Conjunctive Regular Path Queries in Lightweight Description Logics

Meghyn Bienvenu
Laboratoire de Recherche en Informatique
CNRS & Université Paris Sud, France

Magdalena Ortiz and **Mantas Šimkus**
Institute of Information Systems
Vienna University of Technology, Austria

Abstract

Conjunctive regular path queries are an expressive extension of the well-known class of conjunctive queries and have been extensively studied in the database community. Somewhat surprisingly, there has been little work aimed at using such queries in the context of description logic (DL) knowledge bases, and all existing results target expressive DLs, even though lightweight DLs are considered better-suited for data-intensive applications. This paper aims to bridge this gap by providing algorithms and tight complexity bounds for answering two-way conjunctive regular path queries over DL knowledge bases formulated in lightweight DLs of the DL-Lite and $\mathcal{EL}$ families.

1 Introduction

Recent years have seen a rapidly growing interest in using description logic (DL) ontologies to query instance data. In databases, similar attention has been paid to the related problem of querying graph databases which, like DL instance data, are sets of ground facts using only unary and binary predicates, i.e., node- and edge-labeled graphs [Consens and Mendelzon, 1990; Barceló *et al.*, 2010]. The relevance of both problems lies in the fact that in many application areas, data can be naturally represented in such form. This applies, in particular, to XML and RDF data. While the DL and database communities share some common research goals, the research agendas they have pursued differ significantly. In DLs, the focus has been on designing efficient algorithms for answering (plain) conjunctive queries in the presence of ontological constraints. By contrast, work on graph databases typically does not consider ontological knowledge, but instead aims at supporting expressive query languages, like regular path queries (RPQs) and their extensions, which enable sophisticated navigation of paths. Such path navigation has long been considered crucial for querying data on the web. Indeed, it lies at the core of the XPath language for querying XML data, and of the *property paths* feature of SPARQL 1.1, the language recently recommended as the new standard for querying RDF data.

In this paper, we are interested in the problem of querying DL knowledge bases using various kinds of regular path queries. We mainly focus on *conjunctive (two-way) regular path queries* (C(2)RPQs), which are one of the most expressive and popular languages for querying graph databases. CRPQs simultaneously extend plain conjunctive queries (CQs) and basic RPQs: they allow conjunctions of atoms that can share variables in arbitrary ways, where the atoms may contain regular expressions that navigate the arcs of the database (or roles, in DL parlance). In the case of 2RPQs and C2RPQs, roles can be navigated in both directions. C2RPQs have already been studied for DLs, but all existing results concern expressive DLs for which reasoning is provably intractable. In particular, algorithms have been proposed for $\mathcal{ZIQ}$, $\mathcal{ZIO}$, and $\mathcal{ZOQ}$ [Calvanese *et al.*, 2007b; 2009], for which query answering is 2-ExpTime hard. Even in data complexity, that is, when the query and ontology are assumed fixed, these algorithms need exponential time. More recently, algorithms for answering C2RPQs in Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ were proposed [Ortiz *et al.*, 2011]. These algorithms run in polynomial time in the size of the data, but still require exponential time in the size of the ontology. By contrast, to the best of our knowledge, path queries have not yet been considered for the lightweight DLs of the DL-Lite [Calvanese *et al.*, 2007a] and $\mathcal{EL}$ [Baader *et al.*, 2005] families, which underly the OWL 2 QL and EL profiles. This is surprising given that the low complexity of these DLs makes them better suited for data-intensive applications.

This paper aims to remedy this situation by providing algorithms and precise complexity bounds for answering (C)2RPQs in the $\mathcal{EL}$ and DL-Lite families of lightweight DLs. We show that in data complexity, the query answering problem for CR2PQs is NL-complete for DL-Lite and P-complete for $\mathcal{EL}$, which in both cases is the lowest complexity that could be expected. For combined complexity, we prove PSpace-completeness for both DL-Lite and $\mathcal{EL}$, but somewhat surprisingly obtain a tractability result for 2RPQs for both DLs. All of our upper bounds apply to extensions of $\mathcal{EL}$ and DL-Lite with role inclusions. Full proofs of all results can be found in a technical report [Bienvenu *et al.*, 2013].

2 Preliminaries

We briefly recall the syntax of DL-Lite$_\mathcal{R}$ [Calvanese *et al.*, 2007a] and $\mathcal{ELH}$ [Baader *et al.*, 2005] (and relevant sublogics). As usual, we assume sets $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ of *concept names*, *role names*, and *individuals*. We will use $\overline{\mathsf{N_R}}$ to refer

to $\mathsf{N_R} \cup \{r^- \mid r \in \mathsf{N_R}\}$, and if $R \in \overline{\mathsf{N_R}}$, we use R^- to mean r^- if $R = r$ and r if $R = r^-$. An *ABox* is a set of assertions of the form $A(b)$ or $r(b, c)$, where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $b, c \in \mathsf{N_I}$. We use $\mathsf{Ind}(\mathcal{A})$ to refer to the set of individuals in $\mathcal{A}$. A *TBox* is a set of inclusions, whose form depends on the DL in question. In DL-Lite, inclusions take the form $B_1 \sqsubseteq (\neg)B_2$, where each B_i is either A (where $A \in \mathsf{N_C}$) or $\exists R$ (where $R \in \overline{\mathsf{N_R}}$). DL-Lite$_\mathcal{R}$ additionally allows role inclusions of the form $R_1 \sqsubseteq (\neg)R_2$, where $R_1, R_2 \in \overline{\mathsf{N_R}}$. DL-Lite$_\mathsf{RDFS}$ is obtained from DL-Lite$_\mathcal{R}$ by disallowing inclusions which contain negation or have existential concepts ($\exists R$) on the right-hand side. In $\mathcal{EL}$, inclusions have the form $C_1 \sqsubseteq C_2$, where C_1, C_2 are complex concepts constructed as follows: $C := \top \mid A \mid C \sqcap C \mid \exists r.C$. The DL $\mathcal{ELH}$ additionally allows role inclusions of the form $r \sqsubseteq s$, where $r, s \in \mathsf{N_R}$. Note that in $\mathcal{EL}(\mathcal{H})$ TBoxes, inverse roles are not permitted. A *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

As usual, the semantics is based upon *interpretations*, which take the form $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set and $\cdot^\mathcal{I}$ maps each $a \in \mathsf{N_I}$ to $a^\mathcal{I} \in \Delta^\mathcal{I}$, each $A \in \mathsf{N_C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and each $r \in \mathsf{N_R}$ to $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is straightforwardly extended to general concepts and roles, e.g. $(\neg A)^\mathcal{I} = \Delta^\mathcal{I} \setminus A^\mathcal{I}$, $(\exists r.C)^\mathcal{I} = \{c \mid \exists d : (c, d) \in r^\mathcal{I}, d \in C^\mathcal{I}\}$, and $(P^-)^\mathcal{I} = \{(c, d) \mid (d, c) \in P^\mathcal{I}\}$. An interpretation $\mathcal{I}$ satisfies $G \sqsubseteq H$ if $G^\mathcal{I} \subseteq H^\mathcal{I}$; it satisfies $A(a)$ (resp. $r(a, b)$) if $a^\mathcal{I} \in A^\mathcal{I}$ (resp. $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$). $\mathcal{I}$ is a *model* of $(\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ satisfies all inclusions in $\mathcal{T}$ and assertions in $\mathcal{A}$.

To simplify the presentation, we will assume that $\mathcal{ELH}$ TBoxes are in *normal form*, meaning that all concept inclusions are of one of the following forms:

$$A \sqsubseteq B \quad A_1 \sqcap A_2 \sqsubseteq B \quad A \sqsubseteq \exists r.B \quad \exists r.B \sqsubseteq A$$

with $A, A_1, A_2, B \in \mathsf{N_C} \cup \{\top\}$. It is well-known (cf. [Baader *et al.*, 2005] that for every $\mathcal{ELH}$ TBox $\mathcal{T}$, one can construct in polynomial time an $\mathcal{ELH}$ TBox $\mathcal{T}'$ in normal form (possibly using new concept names) such that $\mathcal{T}' \models \mathcal{T}$ and every model of $\mathcal{T}$ can be expanded to a model of $\mathcal{T}'$.

We use $\mathsf{sig}(\mathcal{T})$ to denote the set of all concept and role names appearing in a TBox $\mathcal{T}$. For ease of reference, we also define sets $\mathsf{BC}_\mathcal{T}$ of *basic concepts* and $\mathsf{TC}_\mathcal{T}$ of *tail concepts* for $\mathcal{T}$ as follows: $\mathsf{BC}_\mathcal{T} = \mathsf{TC}_\mathcal{T} = \mathsf{N_C} \cap \mathsf{sig}(\mathcal{T})$ if $\mathcal{T}$ is an $\mathcal{ELH}$ TBox, and $\mathsf{TC}_\mathcal{T} = \{\exists r, \exists r^- \mid r \in \mathsf{N_R} \cap \mathsf{sig}(\mathcal{T})\}$ and $\mathsf{BC}_\mathcal{T} = (\mathsf{N_C} \cap \mathsf{sig}(\mathcal{T})) \cup \mathsf{TC}_\mathcal{T}$ for a DL-Lite$_\mathcal{R}$ TBox $\mathcal{T}$.

Canonical Models We recall the definition of canonical models for DL-Lite$_\mathcal{R}$ and $\mathcal{ELH}$ KBs. For both DLs, the domain of the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ for a KB $(\mathcal{T}, \mathcal{A})$ consists of *paths* of the form $aR_1C_1 \dots R_nC_n$ ($n \geq 0$), where $a \in \mathsf{Ind}(\mathcal{A})$, each C_i is a tail concept, and each R_i a (possibly inverse) role. When $\mathcal{T}$ is a DL-Lite$_\mathcal{R}$ TBox, the domain $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ contains exactly those paths $aR_1\exists R_1^- \dots R_n\exists R_n^-$ which satisfy:
- if $n \geq 1$, then $\mathcal{T}, \mathcal{A} \models \exists R_1(a)$;
- for $1 \leq i < n$, $\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$.

When $\mathcal{T}$ is an $\mathcal{ELH}$ TBox in normal form, the domain $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ contains exactly those paths $ar_1A_1 \dots r_nA_n$ for which each $r_i \in \mathsf{N_R}$, and:
- if $n \geq 1$, then $\mathcal{T}, \mathcal{A} \models \exists r_1.A_1(a)$;
- for $1 \leq i < n$, $\mathcal{T} \models A_i \sqsubseteq \exists r_{i+1}.A_{i+1}$.

We denote the last concept in a path p by $\mathsf{tail}(p)$, and define $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ by taking:

$$\begin{aligned}
a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= a \text{ for all } a \in \mathsf{Ind}(\mathcal{A}) \\
A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \{a \in \mathsf{Ind}(\mathcal{A}) \mid \mathcal{T}, \mathcal{A} \models A(a)\} \\
&\quad \cup \{p \in \Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} \setminus \mathsf{Ind}(\mathcal{A}) \mid \mathcal{T} \models \mathsf{tail}(p) \sqsubseteq A\} \\
r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \{(a, b) \mid r(a, b) \in \mathcal{A}\} \cup \\
&\quad \{(p_1, p_2) \mid p_2 = p_1 S\, C \text{ and } \mathcal{T} \models S \sqsubseteq r\} \cup \\
&\quad \{(p_2, p_1) \mid p_2 = p_1 S\, C \text{ and } \mathcal{T} \models S \sqsubseteq r^-\}
\end{aligned}$$

Note that $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ is composed of a core consisting of the ABox individuals and an *anonymous part* consisting of (possibly infinite) trees rooted at ABox individuals. We will use $\mathcal{I}_{\mathcal{T},\mathcal{A}|e}$ to denote the submodel of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ obtained by restricting the universe to paths having e as a prefix.

Regular Languages We assume the reader is familiar with regular languages, represented either by regular expressions or nondeterministic finite state automata (NFAs). An NFA over an *alphabet* Σ is a tuple $\alpha = \langle S, \Sigma, \delta, s_0, F\rangle$, where S is a finite set of *states*, $\delta \subseteq S \times \Sigma \times S$ the *transition relation*, $s_0 \in S$ the *initial state*, and $F \subseteq S$ the set of *final states*. We use $L(\alpha)$ to denote the language defined by an NFA α, and when the way a regular language is represented is not relevant, we denote it simply by L.

3 Path Queries

We now introduce the query languages studied in this paper.

Definition 1. A *conjunctive (two-way) regular path query (C2RPQ)* has the form $q(\vec{x}) = \exists \vec{y}\ \varphi$ where $\vec{x}$ and $\vec{y}$ are tuples of variables, and φ is a conjunction of atoms of the forms:

(i) $A(t)$, where $A \in \mathsf{N_C}$ and $t \in \mathsf{N_I} \cup \vec{x} \cup \vec{y}$, and

(ii) $L(t, t')$, where L is (an NFA or regular expression defining) a regular language over $\overline{\mathsf{N_R}} \cup \{A? \mid A \in \mathsf{N_C}\}$, and $t, t' \in \mathsf{N_I} \cup \vec{x} \cup \vec{y}$.

As usual, variables and individuals are called *terms*, and the variables in $\vec{x}$ are called *answer variables*. A query with no answer variables is called a *Boolean query*.

Conjunctive (one-way) regular path queries (CRPQs) are obtained by disallowing symbols from $\overline{\mathsf{N_R}} \setminus \mathsf{N_R}$ in atoms of type (ii), and *conjunctive queries (CQs)* result from only allowing type-(ii) atoms of the form $r(t, t')$ with $r \in \mathsf{N_R}$. *Two-way regular path queries (2RPQs)* consist of a single atom of type (ii), and *regular path queries (RPQs)* further disallow symbols from $\overline{\mathsf{N_R}} \setminus \mathsf{N_R}$. Finally, *instance queries (IQs)* take the form $A(x)$ with $A \in \mathsf{N_C}$, or $r(x, y)$ with $r \in \mathsf{N_R}$.

We now define the semantics of C2RPQs. For a regular language L over the alphabet $\mathsf{N_R} \cup \overline{\mathsf{N_R}} \cup \{A? \mid A \in \mathsf{N_C}\}$, we call d_2 an *L-successor* of d_1 in $\mathcal{I}$ if there is some $w = u_1 \dots u_n \in L$ and some sequence $e_0, \dots, e_n$ of elements in $\Delta^\mathcal{I}$ such that $e_0 = d_1$, $e_n = d_2$, and, for all $1 \leq i \leq n$:
- if $u_i = A?$, then $e_{i-1} = e_i \in A^\mathcal{I}$
- if $u_i = R \in \mathsf{N_R} \cup \overline{\mathsf{N_R}}$, then $\langle e_{i-1}, e_i\rangle \in R^\mathcal{I}$

A *match* for a Boolean C2RPQ q in an interpretation $\mathcal{I}$ is a mapping π from the terms in q to elements in $\Delta^\mathcal{I}$ such that:

	IQ		(2)RPQ		CQ		C(2)RPQ	
	data	combined	**data**	**combined**	data	combined	**data**	**combined**
DL-Lite$_{\mathsf{RDFS}}$	in AC_0	NL-c	**NL-c**	**NL-c**	in AC_0	NP-c	**NL-c**	**NP-c**
DL-Lite$_{(\mathcal{R})}$	in AC_0	NL-c	**NL-c**	**P-c**†	in AC_0	NP-c	**NL-c**	**PSPACE-c**
$\mathcal{EL}(\mathcal{H})$	P-c	P-c	**P-c**	**P-c**	P-c	NP-c	**P-c**	**PSPACE-c**

Figure 1: Complexity of Boolean query entailment. The 'c' indicates completeness results. New results are marked in bold. For existing results, we refer to [Baader *et al.*, 2005; Calvanese *et al.*, 2007a; Rosati, 2007; Krisnadhi and Lutz, ; Krötzsch and Rudolph, 2007; Artale *et al.*, 2009] and references therein. † P-hardness for RPQs applies only to DL-Lite$_{\mathcal{R}}$.

– $\pi(c) = c^{\mathcal{I}}$ if $c \in \mathsf{N_I}$,
– $\pi(t) \in A^{\mathcal{I}}$ for each atom $A(t)$ in q, and
– $\pi(t')$ is an L-successor of $\pi(t)$, for each $L(t, t')$ in q.

We write $\mathcal{I} \models q$ if there is a match for q in $\mathcal{I}$, and $\mathcal{T}, \mathcal{A} \models q$ if $\mathcal{I} \models q$ for every model $\mathcal{I}$ of $\mathcal{T}, \mathcal{A}$.

Given an C2RPQ q with answer variables $v_1, \ldots, v_k$, we say that a tuple of individuals $(a_1, \ldots, a_k)$ is a *certain answer* for q w.r.t. $\mathcal{T}, \mathcal{A}$ just in the case that in every model $\mathcal{I}$ of $\mathcal{T}, \mathcal{A}$ there is a match π for q such that $\pi(v_i) = a_i^{\mathcal{I}}$ for every $1 \leq i \leq k$. Deciding whether a tuple of individuals is a certain answer for an C2RPQ can be linearly reduced to Boolean C2RPQ entailment. For this reason, we consider only the latter problem in what follows.

It is well known that the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ can be homomorphically embedded into any model of $\mathcal{T}, \mathcal{A}$, hence a CQ q is entailed from $\mathcal{T}, \mathcal{A}$ if and only if there is a match for q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. This result can be easily lifted from CQs to C2RPQs, as C2RPQs are also monotonic and their matches are preserved under homomorphisms.

Lemma 2. *For every DL-Lite$_{\mathcal{R}}$ or $\mathcal{ELH}$ KB $(\mathcal{T}, \mathcal{A})$ and Boolean C2RPQ q: $\mathcal{T}, \mathcal{A} \models q$ if and only if $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q$.*

This property will be a crucial element in establishing our main theorem:

Theorem 3. *The complexity results in Figure 1 hold.*

We split the proof of this theorem into parts, with the lower bounds shown in the next section, and the (more involved) proofs of the upper bounds outlined in Section 5.

4 Lower Bounds

We start by establishing the required lower bounds.

Proposition 4. *Boolean CRPQ entailment is*

1. NL*-hard in data complexity for DL-Lite$_{\mathsf{RDFS}}$;*
2. P*-hard in data complexity for $\mathcal{EL}$;*
3. NP*-hard in combined complexity for DL-Lite$_{\mathsf{RDFS}}$;*
4. PSPACE*-hard in combined complexity for DL-Lite & $\mathcal{EL}$.*

Statements (1) and (2) hold even for RPQs.

Proof. Statement (1) follows from the analogous result for graph databases [Consens and Mendelzon, 1990]. It can be shown by a simple reduction from the NL-complete directed reachability problem: y is reachable from x in a directed graph G if and only if (x, y) is an answer to $r^*(x, y)$ w.r.t. the ABox $\mathcal{A}_G$ encoding G. Statement (2) is immediate given the P-hardness in data complexity of instance checking in $\mathcal{EL}$ [Calvanese *et al.*, 2006], and (3) follows from the well-known NP-hardness in combined complexity of CQ entailment for databases [Abiteboul *et al.*, 1995].

For statement (4), we give a reduction from the problem of emptiness of the intersection of an arbitrary number of regular languages, which is known to be PSPACE-complete [Kozen, 1977]. Let $L_1, \ldots, L_n$ be regular languages over alphabet Σ. We will use the symbols in Σ as role names, and we add a concept name A. Let $\mathcal{A} = \{A(a)\}$ and $q = \exists x\, L_1(a, x) \wedge \ldots \wedge L_n(a, x)$. For DL-Lite, we will use the following TBox: $\mathcal{T} = \{A \sqsubseteq \exists r \mid r \in \Sigma\} \cup \{\exists r^- \sqsubseteq \exists s \mid r, s \in \Sigma\}$. For $\mathcal{EL}$, we can use $\mathcal{T} = \{A \sqsubseteq \exists r.A \mid r \in \Sigma\}$. Notice that in both cases the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ consists of an infinite tree rooted at a such that every element in the interpretation has a unique r-child for each $r \in \Sigma$ (and no other children). Thus, we can associate to every domain element the word over Σ given by the unique path from a, and moreover, for every word $w \in \Sigma^*$ we can find an element e_w whose path from a is exactly w. This means that if $w \in L_1 \cap \ldots \cap L_n$, we obtain a match for q in the canonical model by mapping x to e_w. Conversely, if q is entailed, then any match in the canonical model defines a word which belongs to every L_i, which means $L_1 \cap \ldots \cap L_n$ is non-empty. □

For (2)RPQs in DL-Lite$_{\mathsf{RDFS}}$ and $\mathcal{EL}$, we inherit combined complexity lower bounds of NL and P respectively from IQs. For DL-Lite, we establish a P lower bound for 2RPQs, which contrasts with the NL-completeness of instance checking.

Proposition 5. *Boolean 2RPQ entailment in DL-Lite is P-hard in combined complexity, assuming an NFA representation of the regular language.*

Proof sketch. Consider the P-complete entailment problem in which one is given a propositional formula $T = \rho_1 \wedge \ldots \wedge \rho_m \wedge v_1$ over variables $v_1, \ldots, v_n$ with $\rho_i = v_{i_1} \wedge v_{i_2} \rightarrow v_{i_3}$, and the problem is to decide whether $T \models v_n$. We construct a DL-Lite TBox $\mathcal{T}$ and 2RPQ q such that $\mathcal{T}, \{A(a)\} \models q$ if and only if $T \models v_n$. We let $\mathcal{T}$ consist of the axioms:

– $A \sqsubseteq \exists r_{i,j}$, for $1 \leq i \leq m$, $j \in \{1, 2\}$
– $\exists r^-_{i_1,j_1} \sqsubseteq \exists r_{i_2,j_2}$, for $1 \leq i_1, i_2 \leq m$ and $j_1, j_2 \in \{1, 2\}$

and $q = \exists x\, \alpha(x, x)$, where $\alpha = (S, \Sigma, \delta, s_0, \{v_n^{out}\})$ is the NFA defined as follows:

– $S = \{s_0\} \cup \{v_1\} \cup \{v_i^{in}, v_i^{out} \mid 2 \leq i \leq n\} \cup \{\rho_i \mid 1 \leq i \leq m\}$

– $\Sigma = \{A?\} \cup \{r_{i,j}, r_{i,j}^- \mid 1 \leq i \leq m, 1 \leq j \leq 2\}$

– δ contains $(s_0, A?, v_n^{in})$, and for each $\rho_i = v_j \wedge v_k \to v_\ell$, the following transitions: $(v_\ell^{in}, r_{i,1}, v_j^{in})$, $(v_j^{out}, r_{i,1}^-, \rho_i)$, $(\rho_i, r_{i,2}, v_k^{in})$, and $(v_k^{out}, r_{i,2}^-, v_\ell^{out})$. Note: we use v_1 in place of v_1^{in} and v_1^{out}.

The first transition in δ enforces that x must be mapped to a and that there must be a loop at a from state v_n^{in} to v_n^{out}. Intuitively, a state v_ℓ^{in} indicates that v_ℓ needs to be proven, and v_ℓ^{out} signals that v_ℓ has been successfully derived. From a state v_ℓ^{in}, the available transitions correspond to the rules in T which conclude on v_ℓ: selecting transition $(v_\ell^{in}, r_{i,1}, v_j^{in})$ means choosing to use ρ_i to derive v_ℓ. The transitions $(v_\ell^{out}, r_{i,1}^-, \rho_i)$ and $(\rho_i, r_{i,2}, v_k^{in})$ allow us to move to the second variable of ρ_i once the first variable of ρ_i has been derived. When both variables have been proven, the transition $(v_k^{out}, r_{i,2}^-, v_\ell^{out})$ allows us to exit the derivation of v_ℓ. Thus, any loop from v_n^{in} to v_n^{out} in $\mathcal{I}_{\mathcal{T},\{A(a)\}}$ corresponds to a derivation of v_n, and conversely, any derivation of v_n yields a path witnessing the entailment of q. □

As a corollary, we get P-hardness of RPQs in DL-Lite$_\mathcal{R}$, by using role inclusions to simulate the inverse roles in the query. We leave open whether the preceding hardness result applies when regular languages are given as regular expressions.

5 Upper Bounds

The main objective of this section will be to define a procedure for deciding $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q$ for a KB $(\mathcal{T}, \mathcal{A})$ and C2RPQ q. The procedure comprises two main steps. First, we rewrite q into a set Q of C2RPQs such that $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q$ if and only if $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q'$ for some $q' \in Q$. The advantage of the rewritten queries is that in order to decide whether $\mathcal{I}_{\mathcal{T},\mathcal{A}} \models q'$, we will only need to consider matches which map the variables to $\mathsf{Ind}(\mathcal{A})$. The second step decides the existence of such restricted matches for the rewritten queries.

In order to more easily manipulate regular languages, it will prove convenient to use NFAs rather than regular expressions. Thus, in what follows, we assume all binary atoms take the form $\alpha(t, t')$, where α is an NFA over $\mathsf{N_R} \cup \overline{\mathsf{N_R}} \cup \{A? \mid A \in \mathsf{N_C}\}$. Given $\alpha = \langle S, \Sigma, \delta, s_0, F\rangle$, we use $\alpha_{s,G}$ to denote the NFA $\langle S, \Sigma, \delta, s, G\rangle$, i.e., the NFA with the same states and transitions as α but with initial state s and final states G.

5.1 Loop Computation

A key to defining our rewriting procedure will be to understand how an atom $L(t, t')$ can be satisfied in the anonymous part of the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. A subtlety arises from the fact that the path witnessing the satisfaction of an atom $L(t, t')$ may be quite complicated: it may move both up and down, passing by the same element multiple times, and possibly descending below t'. This will lead us to decompose an atom $L(t, t')$ into multiple "smaller" atoms corresponding to segments of the L-path which are situated wholly above or below an element. Importantly, we know that the canonical model displays a high degree of regularity, since whenever two elements p_1 and p_2 in the anonymous part end with the same concept (i.e., $\mathsf{Tail}(p_1) = \mathsf{Tail}(p_2)$), the submodels $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_{p_1}$ and $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_{p_2}$ are isomorphic. In particular, this means that if $\mathsf{Tail}(p_1) = \mathsf{Tail}(p_2)$, then p_1 is an L-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_{p_1}$ just in the case that p_2 is an L-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_{p_2}$.

We will require a way of testing for a given TBox $\mathcal{T}$ and NFA α with states s, s' whether $\mathsf{Tail}(e) = C$ ensures that there is a loop from e back to itself, situated wholly within $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_e$, which takes α from state s to state s'. To this end, we construct a table Loop_α which contains for each pair s, s' of states in α, a subset of $\mathsf{TC}_\mathcal{T}$. If $\mathcal{T}$ is a DL-Lite$_\mathcal{R}$ TBox, then Loop_α is defined inductively using the following rules:

1. for every $s \in S$: $\mathsf{Loop}_\alpha[s, s] = \mathsf{TC}_\mathcal{T}$
2. if $C \in \mathsf{Loop}_\alpha[s_1, s_2]$ and $C \in \mathsf{Loop}_\alpha[s_2, s_3]$, then $C \in \mathsf{Loop}_\alpha[s_1, s_3]$
3. if $C \in \mathsf{TC}_\mathcal{T}$, $\mathcal{T} \models C \sqsubseteq A$, and $(s_1, A?, s_2) \in \delta$, then $C \in \mathsf{Loop}_\alpha[s_1, s_2]$
4. if $\mathcal{T} \models C \sqsubseteq \exists R$, $\mathcal{T} \models R \sqsubseteq R'$, $\mathcal{T} \models R^- \sqsubseteq R''$, $(s_1, R', s_2) \in \delta$, $\exists R^- \in \mathsf{Loop}_\alpha[s_2, s_3]$, $(s_3, R'', s_4) \in \delta$, $C \in \mathsf{TC}_\mathcal{T}$, and $C \neq \exists R$, then $C \in \mathsf{Loop}_\alpha[s_1, s_4]$

For $\mathcal{ELH}$, we replace the last rule by:

4'. if $\mathcal{T} \models C \sqsubseteq \exists r.D$, $\mathcal{T} \models r \sqsubseteq r'$, $\mathcal{T} \models r \sqsubseteq r''$, $(s_1, r', s_2) \in \delta$, $D \in \mathsf{Loop}_\alpha[s_2, s_3]$, $(s_3, r''^-, s_4) \in \delta$, and $C \in \mathsf{TC}_\mathcal{T}$, then $C \in \mathsf{Loop}_\alpha[s_1, s_4]$

Example 6. Consider a DL-Lite$_\mathcal{R}$ TBox $\mathcal{T}$ containing the inclusions $B \sqsubseteq \exists r$, $\exists r^- \sqsubseteq B$, $B \sqsubseteq \exists t_1$, and $t_1 \sqsubseteq t_2^-$, and consider the query $q = \exists xy.r^* t_1 t_2 r^-(x, y)$, $B(y)$, or equivalently, $q = \exists xy.\alpha(x, y)$, $B(y)$, where $\alpha = \langle\{s_0, s_1, s_2, s_3\}, \{r, t_1, t_2, r^-\}, \delta, s_0, \{s_3\}\rangle$ and $\delta = \{(s_0, r, s_0), (s_0, t_1, s_1), (s_1, t_2, s_2), (s_2, r^-, s_3)\}$. In the first step of the loop computation, we infer that $\mathsf{Loop}_\alpha[s_i, s_i]$ is the set of all tail concepts for $0 \leq i \leq 4$. Next, by rule 4, and using $\mathcal{T} \models B \sqsubseteq \exists t_1$, $\mathcal{T} \models \exists r^- \sqsubseteq \exists t_1$, $\mathcal{T} \models t_1^- \sqsubseteq t_2$, (s_0, t_1, s_1), $\exists t_1^- \in \mathsf{Loop}_\alpha[s_1, s_1]$, and (s_1, t_2, s_2), we can infer that $B \in \mathsf{Loop}_\alpha[s_0, s_2]$ and $\exists r^- \in \mathsf{Loop}_\alpha[s_0, s_2]$. In a further step, we can use $\mathcal{T} \models B \sqsubseteq \exists r$, (s_0, r, s_0), $\exists r^- \in \mathsf{Loop}_\alpha[s_0, s_2]$, and (s_2, r^-, s_3) to obtain $B \in \mathsf{Loop}_\alpha[s_0, s_3]$.

Note that the table Loop_α can be constructed in polynomial time in $|\mathcal{T}|$ and $|\alpha|$ since entailment of inclusions is polynomial for both DL-Lite$_\mathcal{R}$ and $\mathcal{ELH}$. The following lemma shows that Loop_α has the desired meaning:

Lemma 7. *For every element $p \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \setminus \mathsf{Ind}(\mathcal{A})$: $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s']$ if and only if p is an $L(\alpha_{s,s'})$-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$.*

5.2 Query Rewriting

Our aim is to rewrite our query in such a way that we do not need to map any variables to the anonymous part of the model. We draw our inspiration from a query rewriting procedure for Horn-$\mathcal{SHIQ}$ described in [Eiter *et al.*, 2012]. The main intuition is as follows. Suppose we have a match π for q which maps some variable y to the anonymous part, and no other variable is mapped below $\pi(y)$. Then we modify q so that it has essentially the same match except that variables mapped to $\pi(y)$ are now mapped to the (unique) parent of $\pi(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. The delicate point is that we must "split"

PROCEDURE $\mathsf{rewrite}(q, \mathcal{T})$

1. Choose either to output q or to continue.
2. Choose a non-empty set $\mathsf{Leaf} \subseteq \mathsf{vars}(q)$ and $y \in \mathsf{Leaf}$. Rename all variables in Leaf to y.
3. Choose $C \in \mathsf{TC}_{\mathcal{T}}$ such that $\mathcal{T} \models C \sqsubseteq B$ whenever $B(y)$ is an atom of q. Drop all such atoms from q.
4. For each atom $\alpha(t, t')$ where $\alpha = \langle S, \Sigma, \delta, s, F\rangle$ is an NFA and $y \in \{t, t'\}$,
   - choose a sequence $s_1, \ldots, s_n$ of distinct states from S such that $s_n \in F$,
   - replace $\alpha(t, t')$ by the atoms $\alpha_{s,s_1}(t, y)$, $\alpha_{s_1,s_2}(y, y)$, $\ldots$, $\alpha_{s_{n-2},s_{n-1}}(y, y)$, $\alpha_{s_{n-1},s_n}(y, t')$.
5. Drop all atoms $\alpha_{s,s'}(y, y)$ such that $C \in \mathsf{Loop}_\alpha[s, s']$.
6. Choose some $D \in \mathsf{BC}_{\mathcal{T}}$ and $R, R_1, R_2 \in \overline{\mathsf{N_R}}$ such that:
   (a) $C = \exists R^-$ and $\mathcal{T} \models D \sqsubseteq \exists R$ [for DL-Lite$_R$], or $R \in \mathsf{N_R}$ and $\mathcal{T} \models D \sqsubseteq \exists R.C$ [for $\mathcal{ELH}$].
   (b) $\mathcal{T} \models R \sqsubseteq R_1$ and $\mathcal{T} \models R \sqsubseteq R_2$
   (c) for each atom $\alpha(y, x)$ with $\alpha = \langle S, \Sigma, \delta, s, F\rangle$, there exists $s' \in S$ such that $(s, R_1^-, s') \in \delta$.
   (d) for each atom $\alpha(x, y)$ with $\alpha = \langle S, \Sigma, \delta, s, F\rangle$, there exist $s'' \in S$, $s_f \in F$ with $(s'', R_2, s_f) \in \delta$.

   For atoms of the form $\alpha(y, y)$, both (c) and (d) apply.
7. Replace
   - each atom $\alpha(y, x)$ with $x \neq y$ by $\alpha_{s',F}(y, x)$
   - each atom $\alpha(x, y)$ with $x \neq y$ by $\alpha_{s,s''}(x, y)$
   - each atom $\alpha(y, y)$ by atoms $\alpha_{s',s''}(y, y)$

   with s, s', s'', F as in Step 6.
8. If $D \in \mathsf{N_C}$ is the concept chosen in Step 6, add $D(y)$ to q. If $D = \exists P^-$, add $\alpha_P(z, y)$ to q, where z is a fresh variable and $L(\alpha_P) = \{P\}$. Go to Step 1.

Figure 2: Query rewriting procedure rewrite.

atoms of the form $\alpha(t, t')$ with $y \in \{t, t'\}$ into the parts which are satisfied in the subtree $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_{\pi(y)}$, and those which occur above $\pi(y)$, whose satisfaction still needs to be determined and thus must be incorporated into the new query. With each iteration of the rewriting procedure, we obtain a query which has a match which maps variables "closer" to the core of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, until eventually we find some query that has a match which maps all terms to $\mathsf{Ind}(\mathcal{A})$.

In Figure 2, we give a recursive non-deterministic query rewriting procedure rewrite which implements the above intuition. Slightly abusing notation, we will use $\mathsf{rewrite}(q, \mathcal{T})$ to denote the set of queries which are output by some execution of rewrite on input q,$\mathcal{T}$.

Example 8. We illustrate two different ways to apply the rewriting step to the query $q = \exists xy.r^*t_1t_2r^-(x, y),\ B(y)$ in Example 6. First, let $\mathsf{Leaf} = \{x\}$ be the set chosen in step 2. There is no renaming to do, so we proceed to step 3 and choose $\exists r^-$. In step 4, we choose the sequence s_2, s_3, and replace $\alpha(x, y)$ by $\alpha_{s_0,s_2}(x, x), \alpha_{s_2,s_3}(x, y)$. Since $\exists r^- \in \mathsf{Loop}_\alpha[s_0, s_2]$, we drop the first atom and keep only $\alpha_{s_2,s_3}(x, y)$. In step 6, we can choose B for D, and r for the roles R, R_1, R_2. This ensures (a) and (b). For (c), we can take s_3 since $(s_2, r^-, s_3) \in \delta$. In step 7, we replace $\alpha_{s_2,s_3}(x, y)$ by $\alpha_{s_3,s_3}(x, y)$. At the end of step 8, we are left with the query $q' = \exists xyz.r(z, y), \alpha_{s_3,s_3}(x, y), B(y)$, which is output as a rewriting when we return to step 1. We remark that q' is equivalent modulo $\mathcal{T}$ to the simpler $\exists yz.r(z, y)$ since by choosing $x = y$, the atom $\alpha_{s_3,s_3}(x, y)$ is trivially satisfied, and $B(y)$ is enforced by $r(z, y)$ and the inclusion $\exists r^- \sqsubseteq B$. Intuitively, this rewriting captures the fact that, whenever we have an element e in an interpretation that satisfies $\exists r^-$, then we can map x to e, thereby ensuring that the initial segment $r^*t_1t_2$ is satisfied below e. Moreover, by mapping y to the r-predecessor of e, we satisfy the remaining r^-.

As further illustration, suppose that in step 2, we choose $\mathsf{Leaf} = \{x, y\}$, and let y be the selected variable. After renaming, we obtain $q = \exists y.\alpha(y, y),\ B(y)$. In step 3, we choose $\exists r^-$, which leads us to drop the atom $B(y)$, leaving us with $\exists y.\alpha(y, y)$. In step 4, we choose the sequence that contains only s_3, so the atom $\alpha(y, y)$ is left untouched. Since $\exists r^- \in \mathsf{Loop}_\alpha[s_0, s_3]$, we can drop this atom in step 5, obtaining the empty query. In step 6, we choose $D = B$ and $R = R_1 = R_2 = r$. Step 7 is inapplicable since there are no binary atoms. Finally, in step 8, we add $B(y)$ to obtain the query $q = \exists y.B(y)$. Intuitively, this rewriting captures that if some element e satisfies B, then we can map both x and y to it to obtain a query match in which the regular expression $r^*t_1t_2r^-(x, y)$ is fully satisfied below e.

The next lemma shows that using $\mathsf{rewrite}(q, \mathcal{T})$, we can reduce the problem of finding an arbitrary query match to finding a match involving only ABox individuals.

Lemma 9. *$\mathcal{T}, \mathcal{A} \models q$ if and only if there exists a match π for some query $q' \in \mathsf{rewrite}(q, \mathcal{T})$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ such that $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ for every term t in q'.*

We remark that the number of variables and atoms in each query in $\mathsf{rewrite}(q, \mathcal{T})$ is linearly bounded by $|q|$. This is the key property used to show the following:

Lemma 10. *There are only exponentially many queries in $\mathsf{rewrite}(q, \mathcal{T})$ (up to equivalence), each having size polynomial in $|q|$.*

5.3 Query Evaluation

Even when all terms are mapped to ABox individuals, the paths between them may need to pass by the anonymous part in order to satisfy the regular expressions in the query. This leads us to define a relaxed notion of query entailment, which exploits the fact that if all variables are mapped to $\mathsf{Ind}(\mathcal{A})$, only loops (that is, paths from an individual a to itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_a$) may participate in the paths between them. Hence, we look for paths in the ABox that may use such loops to skip states in the query automata.

As part of our query evaluation procedure, we will need to decide for a given individual a whether a is an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_a$. We cannot use Loop_α directly, since it does not take into account the concepts which are

entailed due to ABox assertions. We note however that the set of loops starting from a given individual is fully determined by the set of basic concepts which the individual satisfies. We thus define a new table ALoop_α such that $\mathsf{ALoop}_\alpha[s, s']$ contains all *subsets* $G \subseteq \mathsf{BC}_\mathcal{T}$ such that a is an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_a$ whenever $G = \{C \in \mathsf{BC}_\mathcal{T} \mid a \in C^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}\}$. Note that the table ALoop_α is exponential in $|\mathcal{T}|$, but the associated decision problem is in P:

Lemma 11. *It can be decided in polytime in $|\mathcal{T}|$ and $|\alpha|$ whether $G \in \mathsf{ALoop}_\alpha[s, s']$.*

We use ALoop_α to define a relaxed notion of query match.

Definition 12. We write $\mathcal{T}, \mathcal{A} \mid\!\approx q$ if there is a mapping π from the terms in q to $\mathsf{Ind}(\mathcal{A})$ such that:

(a) $\pi(c) = c$ for each $c \in \mathsf{N_I}$,

(b) $\mathcal{T}, \mathcal{A} \models A(\pi(t))$ for each atom $A(t)$ in q, and

(c) for each $\alpha(t, t') \in q$ with $\alpha = \langle S, \Sigma, \delta, s, F\rangle$, there is a sequence $(a_0, s_0), \ldots, (a_n, s_n)$ of distinct pairs from $\mathsf{Ind}(\mathcal{A}) \times S$ such that $a_0 = \pi(t)$, $a_n = \pi(t')$, $s_0 = s$, $s_n \in F$, and for every $0 \leq i < n$, either:

(i) $a_i = a_{i+1}$ and $\{C \in \mathsf{BC}_\mathcal{T} \mid \mathcal{T},\mathcal{A} \models C(a_i)\} \in \mathsf{ALoop}_\alpha[s_i, s_{i+1}]$, or

(ii) $\mathcal{T}, \mathcal{A} \models R(a_i, a_{i+1})$ and $(s_i, R, s_{i+1}) \in \delta$ for some R.

The following lemma characterizes C2RPQ entailment in terms of relaxed matches.

Lemma 13. *$\mathcal{T}, \mathcal{A} \models q$ if and only if $\mathcal{T}, \mathcal{A} \mid\!\approx q'$ for some $q' \in \mathsf{rewrite}(q, \mathcal{T})$.*

By applying the preceding characterization, we obtain our C2RPQ upper bounds:

Proposition 14. *Boolean C2RPQ entailment is*

1. NL *in data complexity for DL-Lite$_\mathcal{R}$ and DL-Lite$_\mathsf{RDFS}$;*
2. P *in data complexity for $\mathcal{EL}\mathcal{H}$;*
3. NP *in combined complexity for DL-Lite$_\mathsf{RDFS}$;*
4. PSPACE *in combined complexity for DL-Lite$_\mathcal{R}$ and $\mathcal{EL}\mathcal{H}$.*

Proof sketch. By Lemmas 9 and 13, $\mathcal{T}, \mathcal{A} \models q$ just in the case that $\mathcal{T}, \mathcal{A} \mid\!\approx q'$ for some $q' \in \mathsf{rewrite}(q, \mathcal{T})$. For statements 1 and 2, if $\mathcal{T}$ and q are fixed, then computing $\mathsf{rewrite}(q, \mathcal{T})$ requires only constant time in $|\mathcal{A}|$. To decide whether $\mathcal{T}, \mathcal{A} \mid\!\approx q'$ for $q' \in \mathsf{rewrite}(q, \mathcal{T})$, we guess a mapping π from the terms in q' to $\mathsf{Ind}(\mathcal{A})$ and verify that it satisfies the conditions in Definition 12. Note that for condition (c), we cannot keep the whole sequence $(a_0, s_0), \ldots, (a_n, s_n)$ in memory at once, so we use a binary counter that counts up to $|\mathsf{Ind}(\mathcal{A}) \times S|$ and store only one pair of nodes $(a_i, s_i), (a_{i+1}, s_{i+1})$ at a time. The data complexity of verifying conditions (b) and (c) is the same as for instance checking in the corresponding DL: AC_0 for DL-Lite$_\mathcal{R}$, and P for $\mathcal{EL}\mathcal{H}$. This yields the desired upper bounds of NL and NL^P =P, respectively.

For statement 4, instead of building the whole set $\mathsf{rewrite}(q, \mathcal{T})$, which can be exponential, we generate a single $q' \in \mathsf{rewrite}(q, \mathcal{T})$ non-deterministically. By Lemma 10, every query in $\mathsf{rewrite}(q, \mathcal{T})$ can be generated after at most exponentially many steps, so we can use a polynomial-size counter to check when we have reached this limit. Since each rewritten query is of polynomial size (Lemma 10), and we keep only one query in memory at a time, the generation of a single query in $\mathsf{rewrite}(q, \mathcal{T})$ requires only polynomial space. We can then use the same strategy as above to decide in polynomial space whether $\mathcal{T}, \mathcal{A} \mid\!\approx q'$. This yields a non-deterministic polynomial space procedure for deciding $\mathcal{T}, \mathcal{A} \models q$. Using the well-known fact that NPSPACE = PSPACE, we obtain the desired upper bound.

For statement 3, we note that if $\mathcal{T}$ is an DL-Lite$_\mathsf{RDFS}$ TBox, $\mathsf{rewrite}(q, \mathcal{T}) = \{q\}$. Thus, it suffices to decide $\mathcal{T}, \mathcal{A} \mid\!\approx q$, which can be done by guessing a mapping π and verifying in polytime that π satisfies the conditions of Definition 12. □

By moving to 2RPQs, we can achieve tractability even in combined complexity.

Proposition 15. *Boolean 2RPQ entailment is*

1. NL *in combined complexity for DL-Lite$_\mathsf{RDFS}$;*
2. P *in combined complexity for DL-Lite$_\mathcal{R}$ and $\mathcal{EL}\mathcal{H}$.*

Proof sketch. For (1), we can iterate over all mappings π of the (at most two) query variables, and for each mapping, we check whether the conditions of Definition 12 are met using the same strategy as in the proof of point 1 of Proposition 14. Recall that in DL-Lite$_\mathsf{RDFS}$, instance checking is in NL w.r.t. combined complexity [Calvanese *et al.*, 2007a].

For (2), we first give a polynomial reduction to the problem of deciding whether $\mathcal{T}, \mathcal{A} \mid\!\approx q'$ with q' a 2RPQ. When $q = \exists xy\, L(x, y)$ with $x \neq y$, we can simply replace q by $q' = \exists xy\, \Sigma^* \cdot L \cdot \Sigma^*(x, y)$, where $\Sigma = \mathsf{N_R} \cup \overline{\mathsf{N_R}} \cup \{A? \mid A \in \mathsf{N_C}\}$, since $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{T}, \mathcal{A} \mid\!\approx q'$. For queries of the form $\exists x L(x, x)$, the proof is more involved and passes by the definition of an alternative rewriting procedure for 2RPQs, which is similar in spirit to rewrite but is guaranteed to run in polynomial time. We can then check for a match of a 2RPQ in the ABox using essentially the same strategy as for (1), except that we must now perform some polynomial-time ALoop_α tests to verify condition (c) of Definition 12. □

6 Conclusion and Future Work

In this paper, we established tight complexity bounds for answering various forms of regular path queries over knowledge bases formulated in lightweight DLs from the DL-Lite and $\mathcal{EL}$ families. Our results demonstrate that the query answering problem for these richer query languages is often not much harder than for the CQs and IQs typically considered. Indeed, query answering remains tractable in data complexity for the highly expressive class of C2RPQs, and for 2RPQs, we even retain polynomial combined complexity.

In future work, we plan to explore other useful extensions of regular path queries, such as *nested path expressions* (along the lines of [Pérez *et al.*, 2010]), and the addition of *path variables* (recently explored in [Barceló *et al.*, 2010]).

Acknowledgements The authors were supported by a Université Paris-Sud Attractivité grant and the ANR project PAGODA ANR-12-JS02-007-01 (Bienvenu), the FWF project T515-N23 (Ortiz), and the FWF project P25518-N23 and the WWTF project ICT12-015 (Šimkus).

References

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

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

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI*, pages 364–369, 2005.

[Barceló *et al.*, 2010] Pablo Barceló, Carlos A. Hurtado, Leonid Libkin, and Peter T. Wood. Expressive languages for path queries over graph-structured data. In *Proc. of PODS*, pages 3–14, 2010.

[Bienvenu *et al.*, 2013] Meghyn Bienvenu, Magdalena Ortiz, and Mantas Šimkus. Conjunctive Regular Path Queries in Lightweight Description Logics. Technical Report INFSYS RR-1843-13-01, Institute of Information Systems, Vienna University of Technology. Available at `http://www.kr.tuwien.ac.at/research/reports/`.

[Calvanese *et al.*, 2006] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proc. of KR*, pages 260–270, 2006.

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

[Calvanese *et al.*, 2007b] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proc. of AAAI*, pages 391–396, 2007.

[Calvanese *et al.*, 2009] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In *Proc. of IJCAI*, pages 714–720, 2009.

[Consens and Mendelzon, 1990] Mariano P. Consens and Alberto O. Mendelzon. Graphlog: a visual formalism for real life recursion. In *Proc. of PODS*, pages 404–416, 1990.

[Eiter *et al.*, 2012] Thomas Eiter, Magdalena Ortiz, Mantas Šimkus, TrungKien Tran, and Guohui Xiao. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *Proc. of AAAI*, 2012.

[Kozen, 1977] Dexter Kozen. Lower bounds for natural proof systems. In *Proc. of FOCS*, pages 254–266, 1977.

[Krisnadhi and Lutz,] Adila Krisnadhi and Carsten Lutz. Data complexity in the $\mathcal{EL}$ family of description logics. In *Proc. of LPAR*, pages 333–347.

[Krötzsch and Rudolph, 2007] Markus Krötzsch and Sebastian Rudolph. Conjunctive queries for $\mathcal{EL}$ with composition of roles. In *Proc. of DL*, 2007.

[Ortiz *et al.*, 2011] Magdalena Ortiz, Sebastian Rudolph, and Mantas Simkus. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *Proc. of IJCAI*, pages 1039–1044, 2011.

[Pérez *et al.*, 2010] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. nSPARQL: A navigational language for RDF. *Journal of Web Semantics*, 8(4):255–270, 2010.

[Rosati, 2007] Riccardo Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proc. of DL*, 2007.

Omitted Proofs

Notational conventions. For convenience, we introduce some further notation. We use $\mathsf{terms}(q)$ to refer to the set of terms of a query q. We use $q_1 \subseteq q_2$ to denote query containment (defined in the standard way), and $C \rightsquigarrow D$ (with $C, D \in \mathsf{BC}_\mathcal{T}$) to state that $\exists x C(x) \subseteq \exists x D(x)$.

Finally, given a word $w = u_1 \dots u_n$ and interpretation $\mathcal{I}$, we let $w^\mathcal{I}$ denote that set of all sequences $e_0, \dots, e_n$ of elements in $\mathcal{I}$, such that for every $1 \leq i \leq n$:

- if $u_i = A?$, then $e_{i-1} = e_i \in A^\mathcal{I}$
- if $u_i = R \in \mathsf{N_R} \cup \overline{\mathsf{N_R}}$, then $\langle e_{i-1}, e_i \rangle \in R^\mathcal{I}$

Thus, b is an $L(\alpha)$-successor of c just in the case that there is a word $w \in L(\alpha)$ and $\sigma \in w^\mathcal{I}$ such that σ begins with b and terminates by c.

Proposition 5. 2RPQ answering in DL-Lite is P-hard (combined complexity), assuming an NFA representation of the regular language.

Proof. Consider a propositional Horn theory T over $v_1, \dots, v_n$ consisting of:

- a set of rules $\rho_i = v_{i_1} \wedge v_{i_2} \rightarrow v_{i_3}$ $(1 \leq i \leq m)$
- a single "initialization" fact: v_1

Let v_n be a target variable. Then the problem is to decide whether $T \models v_n$. Note that this problem is P-complete (straightforward reduction from Horn satisfiability).

We construct a DL-Lite TBox $\mathcal{T}$, ABox $\mathcal{A}$, and 2RPQ q such that $\mathcal{T}, \mathcal{A} \models q$ if and only if $T \models v_n$. The ABox $\mathcal{A}$ consists of a single assertion $A(a)$, and $\mathcal{T}$ contains the following axioms:

- $A \sqsubseteq \exists r_{i,j}$, for $1 \leq i \leq m$, $j \in \{1, 2\}$
- $\exists r^-_{i_1,j_1} \sqsubseteq \exists r_{i_2,j_2}$, for $1 \leq i_1, i_2 \leq m$ and $j_1, j_2 \in \{1, 2\}$

We set $q = \exists x \alpha(x, x)$, where $\alpha = (S, \Sigma, \delta, s_0, \{v_n^{out}\})$ is an NFA defined as follows:

- $S = \{s_0\} \cup \{v_1\} \cup \{v_i^{in}, v_i^{out} \mid 2 \leq i \leq n\} \cup \{\rho_i \mid 1 \leq i \leq m\}$
- $\Sigma = \{A?\} \cup \{r_{i,j}, r^-_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq 2\}$
- δ contains the following transitions:
  - $(s_0, A?, v_n^{in})$
  - for each rule $\rho_i = v_j \wedge v_k \rightarrow v_\ell$:
    $(v_\ell^{in}, r_{i,1}, v_j^{in})$, $(v_j^{out}, r^-_{i,1}, \rho_i)$, $(\rho_i, r_{i,2}, v_k^{in})$, $(v_k^{out}, r^-_{i,2}, v_\ell^{out})$
  - note: we replace any occurrences of v_1^{in} and v_1^{out} by v_1

Claim: $\mathcal{T}, \mathcal{A} \models q$ if and only if $T \models v_n$

($\Rightarrow$) Suppose that $\mathcal{T}, \mathcal{A} \models q$, and hence that q holds in the canonical model $\mathcal{I}$ of $\mathcal{T}, \mathcal{A}$. This means that exists a sequence $(s_0, \sigma_1, s_1), (s_1, \sigma_2, s_2), \dots, (s_{p-1}, \sigma_p, s_p) \in \delta$ of transitions and a path $e_0, \dots, e_p \in (\sigma_1 \dots \sigma_p)^\mathcal{I}$ such that $e_0 = e_p$ and $s_p = v_n^{out}$. Note that since there is a single transition from s_0 (namely $(s_0, A?, v_n^{in})$), we must have $(e_0, e_1) \in (A?)^\mathcal{I}$, hence $e_0 = e_1 = e_p = a$ and $s_1 = v_n^{in}$.

We will show by induction on i that if $s_i = v_j^{out}$, then $T \models v_j$, for every $1 \leq i \leq p$. This clearly implies that $T \models v_n$, since $s_p = v_n^{out}$.

For the base case, suppose that $s_i = v_j^{out}$, and there is no $k < i$ such that $s_k = v_{j'}^{out}$. By examining the possible transitions leading to v_j^{out}, then we can infer that there must exist a rule $\rho_\ell = v_1 \wedge v_1 \rightarrow v_j$ in T, and hence that $T \models v_j$.

For the induction step, suppose that $s_{k+1} = v_j^{out}$ and that the statement has been shown to hold for all $1 \leq i \leq k$. Then there must exist a rule $\rho_\ell = v_{\ell_1} \wedge v_{\ell_2} \rightarrow v_j$ such that $\sigma_{k+1} = r^-_{\ell,2}$ and $e_k = e_{k+1} r_{\ell,2}$. We consider the most interesting case in which $v_{\ell_1} \neq v_1$. Then we must have $s_k = v_{\ell_2}^{out}$, and hence $T \models v_{\ell_2}$, by the IH. Moreover, since $e_k = e_{k+1} r_{\ell,2}$, we must have $k' < k$ such that $e_{k'-1} = e_{k+1}$, $\sigma_{k'} = r_{\ell,2}$, and $e_{k'} = e_k$. Examining the transition relation, we can infer that $s_{k'-1} = \rho_\ell$, which in turn means that $\sigma_{k'-1} = r^-_{\ell,1}$ and $s_{k'-2} = v_{\ell_1}^{out}$. Applying the IH, we obtain $T \models v_{\ell_1}$, and hence $T \models v_j$.

($\Leftarrow$) If $T \models v_n$, then there must be exist a proof tree for v_n from T, i.e. a node-labelled tree satisfying the following conditions:

- the root node is labelled v_n
- all leaf nodes are labelled v_1
- if a non-leaf node has label v_i, then T contains a rule $\rho_j = v_{j_1} \wedge v_{j_2} \rightarrow v_{j_3}$ such that $v_{j_3} = v_i$ and the node's two children have labels v_{j_1} and v_{j_2}

It is straightforward to use this proof tree to define a match for q in the canonical model of $\mathcal{T}, \mathcal{A}$. □

Lemma 7. For every element $p \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \setminus \mathsf{Ind}(\mathcal{A})$: $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s']$ if and only if p is an $L(\alpha_{s,s'})$-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$.

Proof. Fix a TBox $\mathcal{T}$, ABox $\mathcal{A}$, and an automaton $\alpha = \langle S, \Sigma, \delta, s, F \rangle$. For the first direction, we need to show that if $p \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \setminus \mathsf{Ind}(\mathcal{A})$ and $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s']$, then p is an $L(\alpha_{s,s'})$-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$. To this end, we fix also a sequence of applications of the rules 1, 2, 3, and 4 (or 4') which generates the full table Loop_α. Let k be the length of this sequence. It then suffices to show the following claim for all $1 \leq i \leq k$:

Claim 1: If C is inserted into $\mathsf{Loop}_\alpha[s, s']$ on the i-th rule application and $p \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \setminus \mathsf{Ind}(\mathcal{A})$ is such that $\mathsf{Tail}(p) = C$, then p is an $L(\alpha_{s,s'})$-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$.

Proof of claim. The proof is by induction on i. First suppose that C is inserted into $\mathsf{Loop}_\alpha[s, s']$ with the first rule application. Then either rule 1 or rule 3 must have been applied. In the former case, we have $s = s'$, so p is trivially an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$ (as witnessed by $\epsilon \in L(\alpha_{s,s'})$). In the latter case, we have $\mathsf{Tail}(p) = C$ and $\mathcal{T} \models C \sqsubseteq A$, which implies that $p \in A^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. As $(s, A?, s') \in \delta$, we have $A? \in L(\alpha_{s,s'})$, and thus, p is an $L(\alpha_{s,s'})$-successor of itself.

For the induction step, suppose that the statement holds for all $1 \leq i < k$, and let $p \in \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \setminus \mathsf{Ind}(\mathcal{A})$ be such

that $C = \mathsf{Tail}(p)$ is inserted into $\mathsf{Loop}_\alpha[s, s']$ on the k-th rule application. The first possibility is that the k-th rule application involves rules 1 or 3, in which case we proceed as in the base case. The next possibility is that rule 2 was applied. Then there must exist s'' such that after the first $k-1$ rule applications, we have $C \in \mathsf{Loop}_\alpha[s, s'']$ and $C \in \mathsf{Loop}_\alpha[s'', s']$. Applying the induction hypothesis, we get that p is both an $L(\alpha_{s,s''})$-successor of itself and an $L(\alpha_{s'',s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$. It follows that there exists words $w_1 \in L(\alpha_{s,s''})$ and $w_2 \in L(\alpha_{s'',s'})$ and sequence of elements $d = d_0 \dots d_n$ and $e = e_0 \dots e_m$ from $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$ with $d_0 = d_n = e_0 = e_m = p$ such that $d \in w_1^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ and $e \in w_2^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ Thus, $w = w_1 w_2$ is a word from $L(\alpha_{s,s'})$ such that $d_0 \dots d_n e_1 \dots e_m \in w^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ with $d_0 = e_m = p$, hence p is an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$.

The final possibility is that the k-th rule application involves rule 4. Here the proof differs depending on whether $\mathcal{T}$ is formulated in DL-Lite$_\mathcal{R}$ or $\mathcal{ELH}$. We give the proof only for DL-Lite$_\mathcal{R}$; the proof for $\mathcal{ELH}$ proceeds analogously. We first note that since an application of rule 4 leads to the insertion of C into $\mathsf{Loop}_\alpha[s, s']$ at stage k, it must be the case that we can find $R, R', R'' \in \overline{\mathsf{N_R}}$ and $s'', s''' \in S$ such that:

- $C \neq \exists R$,
- $\mathcal{T} \models C \sqsubseteq \exists R$,
- $\mathcal{T} \models R \sqsubseteq R'$,
- $\mathcal{T} \models R^- \sqsubseteq R''$,
- $(s, R', s'') \in \delta$ and $(s''', R'', s') \in \delta$,
- $\exists R^- \in \mathsf{Loop}_\alpha[s'', s''']$ (after $k-1$ rule applications)

As $\mathsf{Tail}(p) = C$, $C \neq \exists R$, and $\mathcal{T} \models C \sqsubseteq \exists R$, the path $p' = pR\exists R^-$ must belong to $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Then by applying the induction hypothesis, we can infer that p' is an $L(\alpha_{s'',s'''})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_{p'}$, and hence there is some word $w \in L(\alpha_{s'',s'''})$ and some sequence of elements σ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_{p'}$ such that $\sigma \in w^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. Using the above points, we can show that the word $w' = R'wR''$ and sequence $\sigma' = p\sigma p$ are such that $\sigma' \in (w')^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. We can thus conclude that p is an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$. (end proof of claim)

For the second direction, we will proceed by induction on the length of the word witnessing the successor relationship. To formalize this, we need to introduce some terminology. Given a path p which is an $L(\beta)$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$, a witness for (p, β) is a pair (w, σ) such that $w \in L(\beta)$, σ is a sequence of elements in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$ which starts and ends with p, and $\sigma \in w^{\mathcal{I}}$. We define the *minimal witness length* of (p, β) as the minimum of $|w|$ over all witnesses (w, σ) for (p, β). To establish the second direction of the lemma, it suffices to prove the following claim for all $i \geq 0$:

Claim 2: If p is an $L(\alpha_{s,s'})$-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$, and the minimal witness length of $(p, \alpha_{s,s'})$ is i, then $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s']$.

Proof of claim. The proof is by induction on i. The base case is when $i = 0$, i.e. when there is a witness (w, σ) with w the empty word. Then we must have $\epsilon \in L(\alpha_{s,s'})$, and the statement follows trivially from rule 1 of the definition of Loop_α.

For the induction step, suppose that the statement of the claim holds for all $i < k$, and suppose that p is an $L(\alpha_{s,s'})$-successor of itself in the interpretation $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$ such that the minimal witness length of $(p, \alpha_{s,s'})$ is k. Let $w = u_1 \dots u_k \in L(\alpha_{s,s'})$ and $\sigma = e_0, \dots, e_k$ be a witness for $(p, \alpha_{s,s'})$. First suppose that there exists some e_j with $1 \leq j < k$ such that $e_j = p$. Set $w_1 = u_1 \dots u_j$, $w_2 = u_{j+1} \dots u_k$, $\sigma_1 = e_0, \dots, e_j$, and $\sigma_2 = e_j, \dots, e_k$. Then (w_1, σ_1) is a witness for $(p, \alpha_{s,s''})$ with length $< k$, and (w_2, σ_2) is a witness for $(p, \alpha_{s'',s'})$ with length $< k$. Thus, from the induction hypothesis, we have that $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s'']$ and $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s'', s']$. Hence, by rule 2 of the construction of Loop_α, we must also have $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s']$.

Now let us consider the second possibility, which is that $e_j \neq p$ for all $1 \leq j < k$. A special case is when $k = 1$, in which case the witness takes the form $(A?, pp)$. We thus have $(s, A?, s') \in \delta$ and $p \in A^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. The latter implies that $\mathcal{T} \models \mathsf{Tail}(p) \sqsubseteq A$, so the conditions of rule 3 are satisfied, and hence $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s']$. Next consider the case where $k \geq 2$. Then since $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_p$ is tree-shaped, we must have $e_1 = e_{k-1}$, and also $e_1 = p'$ for some $p' = pRC \in \Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}|_p}}$. At this point, the proof slightly differs depending on whether we are in DL-Lite$_\mathcal{R}$ or $\mathcal{ELH}$. We present the proof for the case of $\mathcal{ELH}$, in which case we have $R = r \in \mathsf{N_R}$ and $\mathcal{T} \models \mathsf{Tail}(p) \sqsubseteq \exists r.C$. As (w, σ) is a witness for $(p, \alpha_{s,s'})$, we also have that:

- $u_1 \in \overline{\mathsf{N_R}}$ and $\mathcal{T} \models r \sqsubseteq u_1$
- $u_k \in \overline{\mathsf{N_R}}$ and $\mathcal{T} \models r^- \sqsubseteq u_k$

Next we note that since $w \in L(\alpha_{s,s'})$, there must exist $s'', s''' \in S$ such that:

- $(s, u_1, s'') \in \delta$
- $u_2 \dots u_{k-1} \in L(\alpha_{s'',s'''})$
- $(s''', u_k, s') \in \delta$

From the second bullet, we obtain that $w' = u_2 \dots u_{k-1}$ and $\sigma' = e_1, \dots, e_{k-1}$ are a witness for $(p', \alpha_{s'',s'''})$. Note moreover, that w' has length less than k, so the induction hypothesis applies, allowing us to infer that $\mathsf{Tail}(p') = C \in \mathsf{Loop}_\alpha[s'', s''']$. We thus satisfy all of the required conditions for applying rule 4 to obtain $\mathsf{Tail}(p) \in \mathsf{Loop}_\alpha[s, s']$. This completes our proof of Claim 2. □

Lemma 10. There are only exponentially many queries in $\mathsf{rewrite}(q, \mathcal{T})$ (up to equivalence), and each one has size polynomial in $|q|$.

Proof. Let n be the number of variables in the input query q. First we argue that by introduction of fresh atoms in Step 8, a rewriting of q may increase the query size only linearly. In particular, it adds at most $2n$ variables, at most n atoms of the form $D(x)$, and at most $4n$ states occurring in fresh atoms of the form $\alpha(x, y)$. These bounds clearly hold after a one-step application of the rewriting, which introduces one atom of the form $D(y)$ or $\alpha_P(z, y)$, that is, at most one fresh variable z and at most two states (we assume that α_P has only two states

s_P and s_{Pf}, and its only transition is (s_P, P, s_{Pf})). We associate the variable z and the atom with the chosen variable $y \in \mathsf{Leaf}$. In subsequent rewriting steps, if y is not chosen to be a member of Leaf, then there is no need to introduce another variable or atom associated to y. If $y \in \mathsf{Leaf}$ and we had introduced $D(y)$, then we must drop $D(y)$ in Step 3 before introducing again some new atom in Step 8. Hence we only have one extra variable and one extra atom associated to the same variable at any given stage. If instead we had introduced the atom $\alpha_P(z, y)$, then this atom will be rewritten in Step 7 into $\alpha_{s_P,s_P}(z, y)$. We may add a new atom $\alpha_{P'}(z', y)$ in Step 8, leaving us with two extra variables and three states. However, in the next iteration in which $y \in \mathsf{Leaf}$ is chosen, Step 5 will eliminate the older $\alpha_{s_P,s_P}(z, y)$ (since $A \in \mathsf{Loop}_{\alpha_P}[s_P, s_P]$ trivially holds for all A), and Step 7 will rewrite the newer one into $\alpha_{s'_P,s'_P}(z', y)$, leaving us again with just one extra variable and two extra states before introducing a new atom in Step 8. Hence, we have at most two variables and four states associated to the same variable at a time.

Second, we give a bound on the number of NFAs that can occur in the atoms introduced by the other steps of the rewriting algorithm. Let m be the size of the (string encoding) the input query. The rewriting only adds atoms $\alpha_{s,s'}(t, t')$ with s, s' states of some α already occuring in the query, hence there are only $\mathcal{O}(m^2)$ different NFAs. Next we observe that with $\mathcal{O}(m^2)$ NFAs and $\mathcal{O}(m)$ variables, at most $\mathcal{O}(m^4)$ atoms can be built. This gives a polynomial bound on size of the largest possible query in $\mathsf{rewrite}(q, \mathcal{T})$, and a total of $\mathcal{O}(2^{m^4})$ different queries (up to renaming of variables). □

Lemma 9. $\mathcal{T}, \mathcal{A} \models q$ if and only if there a match π for some query $q' \in \mathsf{rewrite}(q, \mathcal{T})$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ such that $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ for every $t \in \mathsf{terms}(q')$.

We split Lemma 9 into the two lemmas, a first showing correctness of the procedure rewrite, and a second showing its completeness. In the proofs of these lemmas, it will prove useful to refer to queries that are produced by a single iteration of rewrite. We thus introduce the set $\mathsf{one\text{-}step}(q, \mathcal{T})$ which contains precisely those queries q' for which there is an execution of $\mathsf{rewrite}(q, \mathcal{T})$ such that q' is output the first time that the procedure returns to Step 1.

Lemma 16. *If $\mathcal{T}, \mathcal{A} \models q'$ for some $q' \in \mathsf{rewrite}(q, \mathcal{T})$, then $\mathcal{T}, \mathcal{A} \models q$.*

Proof. It is sufficient to show that if $q' \in \mathsf{one\text{-}step}(q, \mathcal{T})$ and $\mathcal{T}, \mathcal{A} \models q'$, then $\mathcal{T}, \mathcal{A} \models q$. Fix a C2RPQ q and a DL-Lite$_\mathcal{R}$ or $\mathcal{ELH}$ TBox $\mathcal{T}$. Let $q' \in \mathsf{one\text{-}step}(q, \mathcal{T})$ be such that $\mathcal{T}, \mathcal{A} \models q'$, and let π be a match for q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$.

Consider the execution of $\mathsf{rewrite}(q, \mathcal{T})$ which leads to the query q' being output the first time that the procedure returns to Step 1. Let Leaf be the non-empty subset of $\mathsf{vars}(q)$ which was selected in Step 2, let $C \in \mathsf{TC}_\mathcal{T}$ be the concept selected in Step 3, and let $D \in \mathsf{BC}_\mathcal{T}$ and R, R_1, R_2 be the concept and roles selected in Step 6. Because of Step 8, we know that q' either contains an atom $D(y)$, or contains an atom $\alpha_P(z, y)$ such that $D = \exists P^-$, z is a variable not appearing in $\mathsf{vars}(q)$, and α_P is such that $L(\alpha_P) = \{P\}$. In the former case, we know that $\pi(y) \in D^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and that either (i) $\mathcal{T} \models D \sqsubseteq \exists R$ and $C = \exists R^-$, or (ii) $\mathcal{T} \models D \sqsubseteq \exists R.C$, hence there must exist an R-successor e of $\pi(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ with $\mathsf{Tail}(e) = C$. In the latter case, we can find d such that $(d, \pi(y)) \in P^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, which implies that $\pi(y) \in D^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. We then use the fact that $\mathcal{T} \models D \sqsubseteq \exists R$ where $C = \exists R^-$ to find some R-successor e of $\pi(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ with $\mathsf{Tail}(e) = C$. We define a mapping $\pi' : \mathsf{terms}(q) \to \Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ by setting $\pi'(t) = e$ for every $t \in \mathsf{Leaf}$ and setting $\pi'(t) = \pi(t)$ for every $t \in \mathsf{terms}(q') \setminus \{y\}$. This mapping is well-defined since every variable in q either belongs to Leaf or appears in q'.

We aim to show that π' is a match for q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. To this end, consider some concept atom $B(t) \in q$. First suppose that $t \in \mathsf{Leaf}$. Then we know that the concept C selected in Step 3 is such that $\mathcal{T} \models C \sqsubseteq B$. We then use the fact that since $t \in \mathsf{Leaf}$,we have $\pi'(t) = e \in C^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. If $t \notin \mathsf{Leaf}$, then $B(t) \in q'$. As π is a match for q', we have $\pi(t) \in B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Using $\pi'(t) = \pi(t)$, we get $\pi'(t) \in B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$.

Now consider some atom of the form $\alpha(t, t') \in q$, where $\alpha = \langle S, \Sigma, \delta, s, F\rangle$. If both $t \notin \mathsf{Leaf}$ and $t' \notin \mathsf{Leaf}$, then it can be verified that $\alpha(t, t') \in q'$. As π is a match for q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, it must be the case that $\pi(t')$ is an $L(\alpha)$-successor of $\pi(t)$. Since $\pi'(t) = \pi(t)$ and $\pi'(t) = \pi(t)$, the same holds for π'. Let us next consider the more interesting case in which $\{t, t'\} \cap \mathsf{Leaf} \neq \emptyset$. In Step 4, we have a query containing $\alpha(\sigma(t), \sigma(t'))$, where $\sigma(t) = t$ for $t \notin \mathsf{Leaf}$ and $\sigma(t) = y$ for $t \in \mathsf{Leaf}$. Note that since $\{t, t'\} \cap \mathsf{Leaf} \neq \emptyset$, at least one of $\sigma(t)$ and $\sigma(t')$ must be y. It follows that in Step 4, we will select a sequence $s_1, \ldots, s_n$ of distinct states from S such that $s_n \in F$, and we will replace $\alpha(\sigma(t), \sigma(t'))$ by the atoms: $\alpha_{s,s_1}(\sigma(t), y)$, $\alpha_{s_1,s_2}(y, y)$, $\ldots$, $\alpha_{s_{n-2},s_{n-1}}(y, y)$, $\alpha_{s_{n-1},s_n}(y, \sigma(t'))$. Let us denote this set of atoms by Q_α. We now establish the following claim:

Claim 1. If π' is a match for Q_α in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, then π' is a match for $\alpha(t, t')$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$.

Proof of claim. Suppose that π' is a match for the atoms in Q_α in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. Then this means that:

- $\pi'(y)$ is an $L(\alpha_{s,s_1})$ of $\pi'(\sigma(t))$
- for every $1 \leq i < n-1$: $\pi'(y)$ is an $L(\alpha_{s_i,s_{i+1}})$-successor of $\pi'(y)$
- $\pi'(\sigma(t'))$ is an $L(\alpha_{s_{n-1},s_n})$-successor of $\pi'(y)$

We then remark that the language consisting of all words $w_1 \ldots w_n$ such that $w_1 \in L(\alpha_{s,s_1})$, $w_i \in L(\alpha_{s_i,s_{i+1}})$ for every $1 \leq i < n-1$, and $w_n \in L(\alpha_{s_{n-1},s_n})\}$ is a subset of the language $L(\alpha_{s,s_n})$, and hence of $L(\alpha)$. Thus, by composing the paths witnessing each of the above successor relationships, we can show that $\pi'(\sigma(t'))$ is an $L(\alpha)$-successor of $\pi'(\sigma(t))$. Then to complete the proof, we simply note that $\pi'(\sigma(t)) = \pi'(t)$ and $\pi'(\sigma(t')) = \pi'(t')$, because of the way we defined π' and σ.

Because of Claim 1, to complete the proof that π' is a match for q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, it is sufficient to show the following:

Claim 2. For every $\alpha_{s,s'}(u,u') \in Q_\alpha$: $\pi'(u')$ is an $L(\alpha_{s,s'})$-successor of $\pi'(u)$.

Proof of claim. Take some $\alpha_{s,s'}(u,u') \in Q_\alpha$. We start with the case where $u = u' = y$ and $C \in \mathsf{Loop}_\alpha[s,s']$. We know from above that $\mathsf{Tail}(\pi'(y)) = C$, so we can apply Lemma 7 to infer that $\pi'(y)$ is an $L(\alpha_{s,s'})$-successor of $\pi'(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, which yields the desired result given that $u = u' = y$. Next suppose that either $u \neq y$, $u' \neq y$, or $C \notin \mathsf{Loop}_\alpha[s,s']$. Then we will not remove $\alpha_{s,s'}(u,u')$ in Step 5, so it will still be present in Step 6. There are three cases depending on which of u and u' equals y. We treat each case separately:

Case 1: $u = y$ and $u' \neq y$. It follows that $u' = \sigma(t') = t'$ and so $\pi'(u') = \pi(u')$. In Step 7, we will replace $\alpha_{s,s'}(u,u')$ with $\alpha_{s'',s'}(u,u')$ where $s'' \in S$ is such that $(s, R_1^-, s'') \in \delta$. The atom $\alpha_{s'',s'}(u,u')$ belongs to q', so we know that it is satisfied by π. More precisely, we know that $\pi(u')$ is an $L(\alpha_{s'',s'})$-successor of $\pi(u) = \pi(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. Since e is an R-successor of $\pi(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ and $\mathcal{T} \models R \sqsubseteq R_1$ it follows that $\pi(y)$ is an $L(\alpha_{s,s''})$-successor of e (via the word $R_1^- \in L(\alpha_{s,s''})$). We can thus infer that $\pi'(u') = \pi(u')$ is an $L(\alpha_{s,s'})$-successor of $\pi'(u) = \pi'(y) = e$.

Case 2: $u \neq y$ and $u' = y$. It follows that $u = \sigma(t) = t$ and so $\pi'(u) = \pi(u)$. In Step 7, we will replace $\alpha_{s,s'}(u,u')$ with an atom $\alpha_{s,s''}(u,u')$ where $s'' \in S$ is such that $(s'', R_2, s') \in \delta$. The atom $\alpha_{s,s''}(u,u')$ appears in q', so it must be satisfied by π. That means that $\pi(u') = \pi(y)$ is an $L(\alpha_{s,s''})$-successor of $\pi(u)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. We also know that e is an R-successor of $\pi(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ and that $\mathcal{T} \models R \sqsubseteq R_2$. From this, we can infer that e is an $L(\alpha_{s'',s'})$-successor of $\pi(u')$. Combining these, we get that $\pi'(u) = \pi'(y) = e$ is an $L(\alpha_{s,s'})$-successor of $\pi'(u)$.

Case 3: $u = u' = y$. In Step 7, we will replace $\alpha_{s,s'}(u,u')$ with an atom $\alpha_{s'',s'''}(u,u')$ where $(s, R_1^-, s'') \in \delta$ and $(s'', R_2, s') \in \delta$. By combining the arguments used in Cases 1 and 2, we find that $\pi(y)$ is an $\alpha_{s,s''}$-successor of $\pi'(u)$, $\pi(y)$ is an $\alpha_{s'',s'''}$-successor of $\pi(y)$, and $\pi'(u')$ is an $\alpha_{s''',s'}$-successor of $\pi(y)$. From this, we can infer that $\pi'(u')$ is an $\alpha_{s,s'}$-successor of $\pi'(u)$.

□

Lemma 17. *Suppose that $\mathcal{T}, \mathcal{A} \models q$ and π is a match for q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ such that $\pi(v) = pSC$ and there is no $v' \in \mathsf{vars}(q)$ such that $\pi(v)$ is a proper prefix of $\pi(v')$. Then there is a match π' for some $q' \in \mathsf{one\text{-}step}(q, \mathcal{T})$ such that:*

- *$\pi'(u) = \pi(u)$ for every $u \in \mathsf{terms}(q)$ is such that $\pi(u) \neq \pi(v)$*
- *$\pi'(u) = p$ for every $u \in \mathsf{terms}(q)$ with $\pi(u) = \pi(v)$*
- *if $u \notin \mathsf{terms}(q)$, then $\pi'(u)$ is either a proper prefix of p or in $\mathsf{Ind}(\mathcal{A})$*

Proof. Let π be a match for a C2RPQ q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ and v be such that $\pi(v) = pSC$ and there is no $v' \in \mathsf{vars}(q)$ with $\pi(v)$ a proper prefix of $\pi(v')$. We show how to obtain a query $q' \in \mathsf{one\text{-}step}(q, \mathcal{T})$ and match π' with the required properties. In Step 1 of rewrite, we choose to continue on to Step 2, where we set $\mathsf{Leaf} = \{u \in \mathsf{vars}(q_0) \mid \pi(u) = \pi(v)\}$. We define a function σ as follows: $\sigma(u) = u$ if $\pi(u) \neq \pi(v)$, else $\sigma(u) = y$. At the end of Step 2, we have the query:

$$\{B(\sigma(u)) \mid B(u) \in q\} \cup \{\alpha(\sigma(u), \sigma(u')) \mid \alpha(u,u') \in q\}$$

In Step 3, we choose the concept C (note that $C \in \mathsf{TC}_\mathcal{T}$ because $\pi(v) = pSC$). Consider some atom $B(y) \in q_2$ We know that there must be some atom $B(u) \in q$ with $u \in \mathsf{Leaf}$. Since π is a match for q, we must have that $\pi(u) = \pi(v) \in B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Since $\pi(v) = pSC$, it follows from the definition of canonical models that $\mathcal{T} \models C \sqsubseteq B$, as required by Step 3.

Next we show how to select a decomposition of atoms in Step 4. Consider an atom $\alpha(t,t')$ which is present in the query at the start of Step 4, such that $y \in \{t,t'\}$ and $\alpha = \langle S, \Sigma, \delta, s_0, F\rangle$. Then we know from above that there is an atom $\alpha(u,u')$ in the original query q such that $t = \sigma(u)$ and $t' = \sigma(u')$. Since π is a match for q with $\pi(t) = \pi(u)$ and $\pi(t') = \pi(u')$, we know that $\pi(t')$ is an $L(\alpha)$-successor of $\pi(t)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. It follows that we can find $w = u_1 \dots u_m \in L(\alpha)$ and a sequence $\pi(t) = e_0, \dots, e_m = \pi(t')$ of elements in $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ such that for $e_0, \dots, e_m \in w^\mathcal{I}$. We assume without loss of generality that m is minimal, i.e. we cannot find a different word from $L(\alpha)$ and associated sequence of elements satisfying the same properties but with length $< m$. Now let $j_1 < \dots < j_{n-1}$ be all of the indices $i < m$ such that $e_i = \pi(v)$. We define the words $w_1, \dots, w_n$ by setting:

- $w_1 = u_1 \dots u_{j_1}$
- $w_i = u_{j_{i-1}+1} \dots u_{j_i}$ for $2 \leq i < n$
- $w_n = u_{j_{n-1}+1} \dots u_m$

We also define a sequence of states $s_1, \dots, s_n$ such that:

- $w_1 \in L(\alpha_{s_0,s_1})$
- $w_i \in L(\alpha_{s_{i-1},s_i})$ for $2 \leq i \leq n$
- $s_n \in F$

Note that such a sequence must exist since $w = w_1 \dots w_n \in L(\alpha)$, and α has start state s_0 and final states F. Using the fact that $e_{j_i} = \pi(v) = \pi(y)$ for $1 \leq i < n$, we have that ($\star$):

- $\pi(y)$ is an $L(\alpha_{s_0,s_1})$-successor of $\pi(t) = \pi(\sigma(u)) = \pi(u)$
- $\pi(y)$ is an $L(\alpha_{s_{i-1},s_i})$-successor of $\pi(y)$, for $2 \leq i < n$
- $\pi(t') = \pi(\sigma(u')) = \pi(u')$ is an $L(\alpha_{s_{n-1},s_n})$-successor of $\pi(y)$

Suppose for a contradiction that $s_k = s_\ell$ for some $k < \ell$. It follows that $w_1 \dots w_k w_{\ell+1} \dots w_n \in L(\alpha)$. We then note that the word $w_1 \dots w_k w_{\ell+1} \dots w_n$ and sequence of elements $e_0 \dots e_{j_k} e_{j_\ell+1} \dots e_m$ satisfy the aforementioned conditions and have length less than m, contradicting our earlier minimality assumption. Hence, the states in $s_1, \dots, s_n$ are pairwise disjoint. We can thus choose this sequence of states in Step 4, and replace the atom $\alpha(t,t') = \alpha(\sigma(u), \sigma(u'))$ with the atoms: $\alpha_{s_0,s_1}(\sigma(u), y)$, $\alpha_{s_1,s_2}(y,y)$, $\alpha_{s_{n-2},s_{n-1}}(y,y)$, $\alpha_{s_{n-1},s_n}(y, \sigma(u'))$.

The final choices to be made occur in Step 6, where we must choose a concept $D \in \mathsf{BC}_\mathcal{T}$, roles $R, R_1, R_2 \in \overline{\mathsf{N_R}}$, and states such that conditions (a)-(d) are satisfied. We set $R = S$ (recall that $\pi(t) = pSC$). If $p \notin \mathsf{Ind}$, then we choose D

such that $p = p'PD$. Note that if we are in DL-Lite$_\mathcal{R}$, then $D = \exists P^-$. It follows from the definition of canonical models that $\mathcal{T} \models D \sqsubseteq \exists S$ (if we are in DL-Lite$_\mathcal{R}$) or $\mathcal{T} \models D \sqsubseteq \exists S.C$ (for $\mathcal{ELH}$), so condition (a) holds. If instead we have $p \in \mathsf{Ind}$, then the definition of canonical models, together with our normal form for $\mathcal{ELH}$ TBoxes, guarantees that there is some $D \in \mathsf{BC}_\mathcal{T}$ such that $p \in D^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and $\mathcal{T} \models D \sqsubseteq \exists S$ (if we are in DL-Lite$_\mathcal{R}$) or $\mathcal{T} \models D \sqsubseteq \exists S.C$ (for $\mathcal{ELH}$). Note that for DL-Lite$_\mathcal{R}$, it is always possible to choose D such that we also have $\mathcal{A} \models D(p)$, and we assume in what follows that D has this property.

It remains to show that we can find R_1, R_2 and choices of states such that conditions (b), (c), and (d) are verified. For (c), consider some atom of the form $\gamma(y, x)$ which belongs to the query at the start of Step 6. Then we know that there must exist an atom $\alpha(u, u') \in q$ with $\alpha = \langle S, \Sigma, \delta, s_0, F\rangle$ such that $\gamma(y, x)$ is equal to one of the following atoms which replaced $\alpha(\sigma(u), \sigma(u'))$ during Step 4: $\alpha_{s_0,s_1}(\sigma(u), y), \alpha_{s_1,s_2}(y, y), \ldots, \alpha_{s_{n-2},s_{n-1}}(y, y), \alpha_{s_{n-1},s_n}(y, \sigma(u'))$. Thus, we have an atom of the form $\alpha_{s_i,s_{i+1}}(y, x)$. Using property ($\star$) from above, and considering the different possible values for x, we can infer that $\pi(x)$ is an $L(\alpha_{s_i,s_{i+1}})$-successor of $\pi(y) = \pi(v)$. As $s_i \neq s_{i+1}$, there must exist a non-empty word $w = uw'$ and state $s' \in S$ with $(s_i, u, s') \in \delta$ and $w' \in L(\alpha_{s',s_{i+1}})$ which witnesses this successor relationship. Moreover, from the way we chose the states $s_1, \ldots, s_n$, we know that the path associated with w starts at $\pi(v) = \pi(y)$, ends at $\pi(x)$, *and* does not pass by $\pi(v)$ in between. In particular, this means that either the path is entirely contained in the subtree rooted at $\pi(v)$ or the path never visits any element below $\pi(v)$. The former option cannot hold, since it would imply that $x = y$ and that $C \in \mathsf{Loop}_\alpha[s_i, s_{i+1}]$, so the atom would have been removed in Step 5. Thus, it must be the case that the first "step" in the path goes from $\pi(v)$ to its parent p. It follows that $u = R_1^-$ for some R_1 such that $(p, \pi(v)) \in R_1^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Note that since $\pi(v) = pSC$, we must have $\mathcal{T} \models S \sqsubseteq R_1$. This shows that this choice of R_1 satisfies both (b) and (c). In what follows, it will also prove useful to note that $\pi(x)$ is an $L(\alpha_{s',s_{i+1}})$-successor of p (witnessed by the word w').

We now consider condition (d). Take some atom of the form $\gamma(x, y)$ which appears in the query at the start of Step 6. Then we know from above that we can find some atom $\alpha(u, u') \in q$ (where $\alpha = \langle S, \Sigma, \delta, s_0, F\rangle$) such that $\gamma(x, y)$ is equal to one of the following atoms which replaced $\alpha(\sigma(u), \sigma(u'))$ during Step 4: $\alpha_{s_0,s_1}(\sigma(u), y), \alpha_{s_1,s_2}(y, y), \ldots, \alpha_{s_{n-2},s_{n-1}}(y, y), \alpha_{s_{n-1},s_n}(y, \sigma(u'))$. It follows that $\gamma(x, y)$ is an atom of the form $\alpha_{s_i,s_{i+1}}(x, y)$. Using property ($\star$) from above, and considering the two possible values for x, we can deduce that $\pi(y) = \pi(v)$ is an $L(\alpha_{s_i,s_{i+1}})$-successor of $\pi(x)$. As $s_i \neq s_{i+1}$, there must exist a non-empty word $w = w'u$ and state $s'' \in S$ with $(s'', u, s_{i+1}) \in \delta$ and $w' \in L(\alpha_{s_i,s''})$ which witnesses this successor relationship. Moreover, from the way we chose $s_1, \ldots, s_n$, we know that the path associated with w starts at $\pi(x)$, ends at $\pi(v) = \pi(y)$, *and does not pass by* $\pi(v)$ *in between*. So either the path is entirely contained in the subtree rooted at $\pi(v)$ or the path never visits any element below $\pi(v)$. The former option would imply that $x = y$ and that $C \in \mathsf{Loop}_\alpha[s_i, s_{i+1}]$, so the atom would have been removed in Step 5. Thus, it must be the case that the last "step" in the path is from p to $\pi(v)$. This means that $u = R_2$ for some R_2 with $(p, \pi(v)) \in R_2^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Moreover, since $\pi(v) = pSC$, the definition of canonical models implies that $\mathcal{T} \models S \sqsubseteq R_2$. Thus, we have shown that choosing R_2 allows us to satisfy (b) and (d). In what follows, we will use the fact that p is an $L(\alpha_{s_i,s''})$-successor of $\pi(x)$. It is also important to note that if $x = y$, then we can apply the arguments for conditions (c) and (d) together to show that p is an $L(\alpha_{s',s''})$-successor of p (with s' as in (c), and s'' as required for (d)).

Now let q' be the query we obtain at the end of Step 8 when all non-deterministic choices are made in the manner described above. Note that if we are in $\mathcal{ELH}$, then $\mathsf{terms}(q') \subseteq \mathsf{terms}(q)$. For DL-Lite$_\mathcal{R}$, we either have $\mathsf{terms}(q') \subseteq \mathsf{terms}(q)$ (if $D \in \mathsf{N_C}$), or we have $\mathsf{terms}(q') \subseteq \mathsf{terms}(q) \cup \{z\}$, where $\alpha_P(z, y)$ is the atom added in Step 8. In the latter case, we have $D = \exists P^-$. If $p \notin \mathsf{Ind}$, then we saw above that $p = p'PD$, and thus, p' is such that $(p', p) \in P^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. In this case, we set $e_z = p'$. If $p \in \mathsf{Ind}$, then we know that $\mathcal{A} \models D(p)$, and hence there is some $b \in \mathcal{A}$ such that $\mathcal{A} \models P(b, p)$. In this case, we set $e_z = b$.

We aim to find a match π' for q' which satisfies the conditions of the lemma. Let π' be the mapping defined as follows:

- $\pi'(u) = \pi(u)$ for every $u \in \mathsf{terms}(q)$ with $\pi(u) \neq \pi(v)$
- $\pi'(u) = p$ for every $u \in \mathsf{terms}(q)$ with $\pi(u) = \pi(v)$
- if $\{z\} = \mathsf{terms}(q') \setminus \mathsf{terms}(q)$, then $\pi'(z) = e_z$ (with e_z defined as above)

Note that π' satisfies the conditions of the lemma because of the way we chose e_z.

To show that π' is a match, first take some concept atom $B(u) \in q'$. There are two possibilities. Either $B(u)$ appears in q and $u \notin \mathsf{Leaf}$, or $B(u)$ was introduced in Step 7. In the former case, we know that π satisfies $B(u)$, and since $\pi'(u) = \pi(u)$ (since $u \notin \mathsf{Leaf}$), the same is true of π'. In the latter case, we must have $u = y$ and $B = D$. As $\pi'(u) = p$ and D was chosen so that $p \in D^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, it follows that π' satisfies $B(u)$.

Now consider some atom $\gamma(t, t') \in q'$. If $y \notin \{t, t'\}$, then $\gamma(t, t') \in q$. As π is a match for q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, it must be the case that $\pi(t')$ is an $L(\gamma)$-successor of $\pi(t')$. Since $\pi'(t) = \pi(t)$ and $\pi'(t) = \pi(t)$, the same holds for π', and so the atom $\gamma(t, t')$ is satisfied by π'. Next suppose that $y \in \{t, t'\}$ and $\{t, t'\} \subseteq \mathsf{terms}(q)$. A straightforward examination of the procedure rewrite shows that there is an atom $\alpha(u, u') \in q$ (where $\alpha = \langle S, \Sigma, \delta, s_0, F\rangle$) which is replaced in Step 4 by the atoms $\alpha_{s_0,s_1}(\sigma(u), y), \alpha_{s_1,s_2}(y, y), \ldots, \alpha_{s_{n-2},s_{n-1}}(y, y), \alpha_{s_{n-1},s_n}(y, \sigma(u'))$ and such that $\gamma(t, t')$ replaces one of the latter atoms during Step 7. We distinguish three cases:

Case 1: $\gamma(t, t')$ replaces $\alpha_{s_i,s_{i+1}}(y, x)$ with $x \neq y$. Then $\gamma(t, t')$ must have the form $\alpha_{s',s_{i+1}}(y, x)$, where s' is the state which was chosen to ensure condition (c) in Step 6. We recall that s' is such that $\pi(x)$ is an $L(\alpha_{s',s_{i+1}})$-successor of p. Since $x \neq y$, we know that $x \notin \mathsf{Leaf}$, and so $\pi(x) = \pi'(x)$. It follows that $\pi'(x)$ is an $L(\alpha_{s',s_{i+1}})$-successor of $\pi'(y) = p$,

so the atom $\gamma(t,t')$ is satisfied by π'.

Case 2: $\gamma(t,t')$ replaces $\alpha_{s_i,s_{i+1}}(x,y)$ with $x \neq y$. Then $\gamma(t,t')$ must have the form $\alpha_{s_i,s''}(x,y)$, where s'' is the state which was used in condition 6(d). We showed earlier when examining condition 6(d) that p is an $L(\alpha_{s_i,s''})$-successor of $\pi(x)$. Using the fact that $\pi'(x) = \pi(x)$ and $\pi'(y) = p$, we can infer that $\pi'(y)$ is an $L(\alpha_{s_i,s''})$-successor of $\pi'(x)$, hence π' satisfies the atom $\gamma(t,t')$.

Case 3: $\gamma(t,t')$ replaces $\alpha_{s_i s_{i+1}}(y,y)$. Then $\gamma(t,t')$ must have the form $\alpha_{s',s''}(y,y)$, where s' is the state from 6(c) and s'' is the state from 6(d). We have seen above that $p = \pi'(y)$ is an $L(\alpha_{s',s''})$-successor of $p = \pi'(y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, which means that π' satisfies the atom $\gamma(t,t')$.

The remaining possibility is that $\gamma(t,t')$ is the atom $\alpha_P(z,y)$ added during Step 8. Then we use the facts that $\pi'(z) = e_z$, $\pi'(y) = p$, $(e_z,p) \in P^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, and $L(\alpha_P) = \{P\}$ to show that $\pi'(z)$ is an $L(\alpha_P)$-successor of $\pi'(y)$, and hence that π' satisfies $\gamma(t,t')$.

As we have shown that every atom in q' is satisfied by the mapping π', it follows that π' is a match for q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, which completes the proof. □

Lemma 11. It can be decided in polytime in $|\mathcal{T}|$ and $|\alpha|$ whether $G \in \mathsf{ALoop}_\alpha[s,s']$.

Proof. Fix a DL-Lite$_\mathcal{R}$ or $\mathcal{ELH}$ TBox $\mathcal{T}$ and NFA $\alpha = \langle S, \Sigma, \delta, s, F\rangle$, and let G be a subset of $\mathsf{BC}_\mathcal{T}$.

If $\mathcal{T}$ is a DL-Lite$_\mathcal{R}$ TBox, then let L be the subset of $S \times S$ defined inductively using the following rules:

(1) for every $s \in S$, $(s,s) \in L$

(2) if $(s_1,s_2) \in L$ and $(s_2,s_3) \in L$, then $(s_1,s_3) \in L$

(3) if $A \in G$ and $(s_1, A?, s_2) \in \delta$, then $(s_1,s_2) \in L$

(4) if $C \in G$, $\mathcal{T} \models C \sqsubseteq \exists R$, $\mathcal{T} \models R \sqsubseteq R'$, $\mathcal{T} \models R^- \sqsubseteq R''$, $(s_1,R',s_2) \in \delta$, $\exists R^- \in \mathsf{Loop}_\alpha[s_2,s_3]$, and $(s_3,R'',s_4) \in \delta$, then $(s_1,s_4) \in L$

For $\mathcal{ELH}$, we replace the fourth rule by:

(4') if $C \in G, \mathcal{T} \models C \sqsubseteq \exists r.D, \mathcal{T} \models r \sqsubseteq r', \mathcal{T} \models r^- \sqsubseteq r''$, $(s_1,r',s_2) \in \delta$, $D \in \mathsf{Loop}_\alpha[s_2,s_3]$, and $(s_3,r'',s_4) \in \delta$, then $(s_1,s_4) \in L$

Clearly, for both DL-Lite$_\mathcal{R}$ and $\mathcal{ELH}$, the set L can be computed in polynomial time in $\mathcal{T}$ and α. To complete the proof, we establish the following claim:

Claim: $G \in \mathsf{ALoop}_\alpha[s,s']$ if and only if $(s,s') \in L$.

Proof of claim. For the first direction, suppose that $G \in \mathsf{ALoop}_\alpha[s,s']$. Consider some arbitrary individual a such that $G = \{C \in \mathsf{BC}_\mathcal{T} \mid a \in C^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}\}$. Then it follows from the definition of ALoop_α that a is an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_a$. Thus, we can find a word $w = u_1 \dots u_m \in L(\alpha_{s,s'})$ and a sequence $\sigma = e_0, \dots, e_m$ with $a = e_0 = e_m$ of elements in $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}|_a}$ such that $\sigma \in w^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. We assume without loss of generality that m is minimal, i.e. we cannot find a different word from $L(\alpha_{s,s'})$ and associated sequence of elements satisfying the same properties but with length $< m$. Now let $j_1 < \dots < j_{n-1}$ be all of the indices $1 < i < m$ such that $e_i = a$. We define the words $w_1, \dots, w_n$ by setting:

- $w_1 = u_1 \dots u_{j_1}$
- $w_i = u_{j_{i-1}+1} \dots u_{j_i}$ for $2 \leq i < n$
- $w_n = u_{j_{n-1}+1} \dots u_m$

We also define a sequence of states $s_1, \dots, s_{n-1}$ such that:

- $w_1 \in L(\alpha_{s,s_1})$
- $w_i \in L(\alpha_{s_{i-1},s_i})$ for $2 \leq i < n$
- $w_n \in L(\alpha_{s_{n-1},s'})$

We aim to show that all of the pairs $(s,s_1),(s_1,s_2),\dots,(s_{n-2},s_{n-1}),(s_{n-1},s')$ belong to L. The proof is the same for each such pair, so we show it just for a pair of the form (s_{i-1},s_i). The first possibility is that $j_i = j_{i-1}+1$ and $w_i = A?$ for some concept name A. In that case, we must have $a \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, hence $A \in G$. We can thus apply the third rule to obtain $(s_{i-1},s_i) \in L$. The other possibility is that $j_i > j_{i-1}+1$. We know that a is the root of $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_a$ and that a does not appear in $e_{j_{i-1}+1}, \dots, e_{j_i-1}$. From this, we can infer that every element e_ℓ with $j_{i-1} < e_\ell < j_i$ is a (proper) descendant of a. In particular this means that we must have $e_{j_{i-1}+1} = e_{j_i-1} = p$ for some child $p = aRD$ of a. We also know that $w_i \in L(\alpha_{s_{i-1},s_i})$ and $\sigma_i = e_{j_{i-1}} \dots e_{j_i}$ are such that $\sigma_i \in w_i^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Thus, there must exist some states s'' and s''' such that:

- $(s_{i-1}, u_{j_{i-1}+1}, s'') \in \delta$
- $(s''', u_{j_i}, s_i) \in \delta$
- $u_{j_{i-1}+1} \in \overline{\mathsf{N_R}}$ and $(a,p) \in u_{j_{i-1}+1}^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$
- $u_{j_i} \in \overline{\mathsf{N_R}}$ and $(p,a) \in u_{j_i}^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$
- p is an $L(\alpha_{s'',s'''})$-successor of itself in $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_p$.

By Lemma 7, the last item can be rewritten as: $D \in \mathsf{Loop}_\alpha[s'',s''']$. Moreover, from the definition of canonical models, we know that there must exist some $C \in G$ such that $\mathcal{T} \models C \sqsubseteq \exists r.D$ (or $\mathcal{T} \models C \sqsubseteq \exists R$ where $D = \exists R^-$ for DL-Lite$_\mathcal{R}$). It also follows from the canonical model definition and the fact that $(a,p) \in u_{j_{i-1}+1}^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and $(p,a) \in u_{j_i}^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ that we have both $\mathcal{T} \models R \sqsubseteq U\}$ and $\mathcal{T} \models R^- \sqsubseteq u_{j_i}\}$. We thus have all of the necessary conditions to apply the fourth rule, hence (s_{i-1},s_i) must belong to L. Now that we have seen that all of the pairs $(s,s_1),(s_1,s_2),\dots,(s_{n-2},s_{n-1}),(s_{n-1},s')$ belong to L, we can use repeated applications of the second rule to conclude that $(s,s') \in L$.

For the second direction, consider a sequence of applications of the rules 1-4 which generates the set L, and let $p_1, \dots, p_n$ be the pairs in L, in the order that they were inserted into L. We show by induction that for every $1 \leq i \leq n$ the pair $p_i = (s_i, s_i')$ is such that $G \in \mathsf{ALoop}_\alpha[s_i, s_{i+1}]$. The

base case is the first pair p_1, which must result from an application of rule 1, and hence is of the form (s, s) for some state $s \in S$. Then the property trivially holds since every individual a is an $L(\alpha_{s,s})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_a$. Thus, suppose that the property holds for all p_i with $1 \leq i < k$, and consider the pair $p_k = (s, s')$. If p_k resulted from an application of the first rule, then we can use the same argument as for the base case. If the second rule was used, then there exists pairs $p_\ell = (s, s'')$ and $p_j = (s'', s')$ with $\ell < k$ and $j < k$. Thus, by the induction hypothesis, we have $G \in \mathsf{ALoop}_\alpha[s, s'']$ and $G \in \mathsf{ALoop}_\alpha[s'', s']$. Thus, for any individual a which satisfies precisely the concepts in G in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, there is a word $w_1 \in L(\alpha_{s,s''})$ and sequence of elements σ_1 from $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_a$ which starts and ends with a such that $\sigma_1 \in w_1^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Likewise, we can find a word-sequence pair (w_2, σ_2) which witnesses the loop from s'' to s at a. Then the word $w = w_1 w_2 \in L(\alpha_{s,s'})$ and sequence $\sigma = \sigma_1' \sigma_2$ (where σ_1' is σ_1 with final a removed) are such that $\sigma \in w^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. As σ begins and ends with a, we obtain $G \in \mathsf{ALoop}_\alpha[s, s']$. Next suppose that p_k was obtained by applying rule 3. Then there must be $A \in G$ such that $(s, A?, s') \in \delta$. It follows immediately that $G \in \mathsf{ALoop}_\alpha[s, s']$.

The final possibility is that p_k results from an application of rule 4. Here the proof differs depending on whether we are in DL-Lite$_\mathcal{R}$ or $\mathcal{ELH}$. We give the proof only for DL-Lite$_\mathcal{R}$. We know that there must exist some concept $C \in G$, roles R, R', R'', and states s'', s''' such that:

- $\mathcal{T} \models C \sqsubseteq \exists R$
- $\mathcal{T} \models R \sqsubseteq R'$
- $\mathcal{T} \models R^- \sqsubseteq R''$
- $(s, R', s'') \in \delta$
- $\exists R^- \in \mathsf{Loop}_\alpha[s'', s''']$
- $(s''', R'', s') \in \delta$

Let a be some individual which satisfies all concepts in G in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. As $C \in G$ and $\mathcal{T} \models C \sqsubseteq \exists R$, the path $e = aR\exists R^-$ belongs to $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Using $\exists R^- \in \mathsf{Loop}_\alpha[s'', s''']$, we can infer that e is an $L(\alpha_{s'',s'''})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_e$, and hence there is some word $w \in L(\alpha_{s'',s'''})$ and some sequence of elements σ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_e$ such that $w \in \sigma^{\mathcal{I}_{\mathcal{A},\mathcal{T}}|_e}$. Using the above points, we can show that the word $w' = R'wR''$ and sequence $\sigma' = a\sigma a$ are such that $w' \in \sigma'^{\mathcal{I}_{\mathcal{T},\mathcal{A}}|_a}$. We can thus conclude that a is an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_a$. □

Lemma 13. $\mathcal{T}, \mathcal{A} \models q$ if and only if $\mathcal{T}, \mathcal{A} \mathrel{|\!\approx} q'$ for some $q' \in \mathsf{rewrite}(q, \mathcal{T})$.

Proof. For the first direction, suppose that $\mathcal{T}, \mathcal{A} \mathrel{|\!\approx} q'$ for some $q' \in \mathsf{rewrite}(q, \mathcal{T})$. Then we can find a mapping π which satisfies the conditions of Definition 12. We wish to show that π is a match for q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. First we note that because of condition (a), we have $\pi(c) = c$ for every individual c appearing in q'. Next consider some atom $A(t) \in q'$. By condition (b), we know that $\mathcal{T}, \mathcal{A} \models A(\pi(t))$. It follows from the definition of the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ that $\pi(t) \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Finally consider some atom $\alpha(t, t') \in q'$, where $\alpha = \langle S, \Sigma, \delta, s, F \rangle$. Then by condition (c), there is a sequence $(a_0, s_0), \ldots (a_n, s_n)$ of distinct pairs from $\mathsf{Ind}(\mathcal{A}) \times S$ such that $a_0 = \pi(t)$, $a_n = \pi(t')$, $s_0 = s$, $s_n \in F$, and for every $0 \leq i < n$, one of the following holds:

(i) $a_i = a_{i+1}$ and $\{C \in \mathsf{BC}_\mathcal{T} \mid \mathcal{T}, \mathcal{A} \models C(a_i)\} \in \mathsf{ALoop}_\alpha[s_i, s_{i+1}]$

(ii) $\mathcal{T}, \mathcal{A} \models R(a_i, a_{i+1})$ and (s_i, R, s_{i+1}), for some R.

For each $0 \leq i < n$, we select a word and sequence of elements in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. The definition will depend on whether i satisfies condition (i) or condition (ii). First suppose that i is such that condition (i) is satisfied. Then $\{C \in \mathsf{BC}_\mathcal{T} \mid \mathcal{T}, \mathcal{A} \models C(a_i)\} \in \mathsf{ALoop}_\alpha[s_i, s_{i+1}]$, and so we can choose a word $w_i \in L(\alpha_{s_i,s_{i+1}})$ and a sequence σ_i' beginning and ending with a_i satisfying $\sigma_i' \in w_i^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. We let σ_i be σ_i' with the last a_i removed. Now consider the case in which i is such that only condition (ii) is satisfied. Then we set $w_i = R$ and $\sigma_i = a_i$. Now we consider the following word and sequence:

$$w = w_0 \ldots w_{n-1} \qquad \sigma = \sigma_0 \ldots \sigma_{n-1} a_n$$

It is easily verified that $w \in L(\alpha_{s_0,s_n})$ and that $\sigma \in w^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Since σ begins with $a_0 = \pi(t)$ and ends with $a_n = \pi(t')$, we have thus shown that $\pi(t')$ is an $L(\alpha_{s_0,s_n})$-successor of $\pi(t)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, which means that π satisfies the atom $\alpha(t, t')$. We can thus conclude that π is a match for q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. Moreover by definition we have that $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ for every $t \in \mathsf{terms}(q')$. Since $q' \in \mathsf{rewrite}(q, \mathcal{T})$, by applying Lemma 9, we obtain $\mathcal{T}, \mathcal{A} \models q$.

For the second direction, we suppose that $\mathcal{T}, \mathcal{A} \models q$. By Lemma 9, there exists $q' \in \mathsf{rewrite}(q, \mathcal{T})$ and a match π for q' in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ such that $\pi(t) \in \mathsf{Ind}(\mathcal{A})$ for every $t \in \mathsf{terms}(q')$. We claim that the mapping π satisfies the requirements of Definition 12. Condition (a) is trivially satisfied, since π is a match for q', and hence must map every individual in q' to itself. For condition (b), consider some atom $A(t)$ in q'. Since π is a match for q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, we have that $\pi(t) \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. By the construction of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, it follows that $\mathcal{T}, \mathcal{A} \models A(\pi(t))$, so condition (b) holds. To show condition (c), consider some atom $\alpha(t, t') \in q'$ with $\alpha = \langle S, \Sigma, \delta, s_0, F \rangle$. Since π is a match for q' in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, it follows that $\pi(t')$ is an $L(\alpha)$-successor of $\pi(t)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. This means that we can find some word $w = u_1 \ldots u_n \in L(\alpha)$ and some sequence $\sigma = e_0, \ldots, e_n$ of elements in $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ such that $e_0 = \pi(t)$, $e_n = \pi(t')$, and $\sigma \in w^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. We suppose w.l.o.g. that n is minimal, i.e. we cannot find another word and sequence of elements satisfying the above conditions but with length less than n. As $w \in L(\alpha)$ and $\sigma \in w^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, it follows that we can find a sequence of states , $s_1, \ldots, s_n$ such that:

- $(s_0, u_1, s_1) \in \delta$
- $(s_{i-1}, u_i, s_i) \in \delta$ for every $2 \leq i \leq n$
- $s_n \in F$

Now we let $j_1 < \ldots < j_m$ be all of the indices i such that $e_i \in \mathsf{Ind}(\mathcal{A})$. Consider the sequence of pairs

$$(e_{j_1}, s_{j_1}) \ldots (e_{j_m}, s_{j_m})$$

We claim that this sequence satisfies all requirements of condition (c) of Definition 12. First we note that since $e_0 =$

$\pi(t) \in \mathsf{Ind}(\mathcal{A})$ and $e_n = \pi(t') \in \mathsf{Ind}(\mathcal{A})$, we have $j_1 = 0$ and $j_m = n$, and hence $e_{j_1} = \pi(t)$, $s_{j_1} = s_0$, $e_{j_m} = \pi(t')$, and $s_{j_m} \in F$. Next consider some arbitrary index j_k with $k < m$. Then there are two possibilities. First suppose that $j_{k+1} = j_k + 1$. Then there must exist some $R \in \overline{\mathsf{N_R}}$ such that $(e_{j_k}, e_{j_{k+1}}) \in R^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and $(s_{j_k}, R, s_{j_{k+1}}) \in \delta$, which means that condition (c)(ii) holds. The other possibility is that $j_{k+1} > j_k + 1$. Since we know that $e_i \notin \mathsf{Ind}(\mathcal{A})$ for all i between j_k and j_{k+1}, and that the anonymous part is tree-shaped, it follows that $e_{j_k} = e_{j_{k+1}}$, and moreover, every e_i with $j_k < i < j_{k+1}$ belongs to $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_{e_{j_k}}$. Thus, the word $u_{j_k+1} \dots u_{j_{k+1}}$ and sequence $e_{j_k} \dots e_{j_{k+1}}$ witnesses that e_{j_k} is an $L(\alpha_{s_{j_k}, s_{j_{k+1}}})$-successor of itself in $\mathcal{I}_{\mathcal{T},\mathcal{A}}|_{j_k}$. It follows that $\{C \in \mathsf{BC}_\mathcal{T} \mid \mathcal{T}, \mathcal{A} \models C(e_{j_k})\} \in \mathsf{ALoop}_\alpha[s_{j_k}, s_{j_{k+1}}]$, so condition (c)(i) holds. We have thus shown that for every atom $\alpha(t, t') \in q'$, either (i) or (ii) holds, which means that condition (c) is satisfied by π. This yields a mapping with the required properties, so $\mathcal{T}, \mathcal{A} \not\approx q'$. □

We now turn to the proof of point 2 of Proposition 15. To treat 2RPQs of the form $\exists x, y\, \alpha(x, y)$, we employ the following lemma:

Lemma 18. *$\mathcal{T}, \mathcal{A} \models \exists x, y\, \alpha(x, y)$ if and only if there is a match π for $\alpha'(x, y)$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ such that $\pi(x), \pi(y) \in \mathsf{Ind}(\mathcal{A})$, where α' is an NFA such that $L(\alpha') = (\overline{\mathsf{N_R}})^* \cdot L(\alpha) \cdot (\overline{\mathsf{N_R}})^*$.*

Proof. For the first direction, suppose that $\mathcal{T}, \mathcal{A} \models \exists x, y\, \alpha(x, y)$, and let π be a match for $\exists x, y\, \alpha(x, y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. Then there exists $w \in L(\alpha)$ and a sequence σ starting with $\pi(x)$ and ending with $\pi(y)$ such that $\sigma \in w^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. If $\pi(x) \in \mathsf{Ind}(\mathcal{A})$, then set $w_x = \sigma_x = \epsilon$. Otherwise, $\pi(x)$ must be of the form $aR_1C_1 \dots R_nC_n$ for $n \geq 1$. Set $w_x = R_1 \dots R_n$ and $\sigma_x = a(aR_1C_1) \dots (aR_1C_1 \dots R_{n-1}C_{n-1})$. If $\pi(y) \in \mathsf{Ind}(\mathcal{A})$, we set $w_y = \sigma_y = \epsilon$, and otherwise, we let $w_y = R_n^- \dots R_1^-$ and $\sigma_x = (bS_1D_1 \dots S_{n-1}D_{n-1}) \dots (bS_1D_1)b$, where $\pi(y) = bS_1D_1 \dots S_nD_n$. It can be easily verified that $w_x w w_y \in L(\alpha')$ and that $\sigma_x \sigma \sigma_y \in (w_x w w_y)^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, hence b is a $L(\alpha')$-successor of a in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. By defining $\pi'(x) = a$ and $\pi'(b)$, we obtain the desired match for $\alpha'(x, y)$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$.

For the other direction, let π be a match for $\alpha'(x, y)$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. Then there must exist words $w_0 \in (\overline{\mathsf{N_R}})^*$, $w_1 \in L(\alpha)$, and $w_2 \in (\overline{\mathsf{N_R}})^*$ and sequences $\sigma_0, \sigma_1, \sigma_2$ such that:

- σ_0 begins with $\pi(x)$, and σ_2 ends with $\pi(y)$
- the last element of σ_0 is the first element of σ_1
- the last element of σ_1 is the first element of σ_2
- $\sigma_i \in w_i^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ for $0 \leq i \leq 2$

Then by letting $\pi'(x)$ (resp. $\pi'(y)$) be the first (resp. last) element of σ_1, we obtain a match for $\exists x, y\, \alpha(x, y)$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, as witnessed by the word $w_1 \in L(\alpha)$ and sequence σ_1. We thus obtain $\mathcal{T}, \mathcal{A} \models \exists x, y\, \alpha(x, y)$. □

To treat 2RPQs of the form $\exists x\, \alpha(x, x)$, we will utilize the an alternative rewriting algorithm rpq-rewrite presented in Figure 3.

Examining the algorithm rpq-rewrite, we remark that each tuple in $\mathsf{BC}_\mathcal{T} \times S \times S$ is examined at most once by rpq-rewrite,

PROCEDURE rpq-rewrite$(\alpha, \mathcal{T})$

Input: NFA $\alpha = \langle S, \Sigma, \delta, s_0, F \rangle$, TBox $\mathcal{T}$

1. Set $Q = \{\alpha(x, x)\}$, Visited $= \emptyset$, and Frontier $= \{(C, s_0, s_f) \mid C \in \mathsf{BC}_\mathcal{T}, s_f \in F\}$.
2. While Frontier $\neq \emptyset$
   (a) Remove (C, s_1, s_2) from Frontier, add to Visited.
   (b) If $C \in \mathsf{Loop}_\alpha[s_1, s_2]$
   For every $D \in \mathsf{BC}_\mathcal{T}$ such that $D \rightsquigarrow C$.
   Add $D?(x, x)$ to Q.
   (c) If $C \notin \mathsf{Loop}_\alpha[s_1, s_2]$
   Add $C(x) \wedge \alpha_{s_1,s_2}(x, x)$ to Q.
   For every $(s_3, s_4) \in S^2$ such that
   $C \in \mathsf{Loop}_\alpha[s_1, s_3]$ and $C \in \mathsf{Loop}_\alpha[s_4, s_2]$
   For every $(D, R, s_1', s_2') \in \mathsf{BC}_\mathcal{T} \times \overline{\mathsf{N_R}} \times S^2$ such that
   (i) $\mathcal{T} \models D \sqsubseteq \exists R$ and $C = \exists R^-$ [DL-Lite$_R$] or $\mathcal{T} \models D \sqsubseteq \exists R.C$ [$\mathcal{ELH}$];
   (ii) there exist R', R'' with $\mathcal{T} \models R^- \sqsubseteq R'$, $\mathcal{T} \models R \sqsubseteq R''$, $(s_1, R', s_1') \in \delta$, and $(s_2', R'', s_2) \in \delta$;
   (iii) $(D, s_1', s_2') \notin$ (Frontier $\cup$ Visited).
   Add (D, s_1', s_2') to Frontier.
3. Output Q.

Figure 3: Query rewriting procedure rpq-rewrite.

and the loop and entailment checks can be done in polynomial time. We thus obtain:

Lemma 19. *The algorithm* rpq-rewrite *runs in polynomial time, and hence produces a polynomial number of queries.*

In what follows, we will somewhat abuse notation by using rpq-rewrite$(\alpha, \mathcal{T})$ to refer to the set of queries which are output by rpq-rewrite on input $(\alpha, \mathcal{T})$.

Lemma 20. *Then $\mathcal{T}, \mathcal{A} \models \exists x\, \alpha(x, x)$ if and only if there a match π for some query $q(x) \in$ rpq-rewrite$(\alpha, \mathcal{T})$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ such that $\pi(x) \in \mathsf{Ind}(\mathcal{A})$.*

Proof. For the first direction, suppose that $\mathcal{T}, \mathcal{A} \models \exists x \alpha(x, x)$, and let π be a match for $\exists x \alpha(x, x)$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. If $\pi(x) \in \mathsf{Ind}(\mathcal{A})$, then the statement trivially holds since $\alpha(x, x)$ is added to Q in Step 1. Thus, suppose that $\pi(x) \notin \mathsf{Ind}(\mathcal{A})$. We start by proving the following claim:

Claim: If (C, s_1, s_2) is added to Frontier at some point during the execution of rpq-rewrite$(\alpha, \mathcal{T})$, $C \notin \mathsf{Loop}_\alpha[s_1, s_2]$, and there is a match ρ for $C(x) \wedge \alpha_{s_1,s_2}(x, x)$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ such that $\rho(x) \notin \mathsf{Ind}(\mathcal{A})$, then there is a tuple (D, s_3, s_4) which is added to Frontier at some point and such that the query $D(x) \wedge \alpha_{s_3,s_4}(x, x)$ has a match ρ' such that $\rho'(x)$ is the parent of $\rho(x)$ and $\rho'(x) \in D^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$.

Proof of claim. Suppose that (C, s_1, s_2) and ρ satisfy the conditions of the claim. Since $\rho(x) \notin \mathsf{Ind}(\mathcal{A})$, we have that $\rho(x) = pRC$ for some role R and concept C. At some point during Step 2, the tuple (C, s_1, s_2) will be selected, and

we will enter (c) because $C \notin \mathsf{Loop}_\alpha[s_1, s_2]$. Since ρ is a match for $\alpha_{s_1,s_2}(x,x)$, there exists $w = u_1, \ldots, u_n \in L(\alpha)$ and a sequence $e = e_0, \ldots, e_n$ of elements in $\Delta^{\mathcal{I}}$ such that $e_0 = e_n = \pi(x)$, and, for all $1 \leq i \leq n$:

- if $u_i = A?$, then $e_{i-1} = e_i \in A^{\mathcal{I}}$
- if $u_i = R \in \mathsf{N_R} \cup \overline{\mathsf{N_R}}$, then $\langle e_{i-1}, e_i \rangle \in R^{\mathcal{I}}$

Since $w \in L(\alpha)$, we can find a sequence of states $q_0 q_1 \ldots q_n$ with $q_0 = s_1$ and $q_n = s_2$ such that for every $1 \leq i \leq n$, $(q_{i-1}, u_i, q_i) \in \delta$. Because $C \notin \mathsf{Loop}_\alpha[s_1, s_2]$, we know that the match ρ is not fully contained within $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_{\pi(x)}$. Thus, there must be at least two occurrences of the parent p of $\pi(x)$ in the sequence e. Let e_j and e_k respectively be the first and last occurrences of p in e, and let Note that we must thus have $e_{j-1} = e_{k+1} = \pi(x)$. Set $s_3 = q_{j-1}$ and $s_4 = q_{k+1}$. Then the words $u_1 \ldots u_{j-1}$ and $u_{k+2} \ldots u_n$ and sequences $e_0 \ldots e_{j-1}$ and $e_{k+1} \ldots e_n$ witness that $\pi(x)$ is both an $L(\alpha_{s_1,s_3})$-successor and an $L(\alpha_{s_4,s_2})$-successor of itself. As by construction, the sequences $e_0 \ldots e_{j-1}$ and $e_{k+1} \ldots e_n$ are fully contained within $\mathcal{I}_{\mathcal{A},\mathcal{T}}|_{\pi(x)}$, this means that $C \in \mathsf{Loop}_\alpha[s_1, s_3]$ and $C \in \mathsf{Loop}_\alpha[s_4, s_2]$. Thus, $(s_3, s_4) \in S^2$ satisfies the required conditions, and so we will enter the for-loop. By the structure of canonical models, we know that either p has the form $p'SD$, or $p = a \in \mathsf{Ind}(\mathcal{A})$ and $p \in D^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ for some $D \in \mathsf{BC}_\mathcal{T}$ such that $\mathcal{T} \models D \sqsubseteq \exists R$ and $C = \exists R^-$ (if $\mathcal{T}$ is formulated in DL-Lite$_R$), or $\mathcal{T} \models D \sqsubseteq \exists R.C$ (for $\mathcal{T}$ an $\mathcal{ELH}$ TBox). Thus, we can choose $D \in \mathsf{BC}_\mathcal{T}$ so that $p \in D^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ and item (i) of the for-loop condition is satisfied. We set $s_1' = q_j$ and $s_2' = q_k$. We know that $(s_3, u_j, s_1') = (q_{j-1}, u_j, q_j) \in \delta$ and $(s_2', u_{k+1}, s_4) = (q_k, u_{k+1}, q_{k+1}) \in \delta$. As $e_{j-1} \neq e_j$ and $e_k \neq e_{k+1}$, it follows that u_j and u_{k+1} belong to $\overline{\mathsf{N_R}}$. We thus have $(e_{j-1}, e_j) \in u_j^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ and $(e_k, e_{k+1}) \in u_{k+1}^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. Again, because of the properties of canonical models, we can derive that $\mathcal{T} \models R^- \sqsubseteq u_j$ and $\mathcal{T} \models R \sqsubseteq u_{k+1}$, and thus that part (ii) of the for-loop condition is satisfied by the tuple (D, R, s_1', s_2'). If the tuple (D, R, s_1', s_2') has not yet been examined, then it is added to Frontier. Thus, at some point during the execution, the tuple (D, R, s_1', s_2') will be examined, and the query $D(x) \wedge \alpha_{s_1',s_2'}(x,x)$ will be added to Q, and hence belongs to the output of rpq-rewrite$(\alpha, \mathcal{T})$. To complete the proof of the claim, we remark that the word $w_{j+1} \ldots w_k$ and sequence $e_j \ldots e_k$ witnesses that p is an $L(\alpha_{s_1',s_2'})$-successor of itself. Moreover, we know from above that $p \in D^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. It follows that there is a match for $D(x) \wedge \alpha_{s_1',s_2'}(x,x) \in$ rpq-rewrite$(\alpha, \mathcal{T})$ which maps x to the parent p of $\pi(x)$. *(end proof of claim)*

Since $\pi(x) \notin \mathsf{Ind}(\mathcal{A})$, we have $\pi(x) = pRC$ for some $R \in \overline{\mathsf{N_R}}$ and $C \in \mathsf{BC}_\mathcal{T}$. From the definition of canonical models, we have $\pi(x) \in C^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. Since π is a match for $\alpha(x,x)$, π must also be a match for $\alpha_{s_0,s_f}(x,x)$ for some $s_f \in F$. In Step 1, the tuple (C, s_0, s_f) will be added to Frontier. Repeated applications of the above claim either yield a query $q(x) \in$ rpq-rewrite$(\alpha, \mathcal{T})$ having a match ρ mapping x to the ABox, or results in the insertion of a tuple (D, s_1, s_2) into Frontier such that $D(x) \wedge \alpha_{s_1,s_2}(x,x)$ has a match in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ and $D \in \mathsf{Loop}_\alpha[s_1, s_2]$. Let ρ be a match for $D(x) \wedge \alpha_{s_1,s_2}(x,x)$, and let $a \in \mathsf{Ind}(\mathcal{A})$ be such that $\rho(x) \in \mathcal{I}_{\mathcal{A},\mathcal{T}}|_a$. Then there is some $E \in \mathsf{BC}_\mathcal{T}$ such that $E \rightsquigarrow D$ and $a \in E^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. Thus, there is a match for $E?(x,x) \in$ rpq-rewrite$(\alpha, \mathcal{T})$ which maps x to $a \in \mathsf{Ind}(\mathcal{A})$.

For the other direction, we start by establishing the following claim:

Claim: If (C, s_1, s_2) is added to Frontier at some point during the execution of rpq-rewrite$(\alpha, \mathcal{T})$, then $\exists x\, C(x) \wedge \alpha_{s_1,s_2}(x,x) \subseteq \exists x\, \alpha(x,x)$.

Proof of claim. The proof is by induction on the precedence relation obtained by setting $(C, s, s') \prec (D, s'', s''')$ if the tuple (D, s'', s''') is added to Frontier during the examination of tuple (C, s, s'). For the base case, we have the tuples (C, s_1, s_2) which are inserted in Step 1. Every such tuple has the form (C, s_0, s_f), where $s_f \in F$, and thus we trivially have $\exists x\, C(x) \wedge \alpha_{s_0,s_f}(x,x) \subseteq \exists x\, \alpha(x,x)$. Next suppose that we already shown the property for (C, s, s'), and let (D, s'', s''') be such that $(C, s, s') \prec (D, s'', s''')$. Further suppose that there is a match π for $D(x) \wedge \alpha_{s'',s'''}(x,x)$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. Then we can find $w = u_1 \ldots u_n \in L(\alpha)$ and a sequence $e = e_0, \ldots, e_n$ of elements in $\Delta^{\mathcal{I}}$ such that $e_0 = e_n = \pi(x)$, and, for all $1 \leq i \leq n$:

- if $u_i = A?$, then $e_{i-1} = e_i \in A^{\mathcal{I}}$
- if $u_i = R \in \mathsf{N_R} \cup \overline{\mathsf{N_R}}$, then $\langle e_{i-1}, e_i \rangle \in R^{\mathcal{I}}$

We can also find a sequence of states $q_0 q_1 \ldots q_n$ with $q_0 = s''$ and $q_n = s'''$ such that for every $1 \leq i \leq n$, $(q_{i-1}, u_i, q_i) \in \delta$. Because $(C, s, s') \prec (D, s'', s''')$, there must exist some R such that (D, s'', s''') was added to Frontier when examining the tuple (D, R, s'', s'''). We know that this tuple satisfied conditions (i)-(iii), and hence that

- $\mathcal{T} \models D \sqsubseteq \exists R$ and $C = \exists R^-$ [DL-Lite$_R$], or $\mathcal{T} \models D \sqsubseteq \exists R.C$ [$\mathcal{ELH}$]
- there exist R', R'' such that $\mathcal{T} \models R^- \sqsubseteq R', \mathcal{T} \models R \sqsubseteq R''$, $(s_1, R', s_1') \in \delta$, and $(s_2', R'', s_2) \in \delta$

By the first point, the path $p = \pi(x)RC$ belongs to the canonical model. We can then use the word $w' = R'\, w\, R'' \in L(\alpha_{s,s'})$ and sequence $e' = p\, e\, p$ to show that p is an $L(\alpha_{s,s'})$-successor of itself in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. Since we also know that $p \in C^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$, we obtain a match for $\exists x\, C(x) \wedge \alpha_{s,s'}(x,x)$ in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. By the induction hypothesis, $\exists x\, C(x) \wedge \alpha_{s,s'}(x,x) \subseteq \exists x\, \alpha(x,x)$, and so there must also exist a match for $\exists x\, \alpha(x,x)$. This yields the desired containment. (*end proof of claim*)

Now suppose that there is a match for some $q(x) \in$ rpq-rewrite$(\alpha, \mathcal{T})$. There are three possibilities for q. The first is that $q(x) = \alpha(x,x)$, in which case we trivially have $\mathcal{T}, \mathcal{A} \models \exists x \alpha(x,x)$. The next possibility is that $q(x) = \exists x\, C(x) \wedge \alpha_{s,s'}(x,x)$, in which case we can apply the claim to find a match for $\alpha(x,x)$. The final possibility is that $q(x) = D?(x,x)$, which occurs when some $(C, s, s') \in$ Frontier is such that $C \in \mathsf{Loop}_\alpha[s, s']$ and $D \rightsquigarrow C$. In this case, we have $\exists x\, D(x) \subseteq \exists x C(x)$ (since $D \rightsquigarrow C$),

$\exists x\, C(x) \subseteq \exists x\, C(x) \wedge \alpha_{s,s'}(x,x)$ (since $C \in \mathsf{Loop}_\alpha[s,s']$), and $\exists x\, C(x) \wedge \alpha_{s,s'}(x,x) \subseteq \exists x\, \alpha(x,x)$ (by assumption). Putting these statements together, we obtain $\exists x\, D?(x,x) \subseteq \exists x\, \alpha(x,x)$, which yields $\mathcal{T}, \mathcal{A} \models \exists x \alpha(x,x)$. □

Note that the queries output by rpq-rewrite are either 2RPQs, or of the form $\exists x\, C(x) \wedge \alpha(x,x)$, and queries of the latter form are trivially transformed into 2RPQs.

We are now able to prove the second statement in Proposition 15.

Proposition 21. *2RPQ-answering is in P (combined complexity) for DL-Lite$_\mathcal{R}$ and $\mathcal{ELH}$.*

Proof. Lemmas 18, 19, and 20 and the preceding remark provide a polynomial reduction of Boolean 2RPQ-answering to the problem of deciding whether a 2RPQ has a match mapping all variables to the ABox. The latter problem can be decided in P by iterating over the potential mappings and checking whether the conditions of Definition 12 are satisfied. Since 2RPQs contain at most 2 variables, they are only polynomially many mappings to check. □