Answering Conjunctive Queries with Inequalities in $DL$-Lite$_R$

Gianluca Cima, Maurizio Lenzerini, Antonella Poggi

1Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma
2Dipartimento di Lettere e Culture Moderne, Sapienza Università di Roma
{cima, lenzerini, poggi}@diag.uniroma1.it

Abstract

In the context of the Description Logic $DL$-Lite$_R^\neq$, i.e., $DL$-Lite$_R$ without UNA and with inequality axioms, we address the problem of adding to unions of conjunctive queries (UCQs) one of the simplest forms of negation, namely, inequality. It is well known that answering conjunctive queries with unrestricted inequalities over $DL$-Lite$_R$ ontologies is in general undecidable. Therefore, we explore two strategies for recovering decidability, and, hopefully, tractability. Firstly, we weaken the ontology language, and consider the variant of $DL$-Lite$_R^\neq$ corresponding to RDFS enriched with both inequality and disjunction axioms. Secondly, we weaken the query language, by preventing inequalities to be applied to existentially quantified variables, thus obtaining the class of queries named UCQ$^\neq,b$. We prove that in the two cases, query answering is decidable, and we provide tight complexity bounds for the problem, both for data and combined complexity. Notably, the results show that answering UCQ$^\neq,b$ over $DL$-Lite$_R^\neq$ ontologies is still in AC$^0$ in data complexity.

1 Introduction

Description Logics (DLs) (Baader et al. 2003; 2017) allow for defining ontologies in terms of two components, named TBox (general axioms on the concepts and relations in the domain of interest), and ABox (axioms about instances of concepts and relations). In this paper we consider $DL$-Lite$_R$, which is the DL of the $DL$-Lite family (Calvanese et al. 2004; 2007) underpinning the OWL 2 profile OWL 2 QL (Motik et al. 2012), and is arguably one of the most important formalisms in Ontology-Based Data Access (OBDA) (Poggi et al. 2008; Bienvenu 2016; Xiao et al. 2018; Ortiz 2018), where the aim is to access a typically huge amount of data represented as an ABox, either materialized or virtual. In particular, $DL$-Lite$_R$ has been designed so that answering unions of conjunctive queries (UCQs) posed to an ontology expressed in this language can be reduced to evaluating first-order logic queries over the database corresponding to the ABox, and therefore the problem is in AC$^0$ in the size of the ABox, i.e., in the so-called data complexity (Vardi 1982).

Although UCQs constitute the most popular class of queries studied for both databases and ontologies, they have several limitations in expressive power. Notably, they do not allow any form of negation, not even the one expressed by the inequality (i.e., “not equal”) predicate. For instance, the query computing all triangles in an undirected graph cannot be expressed as a conjunctive query (CQ), whereas it can be expressed as the following CQ with inequalities $\{ (x, y, z) | \text{edge}(x, y), \text{edge}(x, z), \text{edge}(y, z), x \neq y, x \neq z, y \neq z \}$, where the predicate edge represents the connections between nodes in the graph.

The above example shows that inequalities are indeed necessary for expressing even very simple properties, like triangle graphs. However, while answering UCQs in $DL$-Lite$_R$ has been extensively studied in recent years (Xiao et al. 2018), the problem of answering CQs with inequalities (CQ$^\neq,b$s) and unions thereof (UCQ$^\neq,b$s) has been rarely investigated. To the best of our knowledge, the basic facts that are known about such problem can be summarized as follows.

- In stark contrast to the UCQ case, answering UCQ$^\neq,b$s is undecidable, even in the case of ontologies expressed in $DL$-Lite$_{core}$, which is the fragment of $DL$-Lite$_R$ without role disjointness and role inclusion axioms (Gutiérrez-Basulto, Ibáñez-García, and Kontchakov 2012). For $DL$-Lite$_R$ ontologies, undecidability holds already for CQ$^\neq,b$s (Gutiérrez-Basulto et al. 2015). Looking at these results, one can easily realize that the main of source of undecidability stems from both the ability of the ontology language to express incomplete information through existential quantifiers, and the possibility of imposing inequalities between existential variables in the query.

- In (Gutiérrez-Basulto et al. 2015) it is also proved that for the subclasses of CQ$^\neq,b$s and UCQ$^\neq,b$s named local CQ$^\neq,b$s and local UCQ$^\neq,b$s, respectively, query answering over $DL$-Lite$_R$ ontologies is decidable, but with a coEXPTIME upper bound in data complexity. Furthermore, it is provably intractable (in general coNP-hard in data complexity) already for local CQ$^\neq,b$s. Local (U)CQs with inequalities, designed in such a way that each inequality atom in the query that contributes to a certain answer with respect to a DL ontology has at least one of its terms bound by an individual in the ABox.

The goal of this paper is to investigate under which conditions, stronger than local UCQs, tractability of answering queries with inequalities is recovered, or at least the com-
plexity is lowered with respect to the one of local UCQs. The basic idea to achieve this goal is to explore ontology languages and query languages ensuring the following property: each inequality atom \( \alpha \neq \beta \) that contributes to the certain answer to a query with respect to a DL ontology, it does so with both terms \( \alpha \) and \( \beta \) bounded by individuals in the ABox. In order to follow this path, we consider as basic language \( \text{DL-Lite}_R \) without the Unique Name Assumption (UNA)\(^1\) and with inequality axioms, called \( \text{DL-Lite}^{=}_R \), and explore two alternative strategies.

- The first strategy is to weaken the ontology language, so as to eliminate all the constructs introducing incomplete information resulting from existentially quantified assertions. The outcome is a sublanguage of \( \text{DL-Lite}^{=}_R \), that we call \( \text{DL-Lite}^{\neq}_{R\text{RDFS}} \) because it extends \( \text{DL-Lite}_{R\text{RDFS}} \) (Cuena Grau 2004; Rosati 2007; Cima, Lenzerini, and Poggi 2019) with both inequality axioms (in the ABox) and disjointness axioms (in the TBox).

- The second strategy is to keep \( \text{DL-Lite}^{=}_R \) as ontology language, but to weaken the query language by restricting the application of the inequality predicate to either individuals or distinguished variables (variables representing output values) only, as done in (Poggi 2016; Cima et al. 2017). The resulting query language is called “(U)CQ\(\neq\)s with bounded inequalities”, and the corresponding class is denoted by (U)CQ\(\neq\)b. Observe that, although limited, the expressive power of (U)CQ\(\neq\)b allow interesting queries to be expressed such as the one computing the triangles in a graph.

For the case of \( \text{DL-Lite}^{\neq}_{R\text{RDFS}} \), we show that answering UCQ\(\neq\)s is decidable, and in particular coNP-complete in data complexity, and \( \Pi_2^p \)-complete in combined complexity (i.e., with respect to the size of the whole input, including the query). We also investigate if the number of inequalities in each disjunct plays a role in falling into intractability. We answer positively to this question, by showing that if the query has at most one inequality per disjunct, answering UCQ\(\neq\)s is PTIME-complete in data complexity, and NP-complete in combined complexity (so, combined complexity class is the same as in the case without inequalities), while it is coNP-hard in data complexity if the query is conjunctive and has at most two inequalities. We also show that going from one to two inequalities causes the jump from NP-hardness to \( \Pi_2^p \)-hardness in combined complexity for (U)CQ\(\neq\)s, and we conjecture that this holds already for CQ\(\neq\)s.

For the case of (U)CQ\(\neq\)b, we show that answering CQ\(\neq\)b\(\neq\)s over \( \text{DL-Lite}^{\neq}_R \) ontologies has the same complexity of the UCQ case, i.e., it is in AC\(^0\) in data complexity and NP-complete in combined complexity. However, perhaps surprisingly, answering UCQ\(\neq\)b\(\neq\)b\(\neq\)s over \( \text{DL-Lite}^{\neq}_R \) ontologies is \( \Pi_2^p \)-complete in combined complexity. Therefore, unless NP = coNP, the presence of union makes the problem of answering queries with inequalities over \( \text{DL-Lite}^{\neq}_R \) ontologies significantly different from the case of UCQs.

We argue that the above results considerably improve our understanding of the implication that the presence of inequalities in queries has in the context of lightweight ontologies. In particular, to the best of our knowledge, our investigation on \( \text{DL-Lite}^{\neq}_{R\text{RDFS}} \) provides the first results on reasoning with inequalities when querying \( \text{DL-Lite}^{\neq}_{R\text{RDFS}} \) ontologies, and they also contribute a new result on containment of UCQs with inequalities in databases (Kolaitis, Martin, and Thakur 1998; Koutris et al. 2017): the problem is in NP (and therefore NP-complete) in the case of at most one inequality for each disjunct, and \( \Pi_2^p \)-complete, in the case of at most two inequalities for each disjunct. On the other hand, our results on (U)CQ\(\neq\)b\(\neq\)s posed to \( \text{DL-Lite}^{\neq}_R \) ontologies show that this class is currently the only class of queries with inequalities that can be answered with AC\(^0\) data complexity. Indeed, the only previously result known for this class was the PTIME algorithm described in (Poggi 2016). We also observe that all the results on CQ\(\neq\)b\(\neq\)s presented in this paper easily extend to UCQ\(\neq\)s posed to OWL 2 QL ontologies interpreted under the Direct Semantics Entailment Regime (Glimm 2011), that is the regime usually adopted for SPARQL queries. So, we are improving on a result reported in (Cima et al. 2017), where it is shown that answering UCQ\(\neq\)s of over OWL 2 QL ontologies can be polynomially reduced to the evaluation of a Datalog program, and therefore is in PTIME in data complexity, and in \( \text{EXP} \text{TIME} \) in combined complexity.

The paper is organized as follows. In Section 2 we provide some details on the notions used in the paper. In Section 3 we illustrate the notion of chase that we use for \( \text{DL-Lite}^{\neq}_R \), which is the basis for some of the technical results presented in this paper. In Section 4 and Section 5 we present our results on \( \text{DL-Lite}^{\neq}_{R\text{RDFS}} \) and (U)CQ\(\neq\)b\(\neq\)s, respectively. Finally, in Section 6 we conclude the paper with a discussion on future work.

## 2 Preliminaries

We define the syntax and the semantics of \( \text{DL-Lite}^{\neq}_R \), and present the query languages considered in the paper.

**DL-Lite\(_R\) and its variants.** Essentially, \( \text{DL-Lite}^{\neq}_R \) generalizes \( \text{DL-Lite}_R \) by removing the UNA, and adding axioms asserting inequalities of individuals\(^2\).

Formally, starting with an alphabet including symbols for individuals, atomic concepts, and atomic roles, and the binary relation symbol \( \neq \), a \( \text{DL-Lite}^{\neq}_R \) ontology, or simply an ontology, is a pair \( O = (T,A) \), such that \( T \), called a TBox, and \( A \), called an ABox, are sets of axioms, that have, respect-
Interpretation function extends to the other basic concepts

\( \exists \vec{x} \) of \( \vec{x} \) or a variable in \( \vec{y} \) appears in some atom of the form \( \phi(\vec{x}, \vec{y}) \), called the body of \( q \), is a finite conjunction of \( \text{DL-Lite}^R \) ABox assertions with variables that can appear in predicate arguments, i.e., atoms of the form \( A(t_1) \), \( P(t_1, t_2) \), or \( t_1 \neq t_2 \), where each \( t_j \) is either an individual of \( O \), or a variable in \( \vec{x} \) or \( \vec{y} \). We impose that every variable in \( \vec{x} \) or \( \vec{y} \) appears in some atom of \( \phi(\vec{x}, \vec{y}) \), as usual (Abiteboul, Hull, and Vianu 1995). If \( \vec{x} \) is empty, then the query is called boolean. A \( \text{CQ}^R \) of \( q \) without atoms of the form \( x_1 \neq x_2 \) in its body is called a conjunctive query (CQ). An intermediate class of queries that lies between CQs and \( \text{CQ}^R \) is the class of conjunctive queries with bound inequalities (\( \text{CQ}^R_b \)). Specifically, a \( \text{CQ}^R_b \) is a \( \text{CQ}^R \) whose inequalities involve only individuals or distinguished variables, i.e., for every atom \( x_1 \neq x_2 \) appearing in \( \phi(\vec{x}, \vec{y}) \), both \( x_1 \) and \( x_2 \) are not in \( \vec{y} \). A UCQ (resp., \( \text{UCQ}^R_b \), \( \text{UCQ}^R \)) is a union of a finite set of CQs (resp., \( \text{CQ}^R_b \)s, \( \text{CQ}^R \)) with same arity.

The set \( \text{cert}(q, O) \) of certain answers of a \( \text{UCQ}^R \) \( q \) over \( O \) is the set of \( n \)-tuples \( t = \langle t_1, \ldots, t_n \rangle \) of individuals in \( O \) such that \( O \models q(t_i) \), i.e., \((t_1^{(I)}, \ldots, t_n^{(I)}) \in q^{(I)} \), also written \( \mathcal{I} \models q \), for every model \( \mathcal{I} \) of \( O \), where \( q^{(I)} \) denotes the extension of \( q \) in \( \mathcal{I} \). When \( q \) is a boolean query, we write \( O \models q \) if \( q^2 = \emptyset \) (i.e., \( q \) is true in \( \mathcal{I} \), also denoted by \( \mathcal{I} \models q \)) for every model \( \mathcal{I} \) of \( O \). Observe that, when \( \mathcal{I} \) is finite it can be seen as a relational database (Abiteboul, Hull, and Vianu 1995), and \( q^2 \) simply denotes the evaluation of the UCQ \( q \) over \( \mathcal{I} \).

When we talk about the problem of answering a query belonging to a class of queries \( Q \) over an \( \mathcal{L} \)-ontology, i.e., an ontology expressed in the DL \( \mathcal{L} \), we implicitly refer to the following decision problem: Given a query \( q \in Q \), an \( \mathcal{L} \)-ontology \( O \), and an \( n \)-tuple \( t \) of individuals of \( O \), check whether \( t \in \text{cert}(q, O) \).

From results of (Calvanese et al. 2007), it is well known that checking whether a \( \text{DL-Lite}^R \) ontology \( O = \langle T, A \rangle \) is satisfiable can be done by evaluating a suitable query over the ABox \( A \) seen as a relational database, in particular it can be done in \( \mathcal{AC}^0 \) in data complexity and in \( \mathbf{PTIME} \) in the size of the TBox \( T \). Furthermore, when a UCQ \( q \) is posed over a satisfiable \( \text{DL-Lite}^R \) ontology \( O = \langle T, A \rangle \), it is possible to compute the set \( \text{cert}(q, O) \) of certain answers by first reformulating \( q \) w.r.t. \( T \), and then by evaluating the reformulated query (which is again a UCQ) over the ABox \( A \) seen as a relational database. This yields the well-known result that answering UCQs over \( \text{DL-Lite}^R \) ontologies is in \( \mathcal{AC}^0 \) in data complexity and \( \mathbf{NP} \)-complete in combined complexity. Observe that, since \( \text{DL-Lite}^R \) is insensitive to the adoption of the UNA for UCQ answering (Artale et al. 2009), the same complexity results hold for the problem of answering UCQs over satisfiable \( \text{DL-Lite}^R \) ontologies.

We end the section with the notion of homomorphism (Chandra and Merlin 1977), that will be used in the following. A homomorphism \( h \) from a \( \text{CQ}^R \) \( q \) to a structure \( B \) is a function from variables and individuals of \( q \) to elements of \( B \) such that (i) \( h(a) = a \) for each individual \( a \) occurring in \( q \); (ii) \( h(t_1) = t_1 \), \( h(t_2) = t_2 \) for each atom of the form \( A(t_1) \) (resp., \( P(t_1, t_2) \)), there is an atom \( A(h(t_1)) \) (resp., \( P(h(t_1), h(t_2)) \)) occurring in \( B \); and (iii) for each atom of the form \( t_1 \neq t_2 \), we have that \( h(t_1) \neq h(t_2) \).
rule is applicable then it will be eventually applied. Finally, we set $Ch_i(O) = \bigcup_{e \in \mathcal{O}} Ch_i(\mathcal{O})$. Note that we make use of the additional binary predicate symbol $ineq$, whose intended role is used to record all inequalities logically implied by $O$.

The rules we use include all the ones illustrated in (Calvanese et al. 2007). For example, if $A_1 \equiv \exists P \in T$, $A_1(e_1)$ is in $Ch_i(O)$, and no $e_2$ exists such that $P(e_1, e_2) \in Ch_i(O)$, then we set $Ch_i^{+1}(O) = Ch_i(O) \cup \{P(e_1, s)\}$, where $s \not\in V$ does not appear in $Ch_i(O)$. There are, however, crucial additions related to the $ineq$ predicate. In what follows, when we say $R(e_1, e_2)$ holds in $Ch_i(O)$, where $R$ is a basic role, we mean (i) $P(e_1, e_2) \in Ch_i(O)$, if $R = P$, or (ii) $P(e_2, e_1) \in Ch_i(O)$, if $R = P^\bot$. Also, when we say that $B_1(e_1)$ holds in $Ch_i(O)$, where $B$ is a basic concept, we mean (i) $A(e_1) \in Ch_i(O)$ if $B = A$, and (ii) $R(e_1, e_2)$ holds in $Ch_i(O)$ for some $e_2$, if $B = \exists R$. The additional rules are as follows:

- If $e_1 \not\equiv e_2$ in $Ch_i(O)$, and $ineq(e_1, e_2)$ is not in $Ch_i(O)$, then $Ch_i^{+1}(O) = Ch_i(O) \cup \{ineq(e_1, e_2)\}$;
- If $ineq(e_1, e_2)$ is in $Ch_i(O)$, and $ineq(e_2, e_1)$ is not in $Ch_i(O)$, then $Ch_i^{+1}(O) = Ch_i(O) \cup \{ineq(e_2, e_1)\}$;
- If $B_1 \equiv \neg B_2$, $B_1(e_1)$, and $B_2(e_2)$ hold in $Ch_i(O)$, and $ineq(e_1, e_2)$ is not in $Ch_i(O)$, then $Ch_i^{+1}(O) = Ch_i(O) \cup \{ineq(e_1, e_2)\}$;
- If $R_1 \equiv \neg R_2$, $R_1(e_1, e_3)$, and $R_2(e_2, e_3)$ hold in $Ch_i(O)$, and $ineq(e_1, e_2)$ is not in $Ch_i(O)$, then $Ch_i^{+1}(O) = Ch_i(O) \cup \{ineq(e_1, e_2)\}$.

From $Ch_i(O)$ it is immediate to define an interpretation $\mathcal{I}_O$ for $O$, extended in order to deal with predicate $ineq$:

- $\Delta^{\mathcal{I}_O} = V^{\mathcal{O}} \cup V$, where $V^{\mathcal{O}}$ is the set of individuals occurring in $O$;
- $e^{\mathcal{I}_O} = e$ for every individual $e \in V^{\mathcal{O}}$;
- $A^{\mathcal{I}_O} = \{e \mid A(e) \text{ occurs in } Ch_i(O)\}$ for every atomic concept $A$;
- $P^{\mathcal{I}_O} = \{(e_1, e_2) \mid P(e_1, e_2) \text{ occurs in } Ch_i(O)\}$ for every atomic role $P$;
- $ineq^{\mathcal{I}_O} = \{(e_1, e_2) \mid ineq(e_1, e_2) \text{ occurs in } Ch_i(O)\}$.

Note that, by definition, $\not\equiv^{\mathcal{I}_O}$ is the set of all sets of distinct individuals in $V^{\mathcal{O}} \cup V$, i.e., $\not\equiv^{\mathcal{I}_O} = \{(e_1, e_2) \mid e_1, e_2 \in V^{\mathcal{O}} \cup V \wedge e_1 \not\equiv e_2\}$.

Obviously, for a $DL$-Lite$^\bot_R$ ontology, $Ch_i(O)$ can be infinite, due to the presence of existential quantifiers in the right-hand side of inclusion axioms, which, by introducing fresh unknown variables, can trigger an infinite number of rule applications. It is easy to see that, on the contrary, for a $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontology $O$, $Ch_i(O)$ is finite, and can be computed in polynomial time in the size of $O$.

We next show that $\mathcal{I}_O$ enjoys some crucial properties for $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontologies $O$.

**Proposition 1.** If $\mathcal{M} = \langle \Delta^{\mathcal{M}}, \cdot^{\mathcal{M}} \rangle$ is a model of a $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontology $O$, then there exists a function $\Psi$ from $\Delta^{\mathcal{I}_O}$ to $\Delta^{\mathcal{M}}$ such that:

1. for every $e \in \Delta^{\mathcal{I}_O}$, if $e \in A^{\mathcal{M}}$, then $\Psi(e) \in A^{\mathcal{M}}$;
2. for every pair $e_1, e_2 \in \Delta^{\mathcal{I}_O}$, if $(e_1, e_2) \in P^{\mathcal{M}}$, then $(\Psi(e_1), \Psi(e_2)) \in P^{\mathcal{M}}$;
3. for every pair $e_1, e_2 \in \Delta^{\mathcal{I}_O}$, if $(e_1, e_2) \in ineq^{\mathcal{I}_O}$, then $\Psi(e_1) \neq \Psi(e_2)$.

The above proposition shows the importance of distinguishing between $\not\equiv$ and $ineq$. Indeed, while by definition of $\mathcal{I}_O$ two different individuals $e_1, e_2$ satisfy $e_1 \not\equiv e_2$, it may happen that for some model $\mathcal{M}$ of $O$, $e_1 \not\equiv e_2$, implying that no function $\Psi$ exists from $\Delta^{\mathcal{I}_O}$ to $\Delta^{\mathcal{M}}$ such that $\Psi(e_1) \neq \Psi(e_2)$. In other words, condition 3 in Proposition 1 does not hold if we replace $ineq$ with $\not\equiv$.

Note that if $\mathcal{I}_O$ satisfies all the axioms of $O$, then it is a model of $O$, and therefore $O$ is satisfiable. Otherwise, it can be easily seen that $\mathcal{I}_O$ violates at least one disjointness or one inequality axiom of $O$. In particular, it can be proved that $\mathcal{I}_O$ violates some inequality axiom if and only if $\not\equiv \in O$ for some $e \in V^{\mathcal{O}}$. As a result, we can devise a satisfiability checking algorithm for $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ by slightly modifying the so-called violation query for $DL$-Lite$^{\equiv,\not\equiv}_R$ and this shows that, similarly to the “canonical interpretation” of a $DL$-Lite$^{\equiv,\not\equiv}_R$ ontology, $\mathcal{I}_O$ is instrumental for checking the satisfiability of a $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontology $O$. In turn, this implies that checking the satisfiability of a $DL$-Lite$^{\equiv,\not\equiv}_R$ ontology $O = (\mathcal{T}, \mathcal{A})$ can be done in AC$^0$ in the size of $\mathcal{A}$ and in PTIME in the size of $\mathcal{T}$, exactly like in $DL$-Lite$^{\equiv,\not\equiv}_R$.

A reasonable question to ask is whether $\mathcal{I}_O$ also is the right tool for query answering. The next theorem provides a positive answer to this question for the class $CQ^{\equiv,\not\equiv}_b$. In what follows, $\delta(q)$ denotes the query obtained by replacing each inequality atom $t_1 \not\equiv t_2$ in $q$ with the atom $ineq(t_1, t_2)$.

**Theorem 1.** Let $t$ be a tuple of individuals of a satisfiable $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontology $O$, and let $q$ be a $CQ^{\equiv,\not\equiv}_b$ over $O$. We have that $t \in \text{cert}(q, O)$ if and only if $\delta(q) \in \text{cert}(t, \mathcal{I}_O)$.

The above theorem states that $\mathcal{I}_O$ is instrumental also for answering $CQ^{\equiv,\not\equiv}_b$ over $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontologies. However, we will see in the next two sections that this theorem is no longer valid when we move from $CQ^{\equiv,\not\equiv}_b$ to either $UCQ^{\equiv,\not\equiv}_b$, or $CQ^{\equiv,\not\equiv}$.

From now on, we implicitly assume to deal with satisfiable ontologies. Moreover, unless otherwise stated and without loss of generality, we consider only boolean $UCQ^{\equiv,\not\equiv}$s. Indeed, given an $n$-ary $UCQ^{\equiv,\not\equiv}$ $q$, a $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontology $O = (\mathcal{T}, \mathcal{A})$, and an $n$-tuple $t$ of individuals of $O$, checking whether $t \in \text{cert}(q, O)$ is equivalent to checking whether $O \models q(\delta(t))$, where $q(\delta(t))$ denotes the boolean $UCQ^{\equiv,\not\equiv}$ obtained by replacing appropriately the distinguished variables of each disjunct of $q$ with the individuals of $t$.

**4 UCQ^{\equiv,\not\equiv}_b over DL-Lite^{\equiv,\not\equiv}_{RDF\bot} ontologies**

We study the problem of answering $UCQ^{\equiv,\not\equiv}_b$ over satisfiable $DL$-Lite$^{\equiv,\not\equiv}_{RDF\bot}$ ontologies.
Theorem 1 tells us that the certain answers to a CQ\(\neq b\) \(q\) over a DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontology \(O\) coincide with \(\delta(q)^O\). However, the following example shows that the problem drastically changes as soon as we consider general CQ\(\neq\)\(s\).

Example 1. Consider the DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontology \(O = \langle T, A \rangle\) where \(T = \{A_1 \subseteq \neg A_2\}\) and \(A = \{A_1(a_1), A_2(a_2), P(b,c_1), P(b,c_2), P(c_1,a_1), P(c_2,a_2)\}\). For the boolean CQ\(\neq\) \(q = \{1\} \cup P(x, y_1) \land P(x, y_2) \land y_1 \neq y_2\), we have that \(\delta(q)^O\) is false because \(\neg c_1 = c_2\) is not in \(Ch(O)\). However \(O \models q\), because in each model \(M\) where \(c_1^M = c_2^M\) the query is true with the bindings \(x, y_1, y_2 \rightarrow c_1, a_1, a_2\), whereas in each model \(M\) where \(c_1^M \neq c_2^M\), \(q\) is true with the bindings \(x, y_1, y_2 \rightarrow b, c_1, c_2\).

The above example provides a hint on how to design an algorithm for our problem. Intuitively, given a boolean UCQ\(\neq\) \(q\), and a DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontology \(O = \langle T, A \rangle\), we check whether \(O \models \neg q\) by searching for a database that can be obtained from \(Ch(O)\) by equating some of the individuals, and that falsifies \(q\). We thus derive the upper bounds for the problem of answering UCQ\(\neq\)\(s\) in DL-Lite\(^{\neq}\)\(\text{RDFS}\).

Theorem 2. Answering UCQ\(\neq\)\(s\) over DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontologies is in conNP in data complexity and in \(\Pi^p_2\) in combined complexity.

We now provide matching lower bounds for both data and combined complexity, showing that they hold already for the case of CQ\(\neq\)\(s\). We start with data complexity.

Theorem 3. Answering CQ\(\neq\)\(s\) over DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontologies is conNP-hard in data complexity.

The proof of the above theorem has two interesting implications. (i) conNP-hardness in data complexity holds even for CQ\(2^{\neq}\)\(s\) over both DL-Lite\(^{\neq}\)\(\text{RDFS}\) and DL-Lite\(^{\neq}\)\(\text{RDFS}\), where CQ\(k^{\neq}\) denotes the class of CQ\(\neq\) including at most \(k\) inequalities, and UCQ\(k^{\neq}\) the class of unions of finite sets of CQ\(k^{\neq}\)\(s\) with same arity. (ii) Answering UCQ\(2^{\neq}\)\(s\) over DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontologies is conNP-hard too, and this corrects an erroneous statement in (Rosati 2007, Theorem 11), where it is claimed that answering UCQ\(2^{\neq}\)\(s\) over DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontologies is in LOGSPACE in data complexity regardless of whether the UNA is adopted or not. It turns out that this latter statement is true only under the UNA.

The following theorem provides the matching lower bound for combined complexity.

Theorem 4. Answering UCQ\(\neq\)\(s\) over DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontologies is \(\Pi^p_2\)-hard in combined complexity.

By looking at the proof of the theorem, one can see that \(\Pi^p_2\)-hardness holds already in the case of both DL-Lite\(^{\neq}\)\(\text{RDFS}\), and DL-Lite\(^{\neq}\)\(\text{RDFS}\). However, the reduction builds a CQ\(\neq\) whose number of inequalities depends on the input of the reduction, and therefore is not fixed a priori. It is thus natural to ask which is the minimum number of inequalities in CQ\(\neq\) that makes the problem \(\Pi^p_2\)-hard in combined complexity. Similarly to the case of the conNP-hardness result in data complexity, we conjecture that such number is 2. Even though we have not been able to prove this conjecture, we show next that \(\Pi^p_2\)-hardness holds for UCQ\(2^{\neq}\)\(s\).

Algorithm CheckGood\(O, q, F\)

Input: DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontology \(O = \langle T, A \rangle\), UCQ\(1^{\neq}\) \(q\), sequence of functions \(F = \{f_1, \ldots, f_m\}\)

Output: true or false

begin
Compute \(B := Ch(O)\)
for each \(i = 1\) to \(m - 1\):
\begin{itemize}
  \item if \(f_i\) is a homomorphism from a disjunct of \(q_1\) to \(B\)
l et \(t_1 \neq t_2\) be any inequality in any of such disjuncts
  \item if \(\neg eq(f_i(t_1), f_i(t_2)) \in B\) return true
  \item else replace each occurrence of \(f_i(t_1)\) appearing
  \hspace{1cm} in \(B\), in \(q\), and in \(\{f_{i+1}, \ldots, f_m\}\) with \(f_i(t_2)\)
  \item else return false
\end{itemize}
return \(f_m\) is a homomorphism from a disjunct of \(q_2\) to \(B\)
end

Figure 1: The algorithm CheckGood\(O, q, F\)

Theorem 5. Answering UCQ\(2^{\neq}\)\(s\) over DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontologies is \(\Pi^p_2\)-hard.

Interestingly, the proof of the previous theorem shows that \(\Pi^p_2\)-hardness holds even if the query is the union of a CQ\(2^{\neq}\) and a CQ without inequalities, and the ontology is expressed in DL-Lite\(^{\neq}\)\(\text{RDFS}\). Observe that, in this language, CQ\(2^{\neq}\)s containing even a single inequality have an empty set of certain answers. Thus, we are observing a surprising jump from constant time to \(\Pi^p_2\)-hardness if we add union to such CQ\(\neq\)\(s\).

To complete the picture of answering UCQ\(2^{\neq}\)\(s\) in DL-Lite\(^{\neq}\)\(\text{RDFS}\), it remains to study the case of UCQ\(1^{\neq}\)\(s\). In what follows, without loss of generality, we assume that each UCQ\(1^{\neq}\) is written as \(q = q_1 \cup q_2\), where \(q_2\) is a UCQ with no inequalities and \(q_1\) is a UCQ\(1^{\neq}\) having exactly one inequality per disjunct.

In principle, for answering UCQ\(1^{\neq}\)\(s\) over DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontologies it is possible to use the algorithm provided in (Fagin et al. 2005, Theorem 5.12) in the context of data exchange. However, this would result in an exponential time algorithm with respect to the size of the query. On the contrary, by elaborating on the idea of (Fagin et al. 2005, Theorem 5.12), we have devised an algorithm that runs in PTIME in data complexity and in NP in combined complexity. We start with the following definition.

Definition 1. Let \(O = \langle T, A \rangle\) be a DL-Lite\(^{\neq}\)\(\text{RDFS}\) ontology, and let \(q\) be a boolean UCQ\(1^{\neq}\) over \(O\). A sequence \(F = \{f_1, \ldots, f_m\}\) of functions from variables and individuals of \(q\) to individuals of \(A\) is good w.r.t. \(O\) and \(q\) if the algorithm CheckGood\(O, q, F\) provided in Figure 1 returns true.

Roughly speaking, starting from \(B := Ch(O)\), in each step \(i\) from 1 to \(m - 1\) such that \(f_i\) is a homomorphism from a disjunct of \(q_1\) to \(B\), the algorithm CheckGood\(O, q, F\) replaces everywhere the individual \(f_i(t_1)\) with the individual \(f_i(t_2)\), to consider the models in which \(f_i(t_1) = f_i(t_2)\) (since there is a homomorphism, \(q\) is true in the models where \(f_i(t_1) \neq f_i(t_2)\)). Afterwards, the algorithm sanctions that \(F\) is a good sequence if and only if either it is not possible to equate two individuals without contradicting an \(\neg eq\).
atom of $B$, or the resulting $B$ and $q_2$ are such that $B \models q_2$. Using the above notion of good sequence, it is possible to derive the following characterization ($n_A$ denotes the number of individuals occurring in the ABox $A$).

**Proposition 2.** Let $O = \langle T, A \rangle$ be a DL-Lite$_{\mathcal{RDF}}$ ontology, and let $q$ be a boolean UCQ$^{1,\neq}$ over $O$. We have that $O \models q$ if and only if there exists a sequence $F = \{f_1, \ldots, f_m\}$ of $m \leq n_A$ functions that is good w.r.t. $O$ and $q$.

We are now ready to establish our result on answering UCQ$^{1,\neq}$s in DL-Lite$_{\mathcal{RDF}}$.

**Theorem 6.** Answering UCQ$^{1,\neq}$s over DL-Lite$_{\mathcal{RDF}}$ ontologies is PTIME-complete in data complexity and NP-complete in combined complexity.

**Proof.** (Sketch) NP-hardness in combined complexity follows from NP-hardness of CQ evaluation over relational databases (Chandra and Merlin 1977). In the rest of this proof sketch, we discuss only the upper bounds. By Proposition 2 it is possible to decide whether $O \models q$ as follows. We guess a sequence $F = \{f_1, \ldots, f_m\}$ (with $m \leq n_A$) of functions from disjuncts of $q_1$ to $Ch(O)$ (note that this can be done in PTIME in the size of $A$). Then, by exploiting the algorithm CheckGood$(O, q, F)$, we check whether $F$ is a good sequence w.r.t. $O$ and $q$ using: (i) a PTIME step in the size of $O$ for computing $B = Ch(O)$; (ii) for each $i \in \{1, m - 1\}$, a PTIME step for checking whether $f_i$ is a homomorphism from a disjunct of $q_1$ to $B$, $\text{ineq}(f_i(t_1), f_i(t_2)) \in B$, and for replacing each occurrence of $f_i(t_1)$ with $f_i(t_2)$; finally, (iii) a PTIME step for checking whether $f_m$ is a homomorphism from some disjunct of $q_2$ to $B$.

Example 2. Consider the DL-Lite$^\neq$ ontology $O = \langle T, A \rangle$ with $T = \{P_1 \sqsubseteq P_2, A_1 \sqsubseteq A_2\}$, and the CQ$^{1,\neq}$,

$$q = \{\{x_1, x_2\} | P_2(x_1, x_2) \land x_1 \neq c\}$$

over $O$. It is easy to see that $\sigma(x_1 \neq c, T)$ is the formula $\text{ineq}(x_1, c) \lor \text{ineq}(x_1, x_2) \lor \text{ineq}(x_2, c)$, then $\tau(q, T)$ is the UCQ$^{1,\neq}$ whose disjuncts are the following:

$\{\{x_1, x_2\} | P_2(x_1, x_2) \land \text{ineq}(x_1, c), \{x_1, x_2\} | P_2(x_1, x_2) \land \text{ineq}(x_2, c)\}$, and $\{\{x_1, x_2\} | P_2(x_1, x_2) \land A_2(c)\}$.}

For a DL-Lite$^\neq$ ontology $O = \langle T, A \rangle$, we denote by $O^{\text{ineq}} = \langle T, A^{\text{ineq}} \rangle$ the DL-Lite$^\neq$ ontology where $\text{ineq}$ is a new atomic role, and $A^{\text{ineq}}$ is the DL-Lite$^\neq$ ABox obtained from $A$ by replacing each assertion $c_1 \neq c_2$ appearing in $A$ with the assertion $\text{ineq}(c_1, c_2)$.

The next proposition, whose proof relies on an extension of (Calvanese et al. 2007, Lemma 39) and on Theorem 1, states that computing $\text{cert}(q, O)$ for a given DL-Lite$^\neq$ ontology $O = \langle T, A \rangle$, and a CQ$^{1,\neq}$ $q$ over $O$, can be reduced to computing the certain answers of the UCQ $\tau(q, T)$ over the DL-Lite$^\neq$ ontology $O^{\text{ineq}}$.

**Proposition 3.** Let $O = \langle T, A \rangle$ be a DL-Lite$^\neq$ ontology, and let $q$ be a CQ$^{1,\neq}$ over $O$. Then, we have that $\text{cert}(q, O) = \text{cert}(\tau(q, T), O^{\text{ineq}})$.

From the above proposition, we immediately derive that answering CQ$^{1,\neq}$s over DL-Lite$^\neq$ ontologies has the same data and combined complexity as answering UCQs over DL-Lite$^\neq$ ontologies.

**Theorem 7.** Answering CQ$^{1,\neq}$s over DL-Lite$^\neq$ ontologies is in AC$^0$ in data complexity, and NP-complete in combined complexity.

Looking at the proof of the two above statements, one realizes the importance of Theorem 1, stating that, similarly to DL-Lite$^\neq$, DL-Lite$^\neq$ admits a model $I_C$ that is representative of all the models of $O$ w.r.t. answering CQ$^{1,\neq}$s. One might therefore think that, analogously to DL-Lite$^\neq$, this property extend to UCQ$^{1,\neq}$s. The following example shows that, surprisingly, this is not the case.
Example 3. Consider the DL-Lite_Ω^Ω ontology \( \mathcal{O} = \langle T, A \rangle \), where \( T = \emptyset \) and \( A = \{ P(a, b) \} \). For the UCQ^{p,b} \( \Sigma \) \( \varnothing = q_1 \cup q_2 \), where \( q_1 = \{ \{ \} \mid P(a, a) \} \) and \( q_2 = \{ \{ \} \mid a \neq b \} \), it is easy to see that \( \delta(\Sigma)_{\mathcal{O}} \) is false. However, one can verify that \( \mathcal{O} \models \Sigma \). Indeed, for any model \( M \) of \( \mathcal{O} \), either \( a^M = b^M \) and \( M \models q_1 \), or \( a^M \neq b^M \) and \( M \models q_2 \).

Proposition 5. Let \( \mathcal{O} = \langle T, A \rangle \) be a DL-Lite_Ω^Ω ontology, and let \( \varnothing \) be a boolean UCQ^{p,b} over \( \mathcal{O} \). We have that \( \mathcal{O} \models \varnothing \) if and only if there exists an equivalence relation \( e \) on the set \( C_\varnothing \) of all individuals appearing in \( \varnothing \) such that \( \mathcal{O} \not\models e \).

Intuitively, to decide \( \mathcal{O} \not\models \varnothing \), it is sufficient to guess an equivalence relation \( e \) between the individuals of \( \varnothing \) for which there exists an \( e \)-model \( \mathcal{I} \) of \( \mathcal{O} \) such that \( \mathcal{I} \not\models \varnothing \). Observe that, by definition, such model exists if and only if \( \mathcal{O} \not\models e \).

The following theorem characterizes the complexity of answering UCQ^{p,b} s over DL-Lite_Ω^Ω ontologies.

**Theorem 8.** Answering UCQ^{p,b} s over DL-Lite_Ω^Ω ontologies is in \( \mathcal{AC}^0 \) in data complexity and \( \Pi_2^p \)-complete in combined complexity.

Proof. (Sketch) As for the upper bounds, we now show how to decide whether \( \mathcal{O} \not\models \varnothing \) in \( \mathcal{AC}^0 \) in data complexity and in \( \Sigma_2^p \) in combined complexity. In particular, observe that by Proposition 5 it is sufficient to: (i) guess an equivalence relation \( e \); (ii) check whether \( \mathcal{O} \not\models \varnothing, e \), where this last step, due to Proposition 4, can be done in \( \mathcal{AC}^0 \) in the size of \( \mathcal{A} \), and with an NP-oracle in the size of the input.

The proof of the above theorem allows us to conclude that the same complexity results hold even for the problem of answering UCQ^{p,b} s over DL-Lite_RDFS ontologies.

6 Conclusion

We have carried a thorough analysis of the problem of answering UCQs with inequalities posed to a DL-Lite_Ω^Ω ontology. The results presented in this paper greatly contribute to clarify how inequalities impact on the problem of answering queries over DL-Lite_Ω^Ω ontologies. In particular, we have presented the first results on dealing with inequalities in queries posed to DL-Lite_RDFS ontologies, and we have deeply investigated a specific class of queries, namely UCQ^{p,b} s, for which query answering over DL-Lite_RDFS ontologies is still in \( \mathcal{AC}^0 \) in data complexity. We have also mentioned the connection between the problems studied here and two other problems, namely containment of conjunctive queries with inequalities in databases, and answering UCQ^{p,b} s over OWL 2 QL ontologies under the direct semantics, although we could not elaborate on these aspects for the lack of space.

There are several issues to consider for continuing the work presented in this paper, the most obvious being trying to decide which is the minimum number of inequalities that makes query answering over DL-Lite_RDFS \( \Pi_2^p \)-hard in combined complexity. Another interesting future work is to look for extensions of both DL-Lite_Ω^Ω and UCQ^{p,b} s for which query answering is still decidable/tractable. Finally, we observe that it is still open whether answering CQ^{p,b} s over DL-Lite_{core} ontologies is decidable.

Acknowledgments

This work has been supported by Sapienza under the project "PRE-O-PRE" and by MIUR, under the SIR project "MODEUS" - grant n. RBSI14TQHQ, and under the PRIN 2017 project "HOPE".

---

1 An equivalence relation \( e \) on a set of individuals \( C \) is a binary relation over \( C \) that is reflexive, symmetric, and transitive.
References


