The Data Complexity of Answering Instance Queries in $\mathcal{FL}_0$

Franz Baader*
TU Dresden, Chair for Automata Theory
Dresden, Germany

Pavlos Marantidis*
TU Dresden, Chair for Automata Theory
Dresden, Germany

Maximilian Pensel*
TU Dresden, Chair for Automata Theory
Dresden, Germany

ABSTRACT

Ontology-mediated query answering can be used to access large data sets through a mediating ontology. It has drawn considerable attention in the Description Logic (DL) community where both the complexity of query answering and practical query answering approaches based on rewriting were investigated in detail. Surprisingly, there is still a gap in what is known about the data complexity of query answering w.r.t. ontologies formulated in the inexpressive DL $\mathcal{FL}_0$. While it is known that the data complexity of answering conjunctive queries w.r.t. $\mathcal{FL}_0$ ontologies is coNP-complete, the exact complexity of answering instance queries was open until now. In the present paper, we show that answering instance queries w.r.t. $\mathcal{FL}_0$ ontologies is in P for data complexity. Together with the known lower bound of P-completeness for a fragment of $\mathcal{FL}_0$, this closes the gap mentioned above.

CCS CONCEPTS

• **Theory of computation → Description logics**; *Database query languages (principles)*;

KEYWORDS

Ontology-Mediated Query Answering, Data Complexity, Description Logic $\mathcal{FL}_0$

ACM Reference Format:
Franz Baader, Pavlos Marantidis, and Maximilian Pensel. 2018. The Data Complexity of Answering Instance Queries in $\mathcal{FL}_0$. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3184558.3191618

1 INTRODUCTION

In the early days of DL research, the inexpressive DL $\mathcal{FL}_0$, which has only conjunction, value restriction and the top concept as concept constructors, was considered to be the smallest possible DL. In fact, when providing a formal semantics for so-called property edges of semantic networks in the first DL system KL-ONE [5], value restrictions were used. For this reason, the language for constructing concepts in KL-ONE and all of the other early DL systems [4, 14, 17, 21] contained $\mathcal{FL}_0$. It came as a surprise when it was shown that subsumption reasoning w.r.t. acyclic $\mathcal{FL}_0$ terminologies (TBoxes) is coNP-hard [15]. The complexity increases when more expressive forms of TBoxes are used: for cyclic TBoxes to PSpace [1, 10] and for general TBoxes consisting of general concept inclusions (GCIs) even to ExpTime [2]. Thus, w.r.t. general TBoxes, subsumption reasoning in $\mathcal{FL}_0$ is as hard as subsumption reasoning in $\mathcal{ALC}$, its closure under negation. These negative complexity results for $\mathcal{FL}_0$ were one of the reasons why the attention in DL research shifted from $\mathcal{FL}_0$ to $\mathcal{EL}$, which is obtained from $\mathcal{FL}_0$ by replacing value restriction with existential restriction as a constructor. In fact, subsumption reasoning in $\mathcal{EL}$ stays polynomial even in the presence of general TBoxes [6].

In the present paper, we are not concerned with subsumption reasoning, but with ontology-mediated query answering [16]. In this setting, one has a DL TBox $\mathcal{T}$ as well as an ABox $\mathcal{A}$ (representing the data) together with a query q, and one wants to know whether TBox and ABox imply that a given tuple is an answer to the query. As queries, one usually considers conjunctive queries or instance queries. In the latter case, the query is actually a concept of the employed DL. As usual in a database context, one distinguishes between combined complexity and data complexity: in the former, the complexity is measured in the combined size of all three inputs $\mathcal{T}$, $\mathcal{A}$, q, whereas in the latter one assumes $\mathcal{T}$ and q to be of constant size, and measures the complexity only in the size of the ABox $\mathcal{A}$.

With respect to combined complexity, $\mathcal{FL}_0$ behaves the same as $\mathcal{ALC}$ also for ontology-mediated query answering. Both answering instance queries and answering conjunctive queries w.r.t. general $\mathcal{FL}_0$ TBoxes are ExpTime-complete, just as for $\mathcal{ALC}$. The upper bound is inherited from $\mathcal{ALC}$ [12], and the lower bound obviously follows from ExpTime-hardness of subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes [2].

However, w.r.t. data complexity, $\mathcal{FL}_0$ exhibits an interesting behaviour, which differs from the one of $\mathcal{ALC}$ and of $\mathcal{EL}$. In fact, in $\mathcal{ALC}$ and in $\mathcal{EL}$, the data complexity for instance queries and for conjunctive queries are the same, namely both coNP-complete for $\mathcal{ALC}$ [11, 20] and P-complete for $\mathcal{EL}$ [8, 19]. In contrast, for $\mathcal{FL}_0$, the data complexity is coNP-complete for conjunctive queries, whereas it is P-complete for instance queries. For the case of conjunctive queries, the coNP upper bound is inherited from $\mathcal{ALC}$, and the coNP lower bound is shown for a certain fragment of $\mathcal{FL}_0$ in the proof of Theorem 4.5 in [8]. In this fragment, GCIs may have a value restriction $\forall r.A$ or a concept name A as left-hand side, and a concept name B as right-hand side. For instance queries, the P lower bound is shown for another fragment of $\mathcal{FL}_0$, where left-hand sides of GCIs are conjunctions of concept names and right-hand sides are value restrictions $\forall r.A$ or concept names A (Theorem 4.3 in [8]). For this fragment, a P upper bound is also established in [8], but the exact data complexity of answering instance queries w.r.t. general $\mathcal{FL}_0$ TBoxes is left open in [8].

*Email contact: {firstname}.{lastname}@tu-dresden.de

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3191618

In the present paper, we close this gap by proving a P upper bound for the data complexity of answering instance queries w.r.t. general $\mathcal{FL}_0$ TBoxes.

2 THE DESCRIPTION LOGIC $\mathcal{FL}_0$

In Description Logics, *concept constructors* are used to build complex *concepts* out of *concept names* (unary predicates) and *role names* (binary predicates). A particular DL is determined by the available constructors. The DL $\mathcal{FL}_0$ has the constructors top concept ($\top$), conjunction ($C \sqcap D$), and value restriction ($\forall r.C$). To be more precise, the set of $\mathcal{FL}_0$ concepts is inductively defined as follows:

- $\top$ and every concept name is an $\mathcal{FL}_0$ concept,
- if C, D are $\mathcal{FL}_0$ concepts and r is a role name, then $C \sqcap D$ and $\forall r.C$ are $\mathcal{FL}_0$ concepts.

The *semantics* of $\mathcal{FL}_0$ is defined using first-order interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consisting of a non-empty domain $\Delta^\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$ that assigns a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ to each concept name A and a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ to each role name r. This function is extended to $\mathcal{FL}_0$ concepts as follows:

$$\top^\mathcal{I} = \Delta^\mathcal{I} \text{ and } (C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I},$$
$$(\forall r.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \forall y \in \Delta^\mathcal{I} : (x, y) \in r^\mathcal{I} \Rightarrow y \in C^\mathcal{I}\}.$$

An $\mathcal{FL}_0$ *TBox* $\mathcal{T}$ is a finite set of *general concept inclusions (GCIs)*, which are expressions of the form $C \sqsubseteq D$ for $\mathcal{FL}_0$ concepts C, D. The interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ if it satisfies all the GCIs in $\mathcal{T}$, i.e., $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds for all GCIs $C \sqsubseteq D$ in $\mathcal{T}$. An $\mathcal{FL}_0$ *ABox* $\mathcal{A}$ is a finite set of *assertions*, which are expressions of the form $C(a)$ (concept assertion) or $r(a, b)$ (role assertion), where C is an $\mathcal{FL}_0$ concept, r a role name, and a, b are elements of an additional set of individual names. The ABox $\mathcal{A}$ is *simple* if all the concepts C in concept assertions $C(a)$ are concept names. An interpretation then additionally assigns to every individual name a in $\mathcal{A}$ an element $a^\mathcal{I} \in \Delta^\mathcal{I}$. The interpretation $\mathcal{I}$ is a *model* of $\mathcal{A}$ if it satisfies all the assertions in $\mathcal{A}$, i.e., $a^\mathcal{I} \in C^\mathcal{I}$ (resp. $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$) holds for all assertions $C(a)$ (resp. $r(a, b)$) in $\mathcal{A}$. A *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ together with an ABox $\mathcal{A}$. The interpretation $\mathcal{I}$ is a *model* of $\mathcal{K}$ if it is a model of both the TBox $\mathcal{T}$ and the ABox $\mathcal{A}$.

Given an $\mathcal{FL}_0$ TBox $\mathcal{T}$ and two $\mathcal{FL}_0$ concepts C, D, we say that *C is subsumed by D* (denoted as $C \sqsubseteq_\mathcal{T} D$) if $C^\mathcal{I} \subseteq D^\mathcal{I}$ for all models $\mathcal{I}$ of $\mathcal{T}$. These two concept descriptions are *equivalent* (denoted as $C \equiv_\mathcal{T} D$) if $C \sqsubseteq_\mathcal{T} D$ and $D \sqsubseteq_\mathcal{T} C$. If the TBox is empty, we write $C \equiv D$ instead of $C \equiv_\emptyset D$. The subsumption problem is the problem of deciding, for a given TBox $\mathcal{T}$ and given concepts C, D, whether $C \equiv_\mathcal{T} D$ holds or not.

In the presence of an ABox $\mathcal{A}$, we can also consider the *instance problem*: given an individual name a and an $\mathcal{FL}_0$ concept C we say that *a is an instance of C in $\mathcal{A}$ w.r.t. $\mathcal{T}$* (written $(\mathcal{T}, \mathcal{A}) \models C(a)$) if $a^\mathcal{I} \in C^\mathcal{I}$ for all models $\mathcal{I}$ of the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. In this case we say that a is a *certain answer* of the *instance query C* on the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. The instance problem is the problem of deciding, for a given knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a given assertion $C(a)$, whether $(\mathcal{T}, \mathcal{A}) \models C(a)$ holds or not.

For the DL $\mathcal{FL}_0$, the subsumption and the instance problem are ExpTime-complete [2]. For the instance problem, this complexity result is w.r.t. combined complexity, where the size of the input is the sum of the sizes of C, $\mathcal{T}$, and $\mathcal{A}$. In this paper, we are more concerned with *data complexity*. In this setting, one assumes that the ABox is simple and that the size of $\mathcal{T}$ and C is constant. Data complexity is thus measured as a function of the size of $\mathcal{A}$.

In the rest of this paper, we assume without loss of generality that all the GCIs in $\mathcal{FL}_0$ TBoxes are of the following normal form:

$$A \sqsubseteq B, \quad A_1 \sqcap A_2 \sqsubseteq B, \quad A \sqsubseteq \forall r.B, \quad \text{or} \quad \forall r.A \sqsubseteq B,$$

where A, A_1, A_2 are concept names or the top-concept $\top$, B is a concept name, and r is a role name. By adapting the normalisation rules for $\mathcal{EL}$ as described in [3], one can transform a given TBox $\mathcal{T}$ into a normalised TBox $\mathcal{T}'$ where all GCIs have this form. This can be achieved in polynomial time and results in a normalised TBox of linear size. This new TBox $\mathcal{T}'$ is a conservative extension (see [3]) of the original TBox $\mathcal{T}$, and thus has the same consequences over the vocabulary (i.e., concepts and role names) of $\mathcal{T}$.

We will make use of the following property of $\mathcal{FL}_0$, which follows from the properties of the so-called least functional models introduced and studied in [18]. However, below we will give a self-contained proof of this property.

LEMMA 2.1. *Let $\mathcal{T}$ be a normalised $\mathcal{FL}_0$ TBox, S the set of concept names occurring in $\mathcal{T}$, and $S' \subseteq S$ a subset of S that is closed under subsumption w.r.t. $\mathcal{T}$, i.e., if $B \in S$ is such that $\bigsqcap S' \sqsubseteq_\mathcal{T} B$, then $B \in S'$. Then there is a model $\mathcal{I}$ of $\mathcal{T}$ and an element $d \in \Delta^\mathcal{I}$ such that $\{B \in S \mid d \in B^\mathcal{I}\} = S'$.*

Before we can prove this lemma, we need to introduce a few notions and show some auxiliary results. Since $\forall r.(C \sqcap D) \equiv \forall r.C \sqcap \forall r.D$, any $\mathcal{FL}_0$ concept description is equivalent to a conjunction of nested value restrictions of the form $\forall r_1.\forall r_2.\cdots\forall r_n.A$, where A is a concept name. Here the top concept $\top$ is assumed to be the empty conjunction. In addition, A is the same as $\forall \varepsilon.A$, where ε denotes the empty word.

In the following, we will write such nested value restrictions $\forall r_1.\forall r_2.\cdots\forall r_n.A$ as $\forall w.A$, where $w = r_1 r_2 \ldots r_n$ is viewed as a word over the alphabet N_R of all role names occurring in a given $\mathcal{FL}_0$ TBox $\mathcal{T}$.

Given an $\mathcal{FL}_0$ concept C that contains only concept and role names occurring in $\mathcal{T}$, we define the *value restriction set* of C w.r.t. $\mathcal{T}$ as

$$V(C, \mathcal{T}) := \{(w, A) \mid C \sqsubseteq_\mathcal{T} \forall w.A\}.$$

We can use this set to construct a tree-shaped model $\mathcal{I}(C, \mathcal{T})$ of $\mathcal{T}$ as follows:

- $\Delta^{\mathcal{I}(C,\mathcal{T})} := N_R^*$, i.e., $\Delta^{\mathcal{I}(C,\mathcal{T})}$ consists of all finite words over N_R;
- $r^{\mathcal{I}(C,\mathcal{T})} := \{(w, wr) \mid w \in N_R^*\}$, i.e., for every role $r \in N_R$, the word w has exactly one r-successor, which is wr;
- $A^{\mathcal{I}(C,\mathcal{T})} := \{w \mid (w, A) \in V(C, \mathcal{T})\}$ for every concept name A.

An example of such a tree-shaped model is illustrated in Figure 1.

The following lemma states that $\mathcal{I}(C, \mathcal{T})$ is indeed a model of $\mathcal{T}$. Later on, we will use $\mathcal{I}(C, \mathcal{T})$ to prove Lemma 2.1: the interpretation $\mathcal{I}$ and the element d of $\Delta^\mathcal{I}$ whose existence is claimed in that lemma will be $\mathcal{I}(C, \mathcal{T})$ and the empty word $\varepsilon \in \Delta^{\mathcal{I}(C,\mathcal{T})}$, respectively.

LEMMA 2.2. *Let $\mathcal{T}$ be an $\mathcal{FL}_0$ TBox and C a conjunction of concept names. Then $\mathcal{I}(C, \mathcal{T})$ is a model of $\mathcal{T}$ and $\varepsilon \in C^{\mathcal{I}(C,\mathcal{T})}$.*

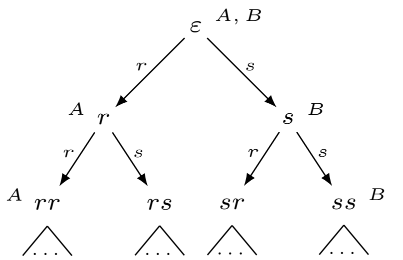

Figure 1: Graph representation of $\mathcal{I}(C,\mathcal{T})$ for $C = A \sqcap B$ and $\mathcal{T} = \{A \sqsubseteq \forall r.A, B \sqsubseteq \forall s.B\}$.

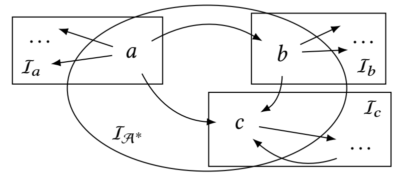

Figure 2: Example of interpretation $\mathcal{I}$ (c.f. Lemma 3.1).

Proof. In order to show that $\mathcal{I}(C,\mathcal{T})$ is a model of $\mathcal{T}$, we consider the different shapes that normalised GCIs can have:

- Consider a GCI of the form $A_1 \sqcap A_2 \sqsubseteq B$, and let $w \in N_R^*$ be such that $w \in (A_1 \sqcap A_2)^{\mathcal{I}(C,\mathcal{T})}$. Then $w \in A_i^{\mathcal{I}(C,\mathcal{T})}$ for $i = 1, 2$, and thus $C \sqsubseteq_{\mathcal{T}} \forall w.A_i$, which implies that $C \sqsubseteq_{\mathcal{T}} \forall w.(A_1 \sqcap A_2)$. Since value restrictions are monotonic w.r.t. subsumption, this implies $C \sqsubseteq_{\mathcal{T}} \forall w.B$, and thus $w \in B^{\mathcal{I}(C,\mathcal{T})}$.
- A GCI of the form $A \sqsubseteq B$ can be treated in the same way.
- Consider a GCI of the form $A \sqsubseteq \forall r.B$, and let $w \in N_R^*$ be such that $w \in A^{\mathcal{I}(C,\mathcal{T})}$. Then $w \in A^{\mathcal{I}(C,\mathcal{T})}$ implies $C \sqsubseteq_{\mathcal{T}} \forall w.A$. Using monotony of value restrictions, we obtain $C \sqsubseteq_{\mathcal{T}} \forall wr.B$, and thus $wr \in B^{\mathcal{I}(C,\mathcal{T})}$. Since wr is the only r-successor of w, this shows that $w \in (\forall r.B)^{\mathcal{I}(C,\mathcal{T})}$.
- Consider a GCI of the form $\forall r.A \sqsubseteq B$, and assume that $w \in (\forall r.A)^{\mathcal{I}(C,\mathcal{T})}$. Since wr is an r-successor of w, this implies that $wr \in A^{\mathcal{I}(C,\mathcal{T})}$, and thus $C \sqsubseteq_{\mathcal{T}} \forall wr.A$. Monotony of value restrictions yields $C \sqsubseteq_{\mathcal{T}} \forall w.B$, and thus $w \in B^{\mathcal{I}(C,\mathcal{T})}$.

Thus, $\mathcal{I}(C,\mathcal{T})$ is a model of $\mathcal{T}$. It remains to show that $\varepsilon \in C^{\mathcal{I}(C,\mathcal{T})}$, i.e., $\varepsilon \in A^{\mathcal{I}(C,\mathcal{T})}$ for all concept names A occurring in the conjunction of concept names C. Indeed, we have $C \sqsubseteq_{\mathcal{T}} A = \forall \varepsilon.A$, and thus $(\varepsilon, A) \in V(C,\mathcal{T})$, which yields $\varepsilon \in A^{\mathcal{I}(C,\mathcal{T})}$. □

We are now ready to prove Lemma 2.1. Thus, let $\mathcal{T}$ be a normalised $\mathcal{FL}_0$ TBox, S the set of concept names occurring in $\mathcal{T}$, and $S' \subseteq S$ a subset of S that is closed under subsumption w.r.t. $\mathcal{T}$. We define $C := \bigsqcap S'$ and claim that $\mathcal{I} := \mathcal{I}(C,\mathcal{T})$ and $d := \varepsilon$ satisfy the properties required by the lemma, i.e., $\{B \in S \mid \varepsilon \in B^{\mathcal{I}(C,\mathcal{T})}\} = S'$. To show this identity, first assume that $\varepsilon \in B^{\mathcal{I}(C,\mathcal{T})}$. The definition of $\mathcal{I}(C,\mathcal{T})$ then yields $C \sqsubseteq_{\mathcal{T}} \forall \varepsilon.B = B$, and thus $B \in S'$ since S' is closed under subsumption. Conversely, assume that $B \in S'$. Then B is a conjunct in C, which yields $C \sqsubseteq_{\mathcal{T}} B = \forall \varepsilon.B$, and thus $\varepsilon \in B^{\mathcal{I}(C,\mathcal{T})}$. This completes the proof of Lemma 2.1.

3 THE DATA COMPLEXITY OF INSTANCE QUERIES IN $\mathcal{FL}_0$

In this section, we will introduce a procedure for answering instance queries, show its correctness, and prove that it has polynomial time data complexity. Thus, we will assume that the ABox $\mathcal{A}$ is simple, and measure the complexity in the size of the ABox only, i.e., assume that the TBox $\mathcal{T}$ and the query C are of constant size. The main idea underlying the procedure is that we use propagation of value restrictions over role assertions as well as subsumption computations to extend the given ABox with additional concept assertions. The claim is then that the assertion $C(a)$ follows from the input knowledge base $\mathcal{K} = (\mathcal{T},\mathcal{A})$ iff $C(a)$ is contained in the extended ABox $\mathcal{A}^*$.

To be more precise, let $\mathcal{T}$ be a normalised $\mathcal{FL}_0$ TBox, $\mathcal{A}$ a simple ABox, and C an instance query. We can assume without loss of generality that C is a concept name.[1] Let S be the set of concept names occurring in $\mathcal{T}$.

Starting with $\mathcal{A}' := \mathcal{A}$, we extend the ABox by applying the following two steps iteratively until the ABox does not change anymore:

(1) If $A(a) \in \mathcal{A}', r(a,b) \in \mathcal{A}', A \sqsubseteq \forall r.B \in \mathcal{T}$ and $B(b) \notin \mathcal{A}'$, then extend $\mathcal{A}'$ with $B(b)$. Apply this rule as long as it adds concept assertions to $\mathcal{A}'$. Continue with Step 2.
(2) If $S' = \{A \in S \mid A(a) \in \mathcal{A}'\}$ and $\bigsqcap S' \sqsubseteq_{\mathcal{T}} B$ for a concept name $B \in S$ such that $B \notin S'$, then extend $\mathcal{A}'$ with $B(a)$. Apply this rule as long as it adds concept assertions to $\mathcal{A}'$. If no new concept assertions were added in this step at all, then terminate; otherwise continue with Step 1.

Since the TBox is assumed to be of constant size, the cardinality of the set S is also constant. For every individual name a, only a constant number of concept assertions can be added. Thus, overall there is only a linear number of additions. Also note that performing each subsumption test $\bigsqcap S' \sqsubseteq_{\mathcal{T}} B$ takes constant time since the size of $\mathcal{T}$ and of S' are bounded by a constant. Let $\mathcal{A}^*$ be the ABox obtained by extending the input ABox $\mathcal{A}$ in this way.

Lemma 3.1. *The individual a is a certain answer of C on $(\mathcal{T},\mathcal{A})$ iff $C(a) \in \mathcal{A}^*$.*

Proof. First, assume that $C(a) \in \mathcal{A}^*$. It is easy to see that all the assertions added during the iteration actually follow from the current knowledge base $(\mathcal{T},\mathcal{A}')$, and thus hold in every model of the original knowledge base $(\mathcal{T},\mathcal{A})$. Consequently, $C(a) \in \mathcal{A}^*$ implies that $C(a)$ holds in every model of $(\mathcal{T},\mathcal{A})$, which shows that a is a certain answer of C on $(\mathcal{T},\mathcal{A})$.

Second, assume that $C(a) \notin \mathcal{A}^*$. We use this to construct a model $\mathcal{I}$ of $(\mathcal{T},\mathcal{A}^*)$ in which $C(a)$ does not hold. Since $\mathcal{A} \subseteq \mathcal{A}^*$, $\mathcal{I}$ is also a model of $(\mathcal{T},\mathcal{A})$, which implies that a is not a certain answer of C on $(\mathcal{T},\mathcal{A})$. The model $\mathcal{I}$ consists of several parts.

(1) First, we construct an interpretation $\mathcal{I}_{\mathcal{A}^*}$ whose domain consists of all individuals occurring in $\mathcal{A}^*$. The role relationships between these individuals are given by $\mathcal{A}^*$, i.e., for two individuals a, b occurring in $\mathcal{A}^*$ we have $(a,b) \in r^{\mathcal{I}_{\mathcal{A}^*}}$ iff

[1] Since both C and $\mathcal{T}$ are assumed to be of constant size, we can simply introduce a name A_C for C using the GCIs $A_C \sqsubseteq C$ and $C \sqsubseteq A_C$.

$r(a,b) \in \mathcal{A}^*$. Similarly, for an individual a occurring in $\mathcal{A}^*$ and a concept name $A \in S$ we have $a \in A^{\mathcal{I}_{\mathcal{A}^*}}$ iff $A(a) \in \mathcal{A}^*$.

(2) For each individual a occurring in $\mathcal{A}^*$, we consider the set $S'(a) := \{A \in S \mid A(a) \in \mathcal{A}^*\}$. This set is closed in the sense of Lemma 2.1 since in Step 2 of the extension procedure no more concept names could be added. Let $\mathcal{I}_a$ be a model of $\mathcal{T}$ and $d_a \in \Delta^{\mathcal{I}_a}$ be such that $\{B \in S \mid d_a \in B^{\mathcal{I}_a}\} = S'(a)$. Such a model exists due to Lemma 2.1.

(3) Without loss of generality we assume that all the domains $\Delta^{\mathcal{I}_a}$ are disjoint from each other, but d_a is actually equal to a. The interpretation $\mathcal{I}$ is now the union of $\mathcal{I}_{\mathcal{A}^*}$ with all the models $\mathcal{I}_a$ for individuals a occurring in $\mathcal{A}^*$ (see Fig. 2 for an illustrating example).

Note that for all concept names A occurring in $\mathcal{T}$ we actually have $a = d_a \in A^{\mathcal{I}_a}$ iff $A(a) \in \mathcal{A}^*$ iff $a \in A^{\mathcal{I}_{\mathcal{A}^*}}$, and thus $a \in A^{\mathcal{I}}$ iff $A(a) \in \mathcal{A}^*$. There might be assertions $B(a) \in \mathcal{A}^*$ for concept names B not occurring in $\mathcal{T}$, and for those we have $a \in B^{\mathcal{I}_{\mathcal{A}^*}}$ and thus also $a \in B^{\mathcal{I}}$, even though $d_a \notin B^{\mathcal{I}_a}$. However, the extension of $\mathcal{I}_a$ with such additional concept memberships does not interfere with the satisfaction of GCIs of $\mathcal{T}$ since B does not occur in $\mathcal{T}$. The role successors of an individual a occurring in $\mathcal{A}^*$ are the role successors in the ABox plus the ones in $\mathcal{I}_a$.

By construction, $\mathcal{I}$ is a model of $\mathcal{A}^*$. We show that it is also a model of $\mathcal{T}$ by case distinction according to the form of the GCI to be satisfied:

- Consider a GCI of the form $A_1 \sqcap A_2 \sqsubseteq B$, and let $d \in \Delta^{\mathcal{I}}$ be such that $d \in A_1^{\mathcal{I}}$ and $d \in A_2^{\mathcal{I}}$. If d is not an individual name from $\mathcal{A}^*$, then there is such an individual name a such that $d \in \Delta^{\mathcal{I}_a} \setminus \{a\}$. According to the definition of $\mathcal{I}$, we have $d \in A_1^{\mathcal{I}_a}$ and $d \in A_2^{\mathcal{I}_a}$. But then the fact that $\mathcal{I}_a$ is a model of $\mathcal{T}$ yields $d \in B^{\mathcal{I}_a}$, which implies $d \in B^{\mathcal{I}}$.
  Assume that $d = a$ is an individual name from $\mathcal{A}^*$. Then $A_1, A_2 \in S'(a)$. Since this set is closed, we also have $B \in S'(a)$, which yields $B(a) \in \mathcal{A}^*$. Thus $a \in B^{\mathcal{I}}$ since we have already seen that $\mathcal{I}$ is a model of $\mathcal{A}^*$.
- A GCI of the form $A \sqsubseteq B$ can be treated in the same way.
- Consider a GCI of the form $A \sqsubseteq \forall r.B$, and let $d \in \Delta^{\mathcal{I}}$ be such that $d \in A^{\mathcal{I}}$. If d is not an individual name from $\mathcal{A}^*$, then there is such an individual name a such that $d \in \Delta^{\mathcal{I}_a} \setminus \{a\}$. Consequently, all the r-successors of d in $\mathcal{I}$ are actually elements of $\Delta^{\mathcal{I}_a}$. Since $\mathcal{I}_a$ is a model of $\mathcal{T}$, all these r-successors must belong to $B^{\mathcal{I}_a}$, and thus to $B^{\mathcal{I}}$.
  Assume that $d = a$ is an individual name from $\mathcal{A}^*$. We need to show that the r-successors of d in $\mathcal{I}$ belong to $B^{\mathcal{I}}$. For the ones actually belonging to $\mathcal{I}_a$, this follows from the fact that $\mathcal{I}_a$ is a model of $\mathcal{T}$. Now, consider an individual b such that $r(a,b) \in \mathcal{A}^*$. But then the fact that no more concept assertions could be added in Step 1 of the extension procedure implies that $B(b) \in \mathcal{A}^*$, and thus $b \in B^{\mathcal{I}}$.
- Consider a GCI of the form $\forall r.A \sqsubseteq B$, and assume that $d \in \Delta^{\mathcal{I}}$ satisfies $d \in (\forall r.A)^{\mathcal{I}}$. Clearly, there exists an individual a from $\mathcal{A}^*$ such that $d \in \Delta^{\mathcal{I}_a}$. Since all the r-successors of d in $\mathcal{I}_a$ are also r-successors of d in $\mathcal{I}$, this implies that $d \in (\forall r.A)^{\mathcal{I}_a}$. Thus, the fact that $\mathcal{I}_a$ is a model of $\mathcal{T}$ implies $d \in B^{\mathcal{I}_a}$, which yields $d \in B^{\mathcal{I}}$.

We have thus shown that $\mathcal{I}$ is a model of the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. In addition, we have $a \notin C^{\mathcal{I}}$ since $C(a) \notin \mathcal{A}^*$. This shows that a cannot be a certain answer of C on $(\mathcal{T}, \mathcal{A})$. □

Since $\mathcal{A}^*$ can be computed in time linear in the size of the data and checking whether $C(a) \in \mathcal{A}^*$ is also possible in linear time this shows our desired data complexity result.

Theorem 3.2. *The data complexity of answering instance queries w.r.t. general $\mathcal{FL}_0$ TBoxes is P-complete.*

Proof. The complexity upper bound (in P) is an immediate consequence of Lemma 3.1 and the fact that $\mathcal{A}^*$ can be computed in time linear in the size of the data. P-hardness has been shown in [8] for a fragment of $\mathcal{FL}_0$. □

4 CONCLUSION

We have shown in this paper that the data complexity of answering instance queries w.r.t. general $\mathcal{FL}_0$ TBoxes is in P. This upper bound matches the known lower bound of P-hardness that holds even for a fragment of $\mathcal{FL}_0$. On the one hand, the P-hardness results is a negative result since it shows that in $\mathcal{FL}_0$ first-order rewritability of queries is not possible [8]. From a practical point of view this means that one cannot simply reduce ontology-mediated query answering w.r.t. $\mathcal{FL}_0$ ontologies to answering SQL queries over the ABox viewed as a relational database. On the other hand, the P upper bound means that Datalog rewritability [7, 9, 13] might be a viable option for obtaining a query answering procedure that is more practical than the one introduced in this paper. As future work we will address this issue, i.e., try to find a Datalog rewriting approach for $\mathcal{FL}_0$ instance queries w.r.t. general $\mathcal{FL}_0$ TBoxes.

ACKNOWLEDGMENTS

This work is partially supported by the German Research Foundation (DFG) within the Research Training Group QuantLA (GRK 1763) and within the Collaborative Research Center SFB 912 – HAEC.

REFERENCES

[1] Franz Baader. 1990. Terminological Cycles in KL-ONE-based Knowledge Representation Languages. In *Proc. of the 8th Nat. Conf. on Artificial Intelligence (AAAI'90)*. Boston (Ma, USA), 621–626.

[2] Franz Baader, Sebastian Brandt, and Carsten Lutz. 2005. Pushing the $\mathcal{EL}$ Envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, Leslie Pack Kaelbling and Alessandro Saffiotti (Eds.). Morgan Kaufmann, Los Altos, Edinburgh (UK), 364–369.

[3] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic.* Cambridge University Press.

[4] Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, Lori Alperin Resnick, and Alexander Borgida. 1991. Living with CLASSIC: When and how to Use a KL-ONE-like Language. In *Principles of Semantic Networks*, John F. Sowa (Ed.). Morgan Kaufmann, Los Altos, 401–456.

[5] Ronald J. Brachman and James G. Schmolze. 1985. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9, 2 (1985), 171–216.

[6] Sebastian Brandt. 2004. Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and—What Else?. In *Proc. of the 16th Eur. Conf. on Artificial Intelligence (ECAI 2004)*, Ramon López de Mántaras and Lorenza Saitta (Eds.). 298–302.

[7] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. 2012. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.* 14 (2012), 57–83.

[8] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. 2013. Data Complexity of Query Answering in Description Logics. *Artificial Intelligence* 195 (2013), 335–360.

[9] Ullrich Hustadt, Boris Motik, and Ulrike Sattler. 2007. Reasoning in Description Logics by a Reduction to Disjunctive Datalog. *J. Autom. Reasoning* 39, 3 (2007), 351–384.

[10] Yevgeny Kazakov and Hans de Nivelle. 2003. Subsumption of concepts in $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete. In *Proc. of the 2003 Description Logic Workshop (DL 2003)*. CEUR Electronic Workshop Proceedings, http://CEUR-WS.org/Vol-81/.

[11] Alon Y. Levy and Marie-Christine Rousset. 1998. Verification of Knowledge Bases based on Containment Checking. *Artificial Intelligence* 101(1-2) (1998), 227–250.

[12] Carsten Lutz. 2008. The Complexity of Conjunctive Query Answering in Expressive Description Logics. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2008) (Lecture Notes in Artificial Intelligence)*, Alessandro Armando, Peter Baumgartner, and Gilles Dowek (Eds.). Springer-Verlag, 179–193.

[13] Carsten Lutz and Frank Wolter. 2017. The Data Complexity of Description Logic Ontologies. *Logical Methods in Computer Science* 13, 4 (2017).

[14] E. Mays, R. Dionne, and R. Weida. 1991. K-REP System Overview. *SIGART Bull.* 2, 3 (1991).

[15] Bernhard Nebel. 1990. Terminological Reasoning is Inherently Intractable. *Artificial Intelligence* 43 (1990), 235–249.

[16] Magdalena Ortiz. 2013. Ontology Based Data Access: The Story So Far. In *Proc. of the 7th Alberto Mendelzon Int. Workshop on Foundations of Data Management (CEUR Workshop Proceedings)*, Loreto Bravo and Maurizio Lenzerini (Eds.), Vol. 1087. http://ceur-ws.org/Vol-1087/keynote3.pdf

[17] Christof Peltason. 1991. The BACK System — an Overview. *SIGART Bull.* 2, 3 (1991), 114–119.

[18] Maximilian Pensel. 2015. *An automata based approach for subsumption w.r.t. general concept inclusions in the description logic $\mathcal{FL}_0$*. Master's thesis. Chair for Automata Theory, TU Dresden, Germany. See http://lat.inf.tu-dresden.de/research/mas.

[19] Riccardo Rosati. 2007. On Conjunctive Query Answering in $\mathcal{EL}$. In *Proc. of the 2007 Description Logic Workshop (DL 2007) (CEUR Workshop Proceedings)*, Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris (Eds.), Vol. 250. CEUR-WS.org.

[20] Andrea Schaerf. 1993. On the Complexity of the Instance Checking Problem in Concept Languages with Existential Quantification. *J. of Intelligent Information Systems* 2 (1993), 265–278.

[21] William A. Woods and James G. Schmolze. 1992. The KL-ONE Family. In *Semantic Networks in Artificial Intelligence*, F. W. Lehmann (Ed.). Pergamon Press, 133–178. Published as a special issue of *Computers & Mathematics with Applications*, Volume 23, Number 2–9.