The Complexity of Explaining Negative Query Answers in *DL-Lite*

Diego Calvanese
Free Univ. of Bozen-Bolzano
Italy
calvanese@inf.unibz.it

Magdalena Ortiz and **Mantas Šimkus**
Vienna University of Technology
Austria
ortiz@kr.tuwien.ac.at, simkus@dbai.tuwien.ac.at

Giorgio Stefanoni
University of Oxford
United Kingdom
gstef@cs.ox.ac.uk

Abstract

In order to meet usability requirements, most logic-based applications provide explanation facilities for reasoning services. This holds also for DLs, where research has focused on the explanation of both TBox reasoning and, more recently, query answering. Besides explaining the presence of a tuple in a query answer, it is important to explain also why a given tuple is missing. We address the latter problem for (conjunctive) query answering over DL-Lite ontologies, by adopting abductive reasoning, that is, we look for additions to the ABox that force a given tuple to be in the result. As reasoning tasks we consider existence and recognition of an explanation, and relevance and necessity of a certain assertion for an explanation. We characterize the computational complexity of these problems for subset minimal and cardinality minimal explanations.

Introduction

Query answering over ontologies formulated in Description Logics (DLs) has received considerable attention in both research and industry. Given an ontology, users pose queries over the conceptual schema and get answers that take into account the constraints specified at the conceptual level. Many efforts have concentrated on lightweight DLs. For instance *DL-Lite*$_\mathcal{A}$, the language at the basis of the OWL 2 QL profile [Motik et al., 2009], has been tailored for query answering over large data sets [Calvanese et al., 2009]. In this setting, expressive power is traded in favour of a better computational behaviour in terms of data-complexity. In fact, conjunctive query answering in *DL-Lite*$_\mathcal{A}$ enjoys FO-rewritability, i.e., it can be reduced to the problem of evaluating a suitable FO query over a database instance.

In order to meet usability requirements set by domain users, most logic-based applications provide explanation algorithms for reasoning services. This holds also for DLs, where research has focused on the explanation of TBox reasoning (cf. [McGuinness and Borgida, 1995; Borgida, Franconi, and Horrocks, 2000; Penaloza and Sertkaya, 2010; Horridge, Parsia, and Sattler, 2008]). Additionally, the problem of explaining positive answers to conjunctive queries over *DL-Lite*$_\mathcal{A}$ ontologies has been studied in [Borgida, Calvanese, and Rodríguez-Muro, 2008], where a procedure for computing the reasons for a tuple to be in the answers to a query is outlined. The same paper advocates the importance of computing explanations also for the absence of query answers. To the best of our knowledge, in the literature, the problem of explaining negative answers has been considered only for relational databases extended with provenance information. In particular, [Chapman and Jagadish, 2009] studied the problem of determining the database operations that prevented a given tuple to be in the answers to the query. Also, [Huang et al., 2008] focused on computing database updates fixing missing answers to given SQL queries. Unfortunately, typical ontologies do not provide provenance information and, thus, negative query answers can not be explained by adapting one of the available solutions.

For this reason, we formalize the problem of explaining the absence of a tuple in the context of query answering over DL ontologies. We adopt *abductive reasoning* [Eiter and Gottlob, 1995; Klarman, Endriss, and Schlobach, 2011], that is, we consider which additions need to be made to the ABox to force the given tuple to be in the result. More precisely, given a TBox $\mathcal{T}$, an ABox $\mathcal{A}$, and a query q, an *explanation* for a given tuple $\vec{c}$ is a new ABox $\mathcal{E}$ such that the answer to q over $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{E} \rangle$ contains $\vec{c}$. An important aspect in explanations is to provide users with explanations that are simple to understand and free of redundancy, hence as small as possible. To address this requirement, we study various restrictions on explanations, in particular, we focus on subset minimal and cardinality minimal ones. We consider standard decision problems associated to logic-based abduction: *(i) existence* of an explanation; *(ii) recognition* of a given ABox as being an explanation; *(iii) relevance* and *(iv) necessity* of an ABox assertion, i.e., whether it occurs in some or all explanations. Additionally, it is important to allow one to restrict the signature of explanations. This can be used to consider only solutions that do not extend the ABox vocabulary: an important property in the context of accessing relational databases through ontologies, where database instances are defined over a small, fixed, vocabulary, and the terminological component is used to enrich that vocabulary. The idea of restricting the explanation signature is an adaptation of a concept introduced in [Baader et al., 2010], which studies among others the $\mathcal{CQ}$-emptiness problem. That is, given a query q

over a TBox $\mathcal{T}$ decide whether for all ABoxes $\mathcal{A}$ over a given signature Σ, we have that evaluating q over $\langle \mathcal{T}, \mathcal{A} \rangle$ leads to an empty result. In our framework, deciding the existence of an explanation generalizes the $\mathcal{CQ}$ non-emptiness problem. In fact, deciding whether there exists an explanation for a negative answer amounts to check whether a query admits a solution w.r.t. a TBox $\mathcal{T}$ *and* an ABox $\mathcal{A}$. In the following we sketch algorithms to solve the relevant reasoning tasks and give a precise characterization of their computational complexity for *DL-Lite*$_\mathcal{A}$. The complexity results for the various reasoning tasks are summarized in Table 1. We provide proof sketches here, and refer to [Calvanese et al., 2012] for full proofs.

Preliminaries

We use the standard notation for *DL-Lite*$_\mathcal{A}$ ontologies $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is the TBox and $\mathcal{A}$ is the ABox. By N_C, N_R, and N_I we denote the alphabets of concept, role, and constant names. When the distinction between concepts and roles is inessential, we call elements of $N_C \cup N_R$ simply *predicates*. As usual, the semantics of *DL-Lite*$_\mathcal{A}$ is based on first-order interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$. We adopt the *unique name assumption* (UNA), i.e., for every interpretation $\mathcal{I}$ and constant pair $c_1 \neq c_2$, we have $c_1^\mathcal{I} \neq c_2^\mathcal{I}$. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{O}$ if it satisfies all the axioms in the TBox $\mathcal{T}$ and all the assertions in the ABox $\mathcal{A}$. We call $\mathcal{O}$ *consistent* if it admits at least one model.

As query language we consider *conjunctive queries (CQs)* and their unions (UCQs). CQs are written as sets of *atoms* of the forms $A(t)$ and $P(t, t')$, where A and P are predicates and t, t' are variables or constants. UCQs are sets of CQs. For a CQ or UCQ $q(\vec{x})$ with answer variables $\vec{x}$, we use $\mathsf{cert}(q, \mathcal{O})$ to denote the *certain answers* over $\mathcal{O}$, that is, the tuples $\vec{c}$ of constants of arity $|\vec{x}|$ that are an answer to q in each model $\mathcal{I}$ of $\mathcal{O}$.

Complexity Theory. We briefly outline the definition of some non-canonical complexity classes used in the paper, and refer to [Papadimitriou, 1994] for more details. The class $\mathrm{P}_\parallel^{\mathrm{NP}}$ contains all the decision problems that can be solved in polynomial time with an NP oracle, where all oracle calls must be first prepared and then issued in parallel. The class DP contains all problems that, considered as languages, can be characterized as the intersection of a language in NP and a language in CONP. It is believed that PTIME $\subseteq$ NP $\subseteq$ DP $\subseteq \mathrm{P}_\parallel^{\mathrm{NP}} \subseteq \Sigma_2^{\mathrm{P}}$ is a strict hierarchy of inclusions. Here we make such an assumption.

Explaining Negative Query Answers

In this paper we deal with the following problem:

Definition 1. Let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a *DL-Lite*$_\mathcal{A}$ ontology, $q(\vec{x})$ a UCQ, and $\vec{c}$ a tuple of constants of arity $|\vec{x}|$. Further, assume a set $\Sigma \subseteq N_C \cup N_R$. We call $\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, q, \vec{c}, \Sigma \rangle$ a *Query Abduction Problem* (QAP). An *explanation* for (or, a solution to) $\mathcal{P}$ is an ABox $\mathcal{E}$ such that:

(i) the concept and role names of $\mathcal{E}$ are contained in Σ,

(ii) the ontology $\mathcal{O}' = \langle \mathcal{T}, \mathcal{A} \cup \mathcal{E} \rangle$ is consistent, and

(iii) $\vec{c} \in \mathsf{cert}(q, \mathcal{O}')$.

The set of all explanations for $\mathcal{P}$ is denoted by $\mathsf{expl}(\mathcal{P})$. If $\Sigma = N_C \cup N_R$, we say that $\mathcal{P}$ has *an unrestricted explanation signature*. ■

The predicates in Σ are the ones allowed in explanations, hence we call them *abducible predicates*. If $\vec{c} \notin \mathsf{cert}(q, \mathcal{O})$, we call $\vec{c}$ a *negative answer* to q over $\mathcal{O}$. Note that a query over the ontology can have a negative answer only if the ontology is consistent. Also, by condition *(ii)*, if the ontology is inconsistent then $\mathcal{P}$ does not have any explanation. Note also that $\mathcal{E}$ may contain constant names not present in $\mathcal{A}$. Next, we provide an example of a QAP.

Example 1. Let $\mathcal{A}$ be the following set of assertions about a particular university:

$$\begin{array}{ll} \mathsf{DPhil}(\mathit{Anna}), & \mathsf{DPhil}(\mathit{Beppe}), \\ \mathsf{enroll}(\mathit{Anna}, \mathit{KR}), & \mathsf{teach}(\mathit{Marco}, \mathit{KR}), \\ \mathsf{enroll}(\mathit{Luca}, \mathit{IDB}), & \mathsf{teach}(\mathit{Carlo}, \mathit{IDB}). \end{array}$$

That is, Anna and Beppe are doctoral students. Anna is enrolled in the KR course, which is taught by Marco, and Luca is enrolled in the introductory DB course (IDB), which is taught by Carlo. Now, consider the following TBox $\mathcal{T}$, in standard *DL-Lite*$_\mathcal{A}$ syntax, formalizing the university domain, of which $\mathcal{A}$ is a (partial) instance:

$$\begin{array}{rclrcl} \exists\mathsf{enroll} & \sqsubseteq & \mathsf{Student}, & \exists\mathsf{teach} & \sqsubseteq & \mathsf{Lecturer}, \\ \exists\mathsf{enroll}^- & \sqsubseteq & \mathsf{Course}, & \exists\mathsf{teach}^- & \sqsubseteq & \mathsf{Course}, \\ \mathsf{DPhil} & \sqsubseteq & \mathsf{Student}, & \mathsf{Course} & \sqsubseteq & \exists\mathsf{teach}^-. \end{array}$$

$\mathcal{T}$ models that objects in the domain of enroll are Students, and objects in the domain of teach are Lecturers, while objects in the range of enroll or of teach are Courses. Among the students we have DPhil students. Finally, every Course must be taught by someone.

Now, assume that the university administration is interested in finding all those who are teaching a course in which at least one of the enrolled students is a doctoral student, which is captured by the following query:

$$q(x) \leftarrow \mathsf{teach}(x, y), \mathsf{enroll}(z, y), \mathsf{DPhil}(z).$$

Assume that Carlo is expected to be part of the result, i.e., $\mathit{Carlo} \in \mathsf{cert}(q, \langle \mathcal{T}, \mathcal{A} \rangle)$. This is not the case, as Luca is the only student of Carlo and he is not known to be a doctoral student. Suppose that we have complete information on all the predicates but enroll and teach, i.e., only the latter predicates are abducible. It is easy to see that:

$$\{\mathsf{teach}(\mathit{Carlo}, \mathit{AI}), \mathsf{enroll}(\mathit{Beppe}, \mathit{AI}), \mathsf{enroll}(\mathit{Luca}, \mathit{AI})\}$$

is an explanation for the given negative answer, which suggests the existence of a course, AI, not present in $\mathcal{A}$. ■

This example shows that certain explanations may be too assumptive in that they include assertions that are not required to solve the problem. Indeed, in the example's explanation there is no reason to assume that Luca is enrolled in the freshly introduced course on Artificial Intelligence. In the following, we will examine various restrictions to $\mathsf{expl}(\mathcal{P})$ to reduce redundancy in explanations, achieved by introducing a preference relation among explanations. This relation is reflexive and transitive, i.e., we have a pre-order among explanations. For such a pre-order $\preceq$ on $\mathsf{expl}(\mathcal{P})$, we write $\mathcal{E} \prec \mathcal{E}'$ if $\mathcal{E} \preceq \mathcal{E}'$ and $\mathcal{E}' \not\preceq \mathcal{E}$.

$\preceq$	$\preceq$-EXIST	$\preceq$-NEC	$\preceq$-REL	$\preceq$-REC
none	NP	coNP	NP	NP
$\leq$	NP	$\mathrm{P}^{\mathrm{NP}}_{\parallel}$	$\mathrm{P}^{\mathrm{NP}}_{\parallel}$	DP
$\subseteq$	NP	coNP	Σ^{P}_{2}	DP

Table 1: Summary of main complexity results for *DL-Lite*$_{\mathcal{A}}$ explanation (all are completeness results)

Definition 2. The preferred explanations $\mathsf{expl}_{\preceq}(\mathcal{P})$ of a QAP $\mathcal{P}$ under the pre-order $\preceq$, called $\preceq$-*explanations*, are defined as follows: $\mathsf{expl}_{\preceq}(\mathcal{P}) = \{\, \mathcal{E} \in \mathsf{expl}(\mathcal{P}) \mid$ there is no $\mathcal{E}' \in \mathsf{expl}(\mathcal{P})$ s.t. $\mathcal{E}' \prec \mathcal{E} \,\}$, i.e., $\mathsf{expl}_{\preceq}(\mathcal{P})$ contains all explanations the that are *minimal* under $\preceq$. ■

We consider two preference orders that are commonly adopted when comparing abductive solutions: the *subset-minimality order*, denoted by $\subseteq$, and the *minimum explanation size order*, denoted by $\leq$. The latter order is defined by $\mathcal{E} \leq \mathcal{E}'$ iff $|\mathcal{E}| \leq |\mathcal{E}'|$. Observe that $\mathsf{expl}_{\leq}(\mathcal{P}) \subseteq \mathsf{expl}_{\subseteq}(\mathcal{P})$.

Example 2. $\{\mathsf{teach}(\mathit{Carlo}, \mathit{AI}), \mathsf{enroll}(\mathit{Beppe}, \mathit{AI})\}$ is a $\subseteq$-explanation in our example, while $\{\mathsf{enroll}(\mathit{Beppe}, \mathit{IDB})\}$ is a $\leq$-explanation (and hence also a $\subseteq$-explanation). ■

We study here the four basic decision problems related to (minimal) explanations [Eiter and Gottlob, 1995], which are parametric w.r.t. the chosen preference order $\preceq$.

Definition 3. Given a QAP $\mathcal{P}$, define the following decision problems:

- $\prec$-EXIST(ENCE): Does there exist a $\preceq$-explanation for $\mathcal{P}$?
- $\prec$-NEC(ESSITY): Does a given assertion α occur in all $\preceq$-explanations for $\mathcal{P}$?
- $\prec$-REL(EVANCE): Does a given assertion α occur in some $\preceq$-explanation for $\mathcal{P}$?
- $\prec$-REC(OGNITION): Is a given ABox $\mathcal{E}$ a $\preceq$-explanation for $\mathcal{P}$?

Whenever no preference is applied (i.e., when $\preceq$ is the identity) we omit to write $\preceq$ in front of the problems' names. ■

In the next section, we study the complexity of these four problems in the light of the different preference relations.

Complexity of Explanations

In Table 1 we give an overview of our complexity results for explanation in *DL-Lite*$_{\mathcal{A}}$. We measure the complexity of a QAP $\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, q, \vec{c}, \Sigma \rangle$ in terms of the combined size of $\mathcal{T}$, $\mathcal{A}$, and q, i.e., we consider *combined complexity*. Notice that we do not explicitly count the explanation signature Σ towards the complexity, since when it is restricted, its size is bounded by the size of the other parameters (see Proposition 1), and when it is unrestricted it is actually countably infinite and defined outside of the actual problem instance.

Existence of Explanations

Before discussing our complexity results, we show that whenever a QAP $\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, q, \vec{c}, \Sigma \rangle$ has an explanation, then $\mathcal{P}$ has an explanation $\mathcal{E}$ that is small in two senses.

First, all concepts and roles occurring in $\mathcal{E}$ occur either in $\mathcal{T}$ or in q. Indeed, we can remove from an arbitrary explanation all assertions that make use of predicates not in $\mathcal{T}$ or in q and all conditions in Definition 1 continue to be satisfied (the removed assertions are irrelevant for certain answers and for ontology consistency). Second, $\mathcal{E}$ is built from a small number of fresh constants. This can be shown using the FOL-rewritability of queries in *DL-Lite*$_{\mathcal{A}}$ [Calvanese et al., 2009], which states that the certain answers to a UCQ over an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ can be computed by rewriting each CQ q_i in it into a UCQ Q'_i to be evaluated over $\mathcal{A}$ alone, seen as a standard relational database. From this and Definition 1 it follows that if there is a solution to $\mathcal{P}$, then there exists an explanation $\mathcal{E}$ such that some CQ q'_i in Q'_i has a match in $\mathcal{E} \cup \mathcal{A}$. Furthermore, $|\mathcal{E}|$ is bounded by $|q'_i|$, while the number of fresh constants in $\mathcal{E}$ is bounded by the number of variables in q'_i. Indeed, a match for q'_i needs to map only the terms and the atoms occurring in the query. Since it follows from [Calvanese et al., 2009] that each q'_i in Q'_i has at most $|q_i|$ atoms and $2 \cdot |q_i|$ terms, we obtain:

Proposition 1. *If $\mathcal{P} = \langle \mathcal{T}, \mathcal{A}, q, \vec{c}, \Sigma \rangle$ has an explanation, then $\mathcal{P}$ has an explanation $\mathcal{E}$ with concepts and roles only from $\mathcal{T}$ and q, at most $\max(q)$ atoms, and at most $2 \cdot \max(q)$ fresh ABox constants, where $\max(q) = \max_{q_i \in q} |q_i|$.*

We now turn to the complexity of finding explanations. An algorithm for EXIST can non-deterministically guess an ABox $\mathcal{E}$ and check in polynomial time whether $\mathcal{E}$ is an explanation for the given QAP. The NP-hardness is proved by reducing the well-known problem of finding a homomorphism between two graphs. Note that existence of an explanation for a QAP $\mathcal{P}$ implies existence of a $\subseteq$-minimal and a $\leq$-minimal explanation for $\mathcal{P}$, thus all bounds for EXIST hold also for $\subseteq$-EXIST and $\leq$-EXIST.

Theorem 2. EXIST, $\subseteq$-EXIST, $\leq$-EXIST *are NP-complete. NP-hardness holds already for QAPs with an empty TBox and a CQ.*

The NP-hardness of EXIST is caused by the restriction of the alphabet over which solutions can be found. In fact, this forbids us to explicitly encode the body of the query into the ABox, and forces us to search for a match. However, if the signature is not constrained, i.e., $\Sigma = N_C \cup N_R$, the problem can be solved in polynomial time. To see this, keep in mind that CQs, seen as FO formulae, are always satisfiable. Then an explanation does not exist only if the structure of the query is not compliant with the constraints expressed in the ontology. A naïve method to check whether a UCQ q is compliant with the ontological constraints is to iteratively go through all the CQs in q and instantiate them in the ABox, introducing fresh constants for the variables. If for none of the CQs we obtain a consistent ontology, then the query violates some of the constraints imposed at the conceptual level. However, we need to take into account that the introduced constants might not correspond to distinct individuals. Indeed, it can be proved that EXIST is equivalent to the PTIME-complete consistency problem for *DL-Lite*$_{\mathcal{A}}$ without the unique name assumption [Artale et al., 2009].

Theorem 3. *For QAPs with unrestricted signatures,* EXIST, $\subseteq$-EXIST, *and* $\leq$-EXIST *are* PTIME-*complete.*

Deciding Necessity

The NP-hardness of EXIST implies the intractability of NEC. Indeed, one can reduce EXIST to non-NEC and obtain a coNP-lower bound on NEC. To decide in coNP whether a given assertion $\varphi(\vec{t})$ is necessary for a QAP $\mathcal{P} = \langle \mathcal{O}, q, \vec{c}, \Sigma \rangle$, we first check whether $\mathcal{O}$ is consistent and entails $\varphi(\vec{t})$, as in this case $\varphi(\vec{t})$ is not necessary. If not, we construct a new ontology $\mathcal{O}'$ whose models are those models of $\mathcal{O}$ in which $\varphi(\vec{t})$ does not hold, and then check EXIST for the new QAP $\mathcal{P}' = \langle \mathcal{O}', q, \vec{c}, \Sigma \rangle$. Clearly, $\varphi(\vec{t})$ is necessary iff there is no explanation for $\mathcal{P}'$. These bounds hold also for $\subseteq$-NEC, since $\varphi(\vec{t})$ occurs in all explanations for $\mathcal{P}$ iff $\varphi(\vec{t})$ occurs in all $\subseteq$-minimal explanations for $\mathcal{P}$. Hence:

Theorem 4. NEC *and* $\subseteq$-NEC *are* coNP*-complete. For QAPs with unrestricted explanation signature,* NEC *and* $\subseteq$-NEC *are* PTIME*-complete.*

Now, we consider necessity under the minimum explanation size order and we show that under common assumptions the problem is harder than NEC. Intuitively, this is because one has to compute first the minimal size of an explanation and, then, inspect all the explanations of that size. This intuition can be directly translated into an algorithm, which uses a polynomial number of parallel calls to an NP oracle. Consider an assertion α and a QAP $\mathcal{P}$. By Proposition 1, we know that the size of explanations for $\mathcal{P}$ is bounded by some m which depends polynomially on the size of the query. Our algorithm prepares and then issues in parallel m calls to the oracle of the form $\langle A_i, B_i \rangle$, where A_i asks whether there exists an explanation for $\mathcal{P}$ of size i that contains α, and B_i asks whether there exists no explanation for $\mathcal{P}$ that is strictly smaller than i. One can show that α is necessary for $\mathcal{P}$ iff there is no index $j \leq m$, such that the oracle for input A_j and B_j returns respectively true and false. The hardness can be proved by reducing the $\mathsf{P}^{\mathsf{NP}}_{\parallel}$-complete problem ODDMINVERTEXCOVER [Wagner, 1987].

Theorem 5. $\leq$-NEC *is* $P^{\mathsf{NP}}_{\parallel}$*-complete.* $P^{\mathsf{NP}}_{\parallel}$*-hardness holds already for QAPs with an empty TBox, a CQ, and an unrestricted explanation signature.*

Deciding Relevance

The NP-hardness of NEC extends also to REL. Membership in NP can be shown by a reduction to EXIST. For a given QAP $\mathcal{P}$ and an assertion α, construct from $\mathcal{P}$ a new QAP $\mathcal{P}'$ by extending the ABox in $\mathcal{P}$ with the assertion α. Intuitively, we restrict the search to those explanations that do not contradict α. Then α is relevant for $\mathcal{P}$ iff an explanation for $\mathcal{P}'$ exists.

Theorem 6. REL *is NP-complete. NP-hardness holds already for QAPs with an empty TBox and a CQ.*

When dealing with $\subseteq$-REL there is another increase in complexity. Intuitively, this is because there is an exponential number of candidate explanations to examine and for each of them one has to check that none of its subsets is itself an explanation, which requires a coNP computation. Formally, a Σ^{P}_2 algorithm for the problem can be obtained similarly as for existence, by first non-deterministically guessing an ABox $\mathcal{E}$ containing α and checking that it is an explanation. Then we must additionally check that none of the subsets of $\mathcal{E}$ is itself an explanation, from which it follows that α is $\subseteq$-relevant. Since deciding whether an explanation has a subset that is itself an explanation can be easily done in NP, the problem is solvable in non-deterministic polynomial time by a TM with an NP oracle. A matching lower bound can be shown by reducing the Σ^{P}_2-complete problem co-CERT3COL [Stewart, 1991].

Theorem 7. $\subseteq$-REL *is* Σ^{P}_2*-complete.* Σ^{P}_2*-hardness holds already for (i) QAPs with an empty TBox and a CQ, and (ii) QAPs with an empty TBox, a UCQ, and an unrestricted explanation signature.*

Unsurprisingly, $\leq$-REL has the same complexity as $\leq$-NEC. The two problems share the same source of complexity, namely the need to inspect all explanations up to a computed size, which allows us to reduce the ODDMINVERTEXCOVER problem. In fact, $\mathsf{P}^{\mathsf{NP}}_{\parallel}$-hardness can be shown using the same reduction as in the proof of Theorem 5. A matching upper bound can also be obtained by slightly modifying the algorithm for $\leq$-NEC.

Theorem 8. $\leq$-REL *is* $P^{\mathsf{NP}}_{\parallel}$*-complete.* $P^{\mathsf{NP}}_{\parallel}$*-hardness holds already for QAPs with an empty TBox, a CQ, and an unrestricted explanation signature.*

Recognizing Explanations

To solve REC one needs to check consistency of the explanation with the ontology, and check whether the tuple is in the certain answer to the query. The former is polynomial and the latter in NP, therefore REC is in NP. One can show NP-hardness by reducing the problem of finding a homomorphism between two directed graphs.

Theorem 9. REC *is NP-complete. NP-hardness holds already for QAPs with an empty TBox, a CQ, and an unrestricted explanation signature.*

In case a preference order is in place, to recognize an explanation one has to check minimality as well. This check is coNP-hard for $\subseteq$- and $\leq$-minimality, leading to completeness for DP. Membership in DP can be shown by providing two languages $L_1 \in$ NP and $L_2 \in$ coNP, such that the set of all yes-instances of our problem is $L_1 \cap L_2$. For $\leq$-REC, we simply let $L_1 = \{(\mathcal{P}, \mathcal{E}) \mid \mathcal{E} \text{ is an explanation for } \mathcal{P}\}$ and $L_2 = \{(\mathcal{P}, \mathcal{E}) \mid \mathcal{P} \text{ has no explanation } \mathcal{E}' \text{ s.t. } |\mathcal{E}'| \leq |\mathcal{E}|\}$. For $\subseteq$-REC, we take L_1 as above and $L_2 = \{(\mathcal{P}, \mathcal{E}) \mid \mathcal{P} \text{ does not have an explanation } \mathcal{E}' \text{ s.t. } \mathcal{E}' \subset \mathcal{E}\}$. Matching hardness is shown by a reduction from the problem HP-NOHP, that is, given two directed graphs G and G', decide whether G has a Hamilton path and G' does not have one.

Theorem 10. $\leq$-REC *and* $\subseteq$-REC *are DP-complete. DP-hardness holds already for QAPs with an empty TBox, a CQ, and an unrestricted explanation signature.*

Computing Explanations We discuss now the problem of actually computing a solution to a QAP $\mathcal{P}$. The complexity of this problem is determined by the established complexity bounds for reasoning tasks over QAPs. Consider first the problem of finding an arbitrary solution $\mathcal{E}$ to $\mathcal{P}$ with unrestricted explanation signature. By Theorem 3, one can do so in polynomial-time by creating a suitable instantiation of the query in the ABox. In general, however, one cannot do better than guessing an ABox $\mathcal{E}$ and deciding whether $\mathcal{E} \in \mathsf{expl}_{\preceq}(\mathcal{P})$. The intuition is that the search space for solutions is intrinsically exponential in the size of the query and minimality criteria require a check over all the solutions.

Conclusions

In this paper we characterize the computational complexity of the novel problem of explanation of negative answers to user queries over *DL-Lite*$_{\mathcal{A}}$ ontologies. All the lower bounds proved in the paper do not rely on the notion of FOL rewritability. Our upper bounds rely on FOL rewritability only to argue that solutions are of polynomial size w.r.t. the input query, and on the fact that query answering can be done in NP. For this reason, we expect our results to carry over to other DLs that admit "small" explanations and for which query answering is in NP. For instance, the complexity bounds are applicable to OWL 2-QL, which is obtained from *DL-Lite*$_{\mathcal{A}}$ by forbidding functionality assertions and dropping the unique name assumption (as our results do not rely on functionality axioms, the unique name assumption is irrelevant).

For more expressive DLs, some bounds on the complexity of our reasoning tasks can also be inferred. For QAPs with unrestricted signature, deciding the existence of an explanation has in general the same complexity as checking ontology consistency. If we consider restricted signature, lower bounds follow from results on $\mathcal{CQ}$-emptiness from [Baader et al., 2010], while we expect some upper bounds to be inherited from the complexity of query entailment.

In this work we have focused on combined complexity. With respect to *data complexity* (i.e., when both the query and the TBox are considered fixed), we observe that those inference tasks that we have shown to be NP-complete essentially rely on checking ontology consistency. It follows that they are FOL rewritable, and hence in AC^0 in data complexity. Moreover, given that explanations are bounded by the size of the query (see Proposition 1), it is easy to see that for a fixed query, there are only polynomially many explanations. Hence all our reasoning tasks are polynomial in data complexity and in *ontology complexity* (i.e., when only the query is considered fixed).

Finally, it would be interesting to apply this framework to other lightweight description logics, starting with those of the $\mathcal{EL}$-family. Also, we would like to investigate other minimality criteria. For instance, semantic criteria allow one to reward explanations that are less/more constraining in terms of the models of an ontology.

Acknowledgements. This work was partially supported by the Austrian Science Fund (FWF) grants P20840 and T515, the EU FP7 projects ACSI (FP7-257593), and by Alcatel-Lucent and EPSRC.

References

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

Baader, F.; Bienvenu, M.; Lutz, C.; and Wolter, F. 2010. Query and predicate emptiness in description logics. In *Proc. of KR 2010*.

Borgida, A.; Calvanese, D.; and Rodríguez-Muro, M. 2008. Explanation in the *DL-Lite* family of description logics. In *Proc. of ODBASE 2008*, volume 5332 of *LNCS*, 1440–1457. Springer.

Borgida, A.; Franconi, E.; and Horrocks, I. 2000. Explaining $\mathcal{ALC}$ subsumption. In *Proc. of ECAI 2000*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodríguez-Muro, M.; and Rosati, R. 2009. Ontologies and databases: The *DL-Lite* approach. In Tessaris, S., and Franconi, E., eds., *Semantic Technologies for Informations Systems – 5th Int. Reasoning Web Summer School (RW 2009)*, volume 5689 of *LNCS*. Springer. 255–356.

Calvanese, D.; Ortiz, M.; Simkus, M.; and Stefanoni, G. 2012. The complexity of explaining negative query answers in *DL-Lite*. Technical Report INFSYS RR-1843-12-03, Institut für Informatik, Technische Universität Wien. Available at http://www.kr.tuwien.ac.at/research/reports/rr1203.pdf.

Chapman, A., and Jagadish, H. V. 2009. Why not? In *Proc. of ACM SIGMOD 2009*, 523–534.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *J. of the ACM* 42(1):3–42.

Horridge, M.; Parsia, B.; and Sattler, U. 2008. Laconic and precise justifications in OWL. In *Proc. of ISWC 2008*, volume 5318 of *LNCS*, 323–338. Springer.

Huang, J.; Chen, T.; Doan, A.; and Naughton, J. 2008. On the provenance of non-answers to queries over extracted data. *PVLDB* 1(1):736–747.

Klarman, S.; Endriss, U.; and Schlobach, S. 2011. ABox abduction in the description logic $\mathcal{ALC}$. *J. of Automated Reasoning* 46(1):43–80.

McGuinness, D. L., and Borgida, A. 1995. Explaining subsumption in description logics. In *Proc. of IJCAI'95*, 816–821.

Motik, B.; Fokoue, A.; Horrocks, I.; Wu, Z.; Lutz, C.; and Grau, B. C. 2009. OWL 2 Web Ontology Language Profiles. W3C Recommendation, World Wide Web Consortium.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison Wesley Publ. Co.

Penaloza, R., and Sertkaya, B. 2010. Complexity of axiom pinpointing in the *DL-Lite* family. In *Proc. of DL 2010*, volume 573 of *CEUR,* ceur-ws.org.

Stewart, I. A. 1991. Complete problems involving boolean labelled structures and projection transactions. *J. of Logic and Computation* 1(6):861–882.

Wagner, K. W. 1987. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science* 51(1–2):53–80.