Actively Learning $\mathcal{ELI}$ Queries under DL-Lite Ontologies

Maurice Funk[1], Jean Christoph Jung[2], and Carsten Lutz[1]

[1] Department of Computer Science, University of Bremen, Germany
[2] Department of Computer Science, University of Hildesheim, Germany

Abstract. We show that $\mathcal{ELI}$ queries (ELIQs) are learnable in polynomial time in the presence of a DL-Lite ontology $\mathcal{O}$, in Angluin's framework of active learning. When initially provided with a conjunctive query (CQ) that implies the target ELIQ under $\mathcal{O}$ (in the sense of query containment), it suffices for the learner to only pose membership queries to the oracle, but no equivalence queries. The initial CQ can be obtained by a single equivalence query and is available 'for free' in case that $\mathcal{O}$ does not pose any disjointness constraints on concepts. Our main technical result is that every $\mathcal{ELI}$ concept has only polynomially many most specific subsumers w.r.t. a DL-Lite ontology, generalizing a recent result about homomorphism frontiers by ten Cate and Dalmau.

1 Introduction

Constructing description logic (DL) concepts, ontologies, and queries can be challenging and costly, especially when logic expertise and domain knowledge are not in the same hands. This has prompted many approaches to *learning* such objects, including PAC learning [12,13,14], the construction of the least common subsumer (LCS) and the most specific concept (MSC) [4,6,7,19,28], and learning from data examples [16,18,22,23,27]. In recent years, there has been significant interest in applying *Angluin's framework of exact learning* in a DL context where a learner interacts in a game-like fashion with an oracle [1,2]. In particular, the learner may be a DL expert and the oracle a collaborating domain expert. The main aim is then to find an algorithm that enables the learner to construct the target object in polynomial time based on queries that it poses to the oracle, even when the oracle is not able to answer the queries in the most informative way.

The interest in exact learning in DLs started with an investigation of ontology learning in (the conference version of) [21], see also [20,26] and the survey [25]. This was recently complemented by studies of exactly learning DL concepts and queries: active learning of $\mathcal{ELI}$ concept queries (ELIQs) without ontologies is considered in [11] while [15] studies active learning of $\mathcal{EL}$ concept queries (ELQs), ELIQs, and restricted forms of conjunctive queries (CQs) in the presence of $\mathcal{EL}$

and $\mathcal{ELI}$ ontologies. The purpose of the current paper is to initiate the study of actively learning concepts and queries under ontologies formulated in DL-Lite, a prominent family of DLs that is featured in the OWL 2 family of ontology languages [3]. Our main result is that ELIQs, which can be viewed both as queries and as concepts to be used in an ontology, can be learned under DL-Lite ontologies in polynomial time even when the oracle can pose only a very basic kind of query to the oracle. To make this precise, we introduce the exact learning framework in more detail.

Learner and oracle both know and agree on the ontology $\mathcal{O}$ and the concept and role names that are available for constructing the target ELIQ q_T which must be satisfiable w.r.t. $\mathcal{O}$; we assume that this includes all concept and role names in $\mathcal{O}$. In a *membership query*, the learner provides an ABox $\mathcal{A}$ and a candidate answer $\bar{a}$ and asks whether $\mathcal{A}, \mathcal{O} \models q_T(\bar{a})$; the oracle faithfully answers "yes" or "no". In an *equivalence query*, 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 tuple $\bar{a}$ such that $\mathcal{A}, \mathcal{O} \models q_T(\bar{a})$ and $\mathcal{A}, \mathcal{O} \not\models q_H(\bar{a})$ (*positive counterexample*) or vice versa (*negative counterexample*). One is then interested in *polynomial time learnability*, that is, whether there is a learning algorithm that constructs $q_T(\bar{x})$, up to equivalence w.r.t. $\mathcal{O}$, such that at any given time, the running time of the algorithm is bounded by a polynomial in the sizes of q_T, of $\mathcal{O}$, and of the largest counterexample given by the oracle so far. A weaker requirement is *polynomial query learnability* where only the sum of the sizes of the queries posed to the oracle up to the current time point has to be bounded by such a polynomial.

We can now state our main result more precisely. With *DL-Lite*, we generally refer to the basic member of the DL-Lite family [10] that admits inclusions between basic concepts, concept disjointness constraints, and role disjointness constraints; *DL-Lite*$^-$ then means the fragment without concept disjointness. Our main result is that ELIQs are polynomial time learnable using only membership queries under *DL-Lite*$^-$ ontologies, and that the same is true for *DL-Lite* ontologies provided that we have available an initial CQ q_H^0 such that $q_H^0 \subseteq_{\mathcal{O}} q_T$, that is, the answers to q_H^0 w.r.t. $\mathcal{O}$ are a subset of those to q_T w.r.t. $\mathcal{O}$ on every ABox $\mathcal{A}$. Such a q_H^0 can be obtained by a single initial equivalence query. We also observe that polynomial learnability using only membership queries fails in the presence of concept disjointness.

Let us mention two interesting perspectives on our results. First, they generalize the results in [11] about polynomial time learnability of ELIQs to the case with *DL-Lite* ontologies, in fact borrowing and extending crucial techniques from [11]. And second, the results in [15] demonstrate that inverse roles pose a significant challenge to polynomial time learnability. More precisely, [15] brings forward a polynomial time learning algorithm for symmetry-free ELIQs under $\mathcal{EL}$ ontologies where symmetry-free means that there is no subconcept of the form $\exists r.(C \sqcap \exists r^-.D)$ with r a role name. It is not clear at all how to generalize that algorithm to unrestricted ELIQs. Moreover, it is proved in [15] that ELQs are not polynomial query learnable under $\mathcal{ELI}$ ontologies. Thus, inverse roles tend to be challenging both in the query and in the ontology. In contrast, the result in this paper need not impose any restriction on the use of inverse roles. It seems relevant to recall here that *DL-Lite* is a fragment of $\mathcal{ELI}$.

A core technical result underlying our approach is that the *frontier* of an ELIQ q w.r.t. a *DL-Lite*$^-$ ontology is only of polynomial size and can be computed in polynomial time, generalizing a similar result from [11] that does not encompass ontologies. More precisely, a frontier of an ELIQ q w.r.t. a *DL-Lite* ontology $\mathcal{O}$ is a set of ELIQs $\mathcal{F}$ such that $q \subseteq_{\mathcal{O}} q_F$ and $q_F \not\subseteq_{\mathcal{O}} q$ for all $q_F \in \mathcal{F}$ and 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 $q_F \subseteq_{\mathcal{O}} q'$. Note that if one thinks of q as an $\mathcal{ELI}$ concept, then $\mathcal{F}$ is the set of most specific subsumers of q w.r.t. $\mathcal{O}$.[1] Apart from being essential for our learning algorithm, there is another reason for why one may be interested in the frontier. In fact, it is observed in [11] that if an ELIQ q has a frontier of polynomial size, then q can be characterized up to equivalence by polynomially many data examples. Such an example takes the form $(\mathcal{A}, a)$ and is a positive example if $\mathcal{A} \models q(a)$ and a negative example otherwise. The same is true in the presence of ontologies.

Proof details are given in the appendix of the long version of this paper, available at `http://www.informatik.uni-bremen.de/tdki/research/papers.html`.

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 or the inverse r^- of a role name. A *basic concept* B is $\top$, a concept name A, or of the form $\exists R$, R a role. A *DL-Lite* ontology $\mathcal{O}$ is a finite set of (basic) *concept inclusions* $B_1 \sqsubseteq B_2$, *concept disjointness constraints* $B_1 \sqcap B_2 \sqsubseteq \bot$, and *role disjointness constraints* $R_1 \sqcap R_2 \sqsubseteq \bot$. A *DL-Lite*$^-$ *ontology* is a *DL-Lite* ontology that contains no concept disjointness constraints. A *DL-Lite* ontology is in *normal form* if all concept inclusions in it are of the form $A \sqsubseteq B$ or $B \sqsubseteq A$ with A a concept name or $\top$ and B a basic 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 $\mathcal{A}$.

The semantics is defined as usual 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 $r \in \mathsf{N_R}$. We further set $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ and $(\exists R)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists (a,b) \in R^{\mathcal{I}}\}$ for all roles R. This definition of interpretation is slightly different from the usual one, but equivalent;[2] its virtue is uniformity as every ABox is a finite interpretation. An interpretation $\mathcal{I}$ *satisfies* a concept inclusion $B_1 \sqsubseteq B_2$ if $B_1^{\mathcal{I}} \subseteq B_2^{\mathcal{I}}$, a concept

[1] One could equivalently say that $\mathcal{F}$ is the set of LCSs of a single concept which strictly generalize that concept.

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

disjointness constraint $B_1 \sqcap B_2 \sqsubseteq \bot$ if $B_1^{\mathcal{I}} \cap B_2^{\mathcal{I}} = \emptyset$, and a role disjointness constraint $R_1 \sqcap R_2 \sqsubseteq \bot$ if $R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset$.

An interpretation is a *model* of a *DL-Lite* ontology or an ABox if it satisfies all concept inclusions, disjointness constraints and assertions in it. We write $\mathcal{O} \models B_1 \sqsubseteq B_2$ if every model of $\mathcal{O}$ satisfies the basic concept inclusion $B_1 \sqsubseteq B_2$ and $\mathcal{A}, \mathcal{O} \models B(a)$ if every model of $\mathcal{A}$ and $\mathcal{O}$ satisfies the concept assertion $B(a)$. An ABox $\mathcal{A}$ is *satisfiable* w.r.t. a *DL-Lite* ontology $\mathcal{O}$ if $\mathcal{A}$ and $\mathcal{O}$ have a common model.

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 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.

Conjunctive Queries, ELIQs, Homomorphisms. A *conjunctive query* (CQ) takes the form $q(\bar{x}) \leftarrow \phi(\bar{x}, \bar{y})$ where ϕ is a conjunction of *concept atoms* $A(x)$, with $A \in \mathsf{N_C}$, and *role atoms* $r(x, y)$, with $r \in \mathsf{N_R}$ over variables $x, y \in \bar{x} \cup \bar{y}$. We may write $r^-(x, y)$ in place of $r(y, x)$. We refer to the variables in $\bar{x}$ as *answer variables* and to the variables in $\bar{y}$ as *quantified variables*. We use $\mathsf{var}(q)$ to denote the set of all variables in $\bar{x}$ and $\bar{y}$. We may view a CQ $q(\bar{x})$ as a set of atoms when convenient and write $r(x, y) \in q(\bar{x})$ to mean that $r(x, y)$ occurs in the conjunction ϕ.

A conjunctive query $q(\bar{x})$ is *unary* if $q(\bar{x})$ only has a single answer variable. 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. An *ELIQ* is a unary CQ q that does not contain a cycle and such that the undirected graph $G_q = (\mathsf{var}(q), \{\{y, z\} \mid r(y, z) \in q\})$ is connected. Note that every ELIQ can be seen as an $\mathcal{ELI}$ concept in a straightforward way and vice versa; see [5] for a definition of $\mathcal{ELI}$ concepts. We use $\mathcal{A}_q$ to denote the ABox obtained from q by viewing variables as individuals and atoms as assertions. A CQ q is *satisfiable* w.r.t. a *DL-Lite* ontology $\mathcal{O}$ if $\mathcal{A}_q$ is satisfiable w.r.t. $\mathcal{O}$.

A *homomorphism* h from interpretation $\mathcal{I}_1$ to interpretation $\mathcal{I}_2$ is a mapping from $\Delta^{\mathcal{I}_1}$ to $\Delta^{\mathcal{I}_2}$ such that $d \in A^{\mathcal{I}_1}$ implies $h(d) \in A^{\mathcal{I}_2}$ and $(d, e) \in r^{\mathcal{I}_1}$ implies $(h(d), h(e)) \in r^{\mathcal{I}_2}$. We use $\mathsf{img}(h)$ to denote the set $\{e \in \Delta^{\mathcal{I}_2} \mid \exists d \in \Delta^{\mathcal{I}_1} \colon h(d) = e\}$. For $d_i \in \Delta^{\mathcal{I}_i}$, $i \in \{1, 2\}$, we write $\mathcal{I}_1, d_1 \to \mathcal{I}_2, d_2$ if there is a homomorphism h from $\mathcal{I}_1$ to $\mathcal{I}_2$ with $h(d_1) = d_2$. With a homomorphism from a CQ q to an interpretation $\mathcal{I}$, we mean a homomorphism from $\mathcal{A}_q$ to $\mathcal{I}$. For a unary CQ $q(x)$, we write $q(x) \to (\mathcal{I}, d)$ if there is a homomorphism h from q to $\mathcal{I}$ with $h(x) = d$. Let $q(x)$ be a unary CQ and $\mathcal{I}$ an interpretation. An element $d \in \Delta^{\mathcal{I}}$ is an *answer to q in $\mathcal{I}$*, written $\mathcal{I} \models q(d)$, if $q(x) \to (\mathcal{I}, d)$. Now let $\mathcal{O}$ be a *DL-Lite* ontology and $\mathcal{A}$ an ABox. An individual $a \in \mathsf{ind}(\mathcal{A})$ is an *answer to q on $\mathcal{A}$ w.r.t. $\mathcal{O}$*, written $\mathcal{A}, \mathcal{O} \models q(a)$, if a is an answer to q in every model of $\mathcal{O}$ and $\mathcal{A}$.

For q_1 and q_2 unary CQs and $\mathcal{O}$ a *DL-Lite* ontology, 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 $\mathcal{A}, \mathcal{O} \models q_2(a)$. 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. The following two technical notions are used throughout this paper. A CQ q is *$\mathcal{O}$-saturated*, with $\mathcal{O}$ a *DL-Lite* 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 $S \subsetneq \mathsf{var}(q)$ such that $q \equiv_{\mathcal{O}} q|_S$ with $q|_S$ the restriction of q to the atoms that only contain variables in S.

The following is a consequence of the fact that acyclic CQs over *DL-Lite* ontologies can be answered in polynomial time [8].

Lemma 1. *Given an ELIQ q and a DL-Lite ontology $\mathcal{O}$, we can find in polynomial time an $\mathcal{O}$-saturated and $\mathcal{O}$-minimal ELIQ q' with $q \equiv_{\mathcal{O}} q'$.*

Universal Model. Let $\mathcal{O}$ be a *DL-Lite* ontology and $\mathcal{A}$ an ABox that is satisfiable w.r.t. $\mathcal{O}$. A *trace* for $\mathcal{A}$ and $\mathcal{O}$ is a sequence $t = aR_1 \dots R_n$, $n \geq 0$, such that $a \in \mathsf{ind}(\mathcal{A})$, the basic concepts $\exists R_1, \dots, \exists R_n$ occur in $\mathcal{O}$, $\mathcal{A}, \mathcal{O} \models \exists R_1(a)$, and $\mathcal{O} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ for $1 \leq i < n$. Let $\mathbf{T}$ denote the set of all traces for $\mathcal{A}$ and $\mathcal{O}$. Then the *universal model* of $\mathcal{A}$ and $\mathcal{O}$ is

$$\begin{aligned}\mathcal{U}_{\mathcal{A},\mathcal{O}} = \mathcal{A} \cup \{A(a) \mid \mathcal{A}, \mathcal{O} \models A(a)\} \cup \\ \{A(tR) \mid tR \in \mathbf{T} \text{ and } \mathcal{O} \models \exists R^- \sqsubseteq A\} \cup \{R(t, tR) \mid tR \in \mathbf{T}\}.\end{aligned}$$

For brevity, we write $\mathcal{U}_{q,\mathcal{O}}$ instead of $\mathcal{U}_{\mathcal{A}_q,\mathcal{O}}$ for any conjunctive query q.

3 Computing Frontiers in Polynomial Time

We show that for every ELIQ q and *DL-Lite* 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. This generalizes a result from [11] for the case without ontologies. We also observe that the same is not true when *DL-Lite* is extended with conjunction, and that ELIQs can be characterized up to equivalence by polynomially many data examples in the presence of *DL-Lite* ontologies.

Definition 1. *A* frontier *of an ELIQ q w.r.t. a DL-Lite ontology $\mathcal{O}$ is a finite 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 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 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 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.*

For proving Theorem 1, we first observe that we can concentrate on ontologies that are in normal form.

Lemma 2. *For every DL-Lite ontology $\mathcal{O}$, we can construct in polynomial time a DL-Lite 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 normalization of $\mathcal{O}$ introduces fresh concept names $X_{\exists R}$ that represent the basic concept $\exists R$. In the proof of Lemma 2, we construct the frontier of q w.r.t. $\mathcal{O}'$ by replacing atoms $X_{\exists R}(x)$ with atoms $R(x, y)$, y a fresh variable.

Now we start with the proof of Theorem 1. Let $\mathcal{O}$ and $q(x)$ be as in the formulation of the theorem, $\mathcal{O}$ in normal form. By Lemma 1 and the fact that equivalent queries have the same frontiers, we can assume that q is $\mathcal{O}$-saturated and $\mathcal{O}$-minimal. To construct a frontier of q w.r.t. $\mathcal{O}$, we consider all ways to weaken q in a minimal way where weakening means to construct from q an ELIQ q' such that $q \subseteq_{\mathcal{O}} q'$ and $q' \not\subseteq_{\mathcal{O}} q$.

We start with some notation. We view the answer variable x of q as the root of the undirected tree G_q, thus imposing a direction on this tree which allows us to use notions for directed trees, such as successor, predecessor, and leaf, for the variables in q. Note that the imposed direction is unrelated to the direction of (inverse) roles in atoms in q. For every $z \in \mathsf{var}(q)$, we use q_z to denote the ELIQ obtained from q by taking the subtree of G_q rooted at z and making z the answer variable. For each variable $y \in \mathsf{var}(q)$, we define a set $\mathit{\Gamma}_y$ of atoms in q that mention y and represent options that we have for weakening q. Formally, $\mathit{\Gamma}_y$ contains

- all role atoms $R(y, z) \in q_y$ and
- all concept atoms $A(y) \in q$ such that
  (i) there is no $B(y) \in q$ with $\mathcal{O} \models B \sqsubseteq A$ and $\mathcal{O} \not\models A \sqsubseteq B$ and
  (ii) there is no $R(y, z) \in q$ with $\mathcal{O} \models \exists R \sqsubseteq A$.

Informally, we can weaken q by choosing a $y \in \mathsf{var}(q)$ and then removing a concept atom $A(y) \in \mathit{\Gamma}_y$ or a role atom $R(y, z) \in \mathit{\Gamma}_y$ as well as the subtree of q rooted at variable z. However, such removals alone are not enough to obtain a *minimal* weakening of q and must be accompanied by certain additions, as detailed below. Note that Conditions (i) and (ii) are needed to ensure that removing $A(y)$ indeed weakens q, that is, the resulting query is not equivalent to q w.r.t. $\mathcal{O}$.

We define a set of ELIQs $\mathcal{F}_q(y)$ for every variable $y \in \mathsf{var}(q)$, by induction on the codepth of y in q. The set $\mathcal{F}_q(x)$ ultimately obtained is a frontier of q w.r.t. $\mathcal{O}$. For every $y \in \mathsf{var}(q)$, set

$$\mathcal{F}_q(y) = \{q^\alpha(y) \mid \alpha \in \mathit{\Gamma}_y\}$$

where $q^\alpha(y)$ is constructed by starting with $q_y(y)$ and then doing the following:

1. if $\alpha = A(y)$, remove all atoms $B(y)$ with $\mathcal{O} \models A \equiv B$ (including α);
2. if $\alpha = R(y, z)$, remove α and all atoms of q_z. For each $q^\beta(z) \in \mathcal{F}_q(z)$, add a disjoint copy $\tilde{q}^\beta$ of q^β and the role atom $R(y, \tilde{z})$ where $\tilde{z}$ is the copy of z in $\tilde{q}^\beta$;
3. for each $S(y, z_1) \in q_y$ with $S(y, z_1) \neq \alpha$, add a disjoint copy q' of q and the role atom $S(y', z_1)$ where y' is the copy of y in q';

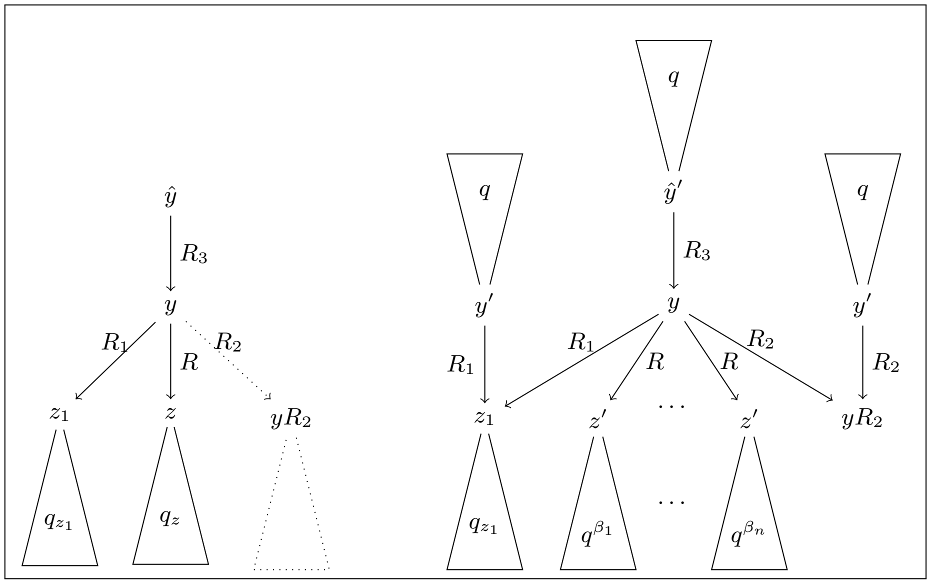

Fig. 1. Construction of $q^{R(y,z)}(y)$ from q

4. for each $S(y, yS) \in \mathcal{U}_{q_y,\mathcal{O}}$ with yS a trace such that $\mathcal{O} \not\models \exists S \sqsubseteq A$ if $\alpha = A(y)$, add a disjoint copy q' of q and the role atoms $S(y, z_1), S(y', z_1)$, where y' is the copy of y in q' and z_1 is a fresh variable name;
5. if there is a $S(\hat{y}, y) \in q$ with $\hat{y}$ the predecessor of y, then add a disjoint copy q' of q and the role atom $S(\hat{y}', y)$ where $\hat{y}'$ is the copy of $\hat{y}$ in q'.

Note that Step 2 is the inductive step, where every subtree rooted at a successor z of y is replaced with all ELIQs from $\mathcal{F}_q(z)$. The construction is illustrated in Figure 1 which on the left side shows a variable y in q with predecessor $\hat{y}$, two successors z_1 and z in q, and one additional successor yR_2 (a trace) in the universal model $\mathcal{U}_{q,\mathcal{O}}$. On the right side, it displays the ELIQ $q^{R(y,z)}(y) \in \mathcal{F}_q(y)$, assuming that $\mathcal{F}_q(z) = \{q^{\beta_1}, \ldots, q^{\beta_n}\}$. We remark that for $y \neq x$, the set $\mathcal{F}_q(y)$ is not necessarily a frontier of the ELIQ $q_y(y)$ because the part of q that is outside of subtree q_y is taken into account in the construction of $\mathcal{F}_q(y)$, in Steps 3-5. Our construction generalizes the construction of frontiers of ELIQs without ontologies given in [11].[3] It indeed yields a frontier.

Lemma 3. *$\mathcal{F}_q(x)$ is a frontier of $q(x)$ w.r.t. $\mathcal{O}$.*

We next observe that the obtained frontier $\mathcal{F}_q(x)$ is of polynomial size. It is then clear that it can be computed in polynomial time as described above since subsumption between basic concepts in *DL-Lite* can be decided in polynomial time [10].

[3] There is actually a small omission in [11] as the counterpart of our Step 5 is missing.

Lemma 4. $\sum_{q^\alpha(x)\in\mathcal{F}_q(x)} |\mathsf{var}(q^\alpha)| \leq |\mathsf{sig}(q)|\cdot|\mathsf{var}(q)|^3\cdot(|\mathsf{var}(q)|+1)\cdot(||\mathcal{O}||+1)$.

We next observe that adding conjunction to *DL-Lite* destroys polynomial frontiers and thus Theorem 1 does not extend to *DL-Lite*$_{\mathsf{horn}}$ ontologies [3]. 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 $q(x) \leftarrow A_1(x) \wedge \cdots \wedge A_n(x)$ 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*.*

The proof is a variation of a proof given in [15] showing that AQ$^\wedge$s are not polynomial time learnable under conjunctive ontologies. It is based on the following AQ$^\wedge$s and ontologies:

$$q_n(x) \leftarrow A_1(x) \wedge A'_1(x) \wedge \cdots \wedge A_n(x) \wedge A'_n(x)$$
$$\mathcal{O}_n = \{A_i \sqcap A'_i \sqsubseteq A_1 \sqcap A'_1 \sqcap \cdots \sqcap A_n \sqcap A'_n \mid 1 \leq i \leq n\}.$$

In fact, the minimal frontier contains all AQ$^\wedge$s that contain exactly one of the conjuncts A_i and A'_i, for $1 \leq i \leq n$. Observe that in the proof of Theorem 1, there are only polynomially many choices for weakening an ELIQ, represented by the sets $\mathit{\Gamma}_y$, $y \in \mathsf{var}(q)$. In contrast, weakening the AQ$^\wedge$ q_n w.r.t. ontology $\mathcal{O}_n$ in a minimal way requires to choose for each $i \in \{1, \ldots, n\}$ whether A_i or A'_i should be removed, and there are exponentially many such choices.

To close this section, we briefly consider the unique characterization of ELIQs in terms of polynomially many data examples. A *data example* takes the form $(\mathcal{A}, a)$ where $\mathcal{A}$ is an ABox and $a \in \mathsf{ind}(\mathcal{A})$. Let E^+, E^- be finite sets of data examples. An ELIQ q *fits* (E^+, E^-) w.r.t. a *DL-Lite* ontology $\mathcal{O}$ if $(\mathcal{A}, a) \in E^+$ implies $\mathcal{A}, \mathcal{O} \models q(a)$ and $(\mathcal{A}, a) \in E^-$ implies $\mathcal{A}, \mathcal{O} \not\models q(a)$. We say that (E^+, E^-) *uniquely characterizes* q w.r.t. $\mathcal{O}$ if q fits (E^+, E^-) and every ELIQ q' that also fits (E^+, E^-) satisfies $q \equiv_{\mathcal{O}} q'$. The following is a consequence of Theorem 1.

Theorem 3. *For every DL-Lite ontology* $\mathcal{O}$ *and every ELIQ* q *that is satisfiable w.r.t.* $\mathcal{O}$*, we can compute in polynomial time data examples* (E^+, E^-) *that uniquely characterize* q *w.r.t.* $\mathcal{O}$*.*

Note that unique characterizability is closely related to the reverse engineering of CQs, also called *query-by-example* and studied in a DL context in [17,24].

4 Learning ELIQs

We use the results from the previous section to show that ELIQs are polynomial time learnable using only membership queries under *DL-Lite* ontologies if the learner is provided with an initial CQ q_H^0 such that q_H^0 is satisfiable w.r.t. the ontology $\mathcal{O}$ and $q_H^0 \subseteq_{\mathcal{O}} q_T$ where q_T is the target ELIQ. Such a q can be constructed in polynomial time if $\mathcal{O}$ is formulated in *DL-Lite*$^-$. Otherwise, it can be produced by a single initial equivalence query with an ELIQ that is not satisfiable w.r.t. $\mathcal{O}$, forcing the learner to provide a positive counterexample $(\mathcal{A}, a)$ from which we can extract the desired q_H^0. Before proving these positive results, however, we first observe that polynomial time learning using only membership queries (but no initial equivalence query) is not possible when $\mathcal{O}$ contains concept disjointness constraints.

A *disjointness ontology* is a *DL-Lite* ontology that only consists of concept disjointness constraints.

Theorem 4. *$AQ^\wedge$s are not polynomial query learnable under disjointness ontologies using only membership queries.*

The proof of Theorem 4 is a variation of that of Theorem 2. We next present the main results of this section.

Theorem 5.

1. *ELIQs are polynomial time learnable under DL-Lite ontologies using only membership queries and a single initial equivalence query.*
2. *ELIQs are polynomial time learnable under DL-Lite$^-$ ontologies using only membership queries.*

Throughout this section, we may assume the ontology to be in normal form.

Lemma 5. *If ELIQs are polynomial time learnable under DL-Lite ontologies in normal form using membership queries and a single initial equivalence query, then this is also true for unrestricted DL-Lite ontologies. The same holds for DL-Lite$^-$ ontologies without the initial equivalence query.*

The idea to prove Lemma 5 is to convert the given ontology into normal form and then run the learning algorithm for ontologies in normal form. Since that algorithm may pose membership queries and equivalence queries that involve fresh concept names introduced during normalization, we need to replace those concept names as described after Lemma 2 before forwarding the query to the oracle (which uses the original non-normalized ontology).

We prove Points 1 and 2 of Theorem 5 simultaneously. Let $\mathcal{O}$ be a *DL-Lite* ontology and Σ a finite signature that contains all symbols in $\mathcal{O}$, both known to the learner and the oracle. Further let $q_T(y)$ be the target ELIQ known to the oracle, formulated in signature Σ and satisfiable w.r.t. $\mathcal{O}$. The algorithm that enables the learner to learn q_T in polynomial time is displayed as Algorithm 1. It takes as input a CQ q_H^0 that is satisfiable w.r.t. $\mathcal{O}$ and satisfies $q_H^0 \subseteq_{\mathcal{O}} q_T$. Note that q_H^0 need not be an ELIQ. The algorithm then constructs and repeatedly updates a hypothesis ELIQ q_H while maintaining that $q_H \subseteq_{\mathcal{O}} q_T$. The initial call to subroutine 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 $q_H \subseteq_{\mathcal{O}} q_T$. Additionally, the algorithm applies the minimize subroutine

Algorithm 1 Algorithm for learning ELIQs under *DL-Lite* ontologies

```
Input A DL-Lite ontology $\mathcal{O}$ 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 $q_H$
```

to ensure that the new q_H is $\mathcal{O}$-minimal and to avoid an excessive blowup while iterating in the while loop.

Before we detail the subroutines treeify and minimize, we explain how to obtain the argument q_H^0 to the algorithm. Suppose first that $\mathcal{O}$ contains neither concept disjointness constraints nor role disjointness constraints. Then

$$q_H^0(x) = \{A(x) \mid A \in \Sigma \cap \mathsf{N_C}\} \cup \{r(x,x) \mid r \in \Sigma \cap \mathsf{N_R}\}. \tag{1}$$

If $\mathcal{O}$ contains concept or role disjointness constraints, then we cannot use the above q_H^0 because it is not satisfiable w.r.t. $\mathcal{O}$. If, however, $\mathcal{O}$ contains only role disjointness constraints, then we can still find a suitable q_H^0. Let $\mathbf{R}$ be the set of all $r \in \Sigma \cap \mathsf{N_R}$ such that $\exists r$ is satisfiable w.r.t. $\mathcal{O}$, introduce four variables $x_r^0, x_r^1, x_{r^-}^0, x_{r^-}^1$ for all $r \in \mathbf{R}$, and fix a linear order $\preceq$ on $\mathbf{R}' = \mathbf{R} \cup \{r^- \mid r \in \mathbf{R}\}$. Fix any variable $x := x_R^i$. Then, q_H^0 is given by

$$\begin{aligned} q_H^0(x) = {} & \{A(x_R^i) \mid A \in \Sigma \cap \mathsf{N_C}, R \in \mathbf{R}', i \in \{0,1\}\} \cup \\ & \{R(x_S^i, x_R^i) \mid R, S \in \mathbf{R}', S \preceq R, i \in \{0,1\}\} \cup \\ & \{R(x_S^i, x_R^{1-i}) \mid R, S \in \mathbf{R}', S \not\preceq R, i \in \{0,1\}\}. \end{aligned}$$

Observe that every variable has an R-successor for every (satisfiable) role R. Therefore, there is a homomorphism from every satisfiable target ELIQ q_T to q_H^0, which shows that indeed $q_H^0 \subseteq_{\mathcal{O}} q_T$.

In the remaining case when $\mathcal{O}$ contains a concept disjointness constraint $A \sqcap B \sqsubseteq \bot$, we pose the ELIQ $A(x) \wedge B(x)$ as an equivalence query to the oracle. Since the target query is satisfiable w.r.t. $\mathcal{O}$, the oracle returns a positive counterexample $(\mathcal{A}, a)$. The desired query q_H^0 is $(\mathcal{A}, a)$ viewed as a CQ with answer variable a. In the algorithm, we may w.l.o.g. assume that q_H^0 is $\mathcal{O}$-saturated due to Lemma 1.

The minimize subroutine. The subroutine takes as input a unary CQ $q(x)$ that is $\mathcal{O}$-saturated, satisfiable w.r.t. $\mathcal{O}$, and satisfies $q \subseteq_{\mathcal{O}} q_T$. It computes an $\mathcal{O}$-minimal unary CQ q' with $q \subseteq_{\mathcal{O}} q' \subseteq_{\mathcal{O}} q_T$ using membership queries. This is done by exhaustively applying the following operation:

Remove successor. Choose a role atom $r(x_1, x_2) \in q$ and let q^- be the restriction of $q \setminus \{r(x_1, x_2)\}$ to the atoms that only contain variables which are reachable from x in $G_{q\setminus\{r(x_1,x_2)\}}$. Pose the membership query $\mathcal{A}_{q^-},\mathcal{O} \models q_T(x)$. If the response is positive, continue with q^- in place of q.

This operation preserves $\mathcal{O}$-saturation and satisfiability w.r.t. $\mathcal{O}$. Since the operation is applied at most once to each atom in q, the running time and number of membership queries is polynomial in $|\mathsf{var}(q)| + |\Sigma|$. The following lemma summarizes important properties of $q' = \mathsf{minimize}(q)$ that we need to show correctness and polynomial running time of Algorithm 1.

Lemma 6. *Let q be a unary CQ that is $\mathcal{O}$-saturated and satisfiable w.r.t. $\mathcal{O}$ such that $q \subseteq q_T$ for the target query $q_T(y)$, and let $q' = \mathsf{minimize}(q)$. Then*

1. *$q \subseteq_{\mathcal{O}} q'$ and $q' \subseteq_{\mathcal{O}} q_T$;*
2. *$\mathsf{var}(q') \subseteq \mathsf{img}(h)$ for every homomorphism h from q_T to $\mathcal{U}_{q',\mathcal{O}}$ with $h(y) = x$ (and consequently $|\mathsf{var}(q')| \leq |\mathsf{var}(q_T)|$);*
3. *q' is $\mathcal{O}$-minimal.*

When q is an ELIQ, $\mathsf{minimize}(q)$ is also an ELIQ. We define $\mathsf{minimize}$ on unrestricted CQs since it is applied to non-ELIQs as part of the $\mathsf{treeify}$ subroutine, described next.

The $\mathsf{treeify}$ subroutine. The subroutine takes as input a unary CQ $q(x)$ that is $\mathcal{O}$-saturated, satisfiable w.r.t. $\mathcal{O}$, and satisfies $q \subseteq_{\mathcal{O}} q_T$. It computes an ELIQ $q' = \mathsf{treeify}(q)$ with $q \subseteq_{\mathcal{O}} q' \subseteq_{\mathcal{O}} q_T$ by repeatedly increasing the length of cycles in q and posing membership queries; similar constructions are used in in [21,15]. Formally, $\mathsf{treeify}$ 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 every cycle. More precisely, p_i' is the result of the following operation.

Double cycle. Choose a cycle $R_1(x_1,x_2),\ldots,R_n(x_n,x_1)$ in p_i and let p_i' be the CQ obtained as follows: start with p_i, introduce copies $x_1',\ldots,x_n'$ of $x_1,\ldots,x_n$, and

– remove all atoms $R(x_n,x_1)$;
– add $A(x_j')$ if $A(x_j) \in p_i$ with $1 \leq j \leq n$;
– add $R(x_j',z)$ if $R(x_j,z) \in p_i$ with $1 \leq j \leq n$ and $z \notin \{x_1,\ldots,x_n\}$;
– add $R(x_j',x_k')$ if $R(x_j,x_k) \in p_i$ with $1 \leq j,k \leq n$ and $\{j,k\} \neq \{1,n\}$;
– add $R(x_n,x_1')$ and $R(x_n',x_1)$ if $R(x_n,x_1) \in p_i$.

Once p_i contains no more cycles, $\mathsf{treeify}$ stops and returns p_i. The next lemma states the central properties of this construction.

Lemma 7. *For all $i \geq 1$,*

1. *p_i is $\mathcal{O}$-saturated and satisfiable w.r.t. $\mathcal{O}$;*
2. *$p_i \subseteq_{\mathcal{O}} q_T$;*
3. *$p_i \subseteq_{\mathcal{O}} p_{i+1}$ and $|\mathsf{var}(p_{i+1})| > |\mathsf{var}(p_i)|$.*

Point 3 of Lemma 7 and the ‘consequently’ part of Point 2 of Lemma 6 imply that $\mathsf{treeify}$ terminates and thus eliminates all cycles in q while maintaining $q \subseteq_{\mathcal{O}} q_T$. The next lemma makes this precise.

Lemma 8. *Let q be a CQ that is $\mathcal{O}$-saturated, satisfiable w.r.t. $\mathcal{O}$, and satisfies $q \subseteq_{\mathcal{O}} q_T$. Then $q' = \mathsf{treeify}(q)$ is an ELIQ that can be computed in time polynomial in $|\mathsf{var}(q_T)| + ||q||$ using membership queries.*

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. Using the properties of frontiers, $\mathsf{minimize}$, and $\mathsf{treeify}$ we can now show that the hypotheses approximate the target query in an increasingly closer way.

Lemma 9. *For all $i \geq 1$,*

1. *$q_i \subseteq_{\mathcal{O}} q_T$;*
2. *$q_i \subseteq_{\mathcal{O}} q_{i+1}$ and $q_{i+1} \not\subseteq_{\mathcal{O}} q_i$;*
3. *$\mathsf{var}(q_i) \subseteq \mathsf{img}(h)$ for every homomorphism h from q_{i+1} to $\mathcal{U}_{q_i,\mathcal{O}}$ with $h(x) = x$.*

Point 3 of Lemma 9 implies that $|\mathsf{var}(q_{i+1})| \geq |\mathsf{var}(q_i)|$, and this can be used to show that the while loop in Algorithm 1 terminates after a polynomial number of iterations, arriving at a hypothesis $q_H \subseteq_{\mathcal{O}} q_T$ such that there is no $q_F \in \mathcal{F}_{q_H}$ with $q_F \subseteq_{\mathcal{O}} q_T$, that is, $q_H \equiv_{\mathcal{O}} q_T$.

Lemma 10. *$q_n \equiv q_T$ for some $n \leq p(|\mathsf{var}(q_T)| + |\Sigma|)$, p a polynomial.*

It follows from Lemma 10, the ‘consequently’ part of Point 2 of Lemma 6, and Lemma 8 that Algorithm 1 is a polynomial time learning algorithm, thus completing the proof of Theorem 5.

5 Outlook

As future work, we are going to consider extensions of the setup studied in this paper. We are optimistic that the approach presented here can be extended to *DL-Lite* ontologies with role inclusions, yielding polynomial size frontiers and polynomial query learnability also for that logic (but not polynomial time learnability since subsumption becomes NP-complete). Natural next steps could then be to replace *DL-Lite* with linear tuple-generating dependencies (TGDs) [9] or with *DL-Lite*$_{\mathsf{krom}}$ ontologies [3]. In contrast, it is clear from Theorem 2 and the results in [15] that our results do not extend to *DL-Lite*$_{\mathsf{horn}}$. It is not ruled out, however, that ELIQs can be learned in polynomial time w.r.t. *DL-Lite*$_{\mathsf{horn}}$ ontologies when both membership and equivalence queries can be used. Another natural question is whether CQs can be learned in polynomial time in the presence of *DL-Lite* ontologies. It is known that this is not possible with only membership queries even without ontologies [11], but it might be possible with both membership and equivalence queries.

Acknowledgement. Carsten Lutz was supported by DFG CRC 1320 EASE.

References

1. Angluin, D.: Learning regular sets from queries and counterexamples. Inf. Comput. **75**(2), 87–106 (1987)
2. Angluin, D.: Queries and concept learning. Mach. Learn. **2**(4), 319–342 (1987)
3. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. of Artifical Intelligence Research **36**, 1–69 (2009)
4. Baader, F.: Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In: Proc. of IJCAI. pp. 319–324. Morgan Kaufmann (2003)
5. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logics. Cambride University Press (2017)
6. Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In: Proc. of IJCAI. pp. 96–103. Morgan Kaufmann (1999)
7. Baader, F., Sertkaya, B., Turhan, A.: Computing the least common subsumer w.r.t. a background terminology. J. Appl. Log. **5**(3), 392–420 (2007)
8. Bienvenu, M., Ortiz, M., Simkus, M., Xiao, G.: Tractable queries for lightweight description logics. In: Proc. of IJCAI. pp. 768–774 (2013)
9. Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. J. Web Semant. **14**, 57–83 (2012)
10. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. of Automated Reasoning **39**(3), 385–429 (2007)
11. ten Cate, B., Dalmau, V.: Conjunctive queries: Unique characterizations and exact learnability. In: Proc. of ICDT. LIPIcs, vol. 186, pp. 9:1–9:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)
12. Cohen, W.W., Hirsh, H.: The learnability of description logics with equality constraints. Mach. Learn. **17**(2-3), 169–199 (1994)
13. Cohen, W.W., Hirsh, H.: Learning the classic description logic: Theoretical and experimental results. In: Proc. of KR. pp. 121–133. Morgan Kaufmann (1994)
14. Frazier, M., Pitt, L.: Classic learning. Mach. Learn. **25**(2-3), 151–193 (1996)
15. Funk, M., Jung, J.C., Lutz, C.: Actively learning concept and conjunctive queries under $\mathcal{EL}^r$-ontologies. In: Proc. of IJCAI (2021)
16. Funk, M., Jung, J.C., Lutz, C., Pulcini, H., Wolter, F.: Learning description logic concepts: When can positive and negative examples be separated? In: Proc. of IJCAI. pp. 1682–1688 (2019)
17. Gutiérrez-Basulto, V., Jung, J.C., Sabellek, L.: Reverse engineering queries in ontology-enriched systems: The case of expressive Horn description logic ontologies. In: Proc. of IJCAI-ECAI. ijcai.org (2018)
18. Jung, J.C., Lutz, C., Pulcini, H., Wolter, F.: Logical separability of incomplete data under ontologies. In: Proc. of KR. pp. 517–528 (2020)
19. Jung, J.C., Lutz, C., Wolter, F.: Least general generalizations in description logic: Verification and existence. In: Proc. of AAAI (2020)
20. Konev, B., Lutz, C., Ozaki, A., Wolter, F.: Exact learning of lightweight description logic ontologies. J. Mach. Learn. Res. **18**(201), 1–63 (2018)
21. Konev, B., Ozaki, A., Wolter, F.: A model for learning description logic ontologies based on exact learning. In: Proc. of AAAI. pp. 1008–1015. AAAI Press (2016)
22. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. Mach. Learn. **78**, 203–250 (2010)

23. Lehmann, J., Völker, J.: Perspectives on Ontology Learning, Studies on the Semantic Web, vol. 18. IOS Press (2014)
24. Ortiz, M.: Ontology-mediated queries from examples: a glimpse at the DL-Lite case. In: Proc. of GCAI. pp. 1–14 (2019)
25. Ozaki, A.: Learning description logic ontologies: Five approaches. where do they stand? KI - Künstliche Intelligenz (2020)
26. Ozaki, A., Persia, C., Mazzullo, A.: Learning query inseparable $\mathcal{ELH}$ ontologies. In: Proc. of AAAI. pp. 2959–2966 (2020)
27. Sarker, M.K., Hitzler, P.: Efficient concept induction for description logics. In: Proc. of AAAI. pp. 3036–3043 (2019)
28. Zarrieß, B., Turhan, A.: Most specific generalizations w.r.t. general $\mathcal{EL}$-TBoxes. In: Proc. of IJCAI. pp. 1191–1197 (2013)