Frontiers and Exact Learning of $\mathcal{ELI}$ Queries under DL-Lite Ontologies

Maurice Funk[1] , **Jean Christoph Jung**[2] and **Carsten Lutz**[1]
[1] Leipzig University, Faculty of Mathematics and Computer Science, Germany
[2] University of Hildesheim, Institute of Computer Science, Germany
mfunk@informatk.uni-leipzig.de, jungj@uni-hildesheim.de, clu@informatik.uni-leipzig.de

Abstract

We study $\mathcal{ELI}$ queries (ELIQs) in the presence of ontologies formulated in the description logic *DL-Lite*. For the dialect *DL-Lite*$^\mathcal{H}$, we show that ELIQs have a frontier (set of least general generalizations) that is of polynomial size and can be computed in polynomial time. In the dialect *DL-Lite*$^\mathcal{F}$, in contrast, frontiers may be infinite. We identify a natural syntactic restriction that enables the same positive results as for *DL-Lite*$^\mathcal{H}$. We use our results on frontiers to show that ELIQs are learnable in polynomial time in the presence of a *DL-Lite*$^\mathcal{H}$/ restricted *DL-Lite*$^\mathcal{F}$ ontology in Angluin's framework of exact learning with only membership queries.

1 Introduction

In the widely studied paradigm of ontology-mediated querying, a database query is enriched with an ontology that provides domain knowledge as well as additional vocabulary for query formulation [Bienvenu *et al.*, 2013b; Calvanese *et al.*, 2009]. We consider ontologies formulated in description logics (DLs) of the *DL-Lite* family and queries that are $\mathcal{ELI}$ queries (ELIQs) or, in other words, tree-shaped unary conjunctive queries (CQs). *DL-Lite* is a prominent choice for the ontology language as it underpins the OWL 2 QL profile of the OWL ontology language [OWL Working Group, 2009]. Likewise, ELIQs are a prominent choice for the query language as they are computationally very well-behaved: without an ontology, they can be evaluated in polynomial time in combined complexity, in contrast to NP-completeness for unrestricted CQs. Moreover, in the form of $\mathcal{ELI}$ concepts they are a central building block of ontologies in several dialects of *DL-Lite* and beyond.

The aim of this paper is to study the related topics of computing least general generalizations (LGGs) of ELIQs under *DL-Lite* ontologies and learning ELIQs under *DL-Lite* ontologies in Angluin's framework of exact learning [Angluin, 1987a; Angluin, 1987b]. Computing generalizations is a natural operation in query engineering that plays a crucial role in learning logical formulas [Plotkin, 1970; Muggleton, 1991], in particular in exact learning [ten Cate and Dalmau, 2021]. Exact learning, in turn, is concerned with constructing queries and ontologies. This can be challenging and costly, especially when logic expertise and domain knowledge are not in the same hands. Aiming at such cases, exact learning provides a systematic protocol for query engineering in which a learner interacts in a game-like fashion with an oracle, which may be a domain expert.

Our results on LGGs concern the notion of a *frontier* of an ELIQ q w.r.t. an ontology $\mathcal{O}$. Such a frontier is a set $\mathcal{F}$ of ELIQs that *generalize* q, that is, $q \subseteq_\mathcal{O} q_F$ and $q_F \not\subseteq_\mathcal{O} q$ for all $q_F \in \mathcal{F}$, where '$\subseteq_\mathcal{O}$' denotes query containment w.r.t. $\mathcal{O}$. Moreover, $\mathcal{F}$ must be *complete* in that for all ELIQs q' with $q \subseteq_\mathcal{O} q'$ and $q' \not\subseteq_\mathcal{O} q$, there is a $q_F \in \mathcal{F}$ such that $q_F \subseteq_\mathcal{O} q'$. We are interested in computing a frontier that contains only polynomially many ELIQs of polynomial size, in polynomial time. This is possible in the case of ELIQs without ontologies as shown in [ten Cate and Dalmau, 2021]; for the simpler $\mathcal{EL}$ queries, the same had been observed earlier (also without ontologies) in [Baader *et al.*, 2018; Kriegel, 2019]. In contrast, unrestricted CQs do not even admit finite frontiers [Nesetril and Tardif, 2000].

In exact learning, the learner and the oracle know and agree on the ontology $\mathcal{O}$, and they also agree on the target query q_T to use only concept and role names from $\mathcal{O}$. The learner may ask *membership queries* where they produce an ABox $\mathcal{A}$ and a candidate answer a and ask whether $\mathcal{A}, \mathcal{O} \models q_T(a)$, that is, whether a is an answer to q_T w.r.t. $\mathcal{O}$ on $\mathcal{A}$. The oracle faithfully answers "yes" or "no". *Polynomial time learnability* then means that the learner has an algorithm for constructing q_T, up to equivalence w.r.t. $\mathcal{O}$, with running time bounded by a polynomial in the sizes of q_T and $\mathcal{O}$.

Learning with only membership queries, as described above and studied in this article, is a strong form of exact learning. In fact, there are not many cases where polynomial time learning with only membership queries is possible, ELIQs without ontologies being an important example [ten Cate and Dalmau, 2021]. Often, one would therefore also admit *equivalence queries* where the learner provides a hypothesis ELIQ q_H and asks whether q_H is equivalent to q_T under $\mathcal{O}$; the oracle answers "yes" or provides a counterexample, that is, an ABox $\mathcal{A}$ and answer a such that $\mathcal{A}, \mathcal{O} \models q_T(a)$ and $\mathcal{A}, \mathcal{O} \not\models q_H(a)$ or vice versa. This is done, for instance, in [Konev *et al.*, 2018; Funk *et al.*, 2021].

We consider as ontology languages the DLs *DL-Lite*$^\mathcal{H}$ and

DL-Lite$^{\mathcal{F}}$, equipped with *role inclusions* (also known as role hierarchies) and *functional roles*, respectively. Both dialects admit concept and role disjointness constraints and $\mathcal{ELI}$ concepts on the right-hand side of concept inclusions [Calvanese *et al.*, 2007; Kikot *et al.*, 2011]. We show that *DL-Lite*$^{\mathcal{H}}$ admits polynomial frontiers that can be computed in polynomial time, and that *DL-Lite*$^{\mathcal{F}}$ does not even admit finite frontiers. We then introduce a fragment *DL-Lite*$^{\mathcal{F}-}$ of *DL-Lite*$^{\mathcal{F}}$ that restricts the use of inverse functional roles on the right-hand side of concept inclusions and show that it is as well-behaved as *DL-Lite*$^{\mathcal{H}}$. Both frontier constructions require a rather subtle analysis. We also note that adding conjunction results in frontiers of exponential size, even for very simple fragments of *DL-Lite*. One application of our results is to show that every ELIQ q can be characterized up to equivalence w.r.t. ontologies formulated in *DL-Lite*$^{\mathcal{H}}$ or *DL-Lite*$^{\mathcal{F}-}$ by only polynomially many data examples of the form $(\mathcal{A}, a)$, labeled as positive if $\mathcal{A}, \mathcal{O} \models q(a)$ and as negative otherwise.

We then consider in detail the application of our results in the context of exact learning and show that ELIQs can be learned in polynomial time w.r.t. ontologies $\mathcal{O}$ formulated in *DL-Lite*$^{\mathcal{H}}$ or *DL-Lite*$^{\mathcal{F}-}$. The learning algorithm uses only membership queries provided that a seed query is available, that is, an ELIQ q_0 such that $q_0 \subseteq_{\mathcal{O}} q_T$. Such a seed query can be constructed using membership queries if $\mathcal{O}$ contains no concept disjointness constraints and obtained by a single initial equivalence query otherwise. We also show that ELIQs cannot be learned at all w.r.t. unrestricted *DL-Lite*$^{\mathcal{F}}$ ontologies using only membership queries, and that they cannot be learned with only polynomially many membership queries when conjunction is admitted.

Proof details are given in the long version [Funk *et al.*, 2022].

Related Work. Exact learning of queries in the context of description logics has been studied in [Funk *et al.*, 2021] while [Konev *et al.*, 2018] considers learning entire ontologies, see also [Ozaki *et al.*, 2020; Ozaki, 2020]. It is shown in [Funk *et al.*, 2021] that a restricted form of CQs (that do not encompass all ELIQs) can be learned in polynomial time under $\mathcal{EL}$ ontologies using both membership and equivalence queries. The results from that paper indicate that inverse roles provide a challenge for exact learning under ontologies and thus it is remarkable that we can handle them without any restrictions in our context. Related forms of learning are the construction of the least common subsumer (LCS) and the most specific concept (MSC) [Baader, 2003; Baader *et al.*, 1999; Baader *et al.*, 2007; Jung *et al.*, 2020b; Zarrieß and Turhan, 2013] which may both be viewed as a form of query generalization. There is also a more loosely related research thread on learning DL concepts from labeled data examples [Funk *et al.*, 2019; Jung *et al.*, 2020a; Lehmann and Hitzler, 2010; Lehmann and Völker, 2014; Sarker and Hitzler, 2019].

2 Preliminaries

Ontologies and ABoxes. Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be countably infinite sets of *concept*, *role*, and *individual names*. A *role* R is a role name $r \in \mathsf{N_R}$ or the inverse r^- of a role name r. An $\mathcal{ELI}$ *concept* is formed according to the syntax rule $C, D ::= \top \mid A \mid C \sqcap D \mid \exists R.C$ where A ranges over concept names and R over roles. A *basic concept* B is an $\mathcal{ELI}$ concept of the form $\top$, A, or $\exists R.\top$. When dealing with basic concepts, for brevity we may write $\exists R$ in place of $\exists R.\top$.

A *DL-Lite*$^{\mathcal{HF}}$ ontology $\mathcal{O}$ is a finite set of *concept inclusions (CIs)* $B \sqsubseteq C$, *role inclusions (RIs)* $R_1 \sqsubseteq R_2$, *concept disjointness constraints* $B_1 \sqcap B_2 \sqsubseteq \bot$, *role disjointness constraints* $R_1 \sqcap R_2 \sqsubseteq \bot$, and *functionality assertions* $\mathsf{func}(R)$. Here, B, B_1, and B_2 range over basic concepts, C over $\mathcal{ELI}$ concepts, and R_1, R_2, R over roles. Superscript $\cdot^{\mathcal{H}}$ indicates the presence of role inclusions (also called role hierarchies) and superscript $\cdot^{\mathcal{F}}$ indicates functionality assertions, and thus it should be clear what we mean with a *DL-Lite*$^{\mathcal{H}}$ ontology and with a *DL-Lite*$^{\mathcal{F}}$ ontology. In fact, we are mainly interested in these two fragments of *DL-Lite*$^{\mathcal{HF}}$.

A *DL-Lite*$^{\mathcal{HF}}$ ontology is in *normal form* if all concept inclusions in it are of one of the forms $A \sqsubseteq B$, $B \sqsubseteq A$, and $A \sqsubseteq \exists R.A'$ with A, A' concept names or $\top$ and B a basic concept. Note that CIs of the form $\exists R \sqsubseteq \exists S$ are not admitted and neither are CIs of the form $A \sqsubseteq \exists R.C$ with C a compound concept. An ABox $\mathcal{A}$ is a finite set of concept assertions $A(a)$ and role assertions $r(a, b)$ with A a concept name or $\top$, r a role name, and a, b individual names. We use $\mathsf{ind}(\mathcal{A})$ to denote the set of individual names used in $\mathcal{A}$.

As usual, the semantics is given in terms of *interpretations* $\mathcal{I}$, which we define to be a (possibly infinite and) non-empty set of concept and role assertions. We use $\Delta^{\mathcal{I}}$ to denote the set of individual names in $\mathcal{I}$, define $A^{\mathcal{I}} = \{a \mid A(a) \in \mathcal{I}\}$ for all $A \in \mathsf{N_C}$, and $r^{\mathcal{I}} = \{(a, b) \mid r(a, b) \in \mathcal{I}\}$ and $(r^-)^{\mathcal{I}} = \{(b, a) \mid r(a, b) \in \mathcal{I}\}$ for all $r \in \mathsf{N_R}$. This definition of interpretation is slightly different from the usual one, but equivalent;[1] its virtue is uniformity as every ABox is a finite interpretation. The interpretation function $\cdot^{\mathcal{I}}$ can be extended from concept names to $\mathcal{ELI}$ concepts in the standard way [Baader *et al.*, 2017]. An interpretation $\mathcal{I}$ *satisfies* a concept or role inclusion $\alpha_1 \sqsubseteq \alpha_2$ if $\alpha_1^{\mathcal{I}} \subseteq \alpha_2^{\mathcal{I}}$, a concept or role disjointness constraint $\alpha_1 \sqcap \alpha_2 \sqsubseteq \bot$ if $\alpha_1^{\mathcal{I}} \cap \alpha_2^{\mathcal{I}} = \emptyset$, and a functionality assertion $\mathsf{func}(R)$ if $R^{\mathcal{I}}$ is a partial function. It *satisfies* a concept or role assertion α if $\alpha \in \mathcal{I}$. Note that, as usual, we thus make the standard names assumption, implying the unique name assumption.

An interpretation is a *model* of an ontology or an ABox if it satisfies all inclusions, disjointness constraints, and assertions in it. We write $\mathcal{O} \models \alpha_1 \sqsubseteq \alpha_2$ if every model of the ontology $\mathcal{O}$ satisfies the concept or role inclusion $\alpha_1 \sqsubseteq \alpha_2$ and $\mathcal{O} \models \alpha_1 \equiv \alpha_2$ if $\mathcal{O} \models \alpha_1 \sqsubseteq \alpha_2$ and $\mathcal{O} \models \alpha_2 \sqsubseteq \alpha_1$. If α_1 and α_2 are basic concepts or roles, then such consequences are decidable in PTIME both in *DL-Lite*$^{\mathcal{H}}$ and in *DL-Lite*$^{\mathcal{F}}$ [Artale *et al.*, 2009]. An ABox $\mathcal{A}$ is *satisfiable* w.r.t. an ontology $\mathcal{O}$ if $\mathcal{A}$ and $\mathcal{O}$ have a common model. Deciding ABox satisfiability is also in PTIME in both *DL-Lite*$^{\mathcal{H}}$ and *DL-Lite*$^{\mathcal{F}}$.

A *signature* is a set of concept and role names, uniformly referred to as symbols. For any syntactic object O such as an ontology or an ABox, we use $\mathsf{sig}(O)$ to denote the symbols

[1]This depends on admitting assertions $\top(a)$ in ABoxes.

used in O and $||O||$ to denote the *size* of O, that is, the length of a representation of O as a word in a suitable alphabet.

Queries. An $\mathcal{ELI}$ concept C can be viewed as an $\mathcal{ELI}$ *query (ELIQ)*. An individual $a \in \mathsf{ind}(\mathcal{A})$ is an *answer* to C on an ABox $\mathcal{A}$ w.r.t. an ontology $\mathcal{O}$, written $\mathcal{A}, \mathcal{O} \models C(a)$, if $a \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{O}$. We shall often view ELIQs as unary *conjunctive queries (CQs)* and also consider CQs that are not ELIQs. In this paper, CQs are always unary. A CQ thus takes the form $q(x_0) = \exists \bar{y}\, \phi(x_0, \bar{y})$ with ϕ a conjunction of *concept atoms* $A(x)$ and *role atoms* $r(x, y)$ where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. We use $\mathsf{var}(q)$ to denote the set of variables that occur in q. We may view q as a set of atoms and may write $r^-(x, y)$ in place of $r(y, x)$. We call x_0 the *answer variable* and use the notion of an answer and the notation $\mathcal{A}, \mathcal{O} \models q(a)$ also for CQs. The formal definition is in terms of homomorphisms as usual, details are in the long version. ELIQs are in 1-to-1 correspondence with CQs whose Gaifman graph is a tree and that contain no self-loops and multi-edges. For example, the ELIQ $C = A \sqcap \exists r^-.(\exists s.B \sqcap \exists r.A)$ is the CQ $q(x_0) = \{A(x), r(y, x), s(y, z), B(z), r(y, z'), A(z')\}$. We use $\mathcal{A}_q$ to denote the ABox obtained from CQ q by viewing variables as individuals and atoms as assertions. A CQ q is *satisfiable* w.r.t. ontology $\mathcal{O}$ if $\mathcal{A}_q$ is.

For CQs q_1 and q_2 and an ontology $\mathcal{O}$, we say that q_1 is *contained in* q_2 w.r.t. $\mathcal{O}$, written $q_1 \subseteq_{\mathcal{O}} q_2$ if for all ABoxes $\mathcal{A}$ and $a \in \mathsf{ind}(\mathcal{A})$, $\mathcal{A}, \mathcal{O} \models q_1(a)$ implies $\mathcal{A}, \mathcal{O} \models q_2(a)$. If q_1, q_2 are ELIQs, then this coincides with q_1 viewed as $\mathcal{ELI}$ concept being subsumed w.r.t. $\mathcal{O}$ by q_2 viewed as an $\mathcal{ELI}$ concept [Baader *et al.*, 2017]. We call q_1 and q_2 equivalent w.r.t. $\mathcal{O}$, written $q_1 \equiv_{\mathcal{O}} q_2$, if $q_1 \subseteq_{\mathcal{O}} q_2$ and $q_2 \subseteq_{\mathcal{O}} q_1$.

$\mathcal{O}$-saturatedness and $\mathcal{O}$-minimality. A CQ q is *$\mathcal{O}$-saturated*, with $\mathcal{O}$ an ontology, if $\mathcal{A}_q, \mathcal{O} \models A(y)$ implies $A(y) \in q$ for all $y \in \mathsf{var}(q)$ and $A \in \mathsf{N_C}$. It is *$\mathcal{O}$-minimal* if there is no $x \in \mathsf{var}(q)$ such that $q \equiv_{\mathcal{O}} q|_{\mathsf{var}(q)\setminus\{x\}}$ with $q|_S$ the restriction of q to the atoms that only contain variables in S. For a CQ q and an ontology $\mathcal{O}$ formulated in *DL-Lite*$^{\mathcal{H}}$ or *DL-Lite*$^{\mathcal{F}}$, one can easily find in polynomial time an $\mathcal{O}$-saturated CQ q' with $q \equiv_{\mathcal{O}} q'$. To achieve $\mathcal{O}$-minimality, we may repeatedly choose variables $x \in \mathsf{var}(q)$, check whether $\mathcal{A}_{q|_{\mathsf{var}(q)\setminus\{x\}}}, \mathcal{O} \models q$, and if so replace q with $q|_{\mathsf{var}(q)\setminus\{x\}}$. For ELIQs, the required checks can be carried out in PTIME in *DL-Lite*$^{\mathcal{F}}$ [Bienvenu *et al.*, 2013a], but are NP-complete in *DL-Lite*$^{\mathcal{H}}$ [Kikot *et al.*, 2011]. We conjecture that in *DL-Lite*$^{\mathcal{H}}$, it is not possible to construct equivalent $\mathcal{O}$-minimal ELIQs in polynomial time.

3 Frontiers in *DL-Lite*$^{\mathcal{H}}$

We show that for every ELIQ q and *DL-Lite*$^{\mathcal{H}}$ ontology $\mathcal{O}$ such that q is satisfiable w.r.t. $\mathcal{O}$, there is a frontier of polynomial size that can be computed in polynomial time. We also observe that this fails when *DL-Lite*$^{\mathcal{H}}$ is extended with conjunction, even in very restricted cases.

Definition 1. *A* frontier *of an ELIQ q w.r.t. a DL-Lite$^{\mathcal{HF}}$ ontology $\mathcal{O}$ is a set of ELIQs $\mathcal{F}$ such that*

1. $q \subseteq_{\mathcal{O}} q_F$ *for all* $q_F \in \mathcal{F}$;
2. $q_F \not\subseteq_{\mathcal{O}} q$ *for all* $q_F \in \mathcal{F}$;
3. *for all ELIQs q' with $q \subseteq_{\mathcal{O}} q' \not\subseteq_{\mathcal{O}} q$, there is a $q_F \in \mathcal{F}$ with $q_F \subseteq_{\mathcal{O}} q'$.*

It is not hard to see that finite frontiers that are minimal w.r.t. set inclusion are unique up to equivalence of the ELIQs in them, that is, if $\mathcal{F}_1$ and $\mathcal{F}_2$ are finite minimal frontiers of q w.r.t. $\mathcal{O}$, then for every $q_F \in \mathcal{F}_1$ there is a $q'_F \in \mathcal{F}_2$ such that $q_F \equiv_{\mathcal{O}} q'_F$ and vice versa. The following is the main result of this section.

Theorem 1. *Let $\mathcal{O}$ be a DL-Lite$^{\mathcal{H}}$ ontology and q an ELIQ that is $\mathcal{O}$-minimal and satisfiable w.r.t. $\mathcal{O}$. Then a frontier of q w.r.t. $\mathcal{O}$ can be computed in polynomial time.*

We note that Theorem 1 still holds when $\mathcal{O}$-minimality is dropped as a precondition and Condition 2 of frontiers is dropped as well. For proving Theorem 1, we first observe that we can concentrate on ontologies that are in normal form.

Lemma 1. *For every DL-Lite$^{\mathcal{H}}$ ontology $\mathcal{O}$, we can construct in polynomial time a DL-Lite$^{\mathcal{H}}$ ontology $\mathcal{O}'$ in normal form such that every $\mathcal{O}$-minimal ELIQ q is also $\mathcal{O}'$-minimal and a frontier of q w.r.t. $\mathcal{O}$ can be constructed in polynomial time given a frontier of q w.r.t. $\mathcal{O}'$.*

We now prove Theorem 1, adapting and generalizing a technique from [ten Cate and Dalmau, 2021]. Let $\mathcal{O}$ and $q(x_0)$ be as in the formulation of the theorem, with $\mathcal{O}$ in normal form. We may assume w.l.o.g. that q is $\mathcal{O}$-saturated. To construct a frontier of q w.r.t. $\mathcal{O}$, we consider all ways to generalize q in a least general way where 'generalizing' means to construct from q an ELIQ q' such that $q \subseteq_{\mathcal{O}} q'$ and $q' \not\subseteq_{\mathcal{O}} q$ and 'least general way' that there is no ELIQ $\widehat{q}$ that generalizes q and satisfies $\widehat{q} \subseteq_{\mathcal{O}} q'$ and $q' \not\subseteq_{\mathcal{O}} \widehat{q}$. We do this in two steps: the actual generalization plus a compensation step, the latter being needed to guarantee that we indeed arrive at a least general generalization.

For $x \in \mathsf{var}(q)$, we use q_x to denote the ELIQ obtained from q by taking the subtree of q rooted at x and making x the answer variable. The construction that follows involves the introduction of fresh variables x, some of which are a 'copy' of a variable from $\mathsf{var}(q)$. We then use $x^{\downarrow}$ to denote that original variable.

Step 1: Generalize. For each variable $x \in \mathsf{var}(q)$, define a set $\mathcal{F}_0(x)$ that contains all ELIQs which can be obtained by starting with $q_x(x)$ and then doing one of the following:

(A) Drop concept atom:

1. choose an atom $A(x) \in q$ such that
   (a) there is no $B(x) \in q$ with $\mathcal{O} \models B \sqsubseteq A$ and $\mathcal{O} \not\models A \sqsubseteq B$ and
   (b) there is no $R(x, y) \in q$ with $\mathcal{O} \models \exists R \sqsubseteq A$;
2. remove all $B(x) \in q$ with $\mathcal{O} \models A \equiv B$, including $A(x)$.

(B) Generalize subquery:

1. choose an atom $R(x, y) \in q$ directed away from x_0;
2. remove $R(x, y)$ and all atoms of q_y;
3. for each $q'(y) \in \mathcal{F}_0(y)$, add a disjoint copy $\widehat{q}'$ of q' and the role atom $R(x, y'')$ with y'' the copy of y in $\widehat{q}'$;

4. for every role S with $\mathcal{O} \models R \sqsubseteq S$ and $\mathcal{O} \not\models S \sqsubseteq R$, add a disjoint copy $\widehat{q}_y$ of q_y and the role atom $S(x, y')$ with y' the copy of y in $\widehat{q}_y$.

The definition of $x^{\downarrow}$ should be clear in all cases. In Point 3 of Case (B), for example, for every variable z in q' that was renamed to z' in $\widehat{q}'$ set $z'^{\downarrow} = z^{\downarrow}$. Note that $z^{\downarrow}$ is defined for all variables z that occur in queries in $\mathcal{F}_0(x)$. Also note that, in Point 1b of (A), it is important to use q rather than q_x as y could be a predecessor of x in q.

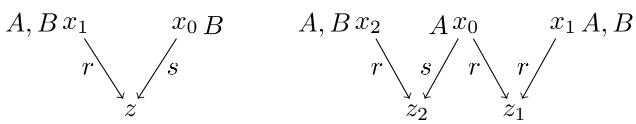
Step 2: Compensate. We construct a frontier $\mathcal{F}$ of $q(x_0)$ by including, for each $p \in \mathcal{F}_0(x_0)$, the ELIQ obtained from p by the following two steps. We write $x \rightsquigarrow^{R}_{q,\mathcal{O}} A$ if $\mathcal{A}_q, \mathcal{O} \models \exists R.A(x)$ and there is no $S(x, y) \in q$ with $\mathcal{O} \models S \sqsubseteq R$ and $\mathcal{A}_q, \mathcal{O} \models A(y)$.

Step 2A. Consider all $x \in \mathsf{var}(p)$, roles R, S, and concept names A such that $x^{\downarrow} \rightsquigarrow^{R}_{q,\mathcal{O}} A$, $\mathcal{O} \models R \sqsubseteq S$, and $\mathcal{O} \models \exists S \sqsubseteq B$ implies $B(x) \in p$ for all concept names B. Add the atoms $S(x, z), A(z), R(x', z)$ where z and x' are fresh variables with $z^{\downarrow}$ undefined, $x'^{\downarrow} = x^{\downarrow}$, and add a disjoint copy $\widehat{q}$ of q, glue the copy of $x^{\downarrow}$ in $\widehat{q}$ to x'.

Step 2B. Consider every $S(x, y) \in p$ directed away from x_0 that was not added in Step 2A. Then $x^{\downarrow}$ and $y^{\downarrow}$ are defined. For every role R with $\mathcal{A}_q, \mathcal{O} \models R(x^{\downarrow}, y^{\downarrow})$, add an atom $R(z, y)$, z a fresh variable with $z^{\downarrow} = x^{\downarrow}$, as well as a disjoint copy $\widehat{q}$ of q and glue the copy of $x^{\downarrow}$ in $\widehat{q}$ to z.

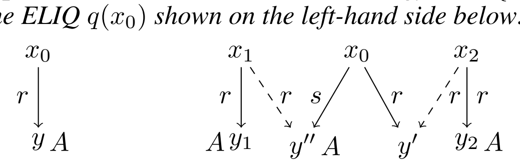
This finishes the construction of the frontier $\mathcal{F}$ of q.

Example 1. *Consider the DL-Lite$^{\mathcal{H}}$ ontology $\mathcal{O} = \{A \sqsubseteq \exists r,\ \exists r \sqsubseteq A,\ r \sqsubseteq s\}$ and the ELIQ $q(x_0) = A(x_0) \wedge B(x_0)$. Then $\mathcal{F}$ contains the ELIQs p_1 and p_2 shown below:*



ELIQ p_1 is the result of dropping the concept atom $A(x_0)$ and p_2 is the result of dropping the concept atom $B(x_0)$. Step 2A adds an r-successor and an s-successor of x_0 in p_2 but only an s-successor in p_1 as $\mathcal{O} \models \exists r \sqsubseteq A$, and then attaches copies of q. Step 2B does nothing, as all role atoms have been added in Step 2A.[2]

Example 2. *Consider the DL-Lite$^{\mathcal{H}}$ ontology $\mathcal{O} = \{r \sqsubseteq s\}$ and the ELIQ $q(x_0)$ shown on the left-hand side below:*



Then $\mathcal{F}$ contains only the ELIQ p shown on the right-hand side. It is the result of dropping the concept atom $A(y)$ in q_y, then generalizing the subquery $r(x_0, y)$ in $q_{x_0} = q$, and then compensating. Step 2A of compensation adds nothing. Step 2B adds the two dashed role atoms and attaches copies of q to x_1 and x_2.

[2]Variables x_2 and z_2 can be dropped from p_2 resulting in an ELIQ that is equivalent w.r.t. $\mathcal{O}$. We did not include such optimizations in the compensation step to avoid making it more complicated.

Lemma 2. *$\mathcal{F}$ is a frontier of $q(x_0)$ w.r.t. $\mathcal{O}$.*

We next show that the constructed frontier is of polynomial size and that its computation takes only polynomial time.

Lemma 3. *The construction of $\mathcal{F}$ runs in time polynomial in $||q|| + ||\mathcal{O}||$ (and thus $\sum_{p \in \mathcal{F}} ||p||$ is polynomial in $||q|| + ||\mathcal{O}||$).*

We next observe that adding conjunction to *DL-Lite* destroys polynomial frontiers and thus Theorem 1 does not apply to *DL-Lite*$_{\mathsf{horn}}$ ontologies [Artale *et al.*, 2009]. In fact, this already holds for very simple queries and ontologies, implying that also for other DLs that support conjunction such as $\mathcal{EL}$, polynomial frontiers are elusive. A *conjunction of atomic queries (AQ$^\wedge$)* is a unary CQ of the form $q(x_0) = A_1(x_0) \wedge \cdots \wedge A_n(x_0)$ and a *conjunctive ontology* is a set of CIs of the form $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A$ where $A_1, \ldots, A_n$ and A are concept names.

Theorem 2. *There are families of AQ$^\wedge$s $q_1, q_2, \ldots$ and conjunctive ontologies $\mathcal{O}_1, \mathcal{O}_2, \ldots$ such that for all $n \geq 1$, any frontier of q_n w.r.t. $\mathcal{O}_n$ has size at least 2^n.*

4 Frontiers in *DL-Lite*$^{\mathcal{F}}$

We start by observing that frontiers of ELIQs w.r.t. *DL-Lite*$^{\mathcal{F}}$ ontologies may be infinite. This leads us to identifying a syntactic restriction on *DL-Lite*$^{\mathcal{F}}$ ontologies that regains finite frontiers. In fact, we show that they are of polynomial size and can be computed in polynomial time.

Theorem 3. *There is an ELIQ q and a DL-Lite$^{\mathcal{F}}$ ontology $\mathcal{O}$ such that q does not have a finite frontier w.r.t. $\mathcal{O}$.*

In the proof of Theorem 3, we use the ELIQ $A(x)$ and

$$\mathcal{O} = \{\ A \sqsubseteq \exists r, \quad \exists r^- \sqsubseteq \exists r, \quad \exists r \sqsubseteq \exists s, \quad \mathsf{func}(r^-)\ \}.$$

The universal model $\mathcal{U}_{q,\mathcal{O}}$ of $\mathcal{A}_q$ and $\mathcal{O}$ is an infinite r-path on which every point has an s-successor. Now consider the following ELIQs $q_1, q_2, \ldots$ that satisfy $q_i \not\subseteq_{\mathcal{O}} q \subseteq_{\mathcal{O}} q_i$:

$$\begin{aligned} q_i(x_1) = {} & r(x_1, x_2), \ldots, r(x_{n-1}, x_n), s(x_n, y), s(x'_n, y), \\ & r(x'_1, x'_2), \ldots, r(x'_{n-1}, x'_n), A(x'_1). \end{aligned}$$

Any frontier $\mathcal{F}$ must contain a p_i with $p_i \subseteq_{\mathcal{O}} q_i$ for all $i \geq 1$. We show that, consequently, there is no bound on the size of the queries in $\mathcal{F}$. We invite the reader to apply the frontier construction from Section 3 after dropping $\mathsf{func}(r^-)$.

The proof actually shows that there is no finite frontier even if we admit the use of unrestricted CQs in the frontier in place of ELIQs. To regain finite frontiers, we restrict our attention to *DL-Lite*$^{\mathcal{F}}$ ontologies $\mathcal{O}$ such that if $B \sqsubseteq C$ is a CI in $\mathcal{O}$, then C contains no subconcept of the form $\exists R.D$ with $\mathsf{func}(R^-) \in \mathcal{O}$. We call such an ontology a *DL-Lite*$^{\mathcal{F}-}$ ontology. We again concentrate on ontologies in normal form.

Lemma 4. *For every DL-Lite$^{\mathcal{F}-}$ ontology $\mathcal{O}$, we can construct in polynomial time a DL-Lite$^{\mathcal{F}-}$ ontology $\mathcal{O}'$ in normal form such that for every ELIQ q, a frontier of q w.r.t. $\mathcal{O}$ can be constructed in polynomial time given a frontier of q w.r.t. $\mathcal{O}'$.*

The main result of this section is as follows.

Theorem 4. *Let $\mathcal{O}$ be a DL-Lite$^{\mathcal{F}-}$ ontology and q an ELIQ that is satisfiable w.r.t. $\mathcal{O}$. Then a frontier of q w.r.t. $\mathcal{O}$ can be computed in polynomial time.*

To prove Theorem 4, let $\mathcal{O}$ and q be as in the theorem, $\mathcal{O}$ in normal form. We may assume w.l.o.g. that q is $\mathcal{O}$-minimal and $\mathcal{O}$-saturated. The construction of a frontier follows the same general approach as for *DL-Lite*$^{\mathcal{H}}$, but the presence of functional roles significantly complicates the compensation step. As before, we introduce fresh variables and rely on the mapping $x^{\downarrow}$.

Step 1: Generalize. For each variable $x \in \mathsf{var}(q)$, define a set $\mathcal{F}_0(x)$ that contains all ELIQs which can be obtained by starting with $q_x(x)$ and then doing one of the following:

(A) Drop concept atom: exactly as for *DL-Lite*$^{\mathcal{H}}$.

(B) Generalize subquery:

1. choose an atom $R(x, y) \in q$ directed away from x_0;
2. remove $R(x, y)$ and all atoms of q_y;
3. if $\mathsf{func}(R) \notin \mathcal{O}$, then for each $q'(y) \in \mathcal{F}_0(y)$ add a disjoint copy $\widehat{q}'$ of q' and the role atom $R(x, y')$ with y' the copy of y in $\widehat{q}'$;
4. if $\mathsf{func}(R) \in \mathcal{O}$ and $\mathcal{F}_0(y) \neq \emptyset$, then choose and add a $q' \in \mathcal{F}_0(y)$ and the role atom $R(x, y)$.

Step 2: Compensate. We construct a frontier $\mathcal{F}$ of $q(x_0)$ by including, for each $p \in \mathcal{F}_0(x_0)$, the CQ obtained from p by the following two steps. For $x \in \mathsf{var}(q)$, R a role, and M a set of concept names from $\mathcal{O}$, we write $x \rightsquigarrow^{R}_{q,\mathcal{O}} M$ if M is maximal with $\mathcal{A}_q, \mathcal{O} \models \exists R. \bigsqcap M(x)$ and there is no $R(x, y) \in q$ with $\mathcal{A}_q, \mathcal{O} \models \bigsqcap M(y)$.

Step 2A. Consider every $x \in \mathsf{var}(p)$, role R, and set of concept names $M = \{A_1, \dots, A_k\}$ with $x^{\downarrow} \rightsquigarrow^{R}_{q,\mathcal{O}} M$. If $\mathcal{O} \models \exists R \sqsubseteq B$ implies $B(x) \in p$ for all concept names B, add the atoms $R(x, z), A_1(z), \dots, A_k(z)$ where z is a fresh variable, and leave $z^{\downarrow}$ undefined.

Step 2B. This step is iterative. For bookkeeping, we mark atoms $R(x, y) \in p$ to be processed in the next round of the iteration. Marking is only applied to atoms $R(x, y)$ directed away from x_0 such that $y^{\downarrow}$ is defined and if $x^{\downarrow}$ is undefined then $\mathsf{func}(R^-) \notin \mathcal{O}$ or q contains no atom of the form $R(y^{\downarrow}, z)$.

To start, consider every $R(x, y) \in p$ directed away from x_0 with $\mathsf{func}(R^-) \notin \mathcal{O}$. Then $x^{\downarrow}$ is defined. Extend p with atom $R^-(y, x')$, x' a fresh variable with $x'^{\downarrow} = x^{\downarrow}$. Mark the new atom.

Then repeatedly choose a marked atom $R(x, y)$ and unmark it. If $\mathsf{func}(R^-) \notin \mathcal{O}$ or q contains no atom of the form $R(y^{\downarrow}, z)$, then add a disjoint copy $\widehat{q}$ of q and glue the copy of $y^{\downarrow}$ in $\widehat{q}$ to y. Otherwise, do the following:

(i) add $A(y)$ whenever $\mathcal{A}_q, \mathcal{O} \models A(y^{\downarrow})$;

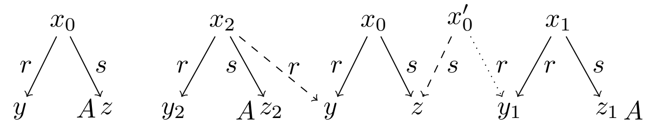
(ii) for all atoms $S(y^{\downarrow}, z) \in q$ with $S(y^{\downarrow}, z) \neq R^-(y^{\downarrow}, x^{\downarrow})$, extend p with atom $S(y, z')$, z' a fresh variable with $z'^{\downarrow} = z$. Mark $S(y, z')$.

(iii) For all roles S and sets $M = \{A_1, \dots, A_k\}$ such that $y^{\downarrow} \rightsquigarrow^{S}_{q,\mathcal{O}} M$, extend p with atoms $S(y, u)$, $S^-(u, y'), A_1(u), \dots, A_k(u)$ where u and y' are fresh variables. Set $y'^{\downarrow} = y^{\downarrow}$ and mark $S^-(u, y')$.

The step is repeated as long as possible. Note that in Point (iii), the role S must occur on the right-hand side of some CI in the *DL-Lite*$^{\mathcal{F}-}$ ontology $\mathcal{O}$. Consequently, $\mathsf{func}(S^-) \notin \mathcal{O}$ and it is not a problem that u receives two S-predecessors. Also in Point (iii), $\mathsf{func}(S) \in \mathcal{O}$ implies that q cannot contain an atom $S(y^{\downarrow}, z)$ due to the definition of '$\rightsquigarrow$' and thus we may leave $u^{\downarrow}$ undefined.

This finishes the construction of the frontier $\mathcal{F}$ of q.

Example 3. *Consider the ontology $\mathcal{O} = \{\mathsf{func}(s)\}$ and ELIQ $q(x_0)$ shown on the left-hand side below:*



The ELIQ $p \in \mathcal{F}$ shown on the right-hand side is the result of dropping the concept atom $A(z)$ in q_z, then generalizing the subquery $s(x_0, z)$ in $q_{x_0} = q$, and then compensating. Step 2A of compensation adds nothing. The start of Step 2B adds the two dashed role atoms and marks them. The step of Step 2B adds the dotted role atom via Point (ii) and marks it. When the step of Step 2B processes role atoms $r^-(y, x_2)$ and $r(x'_0, y)$, it attaches copies of q to x_2 and y_1. Note that directly attaching a copy of q to x'_0 would violate $\mathsf{func}(s)$.

Lemma 5. *$\mathcal{F}$ is a frontier of $q(x_0)$ w.r.t. $\mathcal{O}$.*

As for *DL-Lite*$^{\mathcal{H}}$, the constructed frontier is of polynomial size and its computation takes only polynomial time. Crucially, the iterative process in Point 2B terminates since in Step (ii) a (copy of a) subquery of q is added and the process stops at atoms added in Step (iii).

Lemma 6. *The construction of $\mathcal{F}$ runs in time polynomial in $||q|| + ||\mathcal{O}||$ (and thus $\sum_{p \in \mathcal{F}} ||p||$ is polynomial in $||q|| + ||\mathcal{O}||$).*

One first application of our results on frontiers is to the unique characterization of ELIQs w.r.t. ontologies by labeled data examples. Details are in the long version.

5 Learning ELIQs under Ontologies

We use our results on frontiers to show that ELIQs are polynomial time learnable under ontologies formulated in *DL-Lite*$^{\mathcal{H}}$ and *DL-Lite*$^{\mathcal{F}-}$, using only membership queries. We also present two results on non-learnability.

Theorem 5. *ELIQs are polynomial time learnable under DL-Lite$^{\mathcal{H}}$ ontologies and under DL-Lite$^{\mathcal{F}-}$ ontologies using only membership queries.*

If the ontology contains concept disjointness constraints, then this only holds true if the learner is provided with a seed CQ (definition given below).

For proving Theorem 5, let $\mathcal{O}$ be an ontology formulated in *DL-Lite*$^{\mathcal{H}}$ or *DL-Lite*$^{\mathcal{F}-}$ and $q_T(x_0)$ the target ELIQ known to the oracle. We may again assume $\mathcal{O}$ to be in normal form.

Algorithm 1 Learning ELIQs under *DL-Lite* ontologies

Input An ontology $\mathcal{O}$ in normal form and a CQ q_H^0 satisfiable w.r.t. $\mathcal{O}$ such that $q_H^0 \subseteq_{\mathcal{O}} q_T$
Output An ELIQ q_H such that $q_H \equiv_{\mathcal{O}} q_T$

$q_H := \mathsf{treeify}(q_H^0)$
while there is a $q_F \in \mathcal{F}_{q_H}$ with $q_F \subseteq_{\mathcal{O}} q_T$ **do**
  $q_H := \mathsf{minimize}(q_F)$
end while
return q_H

Lemma 7. *In DL-Lite$^{\mathcal{H}}$ and DL-Lite$^{\mathcal{F}-}$, every polynomial time learning algorithm for ELIQs under ontologies in normal form that uses only membership queries can be transformed into a learning algorithm with the same properties for ELIQs under unrestricted ontologies.*

The learning algorithm is displayed as Algorithm 1. It assumes a *seed CQ* q_H^0, that is, a CQ q_H^0 such that $q_H^0 \subseteq_{\mathcal{O}} q_T$ and q_H^0 is satisfiable w.r.t. $\mathcal{O}$. If $\mathcal{O}$ contains no disjointness constraints, then for $\Sigma = \mathsf{sig}(\mathcal{O})$ we can use as the seed CQ

$$q_H^0(x_0) = \{A(x_0) \mid A \in \Sigma \cap \mathsf{N_C}\} \cup \{r(x_0, x_0) \mid r \in \Sigma \cap \mathsf{N_R}\}.$$

We can still construct a seed CQ q_H^0 in time polynomial in $||\mathcal{O}||$ if $\mathcal{O}$ contains no disjoint constraints on concepts (but potentially on roles); details at in the long version. In the presence of concept disjointness constraints, a seed CQ can be obtained through an initial equivalence query.

The algorithm constructs and repeatedly updates a hypothesis ELIQ q_H while maintaining the invariant $q_H \subseteq_{\mathcal{O}} q_T$. The initial call to subroutine $\mathsf{treeify}$ yields an ELIQ q_H with $q_H^0 \subseteq_{\mathcal{O}} q_H \subseteq_{\mathcal{O}} q_T$ to be used as the first hypothesis. The algorithm then iteratively generalizes q_H by constructing the frontier $\mathcal{F}_{q_H}$ of q_H w.r.t. $\mathcal{O}$ in polynomial time and choosing from it a new ELIQ q_H with $q_H \subseteq_{\mathcal{O}} q_T$. In between, the algorithm applies the $\mathsf{minimize}$ subroutine to ensure that the new q_H is $\mathcal{O}$-minimal and to avoid an excessive blowup while iterating in the while loop.

We next detail the subroutines $\mathsf{treeify}$ and $\mathsf{minimize}$. We define $\mathsf{minimize}$ on unrestricted CQs since it is applied to non-ELIQs as part of the $\mathsf{treeify}$ subroutine.

The $\mathsf{minimize}$ **subroutine.** The subroutine takes as input a unary CQ $q(x_0)$ that is satisfiable w.r.t. $\mathcal{O}$ and satisfies $q \subseteq_{\mathcal{O}} q_T$. It computes a unary CQ q' with $q \subseteq_{\mathcal{O}} q' \subseteq_{\mathcal{O}} q_T$ using membership queries that is minimal in a strong sense. Formally, $\mathsf{minimize}$ first makes sure that q is $\mathcal{O}$-saturated and then exhaustively applies the following operation:

Remove atom. Choose a role atom $r(x, y) \in q$ and let q^- be the maximal connected component of $q \setminus \{r(x, y)\}$ that contains x_0. Pose the membership query $\mathcal{A}_{q^-}, \mathcal{O} \models q_T(x_0)$. If the response is positive, continue with q^- in place of q.

Clearly, the result of $\mathsf{minimize}$ is $\mathcal{O}$-minimal.

The $\mathsf{treeify}$ **subroutine.** The subroutine takes as input a unary CQ $q(x_0)$ that is satisfiable w.r.t. $\mathcal{O}$, and satisfies $q \subseteq_{\mathcal{O}} q_T$. It computes an ELIQ q' with $q \subseteq_{\mathcal{O}} q' \subseteq_{\mathcal{O}} q_T$ by repeatedly increasing the length of cycles in q and minimizing the obtained query; a similar construction is used in [ten Cate and Dalmau, 2021]. The resulting ELIQ is $\mathcal{O}$-minimal.

Formally, $\mathsf{treeify}$ first makes sure that $q(x_0)$ is $\mathcal{O}$-saturated and then constructs a sequence of CQs $p_1, p_2, \ldots$ starting with $p_1 = \mathsf{minimize}(q)$ and then taking $p_{i+1} = \mathsf{minimize}(p_i')$ where p_i' is obtained from p_i by doubling the length of some cycle. Here, a *cycle* in a CQ q is a sequence $R_1(x_1, x_2), \ldots, R_n(x_n, x_1)$ of distinct role atoms in q such that $x_1, \ldots x_n$ are distinct. More precisely, p_i' is the result of the following operation.

Double cycle. Choose a role atom $r(x, y) \in p_i$ that is part of a cycle in p_i and let p be $p_i \setminus \{r(x, y)\}$. The CQ p_i' is then obtained by starting with p, adding a disjoint copy p' of p where x' refers to the copy of $x \in \mathsf{var}(p)$ in p' and adding the role atoms $r(x, y'), r(x', y)$.

If p_i contains no more cycles, $\mathsf{treeify}$ stops and returns p_i.

Returning to Algorithm 1, let $q_1, q_2, \ldots$ be the sequence of ELIQs that are assigned to q_H during a run of the learning algorithm. We show in the long version that for all $i \geq 1$, it holds that $q_i \subseteq_{\mathcal{O}} q_T$, $q_i \subseteq_{\mathcal{O}} q_{i+1}$ while $q_{i+1} \not\subseteq_{\mathcal{O}} q_i$, and $|\mathsf{var}(q_{i+1})| \geq |\mathsf{var}(q_i)|$. This can be used to prove that the while loop in Algorithm 1 terminates after a polynomial number of iterations, arriving at a hypothesis q_H with $q_H \equiv_{\mathcal{O}} q_T$.

We now turn to non-learnability results. Without a seed CQ, ontologies with concept disjointness constraints are not learnable using only polynomially many membership queries. A *disjointness ontology* is a *DL-Lite*$^{\mathcal{HF}}$ ontology that only consists of concept disjointness constraints.

Theorem 6. *AQ$^\wedge$s are not learnable under disjointness ontologies using only polynomially many membership queries.*

We next show that when we drop the syntactic restriction from *DL-Lite*$^{\mathcal{F}-}$, then ELIQs are no longer learnable at all using only membership queries. Note that this is not a direct consequence of Theorem 3 as there could be an alternative approach that does not use frontiers.

Theorem 7. *ELIQs are not learnable under DL-Lite$^{\mathcal{F}}$ ontologies using only membership queries.*

6 Outlook

A natural next step for future work is to generalize the results presented in this paper to *DL-Lite*$^{\mathcal{HF}}$, adopting the same syntactic restriction that we have adopted for *DL-Lite*$^{\mathcal{F}}$, and additionally requiring that functional roles have no proper subroles. The latter serves to control the interaction between functional roles and role inclusions. Even with this restriction, however, that interaction is very subtle and the frontier construction becomes significantly more complex. Other interesting questions are whether ELIQs can be learned in polynomial time w.r.t. *DL-Lite*$_{\mathsf{horn}}$ ontologies and whether CQs can be learned w.r.t. *DL-Lite*$_{\mathsf{core}}$ ontologies when both membership and equivalence queries are admitted.

Acknowledgements

Carsten Lutz was supported by DFG CRC 1320 EASE.

References

[Angluin, 1987a] Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.

[Angluin, 1987b] Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, 1987.

[Artale *et al.*, 2009] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *J. of Artifical Intelligence Research*, 36:1–69, 2009.

[Baader *et al.*, 1999] Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. of IJCAI*, pages 96–103. Morgan Kaufmann, 1999.

[Baader *et al.*, 2007] Franz Baader, Baris Sertkaya, and Anni-Yasmin Turhan. Computing the least common subsumer w.r.t. a background terminology. *J. Appl. Log.*, 5(3):392–420, 2007.

[Baader *et al.*, 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logics*. Cambride University Press, 2017.

[Baader *et al.*, 2018] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. Making repairs in description logics more gentle. In *Proc. of KR*, pages 319–328. AAAI Press, 2018.

[Baader, 2003] Franz Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. of IJCAI*, pages 319–324. Morgan Kaufmann, 2003.

[Bienvenu *et al.*, 2013a] Meghyn Bienvenu, Magdalena Ortiz, Mantas Simkus, and Guohui Xiao. Tractable queries for lightweight description logics. In *Proc. of IJCAI*, pages 768–774, 2013.

[Bienvenu *et al.*, 2013b] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: a study through disjunctive datalog, CSP, and MMSNP. In *Proc. of PODS*, pages 213–224. ACM, 2013.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2009] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, and Riccardo Rosati. Ontologies and databases: The DL-Lite approach. In *Reasoning Web*, volume 5689 of *LNCS*, pages 255–356, 2009.

[Funk *et al.*, 2019] Maurice Funk, Jean Christoph Jung, Carsten Lutz, Hadrien Pulcini, and Frank Wolter. Learning description logic concepts: When can positive and negative examples be separated? In *Proc. of IJCAI*, pages 1682–1688, 2019.

[Funk *et al.*, 2021] Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Actively learning concept and conjunctive queries under $\mathcal{EL}^r$-ontologies. In *Proc. of IJCAI*, pages 1887–1893, 2021.

[Funk *et al.*, 2022] Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Frontiers and exact learning of ELI queries under DL-Lite ontologies. *CoRR*, abs/2204.14172, 2022.

[Jung *et al.*, 2020a] Jean Christoph Jung, Carsten Lutz, Hadrien Pulcini, and Frank Wolter. Logical separability of incomplete data under ontologies. In *Proc. of KR*, pages 517–528, 2020.

[Jung *et al.*, 2020b] Jean Christoph Jung, Carsten Lutz, and Frank Wolter. Least general generalizations in description logic: Verification and existence. In *Proc. of AAAI*, pages 2854–2861, 2020.

[Kikot *et al.*, 2011] Stanislav Kikot, Roman Kontchakov, and Michael Zakharyaschev. On (in)tractability of OBDA with OWL 2 QL. In *Proc. of DL*, 2011.

[Konev *et al.*, 2018] Boris Konev, Carsten Lutz, Ana Ozaki, and Frank Wolter. Exact learning of lightweight description logic ontologies. *J. Mach. Learn. Res.*, 18(201):1–63, 2018.

[Kriegel, 2019] Francesco Kriegel. *Constructing and Extending Description Logic Ontologies using Methods of Formal Concept Analysis*. PhD thesis, TU Dresden, 2019.

[Lehmann and Hitzler, 2010] Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Mach. Learn.*, 78:203–250, 2010.

[Lehmann and Völker, 2014] Jens Lehmann and Johanna Völker. *Perspectives on Ontology Learning*, volume 18 of *Studies on the Semantic Web*. IOS Press, 2014.

[Muggleton, 1991] Stephen Muggleton. Inductive logic programming. *New Generation Comput.*, 8(4):295–318, 1991.

[Nesetril and Tardif, 2000] Jaroslav Nesetril and Claude Tardif. Duality theorems for finite structures (characterising gaps and good characterisations). *J. Comb. Theory, Ser. B*, 80(1):80–97, 2000.

[OWL Working Group, 2009] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 2009. Available at http://www.w3.org/TR/owl2-overview/.

[Ozaki *et al.*, 2020] Ana Ozaki, Cosimo Persia, and Andrea Mazzullo. Learning query inseparable $\mathcal{ELH}$ ontologies. In *Proc. of AAAI*, pages 2959–2966, 2020.

[Ozaki, 2020] Ana Ozaki. Learning description logic ontologies: Five approaches. where do they stand? *KI - Künstliche Intelligenz*, 2020.

[Plotkin, 1970] Gordon Plotkin. A note on inductive generalizations. *Edinburgh University Press*, 1970.

[Sarker and Hitzler, 2019] Md. Kamruzzaman Sarker and Pascal Hitzler. Efficient concept induction for description logics. In *Proc. of AAAI*, pages 3036–3043, 2019.

[ten Cate and Dalmau, 2021] Balder ten Cate and Victor Dalmau. Conjunctive queries: Unique characterizations and exact learnability. In *Proc. of ICDT*, volume 186 of *LIPIcs*, pages 9:1–9:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Zarrieß and Turhan, 2013] Benjamin Zarrieß and Anni-Yasmin Turhan. Most specific generalizations w.r.t. general $\mathcal{EL}$-TBoxes. In *Proc. of IJCAI*, pages 1191–1197, 2013.