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*$^{\mathcal{F}}_{horn}$.

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$^{\mathcal{H}}$ and DL-Lite$^{\mathcal{F}-}$ where the '−' indicates a restriction on the use of inverse functional roles.

DL 2022: 35th International Workshop on Description Logics, August 7–10, 2022, Haifa, Israel

mfunk@informatik.uni-leipzig.de (M. Funk); jungj@uni-hildesheim.de (J. C. Jung); clu@informatik.uni-leipzig.de (C. Lutz)

CEUR Workshop Proceedings (CEUR-WS.org)

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*$^{\mathcal{H}}$ and DL-Lite$^{\mathcal{F}-}$ [21, 24], but it already fails for learning ELIQs under *DL-Lite*$_{horn}$ ontologies [24]; recall that the $\cdot_{horn}$ 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 $q_H \subseteq_\mathcal{O} p$, $q_\mathcal{A} \subseteq_\mathcal{O} p$ where $q_\mathcal{A}$ is $\mathcal{A}$ viewed as a CQ with answer variables $\bar{a}$, and $p \subseteq_\mathcal{O} p'$ for every $p' \in \mathcal{Q}$ with $q_H \subseteq_\mathcal{O} p'$ and $q_\mathcal{A} \subseteq_\mathcal{O} q'$. This approach has been used to learn unrestricted CQs without ontologies [21] and to learn syntactically restricted CQs under $\mathcal{EL}$ 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-Lite*$^{\mathcal{F}-}_{horn}$ ontologies. In theory, a natural way to construct a $\mathcal{Q}$-LGG of the hypothesis q_H and counterexample $\mathcal{A}$ is to build the universal models of q_H (viewed as an ABox) and of $\mathcal{A}$ 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 $\mathcal{A}$ 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 $\mathcal{EL}$ ontologies [22]. To learn ELIQs under DL-Lite ontologies, however, this approach fails since finite ELIQ-universal models are not guaranteed to exist (even for non-branching ELIQs).

Example 1. *Let $\mathcal{O} = \{\top \sqsubseteq \exists r.\top\}$ and $\mathcal{A} = \{A(a)\}$. The homomorphism-universal model of $\mathcal{A}$ and $\mathcal{O}$ is $\mathcal{A}$ 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 $\mathcal{A}$ 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-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 $\mathcal{O} = \{\exists r^-.\top \sqsubseteq \exists r.\top, \exists r^-.\top \sqsubseteq \exists s.\top\}$. Consider the unary CQs*

$$p(x) = \exists y \exists z\, r(x,x) \wedge s(x,y) \wedge s(z,y) \wedge r(z,z) \wedge A(z) \quad \textit{and} \quad q(x) = \exists y\, A(x) \wedge r(x,y).$$

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 $\widehat{q}(x)$ is an ELIQ-LGG of p and q, and consider all ELIQs of the form $q_{n,m} = \exists r^n.\exists s.\exists s^-.\exists(r^-)^m.A$ 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 $\widehat{q} \subseteq_\mathcal{O} q_{n,m}$ 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 $\widehat{q}$ (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 '$\exists r^i$' prefix of $q_{i,i}$ to the same element of $\mathcal{U}$, then an easy pumping argument shows that $\widehat{q} \subseteq_\mathcal{O} q_{j,i}$ 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 $\mathcal{A}$ and $\mathcal{O}$ admits a homomorphism into every model of $\mathcal{A}$ and $\mathcal{O}$.

generated by an existential quantifier, that is, it is in the tree-shaped 'anonymous' part of $\mathcal{U}$. Since $q_{i,i}$ is rooted, the h_i-homomorphic image of the '$\exists r^i$' prefix of $q_{i,i}$ 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 $\widehat{q}$ from x along an r-path and x reaches an instance of A along an r-path, which means that $\widehat{q} \subseteq_\mathcal{O} \exists r^k.A$ for some $k \geq 1$. But this contradicts $p \subseteq_\mathcal{O} \widehat{q}$.

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 the counterexample. We use our construction as a central ingredient to prove polynomial time learnability of ELIQs in the presence of *DL-Lite*$^{\mathcal{F}-}_{\mathit{horn}}$ 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 $\mathcal{ELIF}$ ontologies in place of *DL-Lite*$^{\mathcal{F}-}_{\mathit{horn}}$ ontologies. This serves to highlight the places where we crucially rely on ELIQs and *DL-Lite*$^{\mathcal{F}-}_{\mathit{horn}}$, and in addition it is potentially useful for future proofs where the lemmas that admit a general formulation do not need to be reproved.

Missing proof details are provided in the appendix.

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, and R^- denotes r when $R = r^-$. A *basic concept* B is of the form $\top$, A, or $\exists R$ where A ranges over $\mathsf{N_C}$ and R over roles. A *DL-Lite*$^{\mathcal{F}}_{\mathit{horn}}$ *ontology* is a set of *concept inclusions (CIs)* $B_1 \sqcap \cdots \sqcap B_n \sqsubseteq B$, *concept disjointness constraints* $B_1 \sqcap \cdots \sqcap B_n \sqsubseteq \bot$, *role disjointness constraints* $R_1 \sqcap R_2 \sqsubseteq \bot$, and *functionality assertions* $\mathsf{func}(R)$ where B_i, B range over basic concepts and R_1, R_2, R over roles. In a *DL-Lite*$^{\mathcal{F}-}_{\mathit{horn}}$ ontology, we additionally require that if $\exists R$ occurs on the right-hand side of a CI, then $\mathsf{func}(R^-) \notin \mathcal{O}$ [24].

An $\mathcal{ELI}$ *concept* is an expression C that is built according to the rule $C ::= \top \mid A \mid C \sqcap C \mid \exists R.C$ where A ranges over concept names and R over roles. An $\mathcal{ELIF}$ *ontology* $\mathcal{O}$ is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$, *emptiness constraints* $C \sqsubseteq \bot$, *role disjointness constraints* $R_1 \sqcap R_2 \sqsubseteq \bot$, and *functionality assertions* $\mathsf{func}(R)$ where C, D range over $\mathcal{ELI}$ concepts and R_1, R_2, R over roles. The basic concept $\exists R$ is a different way to write the $\mathcal{ELI}$ concept $\exists R.\top$. Note that every *DL-Lite*$^{\mathcal{F}}_{\mathit{horn}}$ ontology is an $\mathcal{ELIF}$ ontology. An $\mathcal{ELIF}$ ontology (and thus also a *DL-Lite*$^{\mathcal{F}}_{\mathit{horn}}$ ontology) is in *normal form* if all concept inclusions in it are of the form $A \sqsubseteq C$ or $C \sqsubseteq A$, where A is a concept name. Every *DL-Lite*$^{\mathcal{F}}_{\mathit{horn}}$ ontology $\mathcal{O}$ can be transformed in polynomial time into a *DL-Lite*$^{\mathcal{F}}_{\mathit{horn}}$ ontology $\mathcal{O}'$ in normal form such that $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$, and the same is true for $\mathcal{ELIF}$ ontologies.

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 $\mathcal{A}$. 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 $A \in \mathsf{N_C}$ 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 $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, a constraint $C \sqsubseteq \bot$ if $C^{\mathcal{I}} = \emptyset$, a functionality assertion $\mathsf{func}(R)$ if $R^{\mathcal{I}}$ is a partial function, a concept assertion $A(a)$ if $a \in A^{\mathcal{I}}$, and a role assertion $r(a,b)$ if $(a,b) \in r^{\mathcal{I}}$. 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 $\mathcal{O} \models C \sqsubseteq D$ if every model of the ontology $\mathcal{O}$ satisfies the concept inclusion $C \sqsubseteq D$ 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. 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 O such as an ontology, 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.

Queries. Every $\mathcal{ELI}$ concept C can be viewed as an *$\mathcal{ELI}$ query (ELIQ)*. An individual $a \in \mathsf{ind}(\mathcal{A})$ is an *answer* to C on an ABox $\mathcal{A}$ w.r.t. an ontology $\mathcal{O}$, written $\mathcal{A}, \mathcal{O} \models C(a)$, if $a \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{O}$. We shall often view ELIQs as unary *conjunctive queries (CQs)* and also consider CQs that are not ELIQs. A CQ takes the form $q(\bar{x}) = \exists \bar{y}\, \phi(\bar{x}, \bar{y})$ with ϕ a conjunction of *concept atoms* $A(x)$ and *role atoms* $r(x,y)$ where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. We 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 $\mathsf{var}(q)$ 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 $r(y,x)$. A CQ is *rooted* if in its Gaifman graph $G_q = (\mathsf{var}(q), \{\{y,z\} \mid r(y,z) \in q\})$ 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 CQ q by viewing variables as individuals and atoms as assertions. A CQ q is *satisfiable* w.r.t. ontology $\mathcal{O}$ if $\mathcal{A}_q$ is. For any CQ q and set $U \subseteq \mathsf{var}(q)$, $q|_U$ 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 $\mathsf{ind}(\mathcal{A})$, $\mathcal{A}, \mathcal{O} \models q_1(\bar{a})$ 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. *DL-Lite*$^{\mathcal{F}}_{horn}$ ontologies can be conveniently characterized using universal models. Let $\mathcal{O}$ be an *DL-Lite*$^{\mathcal{F}}_{horn}$ 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 $\bigsqcap M$ as a shorthand for $\bigsqcap_{A \in M} A$. For $a \in \mathsf{ind}(\mathcal{A})$, M, M' sets of concept names, and R a role, we write $a \rightsquigarrow^{R}_{\mathcal{A},\mathcal{O}} M$ if $\mathcal{A}, \mathcal{O} \models \exists R. \bigsqcap M(a)$ and M is maximal with this condition. We write

$M \rightsquigarrow_{\mathcal{O}}^{R} M'$ if $\mathcal{O} \models \bigsqcap M \sqsubseteq \exists R. \bigsqcap M'$ and M' is maximal with this.

A *trace* for $\mathcal{A}$ and $\mathcal{O}$ is a sequence $t = aR_1M_1R_2M_2 \dots R_nM_n$, $n \geq 0$ where $a \in \mathsf{ind}(\mathcal{A})$, $R_1, \dots, R_n$ are roles in $\mathsf{sig}(\mathcal{O})$, and $M_1, \dots, M_n$ are sets of concept names in $\mathsf{sig}(\mathcal{O})$, such that

(i) $a \rightsquigarrow_{\mathcal{A},\mathcal{O}}^{R_1} M_1$ and there is no $b \in \mathsf{ind}(\mathcal{A})$ with $R_1(a, b) \in \mathcal{A}$,

(ii) $M_i \rightsquigarrow_{\mathcal{O}}^{R_{i+1}} M_{i+1}$ and $R_{i+1} \neq R_i^-$, for $1 \leq i < n$.

The set $\mathbf{T}$ of all traces for $\mathcal{A}$ and $\mathcal{O}$ forms the domain of the universal model $\mathcal{U}_{\mathcal{A},\mathcal{O}}$, defined as

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

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

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

$\mathcal{O}$-saturatedness and $\mathcal{O}$-minimality. Let $\mathcal{O}$ be an $\mathcal{ELIF}$ ontology. A CQ q is *$\mathcal{O}$-minimal* if there is no $U \subsetneq \mathsf{var}(q)$ such that $q \equiv_{\mathcal{O}} q|_U$. A CQ q is *$\mathcal{O}$-saturated* if $\mathcal{A}_q, \mathcal{O} \models A(y)$ implies $A(y) \in q$ for all $y \in \mathsf{var}(q)$ and $A \in \mathsf{N_C}$. 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 "$\mathcal{A}, \mathcal{O} \models A(a)$?" is available. In *DL-Lite$^{\mathcal{F}}_{horn}$*, 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 $\widehat{q}$ is a *least general $\mathcal{Q}$-generalization ($\mathcal{Q}$-LGG) of CQs p, q under an ontology $\mathcal{O}$* if $q \subseteq_{\mathcal{O}} \widehat{q}$, $p \subseteq_{\mathcal{O}} \widehat{q}$, and $\widehat{q} \subseteq_{\mathcal{O}} q'$ for every $q' \in \mathcal{Q}$ with $q \subseteq_{\mathcal{O}} q'$ and $p \subseteq_{\mathcal{O}} q'$. We consider the following weakening of $\mathcal{Q}$-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 $\widehat{q}$ is a p-*guided $\mathcal{Q}$-generalization of q under $\mathcal{O}$ *if the following conditions are satisfied:*

1. *$q \subseteq_{\mathcal{O}} \widehat{q}$;*
2. *$\widehat{q} \not\subseteq_{\mathcal{O}} q$;*
3. *$\widehat{q} \subseteq_{\mathcal{O}} q'$, 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 $\mathcal{Q}$-LGG. Intuitively, they mean that $\widehat{q}$ 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 $p \subseteq \widehat{q}$, we only want $\widehat{q}$ 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 $\widehat{q}$.

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 $\mathcal{Q}$-generalizations are not uniquely defined.

Example 5. *Consider $q(x) = A(x) \wedge B(x) \wedge C(x)$ and $p(x) = A(x)$. Then both $q_1(x) = A(x)$ and $q_2(x) = A(x) \wedge B(x)$ 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 $\mathcal{Q}$-LGG of two CQs p, q under an ontology $\mathcal{O}$ is both a p-guided $\mathcal{Q}$-generalization of q under $\mathcal{O}$ and a q-guided $\mathcal{Q}$-generalization of p under $\mathcal{O}$. 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 $\mathcal{O}$ from Example 2, and recall that there is no CQ-LGG for p, q. However, the query*

$$\widehat{q}(x) = \exists y \exists y'\, r(x,y) \wedge r(x,y') \wedge A(y')$$

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

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

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

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

(A1) For every $(z,t) \in \mathsf{var}(\widehat{q})$ with $z \in \mathsf{var}(q)$ and $t \in \Delta^{\mathcal{U}_{p,\mathcal{O}}}$, every atom $R(z,z')$ in q, and every atom $R(t,t') \in \mathcal{U}_{p,\mathcal{O}}$, add the atom $R((z,t),(z',t'))$, and all atoms $A(z',t')$ such that $A(z') \in \mathcal{U}_{q,\mathcal{O}}$ and $A(t') \in \mathcal{U}_{p,\mathcal{O}}$.

(A2) For every $(z,t) \in \mathsf{var}(\widehat{q})$ with $z \in \mathsf{var}(q)$ and $t \in \Delta^{\mathcal{U}_{p,\mathcal{O}}}$ and every role R such that $z \leadsto^{R}_{q,\mathcal{O}} M$ for some M and there is no atom of the form $R(z,z')$ in q, add the atoms

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

with $\widehat{z}$ 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 $\mathcal{O}$-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*$^{\mathcal{F}}_{horn}$ 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 $t \in \mathsf{var}(q)$. 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 *DL-Lite*$^{\mathcal{F}-}_{horn}$ 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 traveling the 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 *DL-Lite*$^{\mathcal{F}-}_{horn}$ 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 *DL-Lite*$^{\mathcal{F}-}_{horn}$: the precondition of (A2) implies that $\exists R$ 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 $\widehat{q}$ is not necessarily an ELIQ.

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

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

Note that p and q are $\mathcal{O}$-saturated. The result of exhaustively applying Step (A1) is $\widehat{q}_0(x) = X(x)$,[4] *which generalizes q, but is too general: the ELIQ*

$$q_T(x) = \exists x' \exists y \exists y' \exists z\, r(x, y) \wedge s(y, z) \wedge s(y', z) \wedge r(x', y') \wedge B(x')$$

satisfies $\widehat{q}_0 \not\subseteq_{\mathcal{O}} q_T$, while $p \subseteq_{\mathcal{O}} q_T$ and $q \subseteq_{\mathcal{O}} q_T$. After additionally applying (A2), we obtain

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

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

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

$$p'(x) = \exists y_1 \exists y_2\, r(x, y_1) \wedge r(x, y_2) \wedge A(y_1) \wedge B(y_2)$$
$$q'(x) = \exists y \exists z\, r(x, y) \wedge r(z, y) \wedge A(y) \wedge B(y)$$

The result of (A1) is the direct product $q' \times p'$ of q' and p' which is $\widehat{q}'(x) = \exists y_1 \exists y_2 \exists z\, r(x, y_1) \wedge r(x, y_2) \wedge r(z, y_1) \wedge r(z, y_2) \wedge A(y_1) \wedge B(y_2)$, 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 *DL-Lite*$^{\mathcal{F}-}_{horn}$ 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.

[4] We have replaced the single answer variable (x_1, x_2) with x for the sake of readability.

Algorithm 1 Algorithm for learning ELIQs under *DL-Lite*$^{\mathcal{F}-}_{horn}$ ontologies

```
Input A DL-Lite$^{\mathcal{F}-}_{horn}$ ontology $\mathcal{O}$ and a unary CQ $q_H^0$ satisfiable w.r.t. $\mathcal{O}$ such that $q_H^0 \subseteq_{\mathcal{O}} q_T$
Output An ELIQ $q_H$ such that $q_H \equiv_{\mathcal{O}} q_T$

$q_H :=$ extract-minimal-ELIQ($q_H^0$)
while the equivalence query "$q_H \equiv_{\mathcal{O}} q_T$?" returns a counterexample $(\mathcal{A}, a)$ do
    $q_D :=$ extract-minimal-ELIQ($q_{\mathcal{A}}$) where $q_{\mathcal{A}}$ is $\mathcal{A}$ viewed as a CQ with answer variable $a$
    $q'_H :=$ a $q_D$-guided ELIQ-generalization of $q_H$ under $\mathcal{O}$
    $q_H :=$ extract-minimal-ELIQ($q'_H$)
end while
return $q_H$
```

Theorem 9. *ELIQs are polynomial time learnable under DL-Lite*$^{\mathcal{F}-}_{horn}$ *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*$^{\mathcal{F}-}_{horn}$ *ontologies in normal form using membership and equivalence queries, the same is true for unrestricted DL-Lite*$^{\mathcal{F}-}_{horn}$ *ontologies.*

Our learning algorithm is listed in Algorithm 1. It takes as input a *DL-Lite*$^{\mathcal{F}-}_{horn}$ 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 $U \subsetneq \mathsf{var}(q')$ with $q'|_U \subseteq_{\mathcal{O}} q_T$. 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 $q \subseteq_{\mathcal{O}} q' \subseteq_{\mathcal{O}} q_T$ 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 $y \in \mathsf{var}(p)$ and let $p' = p|_{\mathsf{var}(p) \setminus \{y\}}$. If the response to the membership query $\mathcal{A}_{p'}, \mathcal{O} \models q_T(x)$ 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

p' of p to p and let x', y' be the copies of x, y in p'. Remove the atoms $r(x, y), r(x', y')$ and add the atoms $r(x, y'), r(x', y)$.

We give preference to the first rule, that is, the second rule is only applied when the first one is not applicable. Clearly, if *Drop variable* is not applicable, then p is $(q_T, \mathcal{O})$-minimal. Once no rule is applicable anymore, $\mathsf{extract\text{-}minimal\text{-}ELIQ}$ returns $q' = p$.

The following lemma collects the relevant properties of $\mathsf{extract\text{-}minimal\text{-}ELIQ}$. All properties except termination are essentially consequences of the definition of the subroutine. The proof of termination after polynomially many steps relies on Theorem 13 below.

Lemma 11. *Let q be a unary CQ with $q \subseteq_{\mathcal{O}} q_T$ that is satisfiable w.r.t. $\mathcal{O}$. Then, $\mathsf{extract\text{-}minimal\text{-}ELIQ}(q)$ 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 $q_1, q_2, \dots$ of CQs is a* generalization sequence towards q_T under $\mathcal{O}$ *if for all $i \geq 0$, $q_i \subseteq_{\mathcal{O}} q_{i+1} \not\subseteq_{\mathcal{O}} q_i$, $q_i \subseteq_{\mathcal{O}} q_T$, and $\mathsf{sig}(q_i) \subseteq \mathsf{sig}(\mathcal{O})$.*

Let $q_1, q_2, \dots$ be the sequence of ELIQs that are assigned to q_H during the run of the algorithm. We show inductively that $q_1, q_2, \dots$ is a generalization sequence towards the target query q_T under $\mathcal{O}$. For the base case, note that $\mathsf{extract\text{-}minimal\text{-}ELIQ}$ (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 $(\mathcal{A}, a)$ 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 $q_{\mathcal{A}} \not\subseteq_{\mathcal{O}} q_H$. The subroutine $\mathsf{extract\text{-}minimal\text{-}ELIQ}$ generalizes $q_{\mathcal{A}}$ into a query q_D, hence $q_D \not\subseteq_{\mathcal{O}} q_H$. Since q'_H is a q_D-guided ELIQ-generalization of q_H, we have $q_H \subseteq q'_H$ (Condition 1 of Definition 4), $q'_H \not\subseteq q_H$ (Condition 2), and $q'_H \subseteq_{\mathcal{O}} q_T$ (Condition 3). It remains to note that $\mathsf{extract\text{-}minimal\text{-}ELIQ}$ 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 $\mathcal{ELIF}$ ontology in normal form, and let $q_1, q_2, \dots$ 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 $|\mathsf{var}(q_T)|^3 \cdot |\mathsf{sig}(\mathcal{O})|$.*

It remains to show that the $\mathsf{extract\text{-}minimal\text{-}ELIQ}$ subroutine terminates after polynomially many steps. For this, consider the sequence $p_1, p_2, \dots$ of queries that *Double cycle* is applied to during a run of $\mathsf{extract\text{-}minimal\text{-}ELIQ}$. 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 $p_1, p_2, \dots$ 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 $\mathcal{EL}$ of $\mathcal{ELI}$ as ontology language [22] and we conjecture the same to be true also for many *DL-Lite* dialects, e.g., *DL-Lite*$^{\mathcal{F}}_{horn}$. However, the extension to unrestricted, that is, possibly non-rooted, CQs is not possible for $\mathcal{ELI}$. 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 $\mathcal{ELI}$ ontology that contains the following concept inclusions, for all i with $1 \leq i \leq n$:*

$$\begin{aligned} &\overline{X_i} \sqsubseteq \exists r.\top \\ \exists r^-.(X_0 \sqcap \cdots \sqcap X_{i-1} \sqcap \overline{X_i}) &\sqsubseteq X_i \qquad\qquad & \exists r^-.(X_0 \sqcap \cdots \sqcap X_{i-1} \sqcap X_i) \sqsubseteq \overline{X_i} \\ \exists r^-.\overline{X_i} \sqcap \overline{X_j} &\sqsubseteq \overline{X_i} & \exists r^-.X_i \sqcap \overline{X_j} \sqsubseteq X_i \end{aligned}$$

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*$^{\mathcal{F}-}_{horn}$ 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*$^{\mathcal{F}}_{horn}$, CQs and any *DL-Lite* dialect, ELIQs and *DL-Lite*$^{\mathcal{H}}$, 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 $\mathcal{ELI}$ 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 $\mathcal{ELI}$ ontologies, then ELIQs would be polynomial query learnable under $\mathcal{ELI}$ 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*$^{\mathcal{F}}_{horn}$ (or even *DL-Lite*) ontologies, and whether ELIQs are efficiently learnable under *DL-Lite*$^{\mathcal{F}}_{horn}$ or *DL-Lite*$^{\mathcal{H}}$ ontologies.

References

[1] W. W. Cohen, H. Hirsh, The learnability of description logics with equality constraints, Mach. Learn. 17 (1994) 169–199.

[2] W. W. Cohen, H. Hirsh, Learning the classic description logic: Theoretical and experimental results, in: Proc. of KR, Morgan Kaufmann, 1994, pp. 121–133.

[3] M. Frazier, L. Pitt, Classic learning, Mach. Learn. 25 (1996) 151–193.

[4] F. Baader, Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles, in: Proc. of IJCAI, Morgan Kaufmann, 2003, pp. 319–324.

[5] F. Baader, R. Küsters, R. Molitor, Computing least common subsumers in description logics with existential restrictions, in: Proc. of IJCAI, Morgan Kaufmann, 1999, pp. 96–103.

[6] F. Baader, B. Sertkaya, A. Turhan, Computing the least common subsumer w.r.t. a background terminology, J. Appl. Log. 5 (2007) 392–420.

[7] J. C. Jung, C. Lutz, F. Wolter, Least general generalizations in description logic: Verification and existence, in: Proc. of AAAI, 2020, pp. 2854–2861.

[8] B. Zarrieß, A. Turhan, Most specific generalizations w.r.t. general $\mathcal{EL}$-TBoxes, in: Proc. of IJCAI, 2013, pp. 1191–1197.

[9] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Learning description logic concepts: When can positive and negative examples be separated?, in: Proc. of IJCAI, 2019, pp. 1682–1688.

[10] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Logical separability of incomplete data under ontologies, in: Proc. of KR, 2020, pp. 517–528.

[11] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, Mach. Learn. 78 (2010) 203–250.

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

[13] M. K. Sarker, P. Hitzler, Efficient concept induction for description logics, in: Proc. of AAAI, 2019, pp. 3036–3043.

[14] V. Gutiérrez-Basulto, J. C. Jung, L. Sabellek, Reverse engineering queries in ontology-enriched systems: The case of expressive Horn description logic ontologies, in: Proc. of IJCAI-ECAI, ijcai.org, 2018, pp. 1847–1853.

[15] D. Angluin, Learning regular sets from queries and counterexamples, Inf. Comput. 75 (1987) 87–106.

[16] D. Angluin, Queries and concept learning, Mach. Learn. 2 (1987) 319–342.

[17] B. Konev, A. Ozaki, F. Wolter, A model for learning description logic ontologies based on exact learning, in: Proc. of AAAI, AAAI Press, 2016, pp. 1008–1015.

[18] B. Konev, C. Lutz, A. Ozaki, F. Wolter, Exact learning of lightweight description logic ontologies, J. Mach. Learn. Res. 18 (2018) 1–63.

[19] A. Ozaki, C. Persia, A. Mazzullo, Learning query inseparable $\mathcal{ELH}$ ontologies, in: Proc. of AAAI, 2020, pp. 2959–2966.

[20] A. Ozaki, Learning description logic ontologies: Five approaches. where do they stand?, KI - Künstliche Intelligenz (2020).

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

ability, in: Proc. of ICDT, volume 186 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 9:1–9:24.

[22] M. Funk, J. C. Jung, C. Lutz, Actively learning concept and conjunctive queries under $\mathcal{EL}^r$-ontologies, in: Proc. of IJCAI, 2021, pp. 1887–1893.

[23] M. Funk, J. C. Jung, C. Lutz, Actively learning ELI queries under DL-Lite ontologies, in: Proc. of DL 2021), volume 2954 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021.

[24] M. Funk, J. C. Jung, C. Lutz, Frontiers and exact learning of $\mathcal{ELI}$ queries under DL-Lite ontologies, in: Proc. of IJCAI, 2022.

[25] B. ten Cate, P. G. Kolaitis, K. Qian, W. Tan, Active learning of GAV schema mappings, in: Proc. of PODS, 2018, pp. 355–368. doi:10.1145/3196959.3196974.

[26] B. ten Cate, V. Dalmau, P. G. Kolaitis, Learning schema mappings, ACM Trans. Database Syst. 38 (2013) 28:1–28:31.

[27] M. Fortin, B. Konev, V. Ryzhikov, Y. Savateev, F. Wolter, M. Zakharyaschev, Unique characterisability and learnability of temporal instance queries, in: Proc. of KR, 2022.

[28] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logics, Cambridge University Press, 2017.

[29] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-Lite family and relations, J. Artif. Intell. Res. 36 (2009) 1–69.

[30] M. Bienvenu, M. Ortiz, M. Simkus, G. Xiao, Tractable queries for lightweight description logics, in: Proc. of IJCAI, 2013, pp. 768–774.

[31] F. Kriegel, Navigating the $\mathcal{EL}$ subsumption hierarchy, in: M. Homola, V. Ryzhikov, R. A. Schmidt (Eds.), Proc. of DL, volume 2954 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021.

[32] E. Botoeva, R. Kontchakov, V. Ryzhikov, F. Wolter, M. Zakharyaschev, Games for query inseparability of description logic knowledge bases, Artif. Intell. 234 (2016) 78–119.

A. Additional Preliminaries

A.1. Direct Product

The *direct product* of interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ is the interpretation $\mathcal{I}_1 \times \mathcal{I}_2$ defined as

$$\begin{aligned}&\{\top(a_1, a_2) \mid a_i \in \Delta^{\mathcal{I}_i} \text{ for } i \in \{1,2\}\} \cup\\ &\{A(a_1, a_2) \mid A(a_i) \in \mathcal{I}_i \text{ for } i \in \{1,2\}\} \cup\\ &\{r((a_1, a_2), (b_1, b_2)) \mid r(a_i, b_i) \in \mathcal{I}_i \text{ for } i \in \{1,2\}\}.\end{aligned}$$

A.2. Semantics of Conjunctive Queries

The semantics of CQs is given in terms of homomorphisms as usual. 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 $A(d) \in \mathcal{I}_1$ implies $A(h(d)) \in \mathcal{I}_2$ and $r(d, e) \in \mathcal{I}_1$ implies $r(h(d), h(e)) \in \mathcal{I}_2$. We use $\mathsf{img}(h)$ to denote the set $\{h(d) \mid d \in \Delta^{\mathcal{I}_1}\}$. For tuples $\bar{d}_i$ over $\Delta^{\mathcal{I}_i}$, $i \in \{1, 2\}$, we write $(\mathcal{I}_1, \bar{d}_1) \to (\mathcal{I}_2, \bar{d}_2)$ if there is a homomorphism h from $\mathcal{I}_1$ to $\mathcal{I}_2$ with $h(\bar{d}_1) = \bar{d}_2$. With a homomorphism from a CQ q to an interpretation $\mathcal{I}$, we mean a homomorphism from $\mathcal{A}_q$ to $\mathcal{I}$.

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

A.3. Universal model

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

(i') $a \rightsquigarrow^{R_1}_{\mathcal{A},\mathcal{O}} M_1$ and if $\mathsf{func}(R_1) \in \mathcal{O}$, then there is no $b \in \mathsf{ind}(\mathcal{A})$ with $R_1(a, b) \in \mathcal{A}$,

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

Let $\mathbf{T}$ denote the set of all traces for $\mathcal{A}$ and $\mathcal{O}$. We always assume that $\mathbf{T}$ depends on the ontology language: if $\mathcal{O}$ is in *DL-Lite*$^{\mathcal{F}}_{horn}$, then traces are defined as in the main body of the paper, otherwise, they are defined as above.

Then the *universal model* $\mathcal{U}_{\mathcal{A},\mathcal{O}}$ of $\mathcal{A}$ and $\mathcal{O}$ is

$$\begin{aligned}\mathcal{U}_{\mathcal{A},\mathcal{O}} = \mathcal{A} &\cup \{A(a) \mid \mathcal{A}, \mathcal{O} \models A(a)\} \cup \{A(tRM) \mid tRM \in \mathbf{T} \text{ and } A \in M\} \cup\\ &\{R(t, tRM) \mid tRM \in \mathbf{T}\}.\end{aligned}$$

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 '$\rightsquigarrow$' are important to ensure that all functionality assertions in $\mathcal{O}$ 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 $\bar{a} \in \mathsf{ind}(\mathcal{A})^{|\bar{x}|}$.*

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 $q(\bar{x}) \to (\mathcal{U}_{p,\mathcal{O}}, \bar{x})$.*

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 $h\colon q(\bar{x}) \to (\mathcal{U}_{p,\mathcal{O}}, \bar{y})$ is also a homomorphism $q(\bar{x}) \to (\mathcal{U}_{p|_U,\mathcal{O}}, \bar{y})$ where $U = \mathsf{img}(h) \cap \mathsf{var}(p)$.*

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

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, $\mathcal{A}_p, \mathcal{O} \models \exists R. \bigsqcap M(y)$ if and only if $\mathcal{A}_{p|_U}, \mathcal{O} \models \exists R. \bigsqcap M(y)$ 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 $x \in \mathsf{var}(q)$ 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 $h\colon q(\bar{x}) \to (\mathcal{U}_{p|_U,\mathcal{O}}, \bar{y})$ as required. ❑

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 $h\colon q(\bar{x}) \to (\mathcal{U}_{q,\mathcal{O}}, \bar{x})$ is a homomorphism $q(\bar{x}) \to (q, \bar{x})$.*

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

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 $h\colon q(\bar{x}) \to (\mathcal{U}_{\mathcal{A},\mathcal{O}}, \bar{a})$ for some tuple $\bar{a}$ over $\mathsf{ind}(\mathcal{A})$ can be extended to a homomorphism $h'\colon (\mathcal{U}_{q,\mathcal{O}}, \bar{x}) \to (\mathcal{U}_{\mathcal{A},\mathcal{O}}, \bar{a})$.*

B. Proofs for Section 3

Lemma 21. *The computation of $\widehat{q}$ terminates after polynomially many steps.*

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

- the root ε 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 $w_0 \dots w_k$ from n_1 to n "goes up" in the tree and the path $v_0 \dots v_m$ from n to n_2 "goes down".[5] Consider the following sequence $\alpha_0, \dots, \alpha_{k+m-1}$ of facts:

(a) for $0 \leq i < k$, let α_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 $\alpha_0, \dots, \alpha_{k+m-1}$ 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 α_i and α_{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 α_i, α_{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 (α_i, α_{i+1} were both defined in (b) or α_i was defined in (a) and α_{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 ρ 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. ❑

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

Proof. We show that $\widehat{q}$ satisfies Conditions 1 to 3 from Definition 4 and that $\widehat{q}$ 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 $(z,t) \in \mathsf{var}(\widehat{q})$ with $z \in \mathsf{var}(q)$ and $t \in \Delta^{\mathcal{U}_{p,\mathcal{O}}}$;
- $g(\widehat{z}) = zRM$, for every variable $\widehat{z}$ 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 $\widehat{q}(x_1, x_2) \to (\mathcal{U}_{q,\mathcal{O}}, x_1)$, and thus $q \subseteq_{\mathcal{O}} \widehat{q}$.

Satisfiability of $\widehat{q}$ 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 $\widehat{q}$ to $\mathcal{U}_{q,\mathcal{O}}$, and that since q satisfies all functionality assertions in $\mathcal{O}$, by construction so does $\widehat{q}$. For the latter, it is important that $\mathcal{O}$ is formulated in *DL-Lite*$^{\mathcal{F}-}_{horn}$ rather than in *DL-Lite*$^{\mathcal{F}}_{horn}$. 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 $\widehat{q} \subseteq_{\mathcal{O}} q$. Since $\widehat{q}$ is satisfiable, we can fix a homomorphism $h\colon q(x_1) \to (\mathcal{U}_{\widehat{q},\mathcal{O}}, (x_1, x_2))$. By Lemma 20, there is an extension of the homomorphism g to a homomorphism g' for $(\mathcal{U}_{\widehat{q},\mathcal{O}}, (x_1, x_2)) \to (\mathcal{U}_{q,\mathcal{O}}, x_1)$. Then, the composition of h and g' is a homomorphism $q(x_1) \to (\mathcal{U}_{q,\mathcal{O}}, x_1)$. Lemma 19 implies that variables $\widehat{z}$ introduced in (A2) are not in the image of h. Indeed, if $h(x) = \widehat{z}$ for some $x \in \mathsf{var}(q)$, 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 $z \in \mathsf{var}(q)$ and some trace t in $\mathcal{U}_{p,\mathcal{O}}$. Consider the projection h' of h to its second component, that is

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

It is routine to show that h' is a homomorphism $q(x_1) \to (\mathcal{U}_{p,\mathcal{O}}, x_2)$, 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 $h_1\colon q'(x_0) \to (\mathcal{U}_{q,\mathcal{O}}, x_1)$ and $h_2\colon q'(x_0) \to (\mathcal{U}_{p,\mathcal{O}}, x_2)$. 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 $z \in \mathsf{var}(q)$ 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 $z' \in \mathsf{var}(q)$. Then (A1) is applicable to z, z', t, t', and we find $R((z,t),(z',t')) \in \widehat{q}$. Set $h(x') = (z',t')$.

2. Otherwise, $z' \notin \mathsf{var}(q)$. Since $z \in \mathsf{var}(q)$, z' takes the form zRM for some M and thus $z \rightsquigarrow^R_{q,\mathcal{O}} M$ for that M. Then (A2) is applicable to (z,t) and R. Let $\widehat{z}$ be the variable introduced in (A2). Using the definition of the universal model, one can show that there is a homomorphism f from $\mathcal{U}_{q,\mathcal{O}}$ to $\mathcal{U}_{\widehat{q},\mathcal{O}}$ which maps zRM to $\widehat{z}$ and q to the copy of q that was added to $\widehat{q}$ 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 $q'(x_0) \to (\mathcal{U}_{\widehat{q},\mathcal{O}}, (x_1, x_2))$, and thus $\widehat{q} \subseteq_{\mathcal{O}} q'$.

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

- If $h(x)$ was defined in Step 1 above, then $h(x) = (z,t)$ for $z = h_1(x) \in \mathsf{var}(q)$ and $t = h_2(x) \in \mathsf{ind}(\mathcal{U}_{p,\mathcal{O}})$. Since both h_1 and h_2 are homomorphisms, both $A(z) \in \mathcal{U}_{q,\mathcal{O}}$ and $A(t) \in \mathcal{U}_{p,\mathcal{O}}$. Thus $A(h(x)) = A(z,t) \in \widehat{q} \subseteq \mathcal{U}_{q,\mathcal{O}}$.
- If $h(x)$ was defined in Step 2 above, then $h(x) = f(h_1(x))$ where f is a homomorphism from q to $\mathcal{U}_{\widehat{q},\mathcal{O}}$. Since additionally, h_1 is a homomorphism, it follows that $A(h(x)) \in \mathcal{U}_{\widehat{q},\mathcal{O}}$.

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), $R((h_1(x), h_2(x)), (h_1(y), h_2(y))) \in \widehat{q}$ and thus $R(h(x), h(y)) \in \widehat{q} \subseteq \mathcal{U}_{\widehat{q},\mathcal{O}}$.
- 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 $\mathcal{U}_{\widehat{q},\mathcal{O}}$. Since additionally, h_1 is a homomorphism, it follows that $R(h(x), h(y)) \in \mathcal{U}_{\widehat{q},\mathcal{O}}$.
- 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 $h(y) = f(h_1(y)) = \widehat{z}$ for the element $\widehat{z}$ that was introduced in the application of (A2) to (z,t) that defined $h(y)$. (A2) additionally implies that $R((z,t), \widehat{z}) \in \widehat{q}$, and thus $R(h(x), h(y)) \in \mathcal{U}_{\widehat{q},\mathcal{O}}$.
- 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'.

❏

C. Proofs for Section 4

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

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

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

Lemma 23.

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

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

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

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

Given a *DL-Lite$^{\mathcal{F}}_{horn}$* ontology $\mathcal{O}$ and a signature $\Sigma = \mathsf{sig}(\mathcal{O})$ with $\mathsf{sig}(q_T) \subseteq \Sigma$, algorithm L first computes the ontology $\mathcal{O}'$ in normal form as per Lemma 23, choosing the fresh concept names so that they are not from Σ. It then runs L' on $\mathcal{O}'$ and $\Sigma' = \Sigma \cup \mathsf{sig}(\mathcal{O}')$. In contrast to L', the oracle still works with the original ontology $\mathcal{O}$. 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 $\mathcal{A}', \mathcal{O}' \models q_T(a)$, we may assume that $\mathcal{A}'$ satisfies the functionality assertions from $\mathcal{O}$, since otherwise the answer is trivially "yes". Then, L asks the membership query $\mathcal{A}, \mathcal{O} \models q_T(a)$, where $\mathcal{A}$ is obtained from $\mathcal{A}'$ by starting with $\mathcal{A} = \mathcal{A}'$ and then extending it as follows:

($*$) for every $X_{\exists R}(b) \in \mathcal{A}'$ with $\mathcal{A} \not\models \exists R(b)$, add $R(b, b')$ 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. $\mathcal{A}', \mathcal{O}' \models q(a)$ iff $\mathcal{A}, \mathcal{O} \models q(a)$ for all ELIQs q that only use symbols from Σ, and all $a \in \mathsf{ind}(\mathcal{A}')$.

Proof of the Claim 1. For "if", suppose that $\mathcal{A}, \mathcal{O} \models q(a)$ and let $\mathcal{I}'$ be a model of $\mathcal{A}'$ and $\mathcal{O}'$. We can assume that $\Delta^{\mathcal{I}'}$ 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, $X_{\exists R}(b) \in \mathcal{A}'$ and, by definition of the normal form, $X_{\exists R} \sqsubseteq \exists R \in \mathcal{O}'$. 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 $\mathsf{func}(R^-) \in \mathcal{O}$. 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 $a_0 R_1 a_1 \dots R_n a_n$ 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 $0 \leq i < n$;
- if $\mathsf{func}(R_i^-) \in \mathcal{O}$, then $R_{i+1} \neq R_i^-$, for all i with $0 \leq i < n$;
- if $R_1 = R^-$ then $\mathsf{func}(R^-) \notin \mathcal{O}$.

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

$$\begin{aligned} A^{\mathcal{J}_c} &= \{a_0 R_1 a_1 \dots R_n a_n \in \Delta^{\mathcal{J}_c} \mid a_n \in A^{\mathcal{I}}\} && \text{for all } A \in \mathsf{N_C}; \\ r^{\mathcal{J}_c} &= \{(\pi, \pi r a) \mid \pi r a \in \Delta^{\mathcal{I}_d}\} \cup \\ &\quad\ \{(\pi r^- a, \pi) \mid \pi r^- a \in \Delta^{\mathcal{J}_c}\} && \text{for all } r \in \mathsf{N_R}. \end{aligned}$$

Note that each $\mathcal{J}_c$ has a homomorphism into $\mathcal{I}'$: just map every sequence $a_0 R_1 \dots a_n$ 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 Σ. 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 $X_{\exists R}(x)$ 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 Σ 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 $\mathsf{sig}(\mathcal{O}')$ and let q be obtained from q' by replacing each assertion $X_{\exists R}(x)$ with an assertion $R(x, x')$, for a fresh variable x'. Then $\mathcal{A}|_\Sigma, \mathcal{O}' \models q'(\bar{a})$ iff $\mathcal{A}, \mathcal{O} \models q(\bar{a})$ for all ABoxes $\mathcal{A}$.

Proof of Claim 2. For “if”, suppose $\mathcal{A}, \mathcal{O} \models q(\bar{a})$ and let $\mathcal{I}$ be a model of $\mathcal{A}|_\Sigma$ and $\mathcal{O}'$. Since q and $\mathcal{O}$ contain only symbols from Σ, $\mathcal{A}|_\Sigma, \mathcal{O} \models q(\bar{a})$. Since $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$, $\mathcal{I}$ is also a model of $\mathcal{O}$. Thus $\mathcal{I} \models q(\bar{a})$ and by Point 3 of Lemma 23, $\mathcal{I} \models q'(\bar{a})$ follows as required.

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

Theorem 13. *Let q_T be a rooted CQ and $\mathcal{O}$ an $\mathcal{ELIF}$ ontology in normal form, and let $q_1, q_2, \dots$ 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 $|\mathsf{var}(q_T)|^3 \cdot |\mathsf{sig}(\mathcal{O})|$.*

Proof. Let $\mathcal{O}$ be an $\mathcal{ELIF}$ ontology in normal form and q_T a rooted CQ. Further, let $q_1(\bar{x}_1), \dots, q_n(\bar{x}_n)$ 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 $1 \leq i \leq n$, $|\mathsf{var}(q_i)| \leq |\mathsf{var}(q_T)|$ and q_i is rooted.

Proof of Claim 1. Assume to the contrary that $|\mathsf{var}(q_i)| > |\mathsf{var}(q_T)|$ or q_i is not rooted. Since $q_i \subseteq_\mathcal{O} q_T$, there is a homomorphism $h\colon q_T(\bar{x}) \to (\mathcal{U}_{q_i,\mathcal{O}}, \bar{x}_i)$. If $|\mathsf{var}(q_i)| > |\mathsf{var}(q_T)|$, there is an $x \in \mathsf{var}(q_i)$ with $x \notin \mathsf{img}(h)$. The same is true if q_i is not rooted, since q_T is rooted. By Lemma 18, h is also a homomorphism witnessing $q_T(\bar{x}) \to (\mathcal{U}_{q_i|_U,\mathcal{O}}, \bar{x}_i)$ where $U = \mathsf{img}(h) \cap \mathsf{var}(q_i)$. 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 $q_1 \subseteq_\mathcal{O} q_i$, 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 $h_i\colon q_{i+1}(\bar{x}_{i+1}) \to (\mathcal{U}_{q_i,\mathcal{O}}, \bar{x}_i)$.

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

Proof of Claim 2. Since $\mathsf{var}(q_i) \subseteq \mathsf{img}(h_i)$ implies $|\mathsf{var}(q_i)| \leq |\mathsf{var}(q_{i+1})|$, it suffices to show the former. Assume to the contrary that there is an $x \in \mathsf{var}(q_i)$ with $x \notin \mathsf{img}(h_i)$. By Lemma 18, h is also a homomorphism from $q_{i+1}(\bar{x}_{i+1}) \to (\mathcal{U}_{q_i|_U,\mathcal{O}}, \bar{x}_i)$ with $U = \mathsf{img}(h) \cap \mathsf{var}(q_i)$. Let h' be the extension of h to a homomorphism from $\mathcal{U}_{q_{i+1},\mathcal{O}}$ to $\mathcal{U}_{q_i|_U,\mathcal{O}}$ which exists by Lemma 20. Composing h' with a homomorphism $g\colon q_T(\bar{x}) \to (\mathcal{U}_{q_{i+1},\mathcal{O}}, \bar{x}_{i+1})$ yields a homomorphism $g'\colon q_T(\bar{x}) \to (\mathcal{U}_{q_i|_U,\mathcal{O}}, \bar{x}_i)$. 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 $n \leq 2 \cdot |\mathsf{var}(q_T)|^3 \cdot |\mathsf{sig}(\mathcal{O})|$. Claim 2 implies that $|\mathsf{var}(q_i)| \leq |\mathsf{var}(q_{i+1})|$ for all $i > 0$. By Claim 1, it suffices to show that the length of any subsequence $q_j, \dots, q_k$ with $|\mathsf{var}(q_j)| = \cdots = |\mathsf{var}(q_k)|$ is bounded by $|\mathsf{var}(q_T)|^2 \cdot |\mathsf{sig}(\mathcal{O})|$.

Consider any $i \in \{j, \dots, k-1\}$. Since $|\mathsf{var}(q_{i+1})| = |\mathsf{var}(q_i)|$ and by Claim 2, h_i is a bijection between $\mathsf{var}(q_{i+1})$ and $\mathsf{var}(q_i)$. Since q_i 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 $N|\mathsf{var}(q_j)|^2 + M|\mathsf{var}(q_j)|$ atoms, N and M are the numbers of role names and concept names, respectively, in $\mathsf{sig}(\mathcal{O})$. Since, by Claim 1, $|\mathsf{var}(q_j)| \leq |\mathsf{var}(q_T)|$, the length of the sequence $q_j, \dots, q_k$ is bounded by

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

❏

Definition 24. *An $\mathcal{ELI}$-*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 $A \in \mathsf{N_C}$: if $A(d_1) \in \mathcal{I}_1$, then $A(d_2) \in S$;*
2. *for all $r \in \mathsf{N_R}$ 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 $\mathcal{ELIF}$ 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 $\mathcal{ELI}$-simulation S from $\mathcal{A}_1$ to $\mathcal{A}_2$ with $(a_1, a_2) \in S$, then $\mathcal{A}_1, \mathcal{O} \models q(a_1)$ implies $\mathcal{A}_2, \mathcal{O} \models q(a_2)$.*

Lemma 14. *The sequence $p_1, p_2, \dots$ 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 precondition 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 $x \in \mathsf{var}(p_i)$ 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.

Now assume that $p_i \subseteq_{\mathcal{O}} q_T$ and thus $\mathcal{A}_{p_i}, \mathcal{O} \models q_T(x_0)$. Let again 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 $p_i' \subseteq_{\mathcal{O}} q_T$. By construction of p_i', the relation

$$S = \{(h_i(x), x) \mid x \in \mathsf{var}(p_i')\},$$

where h_i is the natural homomorphism from p_i' to p_i, is an $\mathcal{ELI}$-simulation from $\mathcal{A}_{p_i}$ to $\mathcal{A}_{p_i'}$ with $(x_0, x_0) \in S$. Thus, $\mathcal{A}_{p_i'}, \mathcal{O} \models q_T(x_0)$ by Lemma 25, and $p_i' \subseteq_{\mathcal{O}} q_T$ as required.

It remains to show that $p_{i+1} \not\subseteq_{\mathcal{O}} p_i$ for $1 \leq i < n$. Similarly to what was done above, it suffices to show that $p_i' \not\subseteq_{\mathcal{O}} p_i$ where p_i' is the result of applying *Double cycle* to p_i. Assume to the contrary that $p_i' \subseteq_{\mathcal{O}} p_i$ for some i. Then there is a homomorphism $g\colon p_i(x_0) \to (\mathcal{U}_{p_i',\mathcal{O}}, x_0)$. Composing g with the extension h_i^+ of the natural homomorphism h_i to a homomorphism from $\mathcal{U}_{p_i'}$ to $\mathcal{U}_{p_i}$, which exists by Lemma 20, yields a homomorphism $\widehat{g}\colon p_i(x_0) \to (\mathcal{U}_{p_i,\mathcal{O}}, x_0)$.

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

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

Since Γ 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 $\{\widehat{g}^j(y_1), \ldots, \widehat{g}^j(y_n)\} = \{y_1, \ldots, y_n\}$. Recall that $\widehat{g}$ is the composition of the homomorphisms h_i^+ and g. Since $(q_T, \mathcal{O})$-minimality of p_i implies that $\widehat{g}$ is injective, g must also be injective. Thus, the composition g' of $\widehat{g}^{j-1}$ 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_j and y_j'. 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 $(q_T, \mathcal{O})$-minimality of p_i. ❏