Exact Learning of $\mathcal{ELI}$ Queries in the Presence of
DL-Lite-Horn Ontologies

Maurice Funk$^1$, Jean Christoph Jung$^2$ and Carsten Lutz$^1$

$^1$Leipzig University
$^2$University of Hildesheim

Abstract
Learning, in Angluin’s framework of exact learning, a query in the presence of a description logic ontology often involves as a crucial (iterated) step the generalization of a hypothesis query. This may be achieved, for example, by constructing a least general generalization of the hypothesis and a counterexample that was provided by the oracle. In this research note, we observe that it may pay off to resort to a more liberal construction that uses the counterexample as a guide to produce a generalization of the hypothesis while not necessarily achieving a generalization of the counterexample. We use this approach to show polynomial time learnability of $\mathcal{ELI}$ concept queries (ELIQs) in the presence of ontologies which are formulated in a mild restriction of DL-Lite$^-$.

Keywords
Exact Learning, Least General Generalizations, DL-Lite-Horn

1. Introduction
Various forms of learning description logic (DL) concepts, ontologies, and queries have been studied in the literature, including PAC learning [1, 2, 3], the construction of the least common subsumer (LCS) and the most specific concept (MSC) [4, 5, 6, 7, 8], and learning from labeled data examples [9, 10, 11, 12, 13, 14]. In this research note, we consider Angluin’s framework of exact learning where a learner interacts in a game-like fashion with an oracle [15, 16]. The main aim is to find an algorithm that enables the learner to construct the target object in polynomial time based on queries that they pose to the oracle, even when the oracle does not 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) [17], see also [18, 19] and the survey [20]. This was complemented by studies of exactly learning DL concepts and queries: learning $\mathcal{ELI}$ concept queries (ELIQs) without ontologies is considered in [21] while [22] studies learning $\mathcal{EL}$ concept queries (ELQs), ELIQs, and restricted forms of conjunctive queries (CQs) in the presence of $\mathcal{EL}$ and $\mathcal{ELI}$ ontologies. Very recently, [23, 24] has investigated learning ELIQs in the presence of ontologies formulated in the DL-Lite dialects DL-Lite$^H$ and DL-Lite$^{F-}$ where the ‘$-$’ indicates a restriction on the use of inverse functional roles.
To explain the contribution of this article, let us introduce exact learning of queries in the presence of DL ontologies. Learner and oracle both 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 $\mathcal{O}$. There are two kinds of queries that the learner may pose to the oracle. 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 query $q_H$ 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). Then, polynomial time learnability means that 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 $q_T$, $\mathcal{O}$, and 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 has to be bounded by such a polynomial.

We next describe, on an informal level, how a typical learning algorithm works. The described strategy has been used, e.g., to learn CQs and mappings in database theory [25, 26], LTL formulas [27], as well as ontologies and queries in a DL context [17, 22, 24]. The algorithm constructs a sequence

$q_0 \subseteq \mathcal{O} q_1 \subseteq \mathcal{O} q_2 \subseteq \mathcal{O} \cdots$

of increasingly general hypothesis queries, where ‘$\subseteq \mathcal{O}$’ denotes query containment under the ontology $\mathcal{O}$. It maintains the invariant that $q_i \subseteq \mathcal{O} q_T$ for all $i \geq 0$ where $q_T$ is the target query to be learned (only known to the oracle). As the initial hypothesis $q_0$, one constructs a very strong query which implies any possible $q_T$. If, for example, the query language is unary CQs and the ontology $\mathcal{O}$ does not express any disjointnesses between concept and role names, then this query might be the single-variable query that has atoms $A(x)$ and $r(x, x)$ for all concept names $A$ and role names $r$. If a more restricted query class such as ELIQs is used or the ontology expresses disjointness constraints, then the construction of the initial hypothesis might be more subtle and also involve interaction with the oracle, see [22, 24].

To move from hypothesis $q_i$ to $q_{i+1}$, the algorithm repeatedly employs a suitable generalization strategy, which may be viewed as the heart of the learning algorithm. In the literature, one finds two main such strategies. To describe them, assume that the class of target queries $\mathcal{Q}$ is a class of CQs such as all CQs or all ELIQs.

When query and ontology language are sufficiently restricted, it may happen that the set of all possible least general generalizations of the current hypothesis $q_H$ can be computed in polynomial time. Then, membership queries to the oracle can be used to identify a generalization that implies the target, and the algorithm does not need to use equivalence queries at all. This strategy has been used, for example, to learn ELIQs in the presence of ontologies formulated in DL-Lite$^H$ and DL-Lite$^F$ [21, 24], but it already fails for learning ELIQs under DL-Lite$^{horn}$ ontologies [24]; recall that the $^{horn}$ subscript indicates the presence of conjunction. The second strategy is to pose the current hypothesis as an equivalence query to the oracle, and to then construct a least general generalization of the hypothesis $q_H$ and the returned counterexample $(\mathcal{A}, \bar{a})$, which must be positive since the hypothesis is contained in the target. What we mean
here is a \( \mathcal{Q} \)-LGG, that is, a query \( p \) such that \( q_H \subseteq_\mathcal{O} p, q_A \subseteq_\mathcal{O} p \) where \( q_A \) is \( A \) viewed as a CQ with answer variables \( a \), and \( p \subseteq_\mathcal{O} p' \) for every \( p' \in \mathcal{Q} \) with \( q_H \subseteq_\mathcal{O} p' \) and \( q_A \subseteq_\mathcal{O} q' \). This approach has been used to learn unrestricted CQs without ontologies [21] and to learn syntactically restricted CQs under \( \mathcal{E}\mathcal{L} \) ontologies [22].

The aim of this note is to introduce a variation of the second approach to generalization, and to demonstrate its usefulness by devising a polynomial time learning algorithm for ELIQs in the presence of \( DL\text{-}Lite^{\infty}_{\text{horn}} \) ontologies. In theory, a natural way to construct a \( \mathcal{Q} \)-LGG of the hypothesis \( q_H \) and counterexample \( A \) is to build the universal models of \( q_H \) (viewed as an ABox) and of \( A \) under the ontology \( \mathcal{O} \), and to then take their direct product. \( ^1 \) If we interpret ‘universal model’ as meaning homomorphism universal,\( ^2 \) then such models are infinite and thus the described construction cannot be used in a learning algorithm. But homomorphism universality is not strictly required to obtain a \( \mathcal{Q} \)-LGG when \( \mathcal{Q} \) is not the class of all CQs. We may then use \( \mathcal{Q} \)-universal models which only require that for every query \( q \) in the target query language \( \mathcal{Q} \), the answers to \( q \) on \( A \) under \( \mathcal{O} \) coincide with the answers to \( q \) on the universal model. Sometimes, it is possible to construct finite \( \mathcal{Q} \)-universal models, an approach that has been used successfully to learn a restricted form of CQs in the presence of \( \mathcal{E}\mathcal{L} \) ontologies [22]. To learn ELIQs under \( DL\text{-}Lite \) ontologies, however, this approach fails since finite ELIQ-universal models are not guaranteed to exist (even for non-branching ELIQs).

**Example 1.** Let \( \mathcal{O} = \{ \top \subseteq_\mathcal{O} \exists r . \top \} \) and \( A = \{ A(a) \} \). The homomorphism-universal model of \( A \) and \( \mathcal{O} \) is \( A \) extended with an infinite \( r \)-path \( r(a, a_1), r(a_1, a_2), \ldots \). Any ELIQ-universal model also needs such a path, and thus the only chance to obtain a finite ELIQ-universal model is to reuse individuals on the path. But such a model cannot be ELIQ-universal: if \( a_n = a_m \), with \( n < m \), then \( a \) is an answer to the ELIQ \( \exists r^m . \exists (r^*)^n . A \) on the universal model, but not on \( A \) under \( \mathcal{O} \).

Of course, there could potentially be ways to construct a \( \mathcal{Q} \)-LGG other than taking the direct product of \( \mathcal{Q} \)-universal models. In the presence of \( DL\text{-}Lite \) ontologies, though, the \( \mathcal{Q} \)-LGG is not guaranteed to exist when \( \mathcal{Q} \) is the class of CQs or the class of non-branching ELIQs extended with reflexive role atoms. For non-extended ELIQs, the existence of \( \mathcal{Q} \)-LGGs remains open.

**Example 2.** Let \( \mathcal{O} = \{ \exists r^- . \top \subseteq_\mathcal{O} \exists r . \top, \exists r^- . \top \subseteq_\mathcal{O} \exists s . \top \} \). Consider the unary CQs

\[
p(x) = \exists y \exists z \, r(x, x) \land s(x, y) \land s(z, y) \land r(z, z) \land A(z) \quad \text{and} \quad q(x) = \exists y \, A(x) \land r(x, y).
\]

We claim that no ELIQ-LGG of \( p \) and \( q \) exists, and thus also no \( \mathcal{Q} \)-LGG for any query class \( \mathcal{Q} \) that contains all ELIQs. To see this, assume that the CQ \( \hat{q}(x) \) is an ELIQ-LGG of \( p \) and \( q \), and consider all ELIQs of the form \( q_{n,m} = \exists r^m . \exists s . s^* . (r^-)^m . A \) with \( n, m \geq 1 \). It is easy to see that \( p \subseteq_\mathcal{O} q_{n,m} \) and \( q \subseteq_\mathcal{O} q_{n,m} \) if and only if \( n = m \), thus \( \hat{q} \subseteq_\mathcal{O} q_{n,m} \) if and only if \( n = m \). For all \( i \geq 1 \), take a homomorphism \( h_i \) from \( q_{i,i} \) to the homomorphism universal model \( \mathcal{U} \) of \( \hat{q} \) (viewed as an ABox) and \( \mathcal{O} \); this model is defined in detail in Section 2. If for some \( i \), \( h_i \) maps two distinct variables in the ‘\( \exists r^+ \)’ prefix of \( q_{i,i} \) to the same element of \( \mathcal{U} \), then an easy pumping argument shows that \( \hat{q} \subseteq_\mathcal{O} q_{j,i} \) for some \( j > i \), a contradiction. Otherwise, there is some \( i \geq 1 \) such that \( h_i \) chooses as the \( s \)-successor required by the ‘\( \exists s^* \)’ infix in \( q_{i,i} \) an element of \( \mathcal{U} \) that was

\( ^1 \)This may yield a CQ that does not fall within \( \mathcal{Q} \), but there are strategies for the learning algorithm to deal with this.

\( ^2 \)that is, a universal model of an ABox \( A \) and \( \mathcal{O} \) admits a homomorphism into every model of \( A \) and \( \mathcal{O} \).
generated by an existential quantifier, that is, it is in the tree-shaped ’anonymous’ part of \( \mathcal{U} \). Since \( q_{i,1} \) is rooted, the \( h_i \)-homomorphic image of the ‘\( \exists \)’ prefix of \( q_{i,1} \) enters the anonymous part from the same non-anonymous element \( y \) where it also leaves it to eventually reach an element that satisfies \( A \) (there are no such elements in the anonymous part). Thus \( y \) is reachable in \( \hat{q} \) from \( x \) along an \( r \)-path and \( x \) reaches an instance of \( A \) along an \( r \)-path, which means that \( \hat{q} \subseteq_\mathcal{O} \exists r^k A \) for some \( k \geq 1 \). But this contradicts \( p \subseteq_\mathcal{O} \hat{q} \).

In this paper, we propose to replace \( \mathcal{Q} \)-LGGs by a more liberal construction, which still achieves a generalization of the hypothesis, though not necessarily of the counterexample. In fact, we use the counterexample only as a guide to identify in polynomial time a generalization of the hypothesis that is contained in the target query. In contrast to the construction of \( \mathcal{Q} \)-LGGs via products, our construction is asymmetric in that it treats the hypothesis differently from concept name or extension of \( \mathcal{O} \)-DL-Lite or \( \mathcal{C} \)-DL-Lite. Note that every \( \mathcal{R} \) concept inclusions (CIs) \( \exists \mathcal{R} \), \( \mathcal{R} \) roles. In a functionality assertions and \( \mathcal{B} \) concept disjointness constraints \( \mathcal{R} \) when \( \mathcal{I} \), then \( \mathcal{R} \) DL-Lite ontologies. For the type of learning algorithm that we pursue, there is a ’natural ensemble’ of lemmas that one may use to prove correctness and termination in polynomial time \([21, 22]\). Whenever possible, we establish these lemmas in a general version, namely for rooted CQs in place of ELIQs and for \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) ontologies in place of \( \mathcal{D} \mathcal{L} \)-LGGs by a more liberal construction, which still allows \( \mathcal{Q} \)-LGGs to be countably infinite sets of concept, role and individual names. A role \( \mathcal{R} \) is a role name \( r \) or the inverse \( r^- \) of a role name, and \( r^- \) denotes \( r \) when \( \mathcal{R} = r^- \). A basic concept \( B \) is of the form \( \top, A \), or \( \exists \mathcal{R} A \) where \( A \) ranges over \( \mathbb{N}_C \) and \( \mathcal{R} \) over roles. A \( \mathcal{D} \mathcal{L} \)-LITE\textsubscript{horn} ontology is a set of concept inclusions (CIs) \( B_1 \sqcap \cdots \sqcap B_n \subseteq B \), concept disjointness constraints \( B_1 \sqcap \cdots \sqcap B_n \subseteq \bot \), role disjointness constraints \( \mathcal{R}_1 \sqcap \mathcal{R}_2 \subseteq \bot \), and functionality assertions \( \text{func}(\mathcal{R}) \) where \( B_1, B \) range over basic concepts and \( \mathcal{R}_1, \mathcal{R}_2, R \) over roles. In a \( \mathcal{D} \mathcal{L} \)-LITE\textsubscript{horn} ontology, we additionally require that if \( \exists \mathcal{R} \) occurs on the right-hand side of a CI, then \( \text{func}(\mathcal{R}^-) \notin \mathcal{O} \) \([24]\).

An \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) concept is an expression \( C \) that is built according to the rule \( C ::= \top \mid A \mid C \sqcap C \mid \exists \mathcal{R} C \) where \( A \) ranges over concept names and \( \mathcal{R} \) over roles. An \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) ontology \( \mathcal{O} \) is a finite set of concept inclusions (CIs) \( C \subseteq D \), emptiness constraints \( C \subseteq \bot \), role disjointness constraints \( \mathcal{R}_1 \sqcap \mathcal{R}_2 \subseteq \bot \), and functionality assertions \( \text{func}(\mathcal{R}) \) where \( C, D \) range over \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) concepts and \( \mathcal{R}_1, \mathcal{R}_2, R \) over roles. The basic concept \( \exists \mathcal{R} \) is a different way to write the \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) concept \( \exists \mathcal{R} \top \). Note that every \( \mathcal{D} \mathcal{L} \)-LITE\textsubscript{horn} ontology is an \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) ontology. An \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) ontology (and thus also a \( \mathcal{D} \mathcal{L} \)-LITE\textsubscript{horn} ontology) is in normal form if all concept inclusions in it are of the form \( A \subseteq C \) or \( C \subseteq A \), where \( A \) is a concept name. Every \( \mathcal{D} \mathcal{L} \)-LITE\textsubscript{horn} ontology \( \mathcal{O} \) can be transformed in polynomial time into a \( \mathcal{D} \mathcal{L} \)-LITE\textsubscript{horn} ontology \( \mathcal{O}' \) in normal form such that \( \mathcal{O}' \) is a conservative extension of \( \mathcal{O} \), and the same is true for \( \mathcal{E} \mathcal{L} \mathcal{I} \mathcal{F} \) ontologies.

An ABox \( \mathcal{A} \) is a finite set of concept assertions \( A(a) \) and role assertions \( \mathcal{R}(a, b) \) with \( A \) a concept name or \( \top, \mathcal{R} \) a role name, and \( a, b \) individual names. We use \( \text{ind}(\mathcal{A}) \) to denote the
set of individual names used in $\mathcal{A}$. We admit concept assertions $\top(a)$ in order to represent interpretations and ABoxes in a uniform way.

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 $\mathcal{I}$ and set $A^\mathcal{I} = \{ a \mid A(a) \in \mathcal{I} \}$ for all $A \in N_C$ and $C^\mathcal{I}$ for compound concepts $C$ in the usual way [28]. Note that every ABox is a finite interpretation and, vice versa, every finite interpretation is an ABox. An interpretation $\mathcal{I}$ satisfies a concept inclusion $C \subseteq D$ if $\mathcal{I}|\mathcal{C}$ satisfies the concept inclusion $C \subseteq D$ and $\mathcal{I}|\mathcal{A}$ satisfies $\mathcal{A}$ if $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{I}|\mathcal{O}$. An interpretation is a model of an $\mathcal{ELIF}$ ontology or an ABox if it satisfies all concept inclusions, constraints, and assertions in it. We write $\mathcal{O} \models C \subseteq D$ if every model of the ontology $\mathcal{O}$ satisfies the concept inclusion $C \subseteq D$ and $\mathcal{A}, \mathcal{O} \models B(a)$ every model of $\mathcal{A}$ and $\mathcal{O}$ satisfies the concept assertion $B(a)$. An ABox $\mathcal{A}$ is satisfiable w.r.t. an $\mathcal{ELIF}$ ontology $\mathcal{O}$ if $\mathcal{A}$ and $\mathcal{O}$ have a common model. We use $\mathcal{I}_1 \times \mathcal{I}_2$ to denote the direct product of two interpretations $\mathcal{I}_1, \mathcal{I}_2$.

A signature is a set of concept and role names, uniformly referred to as symbols. For a syntactic object $\mathcal{O}$ such as an ontology, we use $\text{sig}(\mathcal{O})$ to denote the symbols used in $\mathcal{O}$ and $||\mathcal{O}||$ to denote the size of $\mathcal{O}$, that is, the length of a representation of $\mathcal{O}$ as a word in a suitable alphabet.

**Queries.** Every $\mathcal{ELI}$ concept $C$ can be viewed as an $\mathcal{ELI}$ query (ELIQ). An individual $a \in \text{ind}(\mathcal{A})$ 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. A CQ takes the form $q(x) = \exists y \phi(x, y)$ with $\phi$ a conjunction of concept atoms $A(x)$ and role atoms $r(x, y)$ where $A \in N_C$ and $r \in N_R$. We call the variables in $\bar{x}$ answer variables. The arity of $q$ is the length $|\bar{x}|$ of $\bar{x}$, and a query is Boolean if it has arity 0. We use $\text{var}(q)$ to denote the set of variables that occur in $q$. We may view $q$ as a set of atoms whenever convenient and may write $r^-(x, y)$ in place of $r(x, y)$. A CQ is rooted if in its Gaifman graph $G_q = (\text{var}(q), \{ \{y, z\} \mid r(y, z) \in q\})$ every variable is reachable from some answer variable. It is well-known that ELIQs are in 1-to-1 correspondence with rooted, unary CQs whose Gaifman graph is a tree and that contain no self-loops and multi-edges. 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. ontology $\mathcal{O}$ if $\mathcal{A}_q$. For any CQ $q$ and set $U \subseteq \text{var}(q)$, $q|_U$ is the restriction of $q$ to all atoms that only contain variables in $U$.

The semantics of CQs is given in terms of homomorphisms as usual. As for ELIQs, we will write $\mathcal{A}, \mathcal{O} \models q(\bar{a})$ if the tuple $\bar{a}$ is an answer to $q(\bar{x})$ on $\mathcal{A}$ w.r.t. $\mathcal{O}$. For CQs $q_1$ and $q_2$ and an $\mathcal{ELIF}$ 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 $\bar{a}$ from $\text{ind}(\mathcal{A})$, $\mathcal{A}, \mathcal{O} \models q_1(\bar{a})$ implies $\mathcal{A}, \mathcal{O} \models q_2(\bar{a})$. 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 $q_2 \subseteq \mathcal{O} q_1$.

**Universal Model.** Query answering and query containment w.r.t. $\mathcal{DL-Lite}_{horn}^F$ ontologies can be conveniently characterized using universal models. Let $\mathcal{O}$ be an $\mathcal{DL-Lite}_{horn}^F$ ontology in normal form and $\mathcal{A}$ an ABox that is satisfiable w.r.t. $\mathcal{O}$. For a set $M$ of concept names, we write $\prod M$ as a shorthand for $\prod_{A \in M} A$. For $a \in \text{ind}(\mathcal{A})$, $M, M'$ sets of concept names, and $R$ a role, we write $a \sim_R^{A, \mathcal{O}, M}$ if $\mathcal{A}, \mathcal{O} \models \exists R. \prod M(a)$ and $M$ is maximal with this condition. We write
The set \( T \) of all traces for \( A \) and \( \mathcal{O} \) forms the domain of the universal model \( \mathcal{U}_{A,\mathcal{O}} \), defined as

\[
\mathcal{U}_{A,\mathcal{O}} = A \cup \{ A(a) \mid A, \mathcal{O} \models A(a) \} \cup \{ A(tRM) \mid tRM \in T \text{ and } A \in M \} \cup \{ R(t, tRM) \mid tRM \in T \}.
\]

For a CQ \( q \), we usually write \( \mathcal{U}_{A,\mathcal{O}} \) instead of \( \mathcal{U}_{A_q,\mathcal{O}} \). The following property of \( \mathcal{U}_{A,\mathcal{O}} \) is crucial for our technical development.

**Observation 3.** Let \( \mathcal{O} \) be a DL-Lite\(_{\text{horn}}\) ontology in normal form, \( A \) an ABox, and \( \mathcal{I} = \mathcal{U}_{A,\mathcal{O}} \setminus A \). Then for every role \( R, R' \) is a partial function.

**\( \mathcal{O} \)-saturatedness and \( \mathcal{O} \)-minimality.** Let \( \mathcal{O} \) be an ELTF ontology. A CQ \( q \) is \( \mathcal{O} \)-minimal if there is no \( U \subseteq \text{var}(q) \) such that \( q \equiv_U q' \). A CQ \( q \) is \( \mathcal{O} \)-saturated if \( A_q, \mathcal{O} \models A(y) \) implies \( A(y) \in q \) for all \( y \in \text{var}(q) \) and \( A \in \mathcal{N}_C \). Every CQ (or ELIQ) can be converted into an equivalent \( \mathcal{O} \)-saturated one in polynomial time when an oracle for queries of the form "\( A, \mathcal{O} \models A(a) \)" is available. In DL-Lite\(_{\text{horn}}\), such queries can be answered in polynomial time [29] and thus \( \mathcal{O} \)-saturatedness can be established in polynomial time.

**3. Guided Generalizations**

Recall from the introduction that a CQ \( \hat{q} \) is a least general \( \mathcal{O} \)-generalization (Q-LGG) of CQs \( p, q \) under an ontology \( \mathcal{O} \) if \( q \subseteq_\mathcal{O} \hat{q}, p \subseteq_\mathcal{O} \hat{q}, \) and \( \hat{q} \subseteq_\mathcal{O} q' \) for every \( q' \in \mathcal{Q} \) with \( q \subseteq_\mathcal{O} q' \) and \( p \subseteq_\mathcal{O} q' \). We consider the following weakening of Q-LGGs.

**Definition 4.** Let \( \mathcal{O} \) be an ontology and \( \mathcal{Q} \) a class of queries, and let \( p, q \) be CQs with \( p \not\subseteq_\mathcal{O} q \). A CQ \( \hat{q} \) is a \( p \)-guided \( \mathcal{Q} \)-generalization of \( q \) under \( \mathcal{O} \) if the following conditions are satisfied:

1. \( q \subseteq_\mathcal{O} \hat{q} \);
2. \( \hat{q} \not\subseteq_\mathcal{O} q \);
3. \( \hat{q} \subseteq_\mathcal{O} q' \), for every \( q' \in \mathcal{Q} \) with \( q \subseteq_\mathcal{O} q' \) and \( p \subseteq_\mathcal{O} q' \).

Conditions 1 and 3 match the first and the last condition in the definition of a Q-LGG. Intuitively, they mean that \( \hat{q} \) is a generalization of \( q \) (Condition 1) which preserves all common \( \mathcal{Q} \)-consequences of \( p \) and \( q \) (Condition 3). Condition 2 weakens the second condition in the definition of an LGG: instead of requiring \( p \subseteq_\mathcal{O} q \), we only want \( \hat{q} \) to strictly generalize \( q \). In the context of learning, one may view \( p \) as orthogonal knowledge about how to imply the unknown target, and the goal of guided generalization is to incorporate some of that knowledge into \( \hat{q} \).
Thus, in contrast to LGGs, guided generalizations are an asymmetric notion in that the two queries $p$ and $q$ play different roles: $q$ is the query to be generalized and $p$ acts as the guide for doing so. We start with observing that $p$-guided $Q$-generalizations are not uniquely defined.

**Example 5.** Consider $q(x) = A(x) \land B(x) \land C(x)$ and $p(x) = A(x)$. Then both $q_1(x) = A(x)$ and $q_2(x) = A(x) \land B(x)$ are $p$-guided ELIQ-generalizations of $q$ under the empty ontology.

It is not by accident that in Example 5 the ELIQ-LGG of $q$ and $p$ (which is $q_1$) is also a guided ELIQ-generalization. In fact, it is not difficult to show that each $Q$-LGG of two CQs $p$, $q$ under an ontology $O$ is both a $p$-guided $Q$-generalization of $q$ under $O$ and a $q$-guided $Q$-generalization of $p$ under $O$. The subsequent example shows that the converse direction is not true, that is, there are cases where a guided generalization exists, but LGGs do not.

**Example 6.** Consider again queries $p$ and $q$ and the ontology $O$ from Example 2, and recall that there is no CQ-LGG for $p$, $q$. However, the query

$$\tilde{q}(x) = \exists y \exists y' r(x, y) \land r(x, y') \land A(y')$$

is a $p$-guided CQ-generalization of $q$ under $O$. To illustrate the asymmetry of the notion, observe that $\tilde{q}$ is not a $q$-guided CQ-generalization of $p$ under $O$, since it does not satisfy Condition 1.

We now give our main result, namely that guided ELIQ-generalizations of ELIQs under $DL$-Lite$^F_{horn}$ ontologies always exist and can be computed in polynomial time.

**Theorem 7.** Given a $DL$-Lite$^F_{horn}$ ontology $O$ in normal form and ELIQs $p$, $q$ such that $p$, $q$ are satisfiable w.r.t. $O$ and $q$ is $O$-minimal,\(^3\) we can compute in polynomial time a $p$-guided ELIQ-generalization $\tilde{q}$ of $q$ under $O$ such that $\tilde{q}$ is satisfiable w.r.t. $O$.

Let $q(x_1), p(x_2)$ be ELIQs. We construct a $p$-guided ELIQ-generalization $\tilde{q}$ of $q$ under $O$ in three steps as follows. We start with the query $\tilde{q} = (U_q,O \times U_p,O)|_{\{(x_1,x_2)\}}$, that is, the restriction of $U_q,O \times U_p,O$ to variable $(x_1,x_2)$, which will be the answer variable of $\tilde{q}$. This query is then extended by first exhaustively applying rule (A1) below and then applying rule (A2).

(A1) For every $(z,t) \in \text{var}(\tilde{q})$ with $z \in \text{var}(q)$ and $t \in \Delta^U_{q,O}$, every atom $R(z,z')$ in $q$, and every atom $R(t,t') \in U_p,O$, add the atom $R((z,t),(z',t'))$, and all atoms $A'(z',t')$ such that $A'(z') \in U_q,O$ and $A(t') \in U_p,O$.

(A2) For every $(z,t) \in \text{var}(\tilde{q})$ with $z \in \text{var}(q)$ and $t \in \Delta^U_{q,O}$ and every role $R$ such that $z \rightsquigarrow^R_{q,O} M$ for some $M$ and there is no atom of the form $R(z,z')$ in $q$, add the atoms

$$R((z,t),\hat{z}), R(z',\hat{z})$$

with $\hat{z}$ a fresh variable, and add a copy $q'$ of $q$ in which the copy of $z$ is $z'$.

\(^3\)We conjecture that, given an ELIQ, an equivalent $O$-minimal ELIQ can be computed in polynomial time by extending the techniques for answering tree-shaped queries over $DL$-Lite knowledge bases in polynomial time [30] to $DL$-Lite$^F_{horn}$ knowledge bases. For the purpose of this paper the statement in the theorem suffices.
Recall that $\mathcal{U}_{q,\mathcal{O}} \times \mathcal{U}_{p,\mathcal{O}}$, when viewed as an infinitary CQ, may serve as a CQ-LGG of $p$ and $q$. Intuitively, the above construction may be viewed as producing an approximation of this product from below, in the sense that the product may be more general. It is easy to see that after having applied (A1) exhaustively, we have constructed exactly the restriction of the product $\mathcal{U}_{q,\mathcal{O}} \times \mathcal{U}_{p,\mathcal{O}}$ to the elements $(t, t')$ that are reachable from the element $(x_1, x_2)$ and satisfy $t \in \text{var}(q)$. We will show that this is a finite structure and even of polynomial size, which is essentially due to Observation 3 on the shape of universal models for $\text{DL-Lite}_{\text{horn}}^{-}$ ontologies. What is missing is the infinite part of $\mathcal{U}_{q,\mathcal{O}} \times \mathcal{U}_{p,\mathcal{O}}$ determined by elements $(t, t')$ where $t$ is a proper trace, that is, $t$ is not a variable from $q$. (A2) approximates this part by removing traces of $\mathcal{U}_{q,\mathcal{O}}$ (but not of $\mathcal{U}_{p,\mathcal{O}}$) for only one step and then adding copies of $q$ as described. Note that for $\text{DL-Lite}_{\text{horn}}^{-}$ ontologies $\mathcal{O}$, the first step into the traces of $\mathcal{U}_{q,\mathcal{O}}$ is enough to regenerate via $\mathcal{O}$ the entire universal model $\mathcal{U}_{q,\mathcal{O}}$. Also note that for (A2) to produce a query that is satisfiable w.r.t. $\mathcal{O}$, we rely on the restriction to $\text{DL-Lite}_{\text{horn}}^{-}$: the precondition of (A2) implies $\exists R$ appears on the right-hand side of some concept inclusion in $\mathcal{O}$ and thus $R^-$ is not functional.

We demonstrate our construction on two examples that additionally illustrate (1) that (A2) is indeed needed and (2) that the result $\hat{q}$ is not necessarily an ELIQ.

**Example 8.** (1) Consider the ontology $\mathcal{O} = \{ X \subseteq \exists r, \exists r' \subseteq X, \exists r'' \subseteq \exists s \}$ and ELIQs

$$q(x_1) = B(x_1) \land X(x_1) \quad p(x_2) = \exists x' \exists y X(x_2) \land r(x_2, y) \land r'(x', y) \land B(x') \land X(x').$$

Note that $p$ and $q$ are $\mathcal{O}$-saturated. The result of exhaustively applying Step (A1) is $\hat{q}_0(x) = X(x),$ which generalizes $q$, but is too general: the ELIQ

$$q_{\mathcal{T}}(x) = \exists x' \exists y \exists y' \exists z r(x, y) \land s(y, z) \land s(y', z) \land r'(x', y') \land B(x')$$

satisfies $\hat{q}_0 \subseteq q_{\mathcal{T}}$, while $p \subseteq q_{\mathcal{T}}$ and $q \subseteq q_{\mathcal{T}}$. After additionally applying (A2), we obtain

$$\hat{q}(x) = \exists x' \exists y X(x) \land r(x, y) \land r'(x', y) \land B(x') \land X(x'),$$

which is a $p$-guided ELIQ-generalization of $q$ under $\mathcal{O}$.

(2) Consider the following queries $p'$, $q'$ and the empty ontology.

$$p'(x) = \exists y_1 \exists y_2 r(x, y_1) \land r(x, y_2) \land A(y_1) \land B(y_2)$$

$$q'(x) = \exists y \exists z r(x, y) \land r(z, y) \land A(y) \land B(y)$$

The result of (A1) is the direct product $q' \times p'$ of $q'$ and $p'$ which is $\hat{q}'(x) = \exists y_1 \exists y_2 \exists z r(x, y_1) \land r(x, y_2) \land r(z, y_1) \land r(z, y_2) \land A(y_1) \land B(y_2)$, which is not an ELIQ.

4. Exact Learning with Membership and Equivalence Queries

We apply the notion of guided generalizations to show that ELIQs are polynomial time learnable in the presence of $\text{DL-Lite}_{\text{horn}}^{-}$ ontologies using membership and equivalence queries. It is known that both kinds of queries are needed as otherwise polynomial time learnability fails (already without functional roles) [22]. Our learning algorithm follows the scheme detailed in the introduction. The main result is as follows.

\[\text{We have replaced the single answer variable } (x_1, x_2) \text{ with } x \text{ for the sake of readability.}\]
Algorithm 1 Algorithm for learning ELIQs under $DL$-$Lite^-$$_{horn}$ ontologies

<table>
<thead>
<tr>
<th>Input</th>
<th>A $DL$-$Lite^-$$_{horn}$ ontology $\mathcal{O}$ and a unary CQ $q_H^0$ satisfiable w.r.t. $\mathcal{O}$ such that $q_H^0 \subseteq \mathcal{O} q_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>An ELIQ $q_H$ such that $q_H \equiv_\mathcal{O} q_T$</td>
</tr>
</tbody>
</table>

$q_H := \text{extract-minimal-ELIQ}(q_H^0)$

while the equivalence query “$q_H \equiv_\mathcal{O} q_T$?” returns a counterexample $(\mathcal{A}, a)$ do

$q_D := \text{extract-minimal-ELIQ}(q_A)$ where $q_A$ is $\mathcal{A}$ viewed as a CQ with answer variable $a$

$q_H' := \text{a } q_D\text{-guided ELIQ-generalization of }q_H\text{ under }\mathcal{O}$

$q_H := \text{extract-minimal-ELIQ}(q_H')$

end while

return $q_H$

Theorem 9. ELIQs are polynomial time learnable under $DL$-$Lite^-$$_{horn}$ ontologies using membership and equivalence queries.

To prove the theorem, it suffices to consider ontologies in normal form:

Lemma 10. If ELIQs are polynomial time learnable under $DL$-$Lite^-$$_{horn}$ ontologies in normal form using membership and equivalence queries, the same is true for unrestricted $DL$-$Lite^-$$_{horn}$ ontologies.

Our learning algorithm is listed in Algorithm 1. It takes as input a $DL$-$Lite^-$$_{horn}$ ontology $\mathcal{O}$ in normal form and a seed query $q_H^0$ with $q_H^0 \subseteq \mathcal{O} q_T$. A seed query can be obtained in several ways, depending on the type of disjointness constraints present in $\mathcal{O}$; we refer to [24] for details. As explained in the introduction, the algorithm starts with the seed query and constructs a sequence of increasingly more general hypothesis queries. In each round, the learner asks whether the current hypothesis $q_H$ is the target using an equivalence query. If not, they use the counterexample provided by the oracle as a guide to generalize $q_H$ via the construction from the proof of Theorem 7. Since both the input to that construction and the queries posed as equivalence queries must be ELIQs, the algorithm relies on the subroutine extract-minimal-ELIQ to generalize a CQ $q$ with $q \subseteq \mathcal{O} q_T$ into an ELIQ $q'$ with $q' \subseteq \mathcal{O} q_T$ using membership queries. In order to attain polynomial running time, extract-minimal-ELIQ additionally ensures a strong minimality condition on $q'$, namely that it is $(q_T, \mathcal{O})$-minimal, which means that there is no $\mathcal{U} \subset \text{var}(q')$ with $q'|\mathcal{U} \subseteq \mathcal{O} q_T$. Importantly, a $(q_T, \mathcal{O})$-minimal query may have at most as many variables as $q_T$ (provided that it is $\mathcal{O}$-saturated, a condition that we shall maintain at all times), and it is $\mathcal{O}$-minimal. We next detail the extract-minimal-ELIQ subroutine.

The extract-minimal-ELIQ subroutine takes as input a unary CQ $q$ that satisfies $q \subseteq \mathcal{O} q_T$. It computes an ELIQ $q'$ with $q \subseteq \mathcal{O} q'$ by repeatedly attaining $(q_T, \mathcal{O})$-minimality and increasing the length of cycles in $q$. 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. Now, extract-minimal-ELIQ computes the $\mathcal{O}$-saturation $p$ of $q$ and then modifies $p$ by exhaustively applying the following two rules:

Drop variable. Choose a variable $y \in \text{var}(p)$ and let $p' = p|_{\text{var}(p) \setminus \{y\}}$. If the response to the membership query $A_p', \mathcal{O} \models q_T(x)$ is positive, continue with $p'$ in place of $p$.

Double cycle. Choose a role atom $r(x, y) \in p$ that is part of a cycle. Then add a disjoint copy
Let $q$ be a unary CQ with $q \subseteq_{\mathcal{O}} q_T$ that is satisfiable w.r.t. $\mathcal{O}$. Then, extract-minimal-ELIQ($q$) terminates in time polynomial in $|\mathcal{O}| + |q| + |q_T|$ and returns an ELIQ $q'$ that is $\mathcal{O}$-saturated, $(q_T, \mathcal{O})$-minimal, and satisfies $q \subseteq_{\mathcal{O}} q' \subseteq_{\mathcal{O}} q_T$.

To show termination and correctness of our algorithm, we first formalize the notion of a 'sequence of increasingly general hypotheses which are all contained in $q_T$', which is underlying the general scheme described in the introduction.

**Definition 12.** Let $q_T$ be a CQ and $\mathcal{O}$ an ontology. A sequence $q_1, q_2, \ldots$ of CQs is a generalization sequence towards $q_T$ under $\mathcal{O}$ if for all $i \geq 0$, $q_i \subseteq_{\mathcal{O}} q_{i+1} \subseteq_{\mathcal{O}} q_T$, and $\text{sig}(q_i) \subseteq \text{sig}(\mathcal{O})$.

Let $q_1, q_2, \ldots$ be the sequence of ELIQs that are assigned to $q_H$ during the run of the algorithm. We show inductively that $q_1, q_2, \ldots$ is a generalization sequence towards the target query $q_T$ under $\mathcal{O}$. For the base case, note that extract-minimal-ELIQ ($q_H^0$) computes an initial $q_H$ with $q_H^0 \subseteq_{\mathcal{O}} q_H \subseteq_{\mathcal{O}} q_T$. For the inductive step, let $(\mathcal{A}, \sigma)$ be a counterexample provided by the oracle to the equivalence query “$q_H \equiv_{\mathcal{O}} q_T$?”. We may assume that $\mathcal{A}$ uses only symbols from $\mathcal{O}$ (we can simply drop all assertions mentioning other symbols). Since $q_H \subseteq_{\mathcal{O}} q_T$, the counterexample is positive and thus $q_A \subseteq_{\mathcal{O}} q_H$. The subroutine extract-minimal-ELIQ generalizes $q_A$ into a query $q_D$, hence $q_D \not\subseteq_{\mathcal{O}} q_H$. Since $q_H^0$ is a $q_D$-guided ELIQ-generalization of $q_H$, we have $q_H \subseteq q_H^0$ (Condition 1 of Definition 4), $q_H^0 \subseteq q_H$ (Condition 2), and $q_H^0 \subseteq_{\mathcal{O}} q_T$ (Condition 3). It remains to note that extract-minimal-ELIQ preserves these conditions.

It has been observed that already for ELIQs that do not use inverse roles and under the empty ontology, there is no elementary bound on the length of generalization sequences towards a given query $q_T$ [31]. However, since Lemma 11 guarantees that all $q_i$ are $(q_T, \mathcal{O})$-minimal and $\mathcal{O}$-saturated, the next theorem implies that only polynomially many hypotheses are produced.

**Theorem 13.** Let $q_T$ be a rooted CQ and $\mathcal{O}$ an ELIQ ontology in normal form, and let $q_1, q_2, \ldots$ be a generalization sequence towards $q_T$ under $\mathcal{O}$ such that $q_1$ is satisfiable w.r.t. $\mathcal{O}$. If all $q_i$ are $(q_T, \mathcal{O})$-minimal and $\mathcal{O}$-saturated, then the sequence has length at most $|\text{var}(q_T)|^3 \cdot |\text{sig}(\mathcal{O})|$.

It remains to show that the extract-minimal-ELIQ subroutine terminates after polynomially many steps. For this, consider the sequence $p_1, p_2, \ldots$ of queries that Double cycle is applied to during a run of extract-minimal-ELIQ. All these queries are $\mathcal{O}$-saturated. By the preference imposed on rule application, they are also $(q_T, \mathcal{O})$-minimal. Since an application of Drop Variable decreases the size of the query, there are at most polynomially many such applications between $p_i$ and $p_{i+1}$. Thus, it suffices to show the following lemma and apply Theorem 13.

**Lemma 14.** The sequence $p_1, p_2, \ldots$ is a generalization sequence towards $q_T$ under $\mathcal{O}$.
We conclude the section with some comments regarding the (limits of) generality of the central Theorem 13. It has been shown that Theorem 13 holds for unrestricted CQs when one considers the restriction $\mathcal{E}\mathcal{L}$ of $\mathcal{EL}$ as ontology language [22] and we conjecture the same to be true also for many DL-Lite dialects, e.g., DL-Lite$^{\text{F horn}}$. However, the extension to unrestricted, that is, possibly non-rooted, CQs is not possible for $\mathcal{EL}$-ontologies. The subsequent example illustrates that it fails already for Boolean CQs with a single variable.

Example 15. Let $X_i, \overline{X}_i$ for $1 \leq i \leq n$ be concept names and $r$ a role name. Let $\mathcal{O}$ be an $\mathcal{EL}$ ontology that contains the following concept inclusions, for all $i$ with $1 \leq i \leq n$:

$$
\begin{align*}
X_i &\subseteq \exists r. \top \\
\exists^\ast (X_0 \sqcap \cdots \sqcap X_{i-1} \sqcap \overline{X}_i) &\subseteq X_i \\
\exists^\ast (X_i \sqcap \overline{X}_i) &\subseteq X_i \\
\exists^\ast X_i \sqcap \overline{X}_j &\subseteq X_i \\
\exists^\ast X_i \sqcap X_i &\subseteq X_i \\
\end{align*}
$$

Each subset of $\{X_i, \overline{X}_i \mid 1 \leq i \leq n\}$ containing exactly one of $X_i, \overline{X}_i$ for each $i$ represents a number between 0 and $2^n - 1$ in an obvious way. Let $q_i$ be the Boolean CQ that corresponds to number $i$. Clearly, all $q_i$ are $\mathcal{O}$-saturated and $(q_{2^n-1}, \mathcal{O})$-minimal. The sequence $q_1, q_2, \ldots, q_{2^n-1}$ is a generalization sequence towards $q_{2^n-1}$ under $\mathcal{O}$, but its length is exponential in $n$.

5. Conclusions and Future Work

We have introduced a new form of generalizations and proved its applicability in the context of exact learning of ELIQs in the presence of DL-Lite$^{\text{F horn}}$ ontologies. We believe it is worth investigating the new notion of guided generalizations more thoroughly. On the one hand, there are basic open questions such as whether there always exists a least general or most general $p$-guided ELIQ-generalization of $q$ for all ELIQs $p, q$. On the other hand, we would like to understand whether Theorem 7 holds for other relevant combinations of query class and ontology language, e.g., ELIQs and DL-Lite$^{\text{F horn}}$, CQs and any DL-Lite dialect, ELIQs and DL-Lite$^H$, that is, DL-Lite with role hierarchies. The latter will be challenging since there are no universal models that satisfy Observation 3. We note that Theorem 7 does not extend to ELIQs and $\mathcal{EL}$ ontologies: Our results imply that if we could compute in polynomial time using an oracle for queries of the form “$\mathcal{A}, \mathcal{O} \models A(a)$”, guided ELIQ-generalization of ELIQs under $\mathcal{EL}$ ontologies, then ELIQs would be polynomial query learnable under $\mathcal{EL}$ ontologies which is known not to be the case [22]. In cases where guided generalizations are not guaranteed to exist, it would be interesting to study the induced existence and verification decision problems [7]. Finally, we are wondering whether guided generalizations have other applications, for example in learning from labeled data examples.

As we have shown, positive answers to (some of) these questions would directly lead to polynomial time learnability results. Here, interesting open (and challenging) questions are whether CQs are polynomial time learnable in the presence of DL-Lite$^{\text{F horn}}$ (or even DL-Lite) ontologies, and whether ELIQs are efficiently learnable under DL-Lite$^{\text{F horn}}$ or DL-Lite$^H$ ontologies.
References

[21] B. ten Cate, V. Dalmau, Conjunctive queries: Unique characterizations and exact learn-


A. Additional Preliminaries

A.1. Direct Product

The direct product of interpretations \( I_1 \) and \( I_2 \) is the interpretation \( I_1 \times I_2 \) defined as

\[
\{ \top(a_1, a_2) \mid a_i \in \Delta_{I_i} \text{ for } i \in \{1, 2\} \} \cup \\
\{ A(a_1, a_2) \mid A(a_1) \in I_1 \text{ for } i \in \{1, 2\} \} \cup \\
\{ r((a_1, a_2), (b_1, b_2)) \mid r(a_i, b_i) \in I_i \text{ for } i \in \{1, 2\} \}.
\]

A.2. Semantics of Conjunctive Queries

The semantics of CQs is given in terms of homomorphisms as usual. A homomorphism \( h \) from interpretation \( I \) to interpretation \( I' \) is a mapping from \( \Delta_I \) to \( \Delta_{I'} \) such that \( A(d) \in I \) implies \( A(h(d)) \in I' \) and \( r(d, e) \in I \) implies \( r(h(d), h(e)) \in I' \). We use \( \operatorname{img}(h) \) to denote the set \( \{ h(d) \mid d \in \Delta_I \} \). For tuples \( \bar{d} \) over \( \Delta_I \), \( i \in \{1, 2\} \), we write \( (I_i, \bar{d}_i) \rightarrow (I_2, \bar{d}_2) \) if there is a homomorphism \( h \) from \( I \) to \( I' \) with \( h(d_i) = d_2 \). With a homomorphism from a CQ \( q \) to an interpretation \( I \), we mean a homomorphism from \( A_q \) to \( I \).

Let \( q(\bar{x}) \) be a CQ, \( I \) an interpretation, and let \( \bar{d} \) be a tuple over \( \Delta_I \) (of the same arity as \( \bar{x} \)). We write \( q(\bar{x}) \rightarrow (I, \bar{d}) \) if there is a homomorphism \( h \) from \( q \) to \( I \) with \( h(\bar{x}) = \bar{d} \), and call \( \bar{d} \) an answer to \( q \) in \( I \), written \( I \models q(\bar{d}) \), if \( q(\bar{x}) \rightarrow (I, \bar{d}) \). Let \( O \) be an ELIF ontology and \( A \) an ABox. A tuple of individuals \( \bar{a} \) from \( \text{ind}(A) \) is an answer to \( q \) on \( A \) w.r.t. \( O \), written \( A, O \models q(\bar{a}) \), if \( \bar{a} \) is an answer to \( q \) in every model of \( O \) and \( A \).

A.3. Universal model

For the sake of completeness, we provide the definition of traces in the case of ELIF ontologies. There, a trace for \( A \) and \( O \) is a sequence \( t = aR_1M_1R_2M_2 \ldots R_nM_n \), \( n \geq 0 \) where \( a \in \text{ind}(A) \), \( R_1, \ldots, R_n \) are roles that occur in \( O \), and \( M_1, \ldots, M_n \) are sets of concept names that occur in \( O \), such that

(i') \( a \rightarrow_{A, O}^1 R_1 M_1 \) and if \( \text{func}(R_1) \in O \), then there is no \( b \in \text{ind}(A) \) with \( R_1(a, b) \in A \),

(ii') \( M_i \rightarrow_{O}^1 M_{i+1} \) and if \( \text{func}(R_i^-) \in O \), then \( R_{i+1} \neq R_i^- \), for \( 1 \leq i < n \).

Let \( T \) denote the set of all traces for \( A \) and \( O \). We always assume that \( T \) depends on the ontology language: if \( O \) is in \( DL-Lite_{\text{tom}}^e \), then traces are defined as in the main body of the paper, otherwise, they are defined as above.

Then the universal model \( U_{A, O} \) of \( A \) and \( O \) is

\[
U_{A, O} = A \cup \{ A(a) \mid A, O \models A(a) \} \cup \{ A(tRM) \mid tRM \in T \text{ and } A \in M \} \cup \\
\{ R(t, tRM) \mid tRM \in T \}.
\]

The following lemma states the two most important properties of universal models. Its proof is standard, see, e.g., [32]. Note that the maximality conditions of ‘\( \rightarrow \)’ are important to ensure that all functionality assertions in \( O \) are satisfied.
Lemma 16. For any $\mathcal{ELIF}$ ontology $\mathcal{O}$ in normal form and any ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{O}$,

1. $\mathcal{U}_{\mathcal{A},\mathcal{O}}$ is a model of $\mathcal{A}$ and $\mathcal{O}$;
2. $\mathcal{A}, \mathcal{O} \models q(\bar{a})$ iff $\mathcal{U}_{\mathcal{A},\mathcal{O}} \models q(\bar{a})$, for all CQs $q(\bar{x})$ and all $\bar{a} \in \text{ind}(\mathcal{A})^{\mathcal{O}}$.

Universal models also play an important role for query containment. In fact, Point 2 of Lemma 16 implies the following characterization of query containment.

Lemma 17. Let $\mathcal{O}$ be an $\mathcal{ELIF}$ ontology in normal form and $p(\bar{x}), q(\bar{x})$ CQs such that $p$ is satisfiable w.r.t. $\mathcal{O}$. Then, $p \subseteq \mathcal{O}$ q iff $q(\bar{x}) \rightarrow (\mathcal{U}_{p,\mathcal{O}}, \bar{x})$.

The following lemmas show the connections between an ontology in normal form, $\mathcal{O}$-saturated queries and $\mathcal{O}$-minimal queries.

Lemma 18. Let $\mathcal{O}$ be an $\mathcal{ELIF}$ ontology in normal form, let $q(\bar{x})$ be a rooted CQ, and let $p(\bar{y})$ be an $\mathcal{O}$-saturated CQ. Then every homomorphism $h : q(\bar{x}) \rightarrow (\mathcal{U}_{q,\mathcal{O}}, \bar{y})$ is also a homomorphism $q(\bar{x}) \rightarrow (\mathcal{U}_{p,\mathcal{O}}, \bar{y})$ where $U = \text{img}(h) \cap \text{var}(p)$.

Proof. Let $h$ be a homomorphism $h : q(\bar{x}) \rightarrow (\mathcal{U}_{p,\mathcal{O}}, \bar{y})$, and let $U = \text{img}(h) \cap \text{var}(p)$.

Since $p$ is $\mathcal{O}$-saturated, $A(y) \in \mathcal{U}_{p,\mathcal{O}}$ if and only if $A(y) \in \mathcal{U}_{p|U,\mathcal{O}}$, for all concept names $A$ and every $y \in U$. Since $\mathcal{O}$ is in normal form, $\mathcal{A}_{p, \mathcal{O}} \models \exists R. \bigcap M(y)$ if and only if $\mathcal{A}_{p|U, \mathcal{O}} \models \exists R. \bigcap M(y)$ for all sets of concept names $M$ and $y \in U$. By definition of the universal model, the tree generated below any variable $y \in U$ in $\mathcal{U}_{p|U,\mathcal{O}}$ is identical to the tree generated below $y$ in $\mathcal{U}_{p,\mathcal{O}}$.

Since $q$ is rooted, $h$ maps all variables $x \in \text{var}(q)$ either to $U$ or to the tree below some element in $U$. Finally, because $p|U$ and $p$ also agree on the role atoms, $h$ is a homomorphism $h : q(\bar{x}) \rightarrow (\mathcal{U}_{q|U,\mathcal{O}}, \bar{y})$ as required. \[\square\]

Lemma 19. Let $\mathcal{O}$ be an $\mathcal{ELIF}$ ontology in normal form, and let $q(\bar{x})$ be an $\mathcal{O}$-minimal and $\mathcal{O}$-saturated rooted CQ that is satisfiable w.r.t. $\mathcal{O}$. Then every homomorphism $h : q(\bar{x}) \rightarrow (\mathcal{U}_{q,\mathcal{O}}, \bar{x})$ is a homomorphism $q(\bar{x}) \rightarrow (q, \bar{x})$.

Proof. Suppose the contrary, that is, let $h$ be a homomorphism $h : q(\bar{x}) \rightarrow (\mathcal{U}_{q,\mathcal{O}}, \bar{x})$ which maps some variable $x_0$ in $\text{var}(q)$ to the anonymous part of $\mathcal{U}_{q,\mathcal{O}}$. Let $U$ be the image of $h$ restricted to $\text{var}(q)$, that is, $U = \text{img}(h) \cap \text{var}(q)$. Since $x_0$ is mapped to a trace of length $> 1$, $U \subseteq \text{var}(q)$, and by Lemma 18 $h$ is a homomorphism from $q(\bar{x}) \rightarrow (\mathcal{U}_{q|U,\mathcal{O}}, \bar{x})$. Now, Lemma 17 yields $q|U \subseteq q$ in contradiction to $\mathcal{O}$-minimality of $q$. \[\square\]

The following lemma states that any homomorphism from a CQ $q$ to some universal model $\mathcal{U}_{\mathcal{A},\mathcal{O}}$ can be extended to a homomorphism from $\mathcal{U}_{q,\mathcal{O}}$ to $\mathcal{U}_{\mathcal{A},\mathcal{O}}$.

Lemma 20. Let $\mathcal{O}$ be an $\mathcal{ELIF}$ ontology, $\mathcal{A}$ an ABox, and $q(\bar{x})$ a CQ, such that $\mathcal{A}$ and $q$ are both satisfiable w.r.t. $\mathcal{O}$. Every homomorphism $h : q(\bar{x}) \rightarrow (\mathcal{U}_{\mathcal{A},\mathcal{O}}, \bar{a})$ for some tuple $\bar{a}$ over $\text{ind}(\mathcal{A})$ can be extended to a homomorphism $h' : (\mathcal{U}_{q,\mathcal{O}}, \bar{x}) \rightarrow (\mathcal{U}_{\mathcal{A},\mathcal{O}}, \bar{a})$.\[\square\]
B. Proofs for Section 3

Lemma 21. The computation of \( \hat{q} \) terminates after polynomially many steps.

Proof. The initial \( \hat{q} \) can clearly be computed in polynomial time. For the analysis of (A1), observe that, by definition, (A1) computes an initial fragment of the product \( \mathcal{U}_q \times \mathcal{U}_p, \mathcal{O} \). Thus, it creates at most \(|q \times p|\) facts over variables \((x, y)\) with \(x \in \text{var}(q)\) and \(y \in \text{var}(p)\). The remaining rule applications can be structured into labeled trees \( T_{xy} \), for each \((x, y) \in \text{var}(q) \times \text{var}(p)\), as follows:

- the root \( \varepsilon \) of \( T_{xy} \) is labeled with \( \lambda(\varepsilon) = (x, y) \);
- if some node \( n \) is labeled with \( \lambda(n) = (z, t) \) and (A1) is applicable to some \( R(z, z') \) and \( R(t, t') \), then \( n \) has a successor \( n' \) with \( \lambda(n') = (z', t') \); we additionally associate with \( n' \) another label \( \rho(n') = R(z, z') \).

Clearly, it suffices to bound the sizes of each tree \( T_{xy} \) by a polynomial in the input, which is established in the following claim.

Claim. There are no two nodes \( n_1 \neq n_2 \) in \( T_{xy} \) such that \( \lambda(n_1) = (z_1, t_1) \), \( \lambda(n_2) = (z_2, t_2) \), and \( z_1 = z_2 \).

Proof of the claim. The proof is by contradiction. Suppose there are \( n_1 \neq n_2 \) in \( T_{xy} \) such that \( \lambda(n_1) = (z_1, t_1) \), \( \lambda(n_2) = (z_2, t_2) \), and \( z_1 = z_2 \). Consider the unique shortest path from \( n_1 \) to \( n_2 \) in \( T_{xy} \) and let \( n \) be the node closest to the root on this path, that is, the path \( w_0 \ldots w_k \) from \( n_1 \) to \( n \) “goes up” in the tree and the path \( v_0 \ldots v_m \) from \( n \) to \( n_2 \) “goes down”.

Consider the following sequence \( \alpha_0, \ldots, \alpha_{k+m-1} \) of facts:

(a) for \( 0 \leq i < k \), let \( \alpha_i \) be the fact \( R^-(z, z') \) when \( \rho(w_i) = R(z', z) \);
(b) for \( 0 < i \leq m \), let \( \alpha_{k+i-1} = \rho(v_i) \).

By definition of (A1) and the resulting definition of \( T_{xy} \), the sequence \( \alpha_0, \ldots, \alpha_{k+m-1} \) is a path from \( z_1 \) to \( z_2 \) in \( q \). Since \( z_1 = z_2 \) and \( q \) is acyclic, there has to be some \( i \) such that \( \alpha_i = R(z, z') \) and \( \alpha_{i+1} = R^-(z', z) \), for some role \( R \). We distinguish cases on where \( \alpha_i \) and \( \alpha_{i+1} \) were defined (in (a) or in (b) above).

Suppose first that both were defined in (a) and consider the nodes \( w_i, w_{i+1} \). By definition of \( \alpha_i, \alpha_{i+1} \):

- \( \rho(w_i) = R^-(z, z') \) and \( \rho(w_{i+1}) = R(z', z) \),
- \( \lambda(w_i) = (z', t_1) \), for some \( t_1 \), and \( \lambda(w_{i+1}) = (z, t_2) \), for some \( t_2 \).

Note that \( \rho(w_i) \) and \( \rho(w_{i+1}) \) refer to the same atom. Let \( R(t, t_2) \) be the atom such that \( w_{i+1} \) was added to \( T_{xy} \) via an application of (A1) to \( (z', t), R(z', z), R(t, t_2) \). By Observation 3, \( R^- \) is functional in \( \mathcal{U}_p, \mathcal{O} \setminus p \), and thus \( t_2 \) has no other \( R^- \)-neighbor than \( t \) and thus \( t_1 = t \). But then (A1) is not applicable to \((z, t_2), R^-(z, z'), R^-(t_2, t_1) = R^-(t_2, t) \) since \( R((z, t_2), (z', t_1)) = R((z, t_2), (z', t)) \) is already present, a contradiction.

\(^5\)As usual in computer science, we assume that the root of the tree is at the top and leaves at the bottom.
In the other two cases ($\alpha_i$, $\alpha_{i+1}$ were both defined in (b) or $\alpha_i$ was defined in (a) and $\alpha_{i+1}$ was defined in (b)), a contradiction is derived analogously. This finishes the proof of the claim.

Now, let the result of applying (A1) have domain size $N$. Then (A2) is applied at most $N \cdot \rho$ times, where $\rho$ denotes the number of roles in $\mathcal{O}$. Moreover, for each application, we only add two atoms and a copy of $p$. Thus, the overall construction finishes in polynomial time.  

\[ \square \]

Lemma 22. $\hat{q}$ is satisfiable w.r.t. $\mathcal{O}$ and it is a $p$-guided ELIQ-generalization of $q$ under $\mathcal{O}$.

Proof. We show that $\hat{q}$ satisfies Conditions 1 to 3 from Definition 4 and that $\hat{q}$ is satisfiable w.r.t. $\mathcal{O}$. For the proof, it is convenient to use the map $g$ defined by taking:

- $g(z, t) = z$ for every $(z, t) \in \text{var}(\hat{q})$ with $z \in \text{var}(q)$ and $t \in \Delta_{\mathcal{U},\mathcal{O}}$;
- $g(\hat{z}) = zRM$, for every variable $\hat{z}$ introduced in Step (A2) applied to $(z, t)$ and $R, M$;
- $g(x') = x$, for every copy $x'$ of some variable $x$ in $q$ introduced in Step (A2).

For Condition 1, we observe that $g$ is a homomorphism $\hat{q}(x_1, x_2) \rightarrow (\mathcal{U}_q, \mathcal{O}, x_1)$, and thus $q \subseteq_{\mathcal{O}} \hat{q}$.

Satisfiability of $\hat{q}$ w.r.t. $\mathcal{O}$ follows from the facts that $q$ is satisfiable under $\mathcal{O}$, that the map $g$ defined above is a homomorphism from $\hat{q}$ to $\mathcal{U}_q, \mathcal{O}$, and that since $q$ satisfies all functionality assertions in $\mathcal{O}$, by construction so does $\hat{q}$. For the latter, it is important that $\mathcal{O}$ is formulated in DL-$\text{Lite}_{\text{norm}}^{-}$ rather than in DL-$\text{Lite}_{\text{norm}}$. In particular, this ensures that the role $R$ in Step (A2) is not inverse functional.

For Condition 2, suppose to the contrary of what we have to show that $\hat{q} \subseteq_{\mathcal{O}} q$. Since $\hat{q}$ is satisfiable, we can fix a homomorphism $h: q(x_1) \rightarrow (\mathcal{U}_q, \mathcal{O}, x_1)$). By Lemma 20, there is an extension of the homomorphism $g$ to a homomorphism $g'$ for $(\mathcal{U}_q, (x_1, x_2)) \rightarrow (\mathcal{U}_q, \mathcal{O}, x_1)$. Then, the composition of $h$ and $g'$ is a homomorphism $q(x_1) \rightarrow (\mathcal{U}_q, \mathcal{O}, x_1)$. Lemma 19 implies that variables $\hat{z}$ introduced in (A2) are not in the image of $h$. Indeed, if $h(x) = \hat{z}$ for some $x \in \text{var}(q)$, then $g'(h(x))$ takes the shape $zRM$, in contradiction to Lemma 19. Since $q$ is rooted, all $h(x)$ take the shape $(z, t)$ for some $z \in \text{var}(q)$ and some trace $t$ in $\mathcal{U}_p, \mathcal{O}$. Consider the projection $h'$ of $h$ to its second component, that is

$$h'(x) = t \text{ for all } x \in \text{var}(q) \text{ such that } h(x) = (z, t).$$

It is routine to show that $h'$ is a homomorphism $q(x_1) \rightarrow (\mathcal{U}_p, \mathcal{O}, x_2)$, and thus $p \subseteq_{\mathcal{O}} q$, a contradiction.

For Condition 3, let $q'(x_0)$ be any ELIQ with $q \subseteq_{\mathcal{O}} q'$ and $p \subseteq_{\mathcal{O}} q'$. We can fix homomorphisms $h_1, h_2$ with $\mathcal{O}$ and $h_2: q'(x_0) \rightarrow (\mathcal{U}_p, \mathcal{O}, x_2)$. Based on $h_1$ and $h_2$, we inductively define a map $h$. We start with setting $h(x_0) = (h_1(x_0), h_2(x_0))$.

For the inductive step, suppose $h(x) = (z, t)$ is defined and $z \in \text{var}(q)$ and let $R(x, x')$ be an atom in $q'$ such that $h(x')$ is still undefined. Note that $z' = h_1(x')$ satisfies $R(z, z') \in q$ since $h_1$ is a homomorphism. Similarly, $t' = h_2(x')$ satisfies $R(t, t') \in \mathcal{U}_p, \mathcal{O}$. We distinguish cases.

1. Suppose first that $z' \in \text{var}(q)$. Then (A1) is applicable to $z, z', t, t'$, and we find $R((z, t), (z', t')) \in \hat{q}$. Set $h(x') = (z', t')$. 

2. Otherwise, \( z' \notin \text{var}(q) \). Since \( z \in \text{var}(q) \), \( z' \) takes the form \( zRM \) for some \( M \) and thus \( z \sim_{\mathcal{Q}, \mathcal{O}} M \) for that \( M \). Then (A2) is applicable to \((z, t)\) and \( R \). Let \( \hat{\varphi} \) be the variable introduced in (A2). Using the definition of the universal model, one can show that there is a homomorphism \( f \) from \( \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \) to \( \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \) which maps \( zRM \) to \( \hat{\varphi} \) and \( q \) to the copy of \( q \) that was added to \( \hat{\varphi} \) in this application of (A2). We set

\[
h(x') = f(h_1(x'))
\]

for every node \( x' \) in the subtree rooted at \( x' \) (assuming again that the root \( x_0 \) is at the top).

It remains to argue that \( h \) is a homomorphism from \( q'(x_0) \rightarrow (\mathcal{U}_{\hat{\varphi}, \mathcal{O}}, (x_1, x_2)) \), and thus \( \hat{\varphi} \subseteq \mathcal{O} q' \).

To see this, let first \( A(x) \in q' \).

- If \( h(x) \) was defined in Step 1 above, then \( h(x) = (z, t) \) for \( z = h_1(x) \in \text{var}(q) \) and \( t = h_2(x) \in \text{ind}(\mathcal{U}_{\hat{\varphi}, \mathcal{O}}) \). Since both \( h_1 \) and \( h_2 \) are homomorphisms, both \( A(z) \in \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \) and \( A(t) \in \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \). Thus \( A(h(x)) = A(z, t) \in \hat{\varphi} \subseteq \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \).

- If \( h(x) \) was defined in Step 2 above, then \( h(x) = f(h_1(x)) \) where \( f \) is a homomorphism from \( q \) to \( \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \). Since additionally, \( h_1 \) is a homomorphism, it follows that \( A(h(x)) \in \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \).

Suppose now \( R(x, y) \in q' \) and \( x \) is closer to the root \( x_0 \) than \( y \) in \( q' \).

- If both \( h(x) \) and \( h(y) \) were defined in Step 1. By (A1), \( R((h_1(x), h_2(x)), (h_1(y), h_2(y))) \in \hat{\varphi} \) and thus \( R(h(x), h(y)) \in \hat{\varphi} \subseteq \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \).

- If both \( h(x) \) and \( h(y) \) were defined in Step 2, then \( h(x) = f(h_1(x)) \) and \( h(y) = f(h_1(x)) \) for some homomorphism \( f \) from \( q \) to \( \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \). Since additionally, \( h_1 \) is a homomorphism, it follows that \( R(h(x), h(y)) \in \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \).

- If \( h(x) \) was defined in Step 1 and \( h(y) \) was defined in Step 2, then \( h(x) = (h_1(x), h_2(x)) = (z, t) \) and \( h(y) = f(h_1(y)) = \hat{\varphi} \) for the element \( \hat{\varphi} \) that was introduced in the application of (A2) to \((z, t)\) that defined \( h(y) \). (A2) additionally implies that \( R((z, t), \hat{\varphi}) \in \hat{\varphi} \), and thus \( R(h(x), h(y)) \in \mathcal{U}_{\hat{\varphi}, \mathcal{O}} \).

- The case that \( h(x) \) was defined in Step 2, but \( h(y) \) was defined in Step 1 is not possible since \( x \) is closer to the root than \( y \), by assumption and the fact that \( h \) is defined from root to leaves in \( q' \).

\( \square \)

**C. Proofs for Section 4**

We describe how to convert an \( \mathcal{ELT} \) ontology \( \mathcal{O} \) into an \( \mathcal{ELT} \) ontology \( \mathcal{O}' \) in normal form. We use \( \mathcal{C}(\mathcal{O}) \) to denote the set of all concepts that occur on the right-hand side of a concept inclusion in \( \mathcal{O} \). Note that \( \mathcal{C}(\mathcal{O}) \) is closed under taking sub-concepts. We introduce a fresh
concept name $X_C$ for every complex concept $C \in \mathcal{C}(O)$, and set $X_\bot = \bot$ and $X_A = A$ for concept names $A \in \mathcal{C}(O)$. The ontology $O'$ consists of all functionality assertions in $O$ and the following concept inclusions:

- $X_C \subseteq X_D$ for every $C \subseteq D \in O$;
- $X_{D_1 \cap D_2} \subseteq X_{D_1}$ and $X_{D_1 \cap X_{D_2}} \subseteq X_{D_1 \cap D_2}$, for every $D_1 \cap D_2 \in \mathcal{C}(O)$ and $i \in \{1, 2\}$;
- $X_{\exists R.C} \subseteq \exists R.X_C$ and $\exists R.X_C \subseteq X_{\exists R.C}$, for every $\exists R.C \in \mathcal{C}(O)$.

Clearly, $O'$ can be computed in polynomial time. Regarding the relationship between $O$ and $O'$, we observe the following consequences of the definition of $O'$.

**Lemma 23.**

1. $O'$ is a conservative extension of $O$;
2. $\text{sig}(O') = \text{sig}(O) \cup \{X_C \mid C \in \mathcal{C}(O)\}$;
3. $O' \models X_C \equiv C$, for all $C \in \mathcal{C}(O)$.

Lemma 23 essentially says that $O'$ is a conservative extension of $O$, but is slightly stronger in also making precise how exactly a model of $O$ can be extended to a model of $O'$.

**Lemma 10.** If ELIQs are polynomial time learnable under DL-Lite$^F_{\text{horn}}$ ontologies in normal form using membership and equivalence queries, the same is true for unrestricted DL-Lite$^F_{\text{horn}}$ ontologies.

**Proof.** Let $L'$ be a polynomial time learning algorithm for ELIQs under DL-Lite$^F_{\text{horn}}$ ontologies in normal form. We transform it into a polynomial time learning algorithm $L$ for ELIQs under unrestricted DL-Lite$^F_{\text{horn}}$ ontologies, relying on the normal form provided by Lemma 23.

Given a DL-Lite$^F_{\text{horn}}$ ontology $O$ and a signature $\Sigma = \text{sig}(O)$ with $\text{sig}(q_T) \subseteq \Sigma$, algorithm $L$ first computes the ontology $O'$ in normal form as per Lemma 23, choosing the fresh concept names so that they are not from $\Sigma$. It then runs $L'$ on $O'$ and $\Sigma' = \Sigma \cup \text{sig}(O')$. In contrast to $L'$, the oracle still works with the original ontology $O$. To ensure that the answers to the queries posed to the oracle are correct, $L$ modifies $L'$ as follows.

Whenever $L'$ asks a membership query $\mathcal{A}', O' \models q_T(a)$, we may assume that $\mathcal{A}'$ satisfies the functionality assertions from $O$, since otherwise the answer is trivially “yes”. Then, $L$ asks the membership query $\mathcal{A}, O \models q_T(a)$, where $\mathcal{A}$ is obtained from $\mathcal{A}'$ by starting with $\mathcal{A} = \mathcal{A}'$ and then extending it as follows:

\[ (*) \text{ for every } X_{\exists R}(b) \in \mathcal{A}' \text{ with } \mathcal{A} \not\models \exists R(b), \text{ add } R(b, b') \text{ for a fresh individual } b'. \]

By the following claim, the answer to the modified membership query coincides with that to the original query.

**Claim 1.** $\mathcal{A}', O' \models q(a)$ iff $\mathcal{A}, O \models q(a)$ for all ELIQs $q$ that only use symbols from $\Sigma$, and all $a \in \text{ind}(\mathcal{A}')$.

**Proof of the Claim 1.** For “if”, suppose that $\mathcal{A}, O \models q(a)$ and let $I'$ be a model of $\mathcal{A}'$ and $O'$. We can assume that $\Delta^{I'}$ does not mention any of the individuals that were introduced in the
construction of \( \mathcal{A} \). We will extend \( \mathcal{I} \) to a model \( \mathcal{I}' \) of \( \mathcal{A} \) and \( \mathcal{O} \) and such that \( (\mathcal{I}, a) \to (\mathcal{I}', a) \). This clearly suffices since \( \mathcal{I} \models q(a) \).

We construct \( \mathcal{I}' \) by processing every atom introduced in (*). Let \( R(b, b') \) be such an atom. Then, \( X_{3R}(b) \in \mathcal{A}' \) and, by definition of the normal form, \( X_{3R} \subseteq \exists R \in \mathcal{O}' \). Since \( \mathcal{I}' \) is a model of \( \mathcal{A}' \) and \( \mathcal{O}' \), there is an element \( c \) with \( R(b, c) \in \mathcal{I}' \). Informally, let \( \mathcal{J}_c \) be the unraveling of \( \mathcal{I}' \) at \( c \) which takes into account the functionality assertions in \( \mathcal{O} \), and in which the \( R^- \)-successor of \( c \) is omitted in case \( \text{func}(R^-) \in \mathcal{O} \). Then, add a copy of \( \mathcal{J}_c \) to \( \mathcal{I}' \), rename the root of \( \mathcal{J}_c \) to \( b' \), and add \( R(b, b') \) to \( \mathcal{I} \).

We now give a formal definition of \( \mathcal{J}_d \). Its domain \( \Delta^{\mathcal{J}_c} \) consists of all sequences \( a_0R_1a_1 \ldots R_na_n \) such that

- \( a_0 = c \);
- \( a_i \in \Delta^{\mathcal{I}'} \), for all \( i \) with \( 0 \leq i \leq n \);
- \( (a_i, a_{i+1}) \in R_{i+1}^{\mathcal{I}'} \), for all \( i \) with \( 0 \leq i < n \);
- \( \text{if func}(R_i^-) \in \mathcal{O} \), then \( R_{i+1} \neq R_i^- \), for all \( i \) with \( 0 \leq i < n \);
- \( \text{if } R_1 = R^- \text{ then } \text{func}(R^-) \notin \mathcal{O} \).

The interpretation of concept and role names is then as expected:

\[
\begin{align*}
A^{\mathcal{J}_c} &= \{a_0R_1a_1 \ldots R_na_n \in \Delta^{\mathcal{J}_c} | a_n \in A^{\mathcal{I}'} \} \quad \text{for all } A \in \mathcal{N}_C; \\
r^{\mathcal{J}_c} &= \{(\pi, r\pi a) | \pi \pi a \in \Delta^{\mathcal{I}_a} \} \cup \\
&\quad \{(\pi r^- a, \pi) | \pi r^- a \in \Delta^{\mathcal{J}_c} \} \quad \text{for all } r \in \mathcal{N}_R.
\end{align*}
\]

Note that each \( \mathcal{J}_c \) has a homomorphism into \( \mathcal{I}' \): just map every sequence \( a_0R_1 \ldots a_n \) to \( a_n \).

Let \( \mathcal{I} \) be the result of doing the above for every atom in \( \mathcal{A} \setminus \mathcal{A}' \). Clearly, because each such atom is added (as the final step), we have \( \mathcal{I} \models \mathcal{A} \). It is routine to verify that \( \mathcal{I} \) is also a model of \( \mathcal{O} \) and that there is a homomorphism \( (\mathcal{I}, a) \to (\mathcal{I}', a) \).

For “only if”, suppose that \( \mathcal{A}', \mathcal{O}' \models q(a) \) and let \( \mathcal{I} \) be a model of \( \mathcal{A} \) and \( \mathcal{O} \). Since \( \mathcal{O}' \) is a conservative extension of \( \mathcal{O} \), there is a model \( \mathcal{I}' \) of \( \mathcal{O}' \) that coincides with \( \mathcal{I} \) on \( \Sigma \). Moreover, by Point 3 of Lemma 23, it is also a model of \( \mathcal{A}' \). It follows that \( \mathcal{I} \models q(a) \) as required. This finishes the proof of Claim 1.

Second, whenever \( L' \) asks an equivalence query \( q'_H \equiv_{\mathcal{O}'} q_T \), \( L \) instead asks the equivalence query \( q_H \equiv_{\mathcal{O}} q_T \), where \( q_H \) is obtained from \( q'_H \) by replacing each assertion \( X_{3R}(x) \) with an assertion \( R(x, x') \), \( x' \) a fresh variable. Clearly, \( q_H \) is an ELIQ and can thus be used in an equivalence query. Furthermore, when the counterexample returned is \( \mathcal{A} \), the algorithm replaces it with the restriction \( \mathcal{A}|_{\Sigma} \) to signature \( \Sigma \) before passing it on to \( L' \).

Applying the following claim to both \( q'_H \) and \( q'_T = q_T \) shows that the answer to the modified equivalence query coincides with that to the original query.

**Claim 2.** Let \( q' \) be a CQ that uses only symbols from \( \text{sig}(\mathcal{O}') \) and let \( q \) be obtained from \( q' \) by replacing each assertion \( X_{3R}(x) \) with an assertion \( R(x, x') \), for a fresh variable \( x' \). Then \( \mathcal{A}|_{\Sigma}, \mathcal{O}' \models q'(\bar{a}) \text{ iff } \mathcal{A}, \mathcal{O} \models q(\bar{a}) \) for all ABoxes \( \mathcal{A} \).
Proof of Claim 2. For “if”, suppose \( A, \mathcal{O} \models q(\bar{a}) \) and let \( I \) be a model of \( A|_\Sigma \) and \( \mathcal{O}' \). Since \( q \) and \( \mathcal{O} \) contain only symbols from \( \Sigma, A|_\Sigma, \mathcal{O} \models q(\bar{a}) \). Since \( \mathcal{O}' \) is a conservative extension of \( \mathcal{O}, I \) is also a model of \( \mathcal{O} \). Thus \( I \models q(\bar{a}) \) and by Point 3 of Lemma 23, \( I \models q'(\bar{a}) \) follows as required.

For “only if”, suppose \( A|_\Sigma, \mathcal{O}' \models q'(\bar{a}) \) and let \( I \) be a model of \( A \) and \( \mathcal{O} \). Since \( \mathcal{O} \) contains only symbols from \( \Sigma, I|_\Sigma \) is a model of \( A|_\Sigma \) and since \( \mathcal{O}' \) is a conservative extension of \( \mathcal{O} \), there is a model \( I' \) of \( \mathcal{O}' \) that coincides on all symbols from \( \Sigma \) with \( I|_\Sigma \). Thus \( I' \models q'(\bar{a}) \) and by Point 3 of Lemma 23, \( I|_\Sigma \models q(\bar{a}) \). Then \( I \models q(\bar{a}) \) since \( q \) uses only symbols from \( \Sigma \) as required.

\[ \square \]

**Theorem 13.** Let \( q_T \) be a rooted CQ and \( \mathcal{O} \) an \( \mathcal{ELIF} \) ontology in normal form, and let \( q_1, q_2, \ldots \) be a generalization sequence towards \( q_T \) under \( \mathcal{O} \) such that \( q_1 \) is satisfiable w.r.t. \( \mathcal{O} \). If all \( q_i \) are \((q_T, \mathcal{O})\)-minimal and \( \mathcal{O} \)-saturated, then the sequence has length at most \(|\var(q_T)|^3 \cdot |\text{sig}(\mathcal{O})|\).

**Proof.** Let \( \mathcal{O} \) be an \( \mathcal{ELIF} \) ontology in normal form and \( q_T \) a rooted CQ. Further, let \( q_1(\bar{x}_1), \ldots, q_n(\bar{x}_n) \) be a generalization sequence towards \( q_T(\bar{x}) \) under \( \mathcal{O} \) such that \( q_1 \) is satisfiable w.r.t. \( \mathcal{O} \), and suppose all \( q_i \) are \( \mathcal{O} \)-saturated and \((q_T, \mathcal{O})\)-minimal.

We start with showing that all \( q_i \) are rooted and have at most as many variables as \( q_T \).

**Claim 1.** For all \( i \) with \( 1 \leq i \leq n \), \(|\var(q_i)| \leq |\var(q_T)| \) and \( q_i \) is rooted.

**Proof of Claim 1.** Assume to the contrary that \(|\var(q_i)| > |\var(q_T)| \) or \( q_i \) is not rooted. Since \( q_i \subseteq \mathcal{O} q_T \), there is a homomorphism \( h: q_T(\bar{x}) \rightarrow (U_{q_i,\mathcal{O},\bar{x}_i}) \). If \(|\var(q_i)| > |\var(q_T)| \), there is an \( x \in \var(q_i) \) with \( x \notin \text{img}(h) \). The same is true if \( q_i \) is not rooted, since \( q_T \) is rooted. By Lemma 18, \( h \) is also a homomorphism witnessing \( q_T(\bar{x}) \rightarrow (U_{q_i,\mathcal{O},\bar{x}_i}) \) where \( U = \text{img}(h) \cap \var(q_i) \). Hence, \( q_i|_U \subseteq \mathcal{O} q_T \) which is contradiction to \((q_T, \mathcal{O})\)-minimality. This finishes the proof of Claim 1.

We will show next that the \( q_i \) have an increasing number of variables; we need some preparation. Because \( q_1 \) is satisfiable w.r.t. \( \mathcal{O} \) and \( q_i \subseteq \mathcal{O} q_i \), for all \( i \), all \( q_i \) are satisfiable w.r.t. \( \mathcal{O} \). Thus, we can use the characterization of query containment under \( \mathcal{O} \) in terms of the universal model from Lemma 17. Since \( q_i \subseteq \mathcal{O} q_{i+1} \), for all \( i < n \), we can thus fix homomorphisms \( h_i: q_{i+1}(\bar{x}_{i+1}) \rightarrow (U_{q_i,\mathcal{O},\bar{x}_i}) \).

**Claim 2.** For all \( i < n \), \(|\var(q_i)| \leq |\var(h_i)| \) and \(|\var(q_i)| \leq |\var(q_{i+1})| \).

**Proof of Claim 2.** Since \(|\var(q_i)| \leq |\var(h_i)| \) implies \(|\var(q_i)| \leq |\var(q_{i+1})| \), it suffices to show the former. Assume to the contrary that there is an \( x \in \var(q_i) \) with \( x \notin \text{img}(h_i) \). By Lemma 18, \( h \) is also a homomorphism from \( q_{i+1}(\bar{x}_{i+1}) \rightarrow (U_{q_i,\mathcal{O},\bar{x}_i}) \) with \( U = \text{img}(h) \cap \var(q_i) \). Let \( h' \) be the extension of \( h \) to a homomorphism from \( U_{q_i+1,\mathcal{O}} \) to \( U_{q_{i+1},\mathcal{O}} \) which exists by Lemma 20. Composing \( h' \) with a homomorphism \( g: q_T(\bar{x}) \rightarrow (U_{q_{i+1},\mathcal{O},\bar{x}_{i+1}}) \) yields a homomorphism \( g': q_T(\bar{x}) \rightarrow (U_{q_i,\mathcal{O},\bar{x}_i}) \). Hence, \( q_i|_U \subseteq q_T \), which is in contradiction to \((q_T, \mathcal{O})\)-minimality of \( q_i \). This finishes the proof of Claim 2.

Now, we use the two claims to show that \( n \leq 2 \cdot |\var(q_T)|^3 \cdot |\text{sig}(\mathcal{O})| \). Claim 2 implies that \(|\var(q_i)| \leq |\var(q_{i+1})| \) for all \( i > 0 \). By Claim 1, it suffices to show that the length of any subsequence \( q_1, \ldots, q_k \) with \(|\var(q_i)| = \cdots = |\var(q_k)| \) is bounded by \(|\var(q_T)|^2 \cdot |\text{sig}(\mathcal{O})| \).

Consider any \( i \in \{j, \ldots, k-1\} \). Since \(|\var(q_{i+1})| = |\var(q_i)| \) and by Claim 2, \( h_j \) is a bijection between \( \var(q_{i+1}) \) and \( \var(q_i) \). Since \( q_j \) is \( \mathcal{O} \)-saturated, \( h_i \) is also a bijective homomorphism from \( q_{i+1} \) to \( q_i \). Thus, the number of atoms in \( q_{i+1} \) is at most the number atoms in \( q_i \). By the
definition of generalization sequence, \( q_{i+1} \not\subseteq \mathcal{O} q_i \), and thus \( h_i \) cannot be a homomorphism from \( q_i \) to \( q_{i+1} \). Therefore, one of the following two cases applies:

1. there is a concept atom \( A(x_1) \in q_i \) such that \( A(h_i(x_1)) \notin q_{i+1} \), or
2. there is a role atom \( r(x_1, x_2) \in q_i \) such that \( r(h_i(x_1), h_i(x_2)) \notin q_{i+1} \).

Thus, at least one atom is removed going from any \( q_i \) to \( q_{i+1} \).

Since, by definition of generalization sequences, all symbols in \( q_j \) must occur in \( \mathcal{O} \), \( q_j \) contains at most \( N|\text{var}(q_j)|^2 + M|\text{var}(q_j)| \) atoms, \( N \) and \( M \) are the numbers of role names and concept names, respectively, in \( \text{sig}(\mathcal{O}) \). Since, by Claim 1, \( |\text{var}(q_j)| \leq |\text{var}(q_T)| \), the length of the sequence \( q_j, \ldots, q_k \) is bounded by

\[
N|\text{var}(q_j)|^2 + M|\text{var}(q_j)| \leq (N + M) \cdot |\text{var}(q_j)|^2 \leq |\text{sig}(\mathcal{O})| \cdot |\text{var}(q_T)|^2.
\]

Definition 24. An \( \mathcal{ELI}\!F \)-simulation from interpretation \( \mathcal{I}_1 \) to interpretation \( \mathcal{I}_2 \) is a relation \( S \subseteq \Delta_{\mathcal{I}_1} \times \Delta_{\mathcal{I}_2} \) such that for all \( (d_1, d_2) \in S \), we have:

1. for all \( A \in \mathcal{N}_C \): if \( A(d_1) \in \mathcal{I}_1 \), then \( A(d_2) \in S \);
2. for all \( r \in \mathcal{N}_R \) and \( R \in \{r, r^{-}\} \): if there is some \( d_1' \in \Delta_{\mathcal{I}_1} \) with \( R(d_1, d_1') \in \mathcal{I}_1 \), then there is \( d_2' \in \Delta_{\mathcal{I}_2} \) such that \( (d_1', d_2') \in S \) and \( R(d_2, d_2') \in \mathcal{I}_2 \).

The following lemma gives an important property of simulations. The proof is standard and omitted.

Lemma 25. Let \( \mathcal{O} \) be a \( \mathcal{ELI}\!F \) ontology, \( \mathcal{A}_1, \mathcal{A}_2 \) ABoxes and \( q(x) \) an ELIQ such that \( \mathcal{A}_1, \mathcal{A}_2 \), and \( q \) are satisfiable w.r.t \( \mathcal{O} \). If there is an \( \mathcal{ELI}\!F \)-simulation \( S \) from \( \mathcal{A}_1 \) to \( \mathcal{A}_2 \) with \( \langle a_1, a_2 \rangle \in S \), then \( \mathcal{A}_1, \mathcal{O} \models q(a_1) \) implies \( \mathcal{A}_2, \mathcal{O} \models q(a_2) \).

Lemma 14. The sequence \( p_1, p_2, \ldots \) is a generalization sequence towards \( q_T \) under \( \mathcal{O} \).

Proof. First note that both rules used in extract-minimal-ELIQ preserve satisfiability w.r.t. \( \mathcal{O} \). Since the input \( q \) to extract-minimal-ELIQ is assumed to be satisfiable w.r.t. \( \mathcal{O} \) (recall the preconditions of Lemma 11), all \( p_i \) are satisfiable w.r.t. \( \mathcal{O} \) as well, and we can use the characterization of query containment in terms of homomorphisms to the universal model provided in Lemma 17. We do this without further notice below.

We start by showing \( p_i \subseteq \mathcal{O} p_{i+1} \) for all \( i < n \). Let \( p_i' \) be the result of applying Double cycle to \( p_i \) and recall that \( p_{i+1} \) is the result of exhaustively applying Drop variable to \( p_i' \). Hence, it suffices to show \( p_i \subseteq \mathcal{O} p_{i}' \). To do this, in turn, it is enough to point out that we obtain a homomorphism \( h_i \) from \( p_i' \) to \( p_i \) with \( h_i(x_0) = x_0 \) by setting \( h_i(x) = x \) for all \( x \in \text{var}(p_i) \) and \( h_i(x') = x \) for all variables \( x' \) in the disjoint copy of \( p_i \) that is added in Double Cycle. We shall reuse \( h_i \) below and call it the natural homomorphism from \( p_i' \) to \( p_i \).

Next we show that \( p_i \subseteq \mathcal{O} q_T \) for all \( i \) with \( 1 \leq i \leq n \), by induction on \( i \). In the induction start, \( p_1 \) is the result of exhaustive application of Drop variable to the \( \mathcal{O} \)-saturation of the input \( q \). Since \( q \subseteq \mathcal{O} q_T \) and Drop variable preserves this property, \( p_1 \subseteq \mathcal{O} q_T \) follows.
Assume that \( f \), added in that construction. However, since \( r \) from \( p \), injective homomorphism that maps the cycle \( \gamma \) call that \( x \) there is a variable \( \gamma \), \( \gamma \) with \( |\{1, \ldots, \gamma, \ldots, 0\}| \). Thus, \( \Gamma \), is the composition of the homomorphisms \( \gamma \), \( \gamma \). Since \( f \) is not injective, \( \gamma \) \( \gamma \) is an injective homomorphism that maps the cycle \( \gamma \), \( \gamma \) in \( p \). Assume to the contrary that \( p \), \( p \) is an \( E\ell\ell\ell \)-simulation from \( A_p \) to \( A_p \) with \((x_0, x_0) \in S \). Thus, \( A_p, O \models q_T(x_0) \) by Lemma 25, and \( p \) \( q_T \) as required.

It remains to show that \( p_{i+1} \subseteq_O p_i \) for \( 1 \leq i < n \). Similarly to what was done above, it suffices to show that \( p_i \subseteq_O p_i \) where \( p_i \) is the result of applying Double cycle to \( p_i \). Assume to the contrary that \( p_i \subseteq_O p_i \) for some \( i \). Then there is a homomorphism \( g \) : \( p_i(x_0) \rightarrow (U_{p_i,O}, x_0) \). Composing \( g \) with the extension \( h^+ \) of the natural homomorphism \( h_i \) to a homomorphism from \( U_{p_i} \) to \( U_{p_i} \), which exists by Lemma 20, yields a homomorphism \( \hat{g} \) : \( p_i(x_0) \rightarrow (U_{p_i,O}, x_0) \).

Let \( R(y_1, y_2), \ldots, R_n(y_n, y_1) \) be the cycle that was expanded in the construction of \( p_i \) and consider the set \( \Gamma \) of all sets of variables that form a cycle of length \( n \) in \( p_i \). For example, \( \{y_1, \ldots, y_n\} \in \Gamma \).

Let \( \{x_1, \ldots, x_n\} \) be any element of \( \Gamma \). We show that \( \{\hat{g}(x_1), \ldots, \hat{g}(x_n)\} \in \Gamma \). Since \( \hat{g} \) is a homomorphism, it suffices to show that \(|\{\hat{g}(x_1), \ldots, \hat{g}(x_n)\}| = n \). Assume the contrary. Then there are \( x_j \) and \( x_k \) with \( x_j \neq x_k \) and \( \hat{g}(x_j) = \hat{g}(x_k) \), implying that \( \hat{g} \) is not injective. This, in turn, implies that there is a \( x \in \var(p_i) \) with \( x \notin \text{img}(\hat{g}) \), in contradiction to the \((q_T, O)\)-minimality of \( p_i \).

Hence, we can define a function \( f : \Gamma \rightarrow \Gamma \) by setting \( f(\{x_1, \ldots, x_n\}) = \{\hat{g}(x_1), \ldots, \hat{g}(x_n)\} \). Assume that \( f \) is not injective, that is, there are \( \gamma, \gamma' \in \Gamma \) with \( \gamma \neq \gamma' \) and \( f(\gamma) = f(\gamma') \). Since \( f(\gamma) = f(\gamma') \), there is a \( x' \in \gamma' \) with \( \hat{g}(x) = \hat{g}(x') \), and clearly \( x' \neq x \). This again contradicts \((q_T, O)\)-minimality of \( p_i \) as above. Thus, \( f \) is a bijection from \( \Gamma \) to \( \Gamma \).

Since \( \Gamma \) is finite, it follows that there must be a \( j \geq 1 \) such that \( f^j(\{y_1, \ldots, y_n\}) = \{y_1, \ldots, y_n\} \). By definition of \( f \) this implies that \( \{\hat{g}^j(y_1), \ldots, \hat{g}^j(y_n)\} = \{y_1, \ldots, y_n\} \). Recall that \( \hat{g} \) is the composition of the homomorphisms \( h_i^+ \) and \( g \). Since \((q_T, O)\)-minimality of \( p_i \) implies that \( \hat{g} \) is injective, \( g \) must also be injective. Thus, the composition \( g' \) of \( \hat{g}^{-1} \) and \( g \) is an injective homomorphism that maps the cycle \( \{y_1, \ldots, y_n\} \) in \( p_i \) to some subset of the expanded cycle \( \{y_1, y_1', \ldots, y_n, y_n'\} \) in \( p_i \).

First consider the case where \( \{g'(y_1), \ldots, g'(y_n)\} = \{y_1, \ldots, y_n\} \). By the construction of \( p_i' \) from \( p_i \), the query \( p_i'|_{\{y_1, \ldots, y_n\}} \) contains one less role atom than the query \( p_i|_{\{y_1, \ldots, y_n\}} \), implying that \( g' \) cannot be an injective homomorphism, leading to a contradiction. The case where \( \{g'(y_1), \ldots, g'(y_n)\} = \{y_1, \ldots, y_n\} \) is analogous.

The remaining case is that \( \{g'(y_1), \ldots, g'(y_n)\} \) contains both variables of the form \( y_1 \) and \( y_i' \). Then two different atoms from the cycle \( R(y_1, y_2), \ldots, R_n(y_n, y_1) \) must be mapped by \( g' \) to the role atoms \( r(x, y'), r(x', y) \) that were added by Double cycle to connect the disjoint copy of \( p_i \) added in that construction. However, since \( h_i(x') = h_i(x) \) and \( h_i(y') = h_i(y) \), this implies that the composition of \( g' \) and \( h_i \) is a non-injective homomorphism from \( p_i \) to \( p_i \), again contradicting \((q_T, O)\)-minimality of \( p_i \).