CQ}^{\neg s,2}, \mathsf{CQ}^{\neq_2}, \mathsf{UCQ}^{\neg s,\neq}\}$	coNP-c	Π^p_2-c

Table 1: Data and combined complexity results of answering $\mathcal{Q}$ queries over *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies, where "-c" stands for "-complete". The numbers 1 and 2 indicate the number of negated atoms and inequality atoms allowed in queries.

Thirdly, we present complexity lower bounds for CQs with safe negation ($\mathrm{CQ}^{\neg s}$). Interestingly, $\mathrm{CQ}^{\neg s}$ queries exhibit a behavior similar to $\mathrm{CQ}^{\neq}$ queries: answering $\mathrm{CQ}^{\neg s}$ queries with at most one negative atom can be done in NP, while allowing at most two negative atoms is sufficient to obtain Π^p_2-hardness. To the best of our knowledge, this is the first detailed analysis of this kind in the literature. The main complexity results of this work are summarized in Table 1. Of those listed in the table, the only results that were already present in the literature are the data and combined complexity of $\mathrm{CQ}^{\neq_1}$s and $\mathrm{UCQ}^{\neq_1}$s, and the data complexity of $\mathrm{CQ}^{\neq_2}$s (Cima, Lenzerini, and Poggi 2020). All other completeness results are stated here for the first time. Finally, we analyze the case of the Unique Name Assumption for the semantics of *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies.

Notably, the above results imply that query answering in our setting can be solved via Π^p_2 solvers, and this is the best one can expect under the usual complexity assumptions.

The rest of the paper is organized as follows. Section 2 contains preliminaries; Section 3 discusses the case of $\mathrm{UCQ}^{\neg s,\neq}$ queries; Section 4 presents our results on $\mathrm{CQ}^{\neg s}$ queries, while Section 5 presents those on $\mathrm{CQ}^{\neq}$s; Section 6 discusses the impact of UNA; and in Section 7 we conclude.

2 Preliminaries

We fix the four pairwise disjoint and countably infinite sets of symbols Σ_{AC}, Σ_{AR}, $\Sigma_{\mathcal{C}}$, and $\Sigma_{\mathcal{V}}$ for *atomic concepts* (i.e. unary predicates), *atomic roles* (i.e. binary predicates), constants (a.k.a. individuals), and variables, respectively.

Ontologies: Syntax We consider *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies. A *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontology is a pair $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{A}$ is an ABox, i.e. a non-empty, finite set of assertions (also called *ground atoms*) of the form $A(c)$ and $P(c_1, c_2)$ with $A \in \Sigma_{\mathrm{AC}}$, $P \in \Sigma_{\mathrm{AR}}$, and $c, c_1, c_2 \in \Sigma_{\mathcal{C}}$, and $\mathcal{T}$ is a *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ TBox, i.e. a finite set of assertions of the following form:

$B \sqsubseteq A$ $\quad R_1 \sqsubseteq R_2$ $\quad$ (concept/role inclusion)
$B_1 \sqsubseteq \neg B_2$ $\quad R_1 \sqsubseteq \neg R_2$ $\quad$ (concept/role disjointness),

where $A \in \Sigma_{\text{AC}}$ and for $i = 1$ and $i = 2$ (*i*) R_i is a *basic role*, i.e. either an atomic role $P \in \Sigma_{\text{AR}}$ or the inverse of an atomic role $P \in \Sigma_{\text{AR}}$, which we denoted by P^-, and (*ii*) B_i is a *basic concept*, i.e. either an atomic concept $A \in \Sigma_{\text{AC}}$ or an expression of the form $\exists R$ with R a basic role.

Note that *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ extends the DL-like counterpart of `RDFS` (usually referred as *DL-Lite*$_{\texttt{RDFS}}$ (Cuenca Grau 2004; Rosati 2007; Cima, Poggi, and Lenzerini 2023)) by allowing for disjointness assertions. More precisely, we say that a TBox $\mathcal{T}$ is a *DL-Lite*$_{\texttt{RDFS}}$ TBox if it is a finite set of concept and role inclusion assertions, and we say that $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a *DL-Lite*$_{\texttt{RDFS}}$ ontology if $\mathcal{T}$ is a *DL-Lite*$_{\texttt{RDFS}}$ TBox.

Ontologies: Semantics The semantics of *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies is specified through interpretations: an *interpretation* $\mathcal{I}$ is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty, possibly infinite set of objects, and $\cdot^{\mathcal{I}}$ assigns to each constant $c \in \Sigma_{\mathcal{C}}$ an object $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to each atomic concept $A \in \Sigma_{\text{AC}}$ a set of objects $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to each atomic role $P \in \Sigma_{\text{AR}}$ a set of pairs of objects $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Then $\cdot^{\mathcal{I}}$ extends to other basic concepts and roles as follows: $(\exists P)^{\mathcal{I}} = \{o \mid \exists o'.(o,o') \in P^{\mathcal{I}}\}$, $(P^-)^{\mathcal{I}} = \{(o,o') \mid (o',o) \in P^{\mathcal{I}}\}$, and $(\exists P^-)^{\mathcal{I}} = \{o \mid \exists o'.(o',o) \in P^{\mathcal{I}}\}$.

An interpretation $\mathcal{I}$ satisfies a concept inclusion assertion $B \sqsubseteq A$ (resp. role inclusion assertion $R_1 \sqsubseteq R_2$) if $B^{\mathcal{I}} \subseteq A^{\mathcal{I}}$ (resp. $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$), satisfies a concept disjointness assertion $B_1 \sqsubseteq \neg B_2$ (resp. role disjointness assertion $R_1 \sqsubseteq \neg R_2$) if $B_1^{\mathcal{I}} \cap B_2^{\mathcal{I}} = \emptyset$ (resp. $R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset$), and satisfies a ground atom $A(c)$ (resp. $P(c_1, c_2)$) if $c^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp. $(c_1^{\mathcal{I}}, c_2^{\mathcal{I}}) \in P^{\mathcal{I}}$). We say that an interpretation $\mathcal{I}$ is a *model* of a *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontology $\mathcal{O}$ if $\mathcal{I}$ satisfies every assertion and ground atom occurring in $\mathcal{O}$. We denote by $mod(\mathcal{O})$ the set of models of a *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontology $\mathcal{O}$. We say that $\mathcal{O}$ is satisfiable if $mod(\mathcal{O}) \neq \emptyset$, unsatisfiable otherwise.

We say that an interpretation $\mathcal{I}$ satisfies the *Unique Name Assumption (*UNA*)* if $c_1^{\mathcal{I}} \neq c_2^{\mathcal{I}}$ holds for each pair of constants $(c_1, c_2) \in \Sigma_{\mathcal{C}}^2$. We denote by $modU(\mathcal{O})$ the set of models of a *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontology $\mathcal{O}$ that satisfy the UNA.

Query Languages A *conjunctive query with safe negation and inequalities (CQ*$^{\neg_s, \neq}$*)* q is an expression of the form $q = \{\bar{x} \mid \exists \bar{y}.\phi(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y}) \wedge \xi(\bar{x}, \bar{y})\}$, where (*i*) $\bar{x} = (x_1, \ldots, x_n)$ is a tuple of variables, called *distinguished variables*, and n is the *arity* of q (*ii*) $\bar{y}$ is a tuple of variables, also called *existential variables*, (*iii*) $\phi(\bar{x}, \bar{y})$ is a conjunction of atoms with predicate names from $\Sigma_{\text{AC}} \cup \Sigma_{\text{AR}}$ and terms from $\bar{x} \cup \bar{y} \cup \Sigma_{\mathcal{C}}$, (*iv*) $\psi(\bar{x}, \bar{y})$ is a conjunction of negated atoms (i.e. formulae of the form $\neg\alpha$ where α is an atom) with predicate names from $\Sigma_{\text{AC}} \cup \Sigma_{\text{AR}}$ and terms from $\bar{x} \cup \bar{y} \cup \Sigma_{\mathcal{C}}$ such that each variable mentioned in these atoms occurs at least once in some atom from $\phi(\bar{x}, \bar{y})$ (*safeness*), and (*v*) $\xi(\bar{x}, \bar{y})$ is a conjunction of inequality atoms of the form $t \neq t'$, with both t and t' terms from $\bar{x} \cup \bar{y} \cup \Sigma_{\mathcal{C}}$.

A *CQ with safe negation (CQ*$^{\neg_s}$*)* (resp. a *CQ with inequalities (CQ*$^{\neq}$*)*, and a *CQ*) is a CQ$^{\neg_s, \neq}$ that does not contain inequality atoms (resp. negated atoms, neither inequality nor negated atoms). Furthermore, for an integer $p \geq 1$, a CQ$^{\neg_s, p}$ (resp. a CQ$^{\neq_p}$) is a CQ$^{\neg_s}$ (resp. a CQ$^{\neq}$) with *at most* p negated atoms (resp. inequality atoms). We use $\mathsf{CQ}^{\neg_s, p}$ (resp. $\mathsf{CQ}^{\neq_p}$) for the class of CQ$^{\neg_s, p}$ (resp. CQ$^{\neq_p}$) queries.

A UCQ$^{\neg_s, \neq}$ (resp. a UCQ$^{\neg_s, 1}$, a UCQ$^{\neq_1}$) is a query of the form $q = \{\bar{x} \mid (\exists \bar{y_1}.\phi(\bar{x}, \bar{y_1}) \wedge \psi(\bar{x}, \bar{y_1}) \wedge \xi(\bar{x}, \bar{y_1})) \vee \ldots \vee (\exists \bar{y_k}.\phi(\bar{x}, \bar{y_k}) \wedge \psi(\bar{x}, \bar{y_k}) \wedge \xi(\bar{x}, \bar{y_k}))\}$ such that $q_i = \{\bar{x} \mid \exists \bar{y_i}.\phi(\bar{x}, \bar{y_i}) \wedge \psi(\bar{x}, \bar{y_i}) \wedge \xi(\bar{x}, \bar{y_i})\}$, also called *a disjunct of* q, is a CQ$^{\neg_s, \neq}$ (resp. a CQ$^{\neg_s, 1}$, a CQ$^{\neq_1}$), for each $i = 1, \ldots, k$. We denote by $\mathsf{UCQ}^{\neg_s, \neq}$ (resp. $\mathsf{UCQ}^{\neg_s, 1}$, $\mathsf{UCQ}^{\neq_1}$) the class of UCQ$^{\neg_s, \neq}$s (resp. UCQ$^{\neg_s, 1}$s, UCQ$^{\neq_1}$s).

Query Answering Given an interpretation $\mathcal{I}$ and a UCQ$^{\neg_s, \neq}$ q, we denote $q(\mathcal{I})$ the evaluation of q over the interpretation $\mathcal{I}$. For a satisfiable *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a UCQ$^{\neg_s, \neq}$ q of arity n, we denote by $ans(q, \mathcal{O})$ the so-called *certain answers of* q *w.r.t.* $\mathcal{O}$, i.e. the set of those n-tuples $\bar{c} = (c_1, \ldots, c_n)$ of constants occurring in $\mathcal{A}$ such that $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q(\mathcal{I})$ holds for every $\mathcal{I} \in mod(\mathcal{O})$. Similarly, we denote by $ansU(q, \mathcal{O})$ the set of *certain answers of* q *w.r.t.* $\mathcal{O}$ *under the* UNA, i.e. the set of those n-tuples $\bar{c} = (c_1, \ldots, c_n)$ of constants occurring in $\mathcal{A}$ such that $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q(\mathcal{I})$ holds for every $\mathcal{I} \in modU(\mathcal{O})$.

Computational Complexity In our proofs, we will often use the Π_2^p-complete problem $\forall\exists$-CNF (Stockmeyer 1976) to show Π_2^p-hardness results. Such a problem is defined as follows. A $\forall\exists$3CNF formula ϕ on a set of propositional variables $X = \{x_1, \ldots, x_n\} \cup Y = \{y_1, \ldots, y_m\}$ is of the form $\forall X.\exists Y.(c_1 \wedge \ldots \wedge c_p)$ such that the variables in X (resp. Y) are universally (resp. existentially) quantified, each clause c_i is a disjunction of exactly three literals, and each literal is either a variable $z \in X \cup Y$ or its negation $\neg z$. For $i = 1, \ldots, p$, we denote by $z_{i,1}$, $z_{i,2}$, $z_{i,3}$ the first, the second, and the third, respectively, variable appearing (either positively or negated) in clause c_i. Additionally, and without loss of generality, we assume that no clause mentions the same variable twice. The problem $\forall\exists$-CNF decides, given a $\forall\exists$3CNF formula ϕ, whether ϕ is true.

Decision Problem of Interest We are interested in the following family of decision problems associated with query answering, which is parametric w.r.t. an ontology language $\mathcal{L} \subseteq$ *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ and a query language $\mathcal{Q} \subseteq \mathsf{UCQ}^{\neg_s, \neq}$:

ans$(\mathcal{L}, \mathcal{Q})$

Input: An $\mathcal{L}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, a query $q \in \mathcal{Q}$ of arity n, and an n-tuple $\bar{c}$ of constants occurring in $\mathcal{A}$.

Question: Is $\bar{c} \in ans(q, \mathcal{O})$?

The UNA version family of decision problems, denoted by $\mathsf{ansU}(\mathcal{L}, \mathcal{Q})$, is defined as $\mathsf{ans}(\mathcal{L}, \mathcal{Q})$ except that the question becomes: Is $\bar{c} \in ansU(q, \mathcal{O})$?

We are also interested in the *data complexity* (Vardi 1982) version of these two families of decision problems, which is the complexity where only the ABox $\mathcal{A}$ is regarded as the input, and the other components are assumed to be fixed.

3 An Algorithm for Answering Queries with Safe Negation and Inequalities

In this section, we provide a nondeterministic algorithm for answering UCQ$^{\neg_s, \neq}$s over *DL-Lite*$^{\neg}_{\texttt{RDFS}}$ ontologies.

Consider a *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a UCQ$^{\neg_s,\neq}$ q. Let $dom(\mathcal{A})$ and $sig(\mathcal{O}, q)$ denote, respectively, the set of constants occurring in $\mathcal{A}$ and the set of atomic concepts and atomic roles mentioned in $\mathcal{O}$ and q. We say that an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ is *local w.r.t.* $\mathcal{O}$ *and* q if (*i*) $\Delta^{\mathcal{I}} \subseteq dom(\mathcal{A}) \cup \Sigma_{\mathcal{C}}(q)$, where $\Sigma_{\mathcal{C}}(q)$ denotes the set of constants occurring in q and (*ii*) $S^{\mathcal{I}} = \emptyset$ for every $S \notin sig(\mathcal{O}, q)$. Observe that a local interpretation $\mathcal{I}$ naturally induces an equivalence relation $E_{\mathcal{I}}$ over $dom(\mathcal{A}) \cup \Sigma_{\mathcal{C}}(q)$, that is, $(c_1, c_2) \in E_{\mathcal{I}}$ if and only if $c_1^{\mathcal{I}} = c_2^{\mathcal{I}}$, with the elements in $\Delta^{\mathcal{I}}$ being the objects representing the equivalence classes in $E_{\mathcal{I}}$. The next proposition states that it is enough to look only at local interpretations when dealing with the problem of answering UCQ$^{\neg_s,\neq}$s over *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontologies.

Proposition 1. *Let* $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ *be a DL-Lite*$^{\neg}_{RDFS}$ *ontology,* q *be a UCQ*$^{\neg_s,\neq}$ *of arity* n*, and* $\bar{c} = (c_1, \dots, c_n)$ *be an* n*-tuple of constants from* $dom(\mathcal{A})$*. We have that* $\bar{c} \notin ans(q, \mathcal{O})$ *if and only if there exists a local interpretation* $\mathcal{I}$ *w.r.t.* $\mathcal{O}$ *and* q *such that* (i) $\mathcal{I} \in mod(\mathcal{O})$ *and* (ii) $(c_1^{\mathcal{I}}, \dots, c_n^{\mathcal{I}}) \notin q(\mathcal{I})$.

Example 1. *Let* $\mathcal{T} = \{L \sqsubseteq \neg H\}$, $\mathcal{A} = \{L(b), L(c_1), L(a_1), P(b, c_1), P(b, c_2), P(c_1, a_1), P(c_2, a_2), P(f, e), P(f, d), S(b, f), S(c_1, a_2), S(f, a_2), H(a_2), H(e), H(d)\}$, *and* $q = \{() \mid \exists y, y_1, y_2, z. L(y) \wedge P(y, y_1), P(y, y_2) \wedge S(y, z) \wedge \neg L(z) \wedge y_1 \neq y_2\}$*. One can see that* $\langle \mathcal{T}, \mathcal{A} \rangle \not\models q$ *as there is local interpretation* $\mathcal{I}$ *w.r.t.* q *and* $\langle \mathcal{T}, \mathcal{A} \rangle$ *with* $f^{\mathcal{I}} \in L^{\mathcal{I}}$ *and* $e^{\mathcal{I}} = d^{\mathcal{I}}$ *such that* $\mathcal{I} \in mod(\mathcal{O})$ *and* $\mathcal{I} \not\models q$.

Conversely, one can see that $\langle \mathcal{T}, \mathcal{A}' \rangle \models q$*, where* $\mathcal{A}' = \mathcal{A} \cup \{H(f)\}$*. Indeed,* (i) *there can be no model* $\mathcal{I}$ *with* $f^{\mathcal{I}} \in L^{\mathcal{I}}$ *and* (ii) *any model* $\mathcal{I}'$ *with* $c_1^{\mathcal{I}'} = c_2^{\mathcal{I}'}$ *is such that* $\mathcal{I}' \models q$ *due to the fact that* $a_2^{\mathcal{I}'} \notin L^{\mathcal{I}'}$ *and* $a_1^{\mathcal{I}'} \neq a_2^{\mathcal{I}'}$.

Actually, the above proposition immediately suggests an algorithm for solving ans(*DL-Lite*$^{\neg}_{\text{RDFS}}$, UCQ$^{\neg_s,\neq}$).

Theorem 1. *ans*(*DL-Lite*$^{\neg}_{RDFS}$, *UCQ*$^{\neg_s,\neq}$) *is in* Π_2^p *in combined complexity and in coNP in data complexity.*

Proof Sketch. By Proposition 1, to show $\bar{c} \notin ans(q, \mathcal{O})$, it is enough to guess a local interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ w.r.t. $\mathcal{O}$ and q (by guessing an equivalence relation $E_{\mathcal{I}}$ over $dom(\mathcal{A}) \cup \Sigma_{\mathcal{C}}(q)$, a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ for each atomic concept $A \in sig(\mathcal{O}, q)$, and a set $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for each atomic role $P \in sig(\mathcal{O}, q)$), and then check whether both (*i*) $\mathcal{I} \in mod(\mathcal{O})$ and (*ii*) $(c_1^{\mathcal{I}}, \dots, c_n^{\mathcal{I}}) \notin q(\mathcal{I})$ hold.

Note that the size of $\mathcal{I}$ is polynomially related to the size of $\mathcal{A}$ and checking whether $\mathcal{I} \in mod(\mathcal{O})$ and $(c_1^{\mathcal{I}}, \dots, c_n^{\mathcal{I}}) \notin q(\mathcal{I})$ are both feasible in AC^0 in the size of $\mathcal{I}$ and with an NP-oracle in the size of $\bar{c}$, $\mathcal{I}$, and q. So, the above algorithm for deciding $\bar{c} \notin ans(q, \mathcal{O})$ runs in nondeterministic polynomial time with an NP-oracle in the size of the input and in nondeterministic polynomial time in the size of $\mathcal{A}$. ☐

From (Gutiérrez-Basulto et al. 2015) and (Cima, Lenzerini, and Poggi 2020) it is known that, respectively, two negated atoms and two inequalities are enough for getting matching lower bounds regarding the data complexity. In the remaining sections of this paper, we show that the same applies also regarding the combined complexity (thus proving that all the matching lower bounds for the above result already hold for the classes CQ$^{\neg_s,2}$ and CQ$^{\neq_2}$ of queries). On the other hand, we show that for the class of UCQ$^{\neg_s}$s having at most one negated atom per disjunct (i.e. UCQ$^{\neg_s,1}$) the problem becomes easier under the usual computational complexity assumption. In particular, the problem becomes NP-complete in combined complexity and P-complete in data complexity, which is exactly the same as for the case of the UCQ$^{\neq_1}$ class (Cima, Lenzerini, and Poggi 2020).

4 Safe Negation

In this section, we first provide matching lower bounds for the results of Theorem 1 and show that they already hold for ans(*DL-Lite*$_{\text{RDFS}}$, UCQ$^{\neg_s,2}$) and ans(*DL-Lite*$^{\neg}_{\text{RDFS}}$, CQ$^{\neg_s,2}$). Conversely, by combining (Rosati 2007, Theorem 14) with results from (Calvanese et al. 2007) and with the facts that *DL-Lite*$^{\neg}_{\text{RDFS}}$ is a sub-logic of *DL-Lite*$_{\mathcal{R}}$ and *DL-Lite*$_{\mathcal{R}}$ is insensitive to the UNA for CQ answering (Artale et al. 2009), we observe that ans(*DL-Lite*$_{\text{RDFS}}$, CQ$^{\neg_s}$) is NP-complete in combined complexity and in AC^0 in data complexity.

We start by addressing ans(*DL-Lite*$_{\text{RDFS}}$, UCQ$^{\neg_s,2}$).

Theorem 2. *ans*(*DL-Lite*$_{RDFS}$, *UCQ*$^{\neg_s,2}$) *is* Π_2^p*-hard in combined complexity and coNP-hard in data complexity.*

We illustrate here a LogSpace reduction from ∀∃-CNF to ans(*DL-Lite*$_{\text{RDFS}}$, UCQ$^{\neg_s,2}$). This will give us the desired Π_2^p-hardness. To simplify the presentation, we first assume that ontologies may contain also ternary predicates in their alphabet, and then we will remove such an assumption.

Let ϕ be a ∀∃3CNF formula containing p clauses. We now construct a *DL-Lite*$_{\text{RDFS}}$ ontology $\mathcal{O}_\phi = \langle \emptyset, \mathcal{A}_\phi \rangle$ and a Boolean UCQ$^{\neg_s,2}$ q_ϕ, where $\mathcal{T} = \emptyset$ is the fixed TBox without assertions and $q_\phi = q_B \vee q'_\phi$ with q_B being the fixed Boolean CQ$^{\neg_s,2}$ $q_B = \exists w. U(w) \wedge \neg F(w) \wedge \neg T(w)$.

We now describe how to construct q'_ϕ from ϕ. For every $i = 1, \dots, p$, we include the atom $P_i(w_1, w_2, w_3)$, where, for $j = 1, 2, 3$, w_j is the variable y_k if $z_{i,j} = y_k$ with $y_k \in Y$ and w_j is the variable x_k^i if $z_{i,j} = x_k$ with $x_k \in X$. Additionally, we include the atom $T(x_k^i)$ (resp. $F(x_k^i)$) if the variable x_k appears positively (resp. negated) in c_i.

Example 2. *If* ϕ *contains the clauses* $c_4 = \neg x_1 \vee x_2 \vee y_1$ *and* $c_5 = \neg y_1 \vee y_2 \vee \neg x_2$*, then* q'_ϕ *contains the atoms* $P_4(x_1^4, x_2^4, y_1)$, $F(x_1^4)$, $T(x_2^4)$, $P_5(y_1, y_2, x_2^5)$*, and* $F(x_2^5)$.

We now describe how to construct $\mathcal{A}_\phi$ from ϕ. For every $i = 1, \dots, n$, we make use of the two individuals u_i and $x_{i,\star}$ to represent the universally quantified variable x_i, such that $U(u_i)$, $T(x_{i,\star})$, and $F(x_{i,\star})$ are ground atoms occurring in $\mathcal{A}_\phi$. Furthermore, for every $i = 1, \dots, m$, we make use of the two individuals $y_{i,T}$ and $y_{i,F}$ to represent the existentially quantified variable y_i. Now, for every clause c_i, consider the sets $Z_{i,1}$, $Z_{i,2}$, and $Z_{i,3}$ such that, for $j = 1, 2, 3$, $Z_{i,j} = \{u_k, x_{k,\star}\}$ if $z_{i,j} = x_k$ with $x_k \in X$ and $Z_{i,j} = \{y_{k,T}, y_{k,F}\}$ if $z_{i,j} = y_k$ with $y_k \in Y$. We include into the ABox $\mathcal{A}_\phi$ all the possible ground atoms of the form $P_i(t_1, t_2, t_3)$ such that $t_j \in Z_{i,j}$, for $j = 1, 2, 3$, and one of the following three conditions holds:

- $t_j = u_k$ for some $j = 1, 2, 3$ and individual u_k;

- $t_j = y_{k,T}$ for some $j = 1, 2, 3$ and individual $y_{k,T}$ with y_k appearing positively in c_i (in the j-th position);
- $t_j = y_{k,F}$ for some $j = 1, 2, 3$ and individual $y_{k,F}$ with y_k appearing negated in c_i (in the j-th position).

In other words, from all the possible ground atoms $P_i(t_1, t_2, t_3)$ with $t_j \in Z_{i,j}$ for $j = 1, 2, 3$, we discard from $\mathcal{A}_\phi$ only those such that (i) the assignment for the existentially quantified variables do not make the clause true and (ii) there is no u individual in the terms of the atom.

Example 3. *If ϕ contains the clauses $c_4 = \neg x_1 \lor x_2 \lor y_1$ and $c_5 = \neg y_1 \lor y_2 \lor \neg x_2$, then $\mathcal{A}_\phi$ contains the ground atoms $U(u_1)$, $T(x_{1,\star})$, $F(x_{1,\star})$, $U(u_2)$, $T(x_{2,\star})$, $F(x_{2,\star})$, $T(y_{1,T})$, $F(y_{1,F})$, $T(y_{2,T})$, $F(y_{2,F})$ plus the following ground atoms to represent the clauses c_4 and c_5:*
$P_4(u_1, x_{2,\star}, y_{1,T})$ $P_4(u_1, x_{2,\star}, y_{1,F})$ $P_4(u_1, u_2, y_{1,T})$
$P_4(u_1, u_2, y_{1,F})$ $P_4(x_{1,\star}, u_2, y_{1,T})$ $P_4(x_{1,\star}, u_2, y_{1,F})$
$P_4(x_{1,\star}, x_{2,\star}, y_{1,T})$
$P_5(y_{1,F}, y_{2,T}, x_{2,\star})$ $P_5(y_{1,F}, y_{2,T}, u_2)$ $P_5(y_{1,F}, y_{2,F}, x_{2,\star})$
$P_5(y_{1,F}, y_{2,F}, u_2)$ $P_5(y_{1,T}, y_{2,T}, x_{2,\star})$ $P_5(y_{1,T}, y_{2,T}, u_2)$
$P_5(y_{1,T}, y_{2,F}, u_2)$.

We now show how to modify the presented construction to work with only unary and binary predicates. We modify the query q'_ϕ as follows: for every clause c_i of ϕ, we introduce a new variable w_c^i and replace the atom $P_i(w_1, w_2, w_3)$ (with w_1, w_2, and w_3 as previously defined) with the binary atoms $P_{i,1}(w_c^i, w_1)$, $P_{i,2}(w_c^i, w_2)$, and $P_{i,3}(w_c^i, w_3)$. Finally, we modify the ABox $\mathcal{A}_\phi$ as follows: for every clause c_i of ϕ, we introduce a new individual t_c^i and replace every ground atom of the form $P_i(t_1, t_2, t_3)$ (with t_1, t_2, and t_3 as previously defined) with the atoms $P_{i,1}(t_c^i, t_1)$, $P_{i,2}(t_c^i, t_2)$, and $P_{i,3}(t_c^i, t_3)$. Clearly, both $\mathcal{A}_\phi$ and q'_ϕ can be constructed in LOGSPACE from a $\forall\exists$3CNF formula ϕ. Furthermore, it is not hard to verify (see the appendix) that, given a $\forall\exists$3CNF formula ϕ, we have that $\mathcal{O}_\phi \models q_\phi$ if and only if ϕ is true.

We now turn to address $\mathsf{ans}(\textit{DL-Lite}^{\neg}_{\text{RDFS}}, \mathsf{CQ}^{\neg_{s,2}})$.

Theorem 3. *$\mathsf{ans}(\textit{DL-Lite}^{\neg}_{\text{RDFS}}, \mathsf{CQ}^{\neg_{s,2}})$ is Π_2^p-hard in combined complexity and coNP-hard in data complexity.*

The coNP-hardness comes from (Gutiérrez-Basulto et al. 2015, Lemma 11). The Π_2^p-hardness can be shown through an extension of the previously presented reduction, which we describe here. To simplify the presentation, we first assume that ontologies may contain quaternary predicates in their alphabet, and then we will remove such an assumption.

As before, let ϕ be a $\forall\exists$3CNF formula. We now construct a *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontology $\mathcal{O}_\phi = \langle \mathcal{T}, \mathcal{A}_\phi \rangle$ and a Boolean CQ$^{\neg_{s,2}}$ q_ϕ, where $\mathcal{T}$ is the fixed TBox $\mathcal{T} = \{H \sqsubseteq \neg T,\ H \sqsubseteq \neg F\}$. As for q_ϕ, we have the atoms $U(r, v)$, $\neg T(r)$, and $\neg F(r)$ with r and v variables. Furthermore, for every $i = 1, \ldots, p$, we include the atom $P_i(v, w_1, w_2, w_3)$, where w_j is exactly as in the previous reduction, for $j = 1, 2, 3$.

Example 4. *If ϕ contains the clauses $c_4 = \neg x_1 \lor x_2 \lor y_1$ and $c_5 = \neg y_1 \lor y_2 \lor \neg x_2$, then q_ϕ contains the fixed atoms $U(r, v)$, $\neg T(r)$, and $\neg F(r)$ plus the atoms $P_4(v, x_1^4, x_2^4, y_1)$, $F(x_1^4)$, $T(x_2^4)$, $P_5(v, y_1, y_2, x_2^5)$, and $F(x_2^5)$.*

We now describe how to construct $\mathcal{A}_\phi$ from ϕ. We have the fixed ground atoms $U(s, d)$ and $H(s)$ and, for every $i = 1, \ldots, n$, we have the ground atoms $U(u_i, \mathsf{nop})$, $T(x_{i,\star})$ and $F(x_{i,\star})$. For every clause c_i, we have the ground atoms $P_i(d, t_1, t_2, t_3)$, where t_1, t_2, and t_3 are exactly as in the previous reduction and must satisfy one of the three conditions illustrated in the previous reduction. Intuitively, consider a model $\mathcal{I}$ for which each u_i is such that either $u_i^{\mathcal{I}} \in T^{\mathcal{I}}$ or $u_i^{\mathcal{I}} \in F^{\mathcal{I}}$ hold. This model will satisfy the query (by assigning $s^{\mathcal{I}}$ and $d^{\mathcal{I}}$ to the variables r and v, respectively) if and only if there exists a truth assignment to the variables in Y that makes ϕ true when the variables in X are assigned as the model $\mathcal{I}$ dictates with the various u constants.

It thus remains to include into $\mathcal{A}_\phi$ those ground atoms that will make the query trivially true for those models such that both $u_i^{\mathcal{I}} \notin T^{\mathcal{I}}$ and $u_i^{\mathcal{I}} \notin F^{\mathcal{I}}$ for some $i = 1, \ldots, n$. To this aim, we can just include the atoms $F(\mathsf{nop})$ and $T(\mathsf{nop})$ as well as the atom $P_i(\mathsf{nop}, \mathsf{nop}, \mathsf{nop}, \mathsf{nop})$, for every clause c_i.

To conclude, we now show how to modify the presented construction to work with only unary and binary predicates. We modify q_ϕ as follows: for every clause c_i of ϕ, we introduce a new variable w_c^i and replace $P_i(v, w_1, w_2, w_3)$ (with w_1, w_2, and w_3 as previously defined) with the atoms $P_{i,0}(w_c^i, v)$, $P_{i,1}(w_c^i, w_1)$, $P_{i,2}(w_c^i, w_2)$, and $P_{i,3}(w_c^i, w_3)$. We modify the ABox $\mathcal{A}_\phi$ as follows: for every clause c_i of ϕ, we introduce two new individuals t_c^i and nop_c^i and replace (i) the ground atom $P_i(\mathsf{nop}, \mathsf{nop}, \mathsf{nop}, \mathsf{nop})$ with the ground atoms $P_{i,0}(\mathsf{nop}_c^i, \mathsf{nop})$ $P_{i,1}(\mathsf{nop}_c^i, \mathsf{nop})$, $P_{i,2}(\mathsf{nop}_c^i, \mathsf{nop})$, and $P_{i,3}(\mathsf{nop}_c^i, \mathsf{nop})$ (ii) every ground atom of the form $P_i(d, t_1, t_2, t_3)$ (with t_1, t_2, and t_3 as previously defined) with the atoms $P_{i,0}(t_c^i, d)$ $P_{i,1}(t_c^i, t_1)$, $P_{i,2}(t_c^i, t_2)$, and $P_{i,3}(t_c^i, t_3)$. Clearly, both $\mathcal{A}_\phi$ and q_ϕ can be constructed in LOGSPACE from a $\forall\exists$3CNF formula ϕ. Furthermore, it is not hard to verify (see the appendix) that, given a $\forall\exists$3CNF formula ϕ, we have that $\mathcal{O}_\phi \models q_\phi$ if and only if ϕ is true.

4.1 One Safe Negation

We now show that $\mathsf{ans}(\textit{DL-Lite}^{\neg}_{\text{RDFS}}, \mathsf{UCQ}^{\neg_{s,1}})$ can be polynomially reduced to checking whether a *Datalog program* (Ceri, Gottlob, and Tanca 1989) entails a ground atom.

Let $\mathcal{O}$ be a *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontology, q a UCQ$^{\neg_{s,1}}$, and $\bar{c}$ a tuple of constants. Next, we discuss how to translate $\mathcal{O}$, q, and $\bar{c}$ into a suitable program to check whether $\bar{c} \in \mathit{ans}(q, \mathcal{O})$.

Assume a CQ$^{\neg_{s,1}}$ query q' of the form $\{\bar{x} \mid \exists \bar{y}.\phi(\bar{x}, \bar{y}) \land \psi(\bar{x}, \bar{y})\}$. We denote by $\sigma_{q'(\bar{c})}$ the Datalog rule $\forall y.\phi(\bar{c}, \bar{y}) \to \eta$, where η is the propositional atom w, if $\psi(\bar{x}, \bar{y})$ is empty (i.e. q is a CQ), and $\eta = \alpha(\bar{c}, \bar{y})$ if $\psi(\bar{c}, \bar{y}) = \neg\alpha(\bar{x}, \bar{y})$, where α is a single relational atom. For a UCQ$^{\neg_{s,1}}$ q, we use $\sigma_{q(\bar{c})}$ to denote the set $\{\sigma_{q_i(\bar{c})} \mid q_i \text{ is a disjunct of } q\}$.

We now consider the ontology $\mathcal{O}$. For an inclusion assertion o of the form $A_1 \sqsubseteq A$ (resp. $\exists R \sqsubseteq A$, $R_1 \sqsubseteq R_2$), we let σ_o be $\forall x.A_1(x) \to A(x)$ (resp. $\forall x_1, x_2.R(x_1, x_2) \to A(x_1)$, $\forall x_1, x_2.R_1(x_1, x_2) \to R_2(x_1, x_2)$). For a disjointness assertion o of the form $A_1 \sqsubseteq \neg A_2$ (resp. $\exists R \sqsubseteq \neg A$ or $A \sqsubseteq \neg\exists R$, $\exists R_1 \sqsubseteq \neg\exists R_2$, $R_1 \sqsubseteq \neg R_2$), we let σ_o be $\forall x.A_1(x) \land A_2(x) \to w$ (resp. $\forall x_1, x_2.R(x_1, x_2) \land A(x_1) \to w$, $\forall x_1, x_2, x_3.R_1(x_1, x_2) \land R_2(x_1, x_3) \to w$, $\forall x_1, x_2.R_1(x_1, x_2) \land R_2(x_1, x_2) \to w$). An atom $R(t, t')$ stands for $P(t, t')$ if R is a role $P \in \Sigma_{\text{AC}}$ and for $P(t', t)$ if $R = P^-$ for a role $P \in \Sigma_{\text{AC}}$. Given a *DL-Lite*$^{\neg}_{\text{RDFS}}$ TBox $\mathcal{T}$, we use $\sigma_{\mathcal{T}}$ to denote the set $\{\sigma_o \mid o \in \mathcal{T}\}$.

Additionally, given an ABox $\mathcal{A}$, we let $\sigma_{\mathcal{A}}$ be the set of *Datalog facts* corresponding to $\mathcal{A}$, i.e. $\sigma_{\mathcal{A}} = \{\rightarrow A(c) \mid A(c) \in \mathcal{A}\} \cup \{\rightarrow P(c_1, c_2) \in \mathcal{A}\}$. Finally, we use $\mathcal{P}_{q(\bar{c}),\mathcal{T},\mathcal{A}}$ for the Datalog program $\sigma_{q(\bar{c})} \cup \sigma_{\mathcal{T}} \cup \sigma_{\mathcal{A}}$.

Example 5. *Recall Example 1, and let q' be the CQ$^{\neg_1}$ obtained from q by removing the inequality atom $y_1 \neq y_2$. Then $\sigma_{q'()} \cup \sigma_{\mathcal{T}} = \{\forall x.L(x) \wedge H(x) \rightarrow w,\ \forall y, y_1, y_2, z.L(y) \wedge P(y, y_1) \wedge P(y, y_2) \wedge S(y, z) \rightarrow L(z)\}$. One can verify that $\langle \mathcal{T}, \mathcal{A} \rangle \models q'$ and that $\mathcal{P}_{q'(),\mathcal{T},\mathcal{A}} \models w$.*

Proposition 2. *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$^{\neg}_{RDFS}$ ontology, q be a UCQ$^{\neg_{s,1}}$ of arity n, and $\bar{c}$ be an n-tuple of constants from $dom(\mathcal{A})$. We have that $\bar{c} \in ans(q, \mathcal{O})$ if and only if the Datalog program $\mathcal{P}_{q(\bar{c}),\mathcal{T},\mathcal{A}}$ is such that $\mathcal{P}_{q(\bar{c}),\mathcal{T},\mathcal{A}} \models w$.*

By combining the above property with the well-known results concerning the complexity of Datalog programs with predicates of bounded arity (see, e.g. (Dantsin et al. 2001; Gottlob and Schwentick 2012)), we obtain the next result. Note that NP-hardness trivially follows from NP-hardness of the evaluation of CQs over relational databases (Abiteboul, Hull, and Vianu 1995), while P-hardness in data complexity comes from (Gutiérrez-Basulto et al. 2015, Lemma 10).

Theorem 4. *ans(DL-Lite$^{\neg}_{RDFS}$, $\mathsf{UCQ}^{\neg_{s,1}}$) is NP-complete in combined complexity and P-complete in data complexity. The hardnesses already hold for ans(DL-Lite$^{\neg}_{RDFS}$, $\mathsf{CQ}^{\neg_{s,1}}$).*

5 Inequalities

The work in (Cima, Lenzerini, and Poggi 2020) provides a thorough analysis of the problem of answering (U)CQ$^{\neq}$s over *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontologies analogous to the one of Section 4 for the case of (U)CQ$^{\neg_s}$s. However, the Π^p_2-hardness of ans(*DL-Lite*$^{\neg}_{\text{RDFS}}$, $\mathsf{CQ}^{\neq_2}$) was only conjectured there. In this section, we verify such conjecture, thus providing a fairly complete picture of the complexity of the problem at hand.

Theorem 5. *ans(DL-Lite$^{\neg}_{RDFS}$, $\mathsf{CQ}^{\neq_2}$) is Π^p_2-hard in combined complexity.*

We illustrate here a LogSpace reduction from $\forall\exists$-CNF to ans(*DL-Lite*$^{\neg}_{\text{RDFS}}$, $\mathsf{CQ}^{\neq_2}$). Let $\phi = \forall\bar{\mathsf{U}}.\exists\bar{\mathsf{E}}. \wedge_{i=1}^{m} (\ell^i_1 \vee \ell^i_2 \vee \ell^i_3)$, where each ℓ^i_j is a literal over $\bar{\mathsf{U}}$ and $\bar{\mathsf{E}}$. We now construct a Boolean $\mathsf{CQ}^{\neq_2}$ q_ϕ and a *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontology $\mathcal{O}_\phi$ such that $\mathcal{O}_\phi \models q_\phi$ if and only if ϕ is true.

The Ontology $\mathcal{O}_\phi$. We first illustrate the alphabet of $\mathcal{O}_\phi$. The *concept and role names* used in $\mathcal{O}_\phi$ are the following.

- Concept names T, F;
- One concept name G_i, for each clause γ_i of ϕ;
- One concept name V_j, for each variable $\mathsf{U}_j \in \bar{\mathsf{U}}$;
- Role names A, C_1, C_0, Sat, L_1, L_2, L_3, and $Type$.

Additionally, $\mathcal{O}_\phi$ uses the following *individual names*.

1. nop, 1, 0,
2. 111, 110, 101, 100, 011, 010, 001, 000;
3. One v_i for each $\mathsf{U}_i \in \bar{\mathsf{U}}$
4. g_i and nop_i, for each clause γ_i of ϕ.

In what follows, we will often use the intentional definition

$$\{*_1 *_2 *_3 \mid \text{ for each } *_1, *_2, *_3 \in \{1, 0\}\}$$

to denote (subsets of) the individual names in Item 2. For example, $\{1 *_2 *_3$, for each $*_2, *_3 \in \{1, 0\}\}$, will denote the set of all the above symbols that start with 1. Moreover, we will use $\mathcal{V}$ for the set of individual names defined in Item 3, i.e., $\mathcal{V} = \{v_i \mid \mathsf{U}_i \in \bar{\mathsf{U}}\}$.

Intuitively, the goal of the ontology $\mathcal{O}_\phi$ is twofold: it simulates all the possible propositional assignments for the universally quantified letters $\bar{\mathsf{U}}$ of ϕ, and it defines those propositional assignments for $\bar{\mathsf{U}} \cup \bar{\mathsf{E}}$ that satisfy the (sub)formula $\wedge_{i=1}^{m} (\ell^i_1 \vee \ell^i_2 \vee \ell^i_3)$.

Towards this goal, the TBox $\mathcal{T}$ of $\mathcal{O}_\phi$ consists of only one assertion: $T \sqsubseteq \neg F$. On the contrary, the ABox $\mathcal{A}$ of $\mathcal{O}_\phi$ consists of two parts: one fixed, denoted by $\mathcal{A}^{fix}$, and one whose definition depends on ϕ, denoted by $\mathcal{A}^{for}$. Firstly, we provide a definition for $\mathcal{A}^{fix}$.

1. $T(1)$, $F(0)$, $A(1, 1)$, $C_1(1, 0)$, $C_0(1, 0)$;
2. $Type(c, 1)$, for each $c \in \{*_1 *_2 *_3 \mid *_1, *_2, *_3 \in \{1, 0\}\}$;
3. $L_1(c, 1)$, for each $c \in \{1 *_2 *_3 \mid *_2, *_3 \in \{1, 0\}\}$;
4. $L_1(c, 0)$, for each $c \in \{0 *_2 *_3 \mid *_2, *_3 \in \{1, 0\}\}$;
5. $L_2(c, 1)$, for each $c \in \{*_1 1 *_3 \mid *_1, *_3 \in \{1, 0\}\}$;
6. $L_2(c, 0)$, for each $c \in \{*_1 0 *_3 \mid *_1, *_3 \in \{1, 0\}\}$;
7. $L_3(c, 1)$, for each $c \in \{*_1 *_2 1 \mid *_1, *_2 \in \{1, 0\}\}$;
8. $L_3(c, 0)$, for each $c \in \{*_1 *_2 0 \mid *_1, *_2 \in \{1, 0\}\}$;

Given a clause γ of ϕ and a symbol $*_1 *_2 *_3$ as above, we define the propositional assignment $\chi^{\gamma}_{*_1 *_2 *_3}$ for the variables of γ as follows: for each $j = 1, 2, 3$, $\chi^{\gamma}_{*_1 *_2 *_3}(x_j) = \mathbf{T}$, if $*_j = 1$, and $\chi^{\gamma}_{*_1 *_2 *_3}(x_j) = \mathbf{F}$, if $*_j = 0$, where x_j is the letter of the j-th literal of γ. For example, given $\gamma = (x \vee \neg y \vee z)$, $\chi^{\gamma}_{101}(x) = \mathbf{T}$, $\chi^{\gamma}_{101}(y) = \mathbf{F}$, and $\chi^{\gamma}_{101}(z) = \mathbf{T}$.

Next, we provide a definition for the subset $\mathcal{A}^{for}$ of $\mathcal{A}$. Specifically, such set contains all the following.

1. $A(v_i, 0)$, $C_1(v_i, 1)$, $C_0(v_i, 0)$, $V_i(v_i)$, for each $v_i \in \mathcal{V}$;
2. $G_i(g_i)$, for each clause γ_i of ϕ;
3. $Sat(g_i, c)$, for each $c \in \{*_1, *_2, *_3 \mid \chi^{\gamma_i}_{*_1 *_2 *_3} \models \gamma_i\}$;
4. $Sat(g_i, \mathsf{nop}_i)$, $Type(\mathsf{nop}_i, 0)$, for each clause g_i of ϕ;
5. $L_i(\mathsf{nop}_j, \mathsf{nop})$ if the variable of the i-th literal of the j-th clause is existentially quantified, for each $i = 1, 2, 3$;
6. $L_i(\mathsf{nop}_j, v_k)$ if the variable of the i-th literal of the j-th clause is the letter $\mathsf{U}_k \in \bar{\mathsf{U}}$, for each $i = 1, 2, 3$.

The Query q_ϕ. Next, we define the query q_ϕ, which uses the following *individual variable symbols*:

1. y_0, y_1, y, t,
2. One x_i, for each propositional letter $\mathsf{X}_i \in \bar{\mathsf{U}} \cup \bar{\mathsf{E}}$;
3. Variables z_i and a_i, for each $\gamma_i \in \phi$.

Similarly to $\mathcal{O}_\phi$, the query q_ϕ consists of two parts: one fixed and another dependent on the formula ϕ. We denote by q^{fix} the sets of atoms of the former, and q^{for} the set of atoms in the latter. The query q_ϕ is defined as $\exists \bar{x}. \bigwedge_{\alpha \in q^{fix}} \alpha \wedge \bigwedge_{\beta \in q^{for}} \beta$, where $\bar{x}$ is the set of all variables occurring in $q^{fix} \cup q^{for}$. The set q^{fix} consists of the following atoms:

1. $A(y,t)$, $C_1(y,y_1)$, $C_0(y,y_0)$,
2. $(y \neq y_1)$, $(y \neq y_0)$.

The set q^{for} consists of the following atoms.

1. $V_i(x_i)$, for each propositional letter $\mathsf{U}_i \in \bar{\mathsf{U}}$
2. $G_i(z_i)$, $Sat(z_i,a_i)$, and $Type(a_i,t)$, for each $\gamma_i \in \phi$;
3. $L_1(a_i,x_{i,1})$, $L_2(a_i,x_{i,2})$, and $L_3(a_i,x_{i,3})$, where $\mathsf{X}_{i,j}$ is the letter of the j-th literal of γ_i, for each $\gamma_i \in \phi$;

Observe that those in Item 2 are the only two inequality atoms in q_ϕ. Thus, $q_\phi \in \mathsf{CQ}^{\neq_2}$ as needed.

Example 6. *We illustrate a concrete example of our construction. Let ϕ be the following formula*

$$\forall \mathsf{U}_1, \mathsf{U}_2 . \exists \mathsf{E}_1, \exists \mathsf{E}_2 . (\mathsf{U}_1 \vee \neg \mathsf{U}_2 \vee \mathsf{E}_1) \wedge (\mathsf{U}_1 \vee \neg \mathsf{E}_1 \vee \neg \mathsf{E}_2)$$

The set $\mathcal{A}^{for}$ consists of the following:

- $A(v_1,0)$, $C_1(v_1,1)$, $C_0(v_1,0)$, $V_1(v_1)$;
- $A(v_2,0)$, $C_1(v_2,1)$, $C_0(v_2,0)$, $V_2(v_2)$;
- $G_1(z_1)$, $G_2(z_2)$;
- $Sat(z_1,000)$, $Sat(z_1,001)$, $Sat(z_1,011)$, $Sat(z_1,100)$, $Sat(z_1,101)$, $Sat(z_1,110)$, $Sat(z_1,111)$;
- $Sat(z_2,000)$, $Sat(z_2,001)$, $Sat(z_2,010)$, $Sat(z_2,100)$, $Sat(z_2,101)$, $Sat(z_2,110)$, $Sat(z_2,111)$;
- $Sat(z_1,\mathsf{nop}_1)$, $Type(\mathsf{nop}_1,0)$, $L_1(\mathsf{nop}_1,v_1)$, $L_2(\mathsf{nop}_1,v_2)$, $L_3(\mathsf{nop}_1,\mathsf{nop})$,
- $Sat(z_2,\mathsf{nop}_2)$, $Type(\mathsf{nop}_2,0)$, $L_1(\mathsf{nop}_2,v_1)$, $L_2(\mathsf{nop}_2,\mathsf{nop})$, $L_3(\mathsf{nop}_2,\mathsf{nop})$;

Similarly, the query q^{for} consists of the following:

- $V_1(u_1)$, $V_2(u_2)$, $G_1(z_1)$, $G_2(z_2)$;
- $Sat(g_1,a_1)$, $Type(a_1,t)$, $L_1(a_1,u_1)$, $L_2(a_1,u_2)$, $L_3(a_1,e_1)$;
- $Sat(g_2,a_2)$, $Type(a_2,t)$, $L_1(a_2,u_1)$, $L_2(a_2,e_1)$, $L_3(a_2,e_2)$.

Clearly, both q_ϕ and $\mathcal{O}_\phi$ can be constructed in LOGSPACE from a $\forall\exists$3CNF formula ϕ. Furthermore, it can be verified (see appendix) that, given a $\forall\exists$3CNF formula ϕ, we have that $\mathcal{O}_\phi \models q_\phi$ if and only if ϕ is true.

5.1 Query Containment in Relational Databases

In the query containment problem $\mathsf{cont}(\mathcal{Q},\mathcal{Q}')$, we are given two queries $q \in \mathcal{Q}$ and $q' \in \mathcal{Q}'$, and the problem is to decide whether $q \sqsubseteq q'$, i.e. $q(D) \subseteq q'(D)$ holds for each possible *database* D, where the latter can be viewed as a finite set of ground relational atoms.

By exploiting Theorem 5 and a polynomial time reduction from $\mathsf{ans}(\textit{DL-Lite}^{\neg}_{\text{RDFS}}, \mathsf{CQ}^{\neq})$ to $\mathsf{cont}(\mathsf{CQ}^{\neq}, \mathsf{CQ}^{\neq})$ (see Proposition 3), we provide a tight separation between NP and Π^p_2 cases on the number of inequality atoms in queries.

Proposition 3. *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$^{\neg}_{RDFS}$ ontology without role disjointness assertions. It is possible to construct in polynomial time a Boolean CQ$^{\neq}$ $q_\mathcal{O}$ such that, for any Boolean CQ$^{\neq}$ q: $\mathcal{O} \models q$ if and only if $q_\mathcal{O} \sqsubseteq q$.*

Proof Sketch. For every constant c occurring in $\mathcal{A}$, we make use of a variable x_c. The CQ$^{\neq}$ $q_\mathcal{O}$ is then the conjunction of the following atoms: (*i*) $A(x_c)$ is an atom of $q_\mathcal{O}$ if and only if $\mathcal{O} \models A(c)$ and $A \in \Sigma_{\text{AC}}$; (*ii*) $P(x_c, x_{c'})$ is an atom of $q_\mathcal{O}$ if and only if $\mathcal{O} \models P(c,c')$ and $P \in \Sigma_{\text{AR}}$; (*iii*) $x_c \neq x_{c'}$ is an atom of $q_\mathcal{O}$ if and only if there are basic concepts B and B' such that $\mathcal{O} \models B(c)$, $\mathcal{O} \models B'(c')$, and $\mathcal{T} \models B \sqsubseteq \neg B'$. ☐

Notably, while $\mathsf{cont}(\mathsf{CQ}^{\neq_a}, \mathsf{CQ}^{\neq_b})$ is NP-complete if either $a = 0$ or $b \leq 1$ (Chandra and Merlin 1977; Cima, Lenzerini, and Poggi 2020), by looking at the reduction used to prove Theorem 5 and the reduction illustrated in the proof of Proposition 3, it is not hard to get the next result. We recall that membership in Π^p_2 comes from (Klug 1988).

Theorem 6. *$\mathsf{cont}(\mathsf{CQ}^{\neq_1}, \mathsf{CQ}^{\neq_2})$ is Π^p_2-complete.*

6 Unique Name Assumption

The same upper bounds of Theorem 1 can be derived also under the UNA. To this end, we say that an interpretation $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ is *U-local w.r.t. a DL-Lite$^{\neg}_{RDFS}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a $\mathsf{UCQ}^{\neg_s,\neq}$* q if (*i*) $c^\mathcal{I} = c$ for each $c \in \Sigma_\mathcal{C}$, (*ii*) $A^\mathcal{I} \subseteq dom(\mathcal{A})$ for each atomic concept $A \in sig(\mathcal{O},q)$ and $P^\mathcal{I} \subseteq dom(\mathcal{A}) \times dom(\mathcal{A})$ for each atomic role $P \in sig(\mathcal{O},q)$, and (*iii*) $S^\mathcal{I} = \emptyset$ for every $S \notin sig(\mathcal{O},q)$.

Theorem 7. *$\mathsf{ansU}(\textit{DL-Lite}^{\neg}_{RDFS}, \mathsf{UCQ}^{\neg_s,\neq})$ is in Π^p_2 in combined complexity and in coNP in data complexity.*

Proof Sketch. Similarly to Proposition 1, it is possible to show that $\bar{c} \notin ansU(q,\mathcal{O})$ if and only if there exists a U-local interpretation $\mathcal{I}$ w.r.t. $\mathcal{O}$ and q such that (*i*) $\mathcal{I} \in modU(\mathcal{O})$ and (*ii*) $(c_1^\mathcal{I}, \ldots, c_n^\mathcal{I}) \notin q(\mathcal{I})$. Based on this property, we can employ a nondeterministic algorithm analogous to the one illustrated in the proof of Theorem 1. ☐

Complexity lower bounds for UCQ$^{\neg_s}$ under the UNA can be obtained from the fact that *DL-Lite*$^{\neg}_{\text{RDFS}}$ is insensitive to the UNA for UCQ$^{\neg_s}$ answering, as stated by the next result.

Proposition 4. *Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$^{\neg}_{\text{RDFS}}$ ontology and q be a UCQ$^{\neg_s}$. We have that $ans(q,\mathcal{O}) = ansU(q,\mathcal{O})$.*

By combining Proposition 4 with the construction presented in Section 4, one can easily obtain results analogous to Theorems 2, 3, and 4 for the case of the UNA.

Conversely, it is possible to show that answering UCQ$^{\neq}$s over *DL-Lite*$^{\neg}_{\text{RDFS}}$ is much easier under the UNA than without the UNA, and in fact also tractable in data complexity.

Theorem 8. *$\mathsf{ansU}(\textit{DL-Lite}^{\neg}_{\text{RDFS}}, \mathsf{UCQ}^{\neq})$ is NP-complete in combined complexity and in AC^0 in data complexity.*

7 Conclusions

In this work, we presented a thorough analysis of the complexity of answering queries with safe negation and inequalities over *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontologies. This extension of the widely used (DL counterpart of) `RDFS` with disjointness assertions has the potential to represent a strong theoretical underpinning for AI systems based on Knowledge Graphs. Additionally, one of our results contributes to the extensively studied problem of query containment for CQs with inequalities. This result provides a tight complexity characterization based on the number of inequality atoms in the queries.

As for future work, we observe that decidability of $\mathsf{ans}(\textit{DL-Lite}_{core}, \mathsf{CQ}^{\neq})$ is still open today. Also, answering UCQ$^{\neg_s,\neq}$s over *DL-Lite*$^{\neg}_{\text{RDFS}}$ ontologies could be implemented in practice via systems that solve Π^p_2-complete problems, e.g. ASP solvers (Lifschitz 2019).

Acknowledgements

This work has been supported by PNRR MUR project PE0000013-FAIR.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison Wesley Publishing Company.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The *DL-Lite* Family and Relations. *Journal of Artificial Intelligence Research*, 36: 1–69.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Bienvenu, M.; and Ortiz, M. 2015. Ontology-Mediated Query Answering with Data-Tractable Description Logics. In *Reasoning Web. Semantic Technologies for Intelligent Data Access – Eleventh International Summer School Tutorial Lectures (RW 2015)*, volume 9203 of *Lecture Notes in Computer Science*, 218–307.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family. *Journal of Automated Reasoning*, 39(3): 385–429.

Ceri, S.; Gottlob, G.; and Tanca, L. 1989. What you Always Wanted to Know About Datalog (And Never Dared to Ask). *IEEE Transactions on Knowledge and Data Engineering*, 1(1): 146–166.

Chandra, A. K.; and Merlin, P. M. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of the Ninth ACM Symposium on Theory of Computing (STOC 1977)*, 77–90.

Cima, G.; Lenzerini, M.; and Poggi, A. 2020. Answering Conjunctive Queries with Inequalities in *DL-Lite*$_\mathcal{R}$. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, 2782–2789.

Cima, G.; Poggi, A.; and Lenzerini, M. 2023. The notion of Abstraction in Ontology-based Data Management. *Artificial Intelligence*, 323: 103976.

Cuenca Grau, B. 2004. A possible simplification of the semantic web architecture. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004)*, 704–713.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and Expressive Power of Logic Programming. *ACM Computing Surveys*, 33(3): 374–425.

Glimm, B.; Lutz, C.; Horrocks, I.; and Sattler, U. 2008. Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$. *Journal of Artificial Intelligence Research*, 31: 151–198.

Gottlob, G.; and Schwentick, T. 2012. Rewriting Ontological Queries into Small Nonrecursive Datalog Programs. In *Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012)*.

Gutiérrez-Basulto, V.; Ibáñez-García, Y. A.; Kontchakov, R.; and Kostylev, E. V. 2015. Queries with negation and inequalities over lightweight ontologies. *Journal of Web Semantics*, 35: 184–202.

Khan, A. 2023. Knowledge Graphs Querying. *SIGMOD Record*, 52(2): 18–29.

Klug, A. C. 1988. On Conjunctive Queries Containing Inequalities. *Journal of the ACM*, 35(1): 146–160.

Kolaitis, P. G.; Martin, D. L.; and Thakur, M. N. 1998. On the Complexity of the Containment Problem for Conjunctive Queries with Built-in Predicates. In *Proceedings of the Seventeenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 1998)*, 197–204.

Lifschitz, V. 2019. *Answer Set Programming*. Springer.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7): 3580–3599.

Rosati, R. 2007. The Limits of Querying Ontologies. In *Proceedings of the Eleventh International Conference on Database Theory (ICDT 2007)*, volume 4353 of *Lecture Notes in Computer Science*, 164–178.

Stockmeyer, L. J. 1976. The Polynomial-Time Hierarchy. *Theoretical Computer Science*, 3(1): 1–22.

van der Meyden, R. 1997. The Complexity of Querying Indefinite Data about Linearly Ordered Domains. *Journal of Computer and System Sciences*, 54(1): 113–135.

Vardi, M. Y. 1982. The Complexity of Relational Query Languages (Extended Abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC 1982)*, 137–146.

Xiao, G.; Ding, L.; Cogrel, B.; and Calvanese, D. 2019. Virtual Knowledge Graphs: An Overview of Systems and Use Cases. *Data Intelligence*, 1(3): 201–223.

Yang, L.; Chen, H.; Li, Z.; Ding, X.; and Wu, X. 2024. Give us the Facts: Enhancing Large Language Models With Knowledge Graphs for Fact-Aware Language Modeling. *IEEE Transactions on Knowledge and Data Engineering*, 36(7): 3091–3110.