Journal of Artificial Intelligence Research 82 (2025) 851-899 Submitted 04/2024; published 02/2025

Efficient Ontology-Mediated Query Answering: Extending DL-lite$_\mathcal{R}$ and Linear $\mathcal{ELH}$

Mirko M. Dimartino MIRKO.DIMARTINO@PRIMA.IT
Prima Assicurazioni, 71–73 Carter Lane, London, EC4V 5EQ

Peter T. Wood P.WOOD@BBK.AC.UK
Knowledge Lab, Birkbeck, University of London,
Malet Street, London WC1E 7HX, UK

Andrea Calì ANDREA.CALI@UNINA.IT
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione,
Università degli Studî di Napoli Federico II, Italy

Alexandra Poulovassilis A.POULOVASSILIS@BBK.AC.UK
Knowledge Lab, Birkbeck, University of London,
Malet Street, London WC1E 7HX, UK

Abstract

The OWL 2 QL profile of the OWL 2 Web Ontology Language, based on the family of description logics called DL-Lite, is designed so that data stored in a standard relational database system (RDBMS) can be queried through an ontology via a rewriting mechanism, i.e., by rewriting the query into an SQL query that is then answered by the RDBMS system, without any changes to the data. In this paper we propose a language whose expressive power goes beyond that of DL-Lite while still allowing query answering via rewriting of queries into unions of conjunctive two-way regular path queries (UC2RPQs) instead of SQL queries. Our language is an extension of both OWL 2 QL and linear $\mathcal{ELH}$: OWL 2 QL is extended by allowing qualified existential quantification on the left-hand side of concept inclusion axioms, and linear $\mathcal{ELH}$ by allowing inverses in role inclusion axioms. We identify a syntactic property of the extended language that guarantees UC2RPQ-rewritability. We propose a novel rewriting technique for conjunctive queries (CQs) under our ontology language that makes use of nondeterministic finite state automata. We show that CQ answering in our setting is NLOGSPACE-complete with respect to data complexity and NP-complete for combined complexity; we also show that answering instance queries is NLOGSPACE-complete for data complexity and in PTIME for combined complexity.

1. Introduction

Ontologies have been successfully employed in the conceptual modelling of data in several areas, particularly in Information Integration and the Semantic Web. An ontology is a specification of the domain of interest of an application, and can be specified using logical rules which, on the one hand, restrict the form of the underlying data, and on the other hand allow for *inference* of information that is not explicitly contained in the data. *Description Logic* (DL) is a family of knowledge representation formalisms that are able to capture a wide range of ontological constructs (Baader & Nutt, 2007). DLs are based on *concepts* (unary predicates representing classes of individuals) and *roles* (binary predicates representing relations between classes). A DL knowledge base consists of a TBox (the *terminological* component) and an ABox (the *assertional* component). The former is a conceptual representation of the schema, while the latter is an instance of the schema.

A common assumption in this context is the so-called *open-world* assumption, namely that the information in the ABox is sound but not complete; the TBox, in particular, specifies how the ABox can be expanded with additional information in order to answer queries. Answers to a query in this context are called *certain answers*, as they correspond to the answers that are true in all models of the theory constituted by the knowledge base (Lenzerini, 2002). The set of all models is represented by the so-called *expansion* (or *chase* (Calì, Lembo, & Rosati, 2003)) of an ABox $\mathcal{A}$ according to a TBox $\mathcal{T}$. Note that neither each model nor the set of all models is necessarily finite. The expansion (chase) is illustrated in the following example.

Example 1. Consider the TBox $\mathcal{T}$ comprising the assertions $\mathit{Parent} \sqsubseteq \mathit{Person}$ and $\mathit{Person} \sqsubseteq \exists \mathit{has}.\mathit{Parent}$, where *Person* and *Parent* are concepts. The first assertion states that every individual in the class *Parent* is also in the class *Person*. In the second assertion, the concept $\exists \mathit{has}.\mathit{Parent}$ denotes the individuals connected via the role *has* to some individual belonging to the concept *Parent*; in other words, it contains all x such that $\mathit{has}(x,y)$ and $\mathit{Parent}(y)$ for some y. Thus, the second assertion states that every individual in the class *Person* is also in the class of individuals who have a parent. Now suppose we have the ABox $\mathcal{A} = \{\mathit{Person}(\mathit{alice})\}$; we can *expand* $\mathcal{A}$ according to the TBox $\mathcal{T}$ so as to add to it all atoms entailed by $(\mathcal{T},\mathcal{A})$; we therefore add $\mathit{has}(\mathit{alice}, z_0)$ and $\mathit{Parent}(z_0)$, where z_0 is a so-called *labelled null*, that is, a placeholder for an unknown value of which we know the existence (note that, with this approach, $\mathcal{A}$ can be expanded further). Given the query $\mathbf{q}$ defined as $q(x) \leftarrow \mathit{has}(x,y)$, the answer to $\mathbf{q}$ under $(\mathcal{T},\mathcal{A})$ is $\{\mathit{alice}\}$ because $\mathit{has}(\mathit{alice}, z_0)$ is entailed by $(\mathcal{T},\mathcal{A})$; in fact, the certain answers to $\mathbf{q}$ are obtained by evaluating $\mathbf{q}$ on the expansion and by considering answers that do not contain nulls. If we consider the query $\mathbf{q}_1$ defined as $q_1(x) \leftarrow \mathit{Parent}(x)$, the answer is empty because z_0, though known to exist, is not known.

Answers to queries over DL knowledge bases can be computed, for certain languages, by *query rewriting* (Calvanese, De Giacomo, Lembo, Lenzerini, & Rosati, 2007). In query rewriting, a new query $\mathbf{q}'$ is computed (rewritten) from the given query $\mathbf{q}$ according to the TBox $\mathcal{T}$, such that the answers to $\mathbf{q}$ on $\mathcal{K} = (\mathcal{T},\mathcal{A})$ are obtained by evaluating $\mathbf{q}'$ on $\mathcal{A}$, where $\mathcal{A}$ is seen as a database; it is said that $\mathbf{q}$ is *rewritten* into $\mathbf{q}'$ and that $\mathbf{q}'$ is the *perfect rewriting of* $\mathbf{q}$ *with respect to* $\mathcal{T}$. The language of $\mathbf{q}'$, called the *target language*, can be more expressive than that of $\mathbf{q}$. Query rewriting has been extensively employed in query answering over ontologies (Gottlob, Orsi, & Pieris, 2011; Pérez-Urbina, Horrocks, & Motik, 2009). A common rewriting technique for DLs and other knowledge representation formalisms, inspired by resolution in Logic Programming, has as the target language *unions of conjunctive queries* (Calì et al., 2003).

Example 2. Let us consider again the knowledge base of Example 1. The perfect rewriting of query $\mathbf{q}$ is the query $\mathbf{q}'$ defined as $q(x) \leftarrow \mathit{Person}(x) \cup \mathit{has}(x,y)$; intuitively, $\mathbf{q}'$ captures the fact that, to search for individuals which are connected via the role *has* to some other individual, we need also to consider individuals in *Person*, because the TBox might infer the former from the latter. The evaluation of $\mathbf{q}'$ on $\mathcal{A}$ returns the correct (i.e., certain) answers.

The OWL 2 QL profile of the OWL 2 Web Ontology Language — which is based on the description logic DL-Lite$_\mathcal{R}$ (Calvanese et al., 2007) — is expressly designed so that query answering can be performed via query rewriting. Data (assertions) that are stored in a standard relational database can be queried through an ontology by rewriting the query into an SQL query that is then answered by the RDBMS, without any changes to the data (for example, such a rewriting was presented in (Calvanese et al., 2007)).

Extending the expressiveness of DL-Lite$_\mathcal{R}$ may lead to the need for a more expressive target language than SQL, i.e. than first-order (FO) queries. This occurs, for example, when *qualified existential quantification* is allowed on the left-hand side (LHS) of axioms, i.e., formulae of the form $\exists R.D$ where R is a role and D a concept. In cases such as this, we say that the language is not *FO-rewritable*. The following example illustrates this issue.

Example 3. Consider the TBox $\mathcal{T} = \{\exists \mathit{hasParent}.\mathit{Person} \sqsubseteq \mathit{Person}\}$ and the query $\mathbf{q}$ defined as $q(x) \leftarrow \mathit{Person}(x)$. Note that an expression of the form $\exists \mathit{hasParent}.\mathit{Person}$ is forbidden on the left-hand sides of axioms in DL-Lite$_\mathcal{R}$. It is easy to see that the query rewriting technique described earlier produces an infinite union of conjunctive queries: $q(x) \leftarrow \mathit{Person}(x)$, $q(x) \leftarrow \mathit{hasParent}(x,y), \mathit{Person}(y)$ and all conjunctive queries of the form $q(x) \leftarrow \mathit{hasParent}(x,y_1), \ldots, \mathit{hasParent}(y_k,y_{k+1}), \mathit{Person}(y_{k+1})$, with $k \geqslant 1$. This cannot be captured by an FO-rewriting (see Theorem 7.8 in (Baader, Horrocks, Lutz, & Sattler, 2017), for example).

However, by adopting the semantic web query language SPARQL 1.1 (Harris & Seaborne, 2013), database systems should be able to answer queries that are more expressive than FO queries since the *property paths* of SPARQL 1.1 are able to express navigational queries by defining regular expressions on predicates. In particular, every conjunctive two-way regular path query (C2RPQ) (Calvanese, De Giacomo, Lenzerini, & Vardi, 2000), as well as unions of C2RPQs (UC2RPQs), can be translated to a SPARQL 1.1 query. Building on this, in this paper we propose a language that extends DL-Lite$_\mathcal{R}$ but still allows query answering via a simple rewriting mechanism, with UC2RPQs instead of SQL queries as the target language. We allow qualified existential quantification on the left-hand sides of axioms and identify a property of the resulting language that allows a rewriting into UC2RPQs. The description logic resulting from this extension, which we call *harmless linear* $\mathcal{ELHI}$, denoted by $\mathcal{ELHI}_h^{\ell in}$, is a generalisation of both DL-Lite$_\mathcal{R}$ (Artale, Calvanese, Kontchakov, & Zakharyaschev, 2009) and linear $\mathcal{ELH}$ (which is called DL-Lite$^+$ in (Pérez-Urbina, Motik, & Horrocks, 2010)).

Example 4. Recall the issue in the previous example, where a finite FO-rewriting was not feasible. In order to capture the infinite FO-rewriting, we can produce a rewriting into a C2RPQ $\mathbf{q}'$ defined as $q(x) \leftarrow \mathit{hasParent}^*(x,y), \mathit{Person}(y)$, where $\mathit{hasParent}^*$ is a regular expression denoting all finite compositions of *hasParent* with itself, i.e.,

$$\mathit{hasParent}^*(x,y) = \mathit{hasParent}(x,y) \cup \mathit{hasParent}(x,z_1), \ldots, \mathit{hasParent}(z_i,y)$$

for all $i \geqslant 1$.

Contributions. This paper significantly extends earlier work (Dimartino, Calì, Poulovassilis, & Wood, 2016) where we first proposed exploiting the capabilities of navigational queries in order to allow rewriting of conjunctive queries into CRPQs (not UC2RPQs) under a more restrictive DL, namely linear $\mathcal{ELH}$. We also give here a complete theoretical development and full proofs. In more detail, the contributions are the following:

- We define $\mathcal{ELHI}_h^{\ell in}$ (harmless linear $\mathcal{ELHI}$), an ontology language that generalises both DL-Lite$_\mathcal{R}$ and linear $\mathcal{ELH}$.
- We show that *instance queries* (queries with a single atom in their body) under $\mathcal{ELHI}_h^{\ell in}$ knowledge bases can be rewritten to 2RPQs (two-way regular path queries), using an algorithm based on non-deterministic finite-state automata.

- We show that *conjunctive queries* (CQs) under $\mathcal{ELHI}_h^{\ell in}$ knowledge bases can be rewritten to UC2RPQs. This algorithm combines the *tree-witness* rewriting of (Kikot, Kontchakov, & Zakharyaschev, 2012; Kontchakov & Zakharyaschev, 2014) with the above rewriting technique for instance queries. Since UC2RPQs can be straightforwardly expressed in SPARQL 1.1 by means of property paths, our approach is therefore directly applicable to real-world querying settings.
- We undertake a complexity analysis for query answering under $\mathcal{ELHI}_h^{\ell in}$. We analyse the computational cost of query answering in terms of both *data complexity* (where the TBox and the query are fixed and the ABox alone is the input) and *combined complexity* (where the query, TBox and ABox all constitute the input). We show that answering instance queries under $\mathcal{ELHI}_h^{\ell in}$ is NLOGSPACE-complete for data complexity and in PTIME for combined complexity; we also show that answering CQs under $\mathcal{ELHI}_h^{\ell in}$ is NLOGSPACE-complete for data complexity and NP-complete for combined complexity.
- We formally prove the correctness of our algorithms, and also that they comply with the upper complexity bounds.

2. Related Work

Query rewriting has been extensively employed in query answering over ontologies expressed in a wide range of different DLs. In terms of the data complexity of conjunctive query evaluation, the DLs range from so-called "expressive DLs" such as $\mathcal{ALC}$ and $\mathcal{SHIQ}$, for which evaluation is coNP-hard (Schaerf, 1993), to "Horn DLs" such as $\mathcal{EL}$ and OWL 2 EL, for which evaluation is PTIME-hard, and the DL-Lite family and OWL 2 QL, for which evaluation is in AC^0. The latter two complexity measures correspond to query rewritability as follows: membership in AC^0 is necessary for FO-rewritability, while membership in PTIME is necessary for rewritability into Datalog.

Calvanese et al. (2007) introduced the DL-Lite family of DLs, with the motivation of defining DLs for which both reasoning and query answering were tractable. The family includes DL-Lite$_{core}$, which provides concept inclusions, disjointness between concepts, role typing, participation constraints, and non-participation constraints, DL-Lite$_{\mathcal{F}}$, which adds functionality restrictions on roles to the core, and DL-Lite$_{\mathcal{R}}$, which adds role inclusions and role disjointness assertions. They show that the data complexity of answering unions of conjunctive queries expressed in these DLs is in LOGSPACE, later improved to AC^0 (a proper subclass of LOGSPACE).

In subsequent work, Calvanese, De Giacomo, Lembo, Lenzerini, and Rosati (2013) introduce DLR-Lite$_{\mathcal{A},\sqcap}$, which generalises DL-Lite$_{core}$, DL-Lite$_{\mathcal{F}}$, DL-Lite$_{\mathcal{R}}$ and DL-Lite$_{\mathcal{A}}$ (a non-trivial fusion of DL-Lite$_{\mathcal{F}}$ and DL-Lite$_{\mathcal{R}}$), by allowing n-ary relations rather than only binary roles. They show that DLR-Lite$_{\mathcal{A},\sqcap}$ is FO-rewritable and therefore that query answering for DLR-Lite$_{\mathcal{A},\sqcap}$ is in AC^0. In addition, they show that adding qualified existential quantification to the left-hand side or right-hand side of concept inclusions makes query answering NLOGSPACE-hard, which implies that conjunctive queries are no longer FO-rewritable. Allowing conjunction on the left-hand side of concept inclusion as well makes query answering PTIME-complete (Calvanese et al., 2013).

Rosati (2007) investigates the data complexity of query answering for the $\mathcal{EL}$ family of description logics. He shows that answering unions of conjunctive queries is PTIME-complete for both $\mathcal{EL}$ and $\mathcal{ELH}$ (which adds role inclusions to $\mathcal{EL}$).

Building on previous work on DL-Lite$^+$ in (Pérez-Urbina, Motik, & Horrocks, 2008), Pérez-Urbina et al. (2010) present a resolution-based query rewriting algorithm for $\mathcal{ELHIO}^{\neg}$. The DL

$\mathcal{ELHIO}^{\neg}$ is obtained from $\mathcal{ELH}$ by allowing inverse roles ($\mathcal{I}$), concept assertions of the form $\{a\}$ for some constant a ($\mathcal{O}$), and negative inclusions ($\neg$). DL-Lite^{+} is a fragment of $\mathcal{ELH}$ in which concepts involving conjunction are disallowed, also known as *linear* $\mathcal{ELH}$ (which we will denote by $\mathcal{ELH}^{\ell in}$ from now on). Their algorithm rewrites a conjunctive query and an $\mathcal{ELHIO}^{\neg}$ TBox $\mathcal{T}$ into a Datalog program, also showing that conjunctive query answering for $\mathcal{ELHIO}^{\neg}$ is PTIME-complete in terms of data complexity. Furthermore, if $\mathcal{T}$ is in DL-Lite^{+}, then the perfect rewriting is a union of conjunctive queries and a linear Datalog query, as in (Pérez-Urbina et al., 2008), while if $\mathcal{T}$ is in DL-Lite$_{\mathcal{R}}$, then the perfect rewriting is a union of conjunctive queries, as in (Calvanese et al., 2007).

Pérez-Urbina et al. (2009) compare their resolution-based rewriting algorithm from (Pérez-Urbina et al., 2010) with that of Calvanese et al. (2007), where the DL under consideration is OWL 2 QL, which is based on DL-Lite$_{\mathcal{R}}$. The comparison is performed via an empirical evaluation using ontologies and queries derived from realistic applications. The results indicate that the resolution-based algorithm produces significantly smaller rewritings in most cases, an important consideration in practical applications. Improving the performance of such systems has continued to be an active area of research (Trivela, Stoilos, Chortaras, & Stamou, 2015).

Hansen, Lutz, Seylan, and Wolter (2014, 2015) propose an algorithm for computing FO-rewritings of concept instance queries (when they exist) over $\mathcal{ELH}^{dr}$ TBoxes, where $\mathcal{ELH}^{dr}$ extends $\mathcal{ELH}$ with domain and range restrictions on roles and underlies the OWL 2 EL profile. The algorithm outputs a succinct non-recursive Datalog program if the input (query and TBox) is FO-rewritable and otherwise reports non-FO-rewritability. Experiments show that the algorithm is efficient and widely applicable.

In other work, Bienvenu, Lutz, and Wolter (2013) and Bienvenu, Hansen, Lutz, and Wolter (2016) study FO-rewriting in the presence of ontologies formulated in a description logic that lies between $\mathcal{EL}$ and Horn-$\mathcal{SHIF}$ in expressiveness. The former paper considers only concept instance (atomic) queries, as in (Hansen et al., 2015), whereas the latter extends both of these to conjunctive queries. The latter paper characterises FO-rewritability in terms of the presence of certain (almost) tree-shaped ABoxes.

Recently, Lutz and Sabellek (2022) completely characterised the data complexity of answering conjunctive queries with respect to an $\mathcal{EL}$ ontology by providing a trichotomy into the classes AC^0, NLOGSPACE and PTIME. These classes correspond to rewritability into FO, linear Datalog and Datalog, respectively.

The tree-witness technique that we use in Section 5 of this paper is derived from (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014), which address query rewriting under the DLs underpinning the OWL 2 profiles, namely OWL 2 EL, QL and RL, and propose the tree-witness approach to rewrite conjunctive queries under QL.

In terms of query languages more powerful than conjunctive queries, (Bienvenu, Ortiz, & Simkus, 2015) have studied the complexity of answering C2RPQs under various DLs. In particular, they show (among other results) that the data complexity of answering C2RPQs under DL-Lite$_{\mathcal{R}}$ is NLOGSPACE-complete, the same as answering C2RPQs without an ontology, while that for $\mathcal{EL}(\mathcal{H})$ is PTIME-complete, the same as for conjunctive queries.

In this paper we build on the above work by proposing a language that is an extension of both OWL 2 QL and $\mathcal{ELH}^{\ell in}$. We propose a novel rewriting technique for CQs under our ontology language that makes use of non-deterministic finite-state automata. We show that CQ answering in our setting is NLOGSPACE-complete with respect to data complexity and NP-complete for com-

bined complexity; we also show that answering instance queries is NLOGSPACE-complete for data complexity and in PTIME for combined complexity.

3. Preliminaries

In this section we present the formal concepts that we will use in the rest of the paper: the $\mathcal{ELHI}^{\ell in}$ description logic, regular languages, and conjunctive regular path queries.

3.1 The $\mathcal{ELHI}^{\ell in}$ Description Logic

The $\mathcal{ELHI}^{\ell in}$ description logic that is the focus of this paper is derived from the $\mathcal{EL}$ language (which is the core of the OWL 2 EL profile), extended with the additional features of inverse roles ($\mathcal{I}$) and role inclusion axioms ($\mathcal{H}$), but disallowing conjunction of concepts on the left-hand side of concept inclusion axioms. It can also be seen as extending *QL* (Kontchakov & Zakharyaschev, 2014), a slight simplification of the OWL 2 QL profile, and therefore DL-Lite$_\mathcal{R}$ (Artale et al., 2009) to allow qualified existential quantification on the left-hand side (LHS) of concept inclusion axioms.

The syntax of $\mathcal{ELHI}^{\ell in}$ is as follows. The alphabet contains three pairwise disjoint and countably infinite sets of *concept names* A, *role names* R, and *individual names* I. The alphabet also contains a set of *roles* P, such that each $P \in \mathsf{P}$ is either a role name R or its *inverse*, denoted by R^-. A *complex concept* C is constructed from a special primitive concept $\top$ ('top'), concept names and role names using the following production rules:

$$\begin{array}{lcll} C & ::= & D & | \quad \exists P.C \\ D & ::= & A & | \quad \exists P.\top \end{array}$$

where $A \in \mathsf{A}$ and $P \in \mathsf{P}$. The non-terminal D generates a subset of all complex concepts and is used below. The sets of complex concepts generated by the non-terminals C and D are denoted by C and D, respectively. The alphabet includes two additional sets of *negated complex concepts* E and *negated roles* Q constructed using the following production rules:

$$\begin{array}{lcll} E & ::= & D & | \quad \neg D \\ Q & ::= & P & | \quad \neg P \\ P & ::= & R & | \quad R^- \end{array}$$

In $\mathcal{ELHI}^{\ell in}$, a TBox $\mathcal{T}$ is a finite set of *concept* and *role inclusion axioms* of the form

$$C \sqsubseteq E \qquad \text{and} \qquad P \sqsubseteq Q$$

where $C \in \mathsf{C}$, $E \in \mathsf{E}$ and $P, Q \in \mathsf{P}$.

We observe that negation can only appear on the right-hand side (RHS) of an inclusion axiom. Inclusion axioms in which there is no negation are called *positive inclusions* (PIs), while those in which negation does appear are called *negative inclusions* (NIs).

$\mathcal{ELHI}^{\ell in}$ could be extended to allow qualified existential quantification on the RHS of concept inclusion axioms. However, this can be simulated by making use of inclusions between roles as well as unqualified existential quantification of concepts in inclusions between concepts. For example, the axiom $A \sqsubseteq \exists R.B$ can be simulated by $A \sqsubseteq \exists R_1.\top$, $R_1 \sqsubseteq R$ and $\exists R_1^-.\top \sqsubseteq B$, where R_1 is a new role name. Therefore, in this paper, we do not explicitly consider qualified existential quantification on the RHS of axioms.

An ABox $\mathcal{A}$ is a finite set of *concept* and *role assertions* of the form $A(a)$ and $R(a,b)$, where $A \in \mathsf{A}$, $R \in \mathsf{R}$ and $a,b \in \mathsf{I}$. Given an ABox $\mathcal{A}$, we denote by $\mathsf{ind}(\mathcal{A})$ the set of individual names that occur in $\mathcal{A}$. Taken together, $\mathcal{T}$ and $\mathcal{A}$ comprise a *knowledge base* (or KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

NIs in $\mathcal{ELHI}^{\ell in}$ represent integrity constraints which the KB is expected to satisfy. Results in (Calì, Gottlob, & Lukasiewicz, 2012) show that the satisfiability of such constraints can be checked by evaluating a set of Boolean CQs whose size is linear in the number of NIs. Furthermore, the NIs do not affect the answers to CQs on the KB, and so can be ignored during query answering. Since NIs do not contribute to the query rewriting process and checking satisfiability does not increase the complexity of CQ answering under $\mathcal{ELHI}^{\ell in}$, we assume from now on that TBoxes do not contain any NIs.

In this paper we adopt the semantics of DLs defined in terms of interpretations (Baader & Nutt, 2007). An *interpretation* $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ that consists of a non-empty countable infinite *domain of interpretation* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ which assigns *(i)* an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to each individual name $a \in \mathsf{I}$, *(ii)* a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each concept name $A \in \mathsf{A}$ and *(iii)* a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each role name $R \in \mathsf{R}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended inductively to complex concepts with the following definitions:

$$\begin{aligned}(R^-)^{\mathcal{I}} &= \{(v,u) \mid (u,v) \in R^{\mathcal{I}}\}\\ (\exists P.\top)^{\mathcal{I}} &= \{u \mid \text{there is a } v \text{ such that } (u,v) \in P^{\mathcal{I}}\}\\ (\exists P.C)^{\mathcal{I}} &= \{u \mid \text{there is a } v \in C^{\mathcal{I}} \text{ such that } (u,v) \in P^{\mathcal{I}}\}\end{aligned}$$

The satisfaction relation $\models$ for inclusions and assertions (where $C \in \mathsf{C}$, $D \in \mathsf{D}$ and $P,Q \in \mathsf{P}$) is defined as follows:

$$\begin{aligned}\mathcal{I} \models C \sqsubseteq D \quad &\text{if and only if} \quad C^{\mathcal{I}} \subseteq D^{\mathcal{I}}\,,\\ \mathcal{I} \models P \sqsubseteq Q \quad &\text{if and only if} \quad P^{\mathcal{I}} \subseteq Q^{\mathcal{I}}\,,\\ \mathcal{I} \models C(a) \quad &\text{if and only if} \quad a^{\mathcal{I}} \in C^{\mathcal{I}}\,,\\ \mathcal{I} \models P(a,b) \quad &\text{if and only if} \quad (a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}\,.\end{aligned}$$

An interpretation $\mathcal{I}$ is a *model* of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, written $\mathcal{I} \models \mathcal{K}$, if it satisfies all concept and role inclusions of $\mathcal{T}$ and all concept and role assertions of $\mathcal{A}$. A knowledge base is *satisfiable* if admits at least one model.

We also need to define the notion of entailment of inclusions and assertions from a knowledge base. A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *entails* a concept inclusion $C \sqsubseteq E$, written $\mathcal{K} \models C \sqsubseteq E$, if $\mathcal{I} \models C \sqsubseteq E$ for each model $\mathcal{I}$ of $\mathcal{K}$. Analogous definitions apply for entailment of role inclusions, concept assertions and role assertions.

Now, we first convert $\mathcal{ELHI}^{\ell in}$ TBoxes to a normal form, extending (Baader, Brandt, & Lutz, 2005) to include inverse roles. We do this so as to reduce the complexity of the axiom syntax that we need to consider in developing our methods and proofs.

Definition 1. An $\mathcal{ELHI}^{\ell in}$ TBox is said to be in *normal form* if each of its concept inclusions is of one of the following four forms:

$$A_1 \sqsubseteq A_2, \quad \exists R.\top \sqsubseteq A, \quad \exists R.A_1 \sqsubseteq A_2, \quad A \sqsubseteq \exists R.\top,$$

and each of its role inclusions is of one of the following two forms:

$$R_1 \sqsubseteq R_2, \quad R_1 \sqsubseteq R_2^-,$$

where $A, A_1, A_2 \in \mathsf{A}$ and $R, R_1, R_2 \in \mathsf{R}$.

The normal form limits the use of inverse roles to the RHS of role inclusions, and excludes concept inclusion axioms whose LHS comprises complex concepts of the form $\exists P_1.\exists P_2.\ldots\exists P_n.\top$ or $\exists P_1.\exists P_2.\ldots\exists P_n.A$, where each P_i is either R_i or R_i^-, for $n > 1$. Thus, when a TBox is in normal form, its complex concepts are limited to being of the form A, $\exists R.\top$ and $\exists R.A$, for some concept name A and role name R.

Theorem 1. *Each $\mathcal{ELHI}^{\ell in}$ TBox $\mathcal{T}$ can be transformed in linear time into a TBox $\mathcal{T}'$ in normal form such that the size of $\mathcal{T}'$ is linear in the size of $\mathcal{T}$, and $\mathcal{T}'$ is a model conservative extension of $\mathcal{T}$.*

Proof. The claim follows by showing that each axiom that is not in normal form can be encoded by a set of normal form axioms of linear size. To remove inverse roles on the LHS of role inclusions, each role inclusion axiom of the form $R_1^- \sqsubseteq R_2$ can be replaced by the equivalent axiom $R_1 \sqsubseteq R_2^-$. To remove concept inclusion axioms whose LHS comprises complex concepts, two steps are needed:

1. Each concept inclusion axiom of the form $\exists P_1.\exists P_2.\ldots\exists P_n.\phi \sqsubseteq D$, where ϕ is either a concept name or $\top$ and D is the grammar non-terminal above, is encoded by the following n concept inclusion axioms:

$$\begin{aligned} \exists P_n.\phi &\sqsubseteq A_{n-1}, \\ \exists P_{n-1}.A_{n-1} &\sqsubseteq A_{n-2}, \\ &\ldots, \\ \exists P_1.A_1 &\sqsubseteq D, \end{aligned}$$

   where $A_1, A_2, \ldots, A_{n-1}$ are fresh concept names. For each $1 \leqslant i \leqslant n$, if P_i is R_i, the axiom is in normal form; if P_i is R_i^-, the following step is needed.

2. Each concept inclusion axiom that uses an inverse role R^- can be encoded by a modified concept inclusion axiom and a new role inclusion axiom. The modified concept inclusion axiom is obtained by replacing R^- with a fresh role name R_*, while the new role inclusion axiom is $R \sqsubseteq R_*^-$.

□

As is customary, henceforth we define and work with *canonical models* (also called *universal models* (Baader et al., 2017)) of KBs. We begin by defining the *base model* of a given ABox.

Definition 2 (Base model)**.** The *base model* $\mathcal{I}_\mathcal{A}$ of the ABox $\mathcal{A}$ is defined as follows:

(1) $\Delta^{\mathcal{I}_\mathcal{A}} = \mathsf{ind}(\mathcal{A})$;

(2) $a^{\mathcal{I}_\mathcal{A}} = a$, for $a \in \mathsf{ind}(\mathcal{A})$;

(3) $A^{\mathcal{I}_\mathcal{A}} = \{a \mid A(a) \in \mathcal{A}\}$, for each concept name A;

(4) $R^{\mathcal{I}_\mathcal{A}} = \{(a,b) \mid R(a,b) \in \mathcal{A}\}$, for each role name R.

We then use the base model to generate the canonical model of a KB.

Definition 3 (Canonical model)**.** To build the canonical model for an $\mathcal{ELHI}^{\ell in}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is in normal form, we take the base model $\mathcal{I}_\mathcal{A}$ as $\mathcal{I}_0$ and apply the following rules inductively to obtain $\mathcal{I}_{k+1}$ from $\mathcal{I}_k$:

(0) if $d \in \Delta^{\mathcal{I}_k}$ then d is added to $\Delta^{\mathcal{I}_{k+1}}$;

(i) if $d \in A^{\mathcal{I}_k}$ then d is added to $A^{\mathcal{I}_{k+1}}$;

(ii) if $(d,d') \in R^{\mathcal{I}_k}$ then (d,d') is added to $R^{\mathcal{I}_{k+1}}$;

(iii) if $d \in A_1^{\mathcal{I}_k}$ and $A_1 \sqsubseteq A_2 \in \mathcal{T}$, then d is added to $A_2^{\mathcal{I}_{k+1}}$;

(iv) if $(d,d') \in R_1^{\mathcal{I}_k}$ and $R_1 \sqsubseteq R_2 \in \mathcal{T}$, then (d,d') is added to $R_2^{\mathcal{I}_{k+1}}$;

(v) if $(d,d') \in R_1^{\mathcal{I}_k}$ and $R_1 \sqsubseteq R_2^- \in \mathcal{T}$, then (d',d) is added to $R_2^{\mathcal{I}_{k+1}}$;

(vi) if $(d,d') \in R^{\mathcal{I}_k}$ and either (a) $\exists R.\top \sqsubseteq A \in \mathcal{T}$, or (b) $d' \in D^{\mathcal{I}_k}$ and $\exists R.D \sqsubseteq A \in \mathcal{T}$, then d is added to $A^{\mathcal{I}_{k+1}}$;

(vii) if $d \in A^{\mathcal{I}_k}$, $A \sqsubseteq \exists R.\top \in \mathcal{T}$ and there is no d'' such that $(d,d'') \in R^{\mathcal{I}_k}$, then (d,d') is added to $R^{\mathcal{I}_{k+1}}$ and d' is added to $\Delta^{\mathcal{I}_{k+1}}$, where d' is a *fresh* labelled null (i.e., a new domain element).

We then take a fixpoint interpretation, as $k \to \infty$. The resulting interpretation satisfies all the inclusions in $\mathcal{T}$ and all the assertions in $\mathcal{A}$ — i.e., it is a model for $\mathcal{K}$ — and is called the *canonical model* of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, denoted by $\mathcal{J}_\mathcal{K}$.

We note that the above procedure is also called the *chase procedure*, with the resulting model being called the *chase*.

In terms of notation, we will sometimes view the canonical model $\mathcal{J}_\mathcal{K}$ as a set of assertions, i.e., $A(d) \in \mathcal{J}_\mathcal{K}$ if and only if $d \in A^{\mathcal{J}_\mathcal{K}}$ and $R(d,d') \in \mathcal{J}_\mathcal{K}$ if and only if $(d,d') \in R^{\mathcal{J}_\mathcal{K}}$ for each concept name A and each role name R. Note that, in our setting, such assertions may contain labelled nulls.

3.2 Regular Languages and Conjunctive Regular Path Queries

A *non-deterministic finite-state automaton* (NFA) over a set of symbols Σ is a tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of *states*, $\delta \subseteq Q \times \Sigma \times Q$ is the *transition relation*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*. $L(M)$ denotes the language defined by an NFA M, and Σ^* denotes the set of all strings over symbols in Σ, including the *empty string* ε. A language that is recognised by a NFA is a *regular language* (Berry & Sethi, 1986).

Regular languages can also be denoted by regular expressions. A *regular expression* over Σ is either a symbol $a \in \Sigma$, or is formed by applying the operations of concatenation, alternation and Kleene closure to regular expressions as follows: $\alpha_1\alpha_2$ (*concatenation*), $\alpha_1|\alpha_2$ (*alternation*) and α^* (*Kleene closure*), where $\alpha, \alpha_1, \alpha_2$ are regular expressions. Given a regular expression α, the *language* of α is denoted by $L(\alpha)$.

To define the queries below, we assume that there are countably infinite sets of *variables* V, *individual names* I, *concept names* A and *role names* R. There is also a set of *roles* P, where each $P \in \mathsf{P}$ is either a role name R or its *inverse* R^-. A *term* t is an individual name in I or a variable in V. An *atom* is of the form $\alpha(t,t')$, where t, t' are terms, and α is an NFA or regular expression defining a regular language over the set of symbols $\mathsf{P} \cup \mathsf{A}$. A string $s \in (\mathsf{P} \cup \mathsf{A})^*$ is a *path*.

A *conjunctive two-way regular path query* (C2RPQ) $\mathbf{q}$ of arity n has the form $q(\vec{x}) \leftarrow \exists \vec{y}\, \gamma(\vec{x},\vec{y})$, where $\vec{x} = x_1,\ldots,x_n$ and $\vec{y} = y_1,\ldots,y_m$ are tuples of variables, and $\gamma(\vec{x},\vec{y})$ is a conjunction of atoms with variables from $\vec{x}$ and $\vec{y}$ (Bienvenu et al., 2015). Atom $q(\vec{x})$ is the *head* of $\mathbf{q}$, denoted by $head(\mathbf{q})$, and $\gamma(\vec{x},\vec{y})$ is the *body* of $\mathbf{q}$, denoted by $body(\mathbf{q})$. The variables in $\vec{x}$ are the *answer variables* of $\mathbf{q}$, while those in $\vec{y}$ are the *existentially quantified variables* of $\mathbf{q}$. Subsequently, we usually omit the existential quantifier, and represent conjunction using commas. A *union of conjunctive two-way regular path queries* (UC2RPQ) $\mathbf{q}$ of arity n is a union (or set) of C2RPQs each of which has the same head $q(\vec{x})$.

A *conjunctive (one-way) regular path query* (CRPQ) is a C2RPQ in which only symbols from $\mathsf{R} \cup \mathsf{A}$ are allowed in atoms (i.e., disallowing role inverses). A *Boolean C(2)RPQ* is a C(2)RPQ with no answer variables. A *two-way regular path query* (2RPQ) is a C2RPQ with a single atom in its body. A *regular path query* (RPQ) is a CRPQ with a single atom in its body. A *two-way path query* (2PQ) is a 2RPQ $head(\mathbf{q}) \leftarrow \alpha(x,y)$ such that $\alpha \in (\mathsf{P} \cup \mathsf{A})^*$. A *path query* (PQ) is an RPQ $head(\mathbf{q}) \leftarrow \alpha(x,y)$ such that $\alpha \in (\mathsf{R} \cup \mathsf{A})^*$. In both the latter cases, α is called the *path* of $\mathbf{q}$, denoted by $\mathsf{path}(\mathbf{q})$.

A *conjunctive query* (CQ) $\mathbf{q}$ is a CRPQ such that, for each atom $\alpha(t,t') \in body(\mathbf{q})$, $\alpha \in (\mathsf{P} \cup \mathsf{A})$. Intuitively, a CQ has as body a conjunction of atoms whose predicates are in $\mathsf{A} \cup \mathsf{P}$ (without regular expressions). Given a C(2)RPQ $\mathbf{q}$ with answer variables $\vec{x} = x_1,\ldots,x_n$ and an n-tuple of individuals $\boldsymbol{a} = (a_1,\ldots,a_n)$, we use $\mathbf{q}(\boldsymbol{a})$ to refer to the Boolean C(2)RPQ obtained from $\mathbf{q}$ by replacing x_i with a_i in $body(\mathbf{q})$, for every $1 \leqslant i \leqslant n$. An *instance query* (IQ) takes one of the following two forms: *(i)* $q(x) \leftarrow A(x)$, where $A \in \mathsf{A}$ (*concept instance query*), or *(ii)* $q(x,y) \leftarrow P(x,y)$, where $P \in \mathsf{P}$ (*role instance query*).

3.2.1 SEMANTICS OF C2RPQS

We now define the semantics of C2RPQs (Bienvenu et al., 2015) and UC2RPQs. Given individual names a and b, an interpretation $\mathcal{I}$, and an NFA or regular expression α over the alphabet $\mathsf{P} \cup \mathsf{A}$, we say that b *α-follows* a in $\mathcal{I}$, denoted by $\mathcal{I} \models a \xrightarrow{\alpha} b$, if and only if there is some $w = u_1 \cdots u_n \in L(\alpha)$ and some sequence $e_0,\ldots,e_n$ with $e_i \in \Delta^{\mathcal{I}}$, $0 \leqslant i \leqslant n$, such that $e_0 = a^{\mathcal{I}}$ and $e_n = b^{\mathcal{I}}$, and for all $1 \leqslant i \leqslant n$: *(i)* if $u_i = A \in \mathsf{A}$, then $e_{i-1} = e_i \in A^{\mathcal{I}}$, and *(ii)* if $u_i = P \in \mathsf{P}$, then $(e_{i-1},e_i) \in P^{\mathcal{I}}$. A *match* for a C2RPQ $\mathbf{q}$ in an interpretation $\mathcal{I}$ is a mapping π from the terms in $body(\mathbf{q})$ to the elements in $\Delta^{\mathcal{I}}$ such that:

1. $\pi(c) = c^{\mathcal{I}}$ if $c \in \mathsf{I}$;
2. $\mathcal{I} \models \pi(t) \xrightarrow{\alpha} \pi(t')$ for each atom $\alpha(t,t')$ in $\mathbf{q}$.

This definition is easily extended to UC2RPQs by saying that there is a *match* for a UC2RPQ $\mathbf{q}$ in an interpretation $\mathcal{I}$ if there is a match for some C2RPQ in $\mathbf{q}$ in $\mathcal{I}$. When the query $\mathbf{q}$ is a CQ and the interpretation $\mathcal{I}$ is the canonical model, we will also refer to the mapping π as a *homomorphism*.

To simplify notation, we often view all atoms in the body of the query as being binary, with each atom of the form $A(t)$, where $A \in \mathsf{A}$ and $t \in \mathsf{V} \cup \mathsf{I}$, being viewed as a binary atom $A(t,z)$, where z is a fresh variable (that is, newly invented and not appearing elsewhere). This transformation is reversible since the sets of concept names and of role names are disjoint. The semantics of a transformed query remains that of the original. However, we will continue to use unary atoms in some examples, whenever this improves readability.

Given a KB $\mathcal{K}$, an interpretation $\mathcal{I}$ and a (U)C2RPQ $\mathbf{q}$, we say that $\mathcal{I} \models \mathbf{q}$ if there is a match for $\mathbf{q}$ in $\mathcal{I}$, and that $\mathcal{K} \models \mathbf{q}$ if $\mathcal{I} \models \mathbf{q}$ for every model $\mathcal{I}$ of $\mathcal{K}$. Also, we use $\mathbf{q}^{\mathcal{I}}$ to denote:

$$\mathbf{q}^{\mathcal{I}} := \{t \mid \mathcal{I} \models \mathbf{q}(t)\}$$

and $\mathbf{q}^{\mathcal{K}}$ to denote:

$$\mathbf{q}^{\mathcal{K}} := \{t \mid t \in \mathbf{q}^{\mathcal{I}} \text{ for every model } \mathcal{I} \text{ of } \mathcal{K}\}.$$

Given an ABox $\mathcal{A}$, we use $\mathcal{A} \models \mathbf{q}$ as a shorthand for $(\varnothing, \mathcal{A}) \models \mathbf{q}$, where $(\varnothing, \mathcal{A})$ is a KB with an empty TBox.

Given a (U)C2RPQ $\mathbf{q}$ of arity n, a tuple of individual names $\boldsymbol{a} = (a_1, \ldots, a_n)$ is a *certain answer* for $\mathbf{q}$ with respect to a KB $\mathcal{K}$ if and only if $\mathcal{K} \models \mathbf{q}(\boldsymbol{a})$.

3.3 Perfect Rewritings of CQs

Given a TBox $\mathcal{T}$, in this paper we are concerned with rewriting CQs into UC2RPQs. Given a CQ $\mathbf{q}$, we use the axioms of $\mathcal{T}$ to rewrite $\mathbf{q}$ into a C2RPQ $\mathbf{p}$ that returns, when evaluated over the data instance (ABox) $\mathcal{A}$, all the certain answers of $\mathbf{q}$ with respect to $(\mathcal{T}, \mathcal{A})$. The rewriting $\mathbf{p}$ depends only on the TBox $\mathcal{T}$ and the given query $\mathbf{q}$; it is independent of the ABox $\mathcal{A}$. In query processing, therefore, we use $\mathcal{A}$ only in the final step, when the rewriting is evaluated on it.

We call a CQ $\mathbf{q}$ and a TBox $\mathcal{T}$ *UC2RPQ-rewritable* if there exists a UC2RPQ $\mathbf{p}$ such that, for any ABox $\mathcal{A}$ and any tuple $\mathbf{a}$ of individuals in $\mathsf{ind}(\mathcal{A})$, we have

$$(\mathcal{T}, \mathcal{A}) \models \mathbf{q}(\mathbf{a}) \text{ if and only if } \mathcal{A} \models \mathbf{p}(\mathbf{a}).$$

In this case, we say that $\mathbf{p}$ is a *perfect UC2RPQ rewriting* of $\mathbf{q}$ with respect to $\mathcal{T}$.

4. Harmless $\mathcal{ELHI}^{\ell in}$

Extending DL-Lite$_{\mathcal{R}}$ with qualified existential quantification on the left-hand side of concept inclusion axioms is equivalent to allowing inverse roles in role inclusion axioms in $\mathcal{ELH}^{\ell in}$ (resulting in $\mathcal{ELHI}^{\ell in}$, defined in Section 3). This is shown in (Calvanese et al., 2013) to result in PTIME-completeness of CQ answering with respect to data complexity; therefore a rewriting in C2RPQs for this language is not feasible — if, as is normally assumed, NLOGSPACE is a proper subclass of PTIME — since the data complexity of answering C2RPQs is in NLOGSPACE. In fact, inverse roles allow the encoding of a conjunction of concepts on the left-hand side of axioms (as shown in the example below), which is known to lead to PTIME-hardness ((Calvanese et al., 2013), Theorem 4.3).

Example 5. Consider the KB $\mathcal{K}$ with ABox $\{Teacher(alice), Professor(alice)\}$ and TBox comprising the axioms:

$$\begin{aligned} Teacher &\sqsubseteq \exists teaches.\top \\ teaches &\sqsubseteq taughtBy^- \\ \exists taughtBy.Professor &\sqsubseteq Course \\ \exists teaches.Course &\sqsubseteq Person \end{aligned}$$

Using the chase procedure from Definition 3 to produce the canonical model $\mathcal{J}_{\mathcal{K}}$, the axiom $Teacher \sqsubseteq \exists teaches.\top$ results in $teaches(alice, d_0)$, where d_0 is a fresh labelled null, being added to

$\mathcal{J}_\mathcal{K}$ (using rule (vii)). Then $teaches \sqsubseteq taughtBy^-$ results in $taughtBy(d_0, alice)$ being added to $\mathcal{J}_\mathcal{K}$ (rule (v)), $\exists taughtBy.Professor \sqsubseteq Course$ results in $Course(d_0)$ being added to $\mathcal{J}_\mathcal{K}$ (rule (vi)), and $\exists teaches.Course \sqsubseteq Person$ results in $Person(alice)$ being added to $\mathcal{J}_\mathcal{K}$ (rule (vi)). Thus, the TBox encodes the axiom $Teacher \sqcap Professor \sqsubseteq Person$.

In this paper, we investigate the possibility of finding a sub-language of $\mathcal{ELHI}^{\ell in}$ whose CQ answering problem has NLOGSPACE data complexity. We do so by first identifying a syntactic property of $\mathcal{ELHI}^{\ell in}$ TBoxes, which we call the *harmless* property, that prevents the above encoding of rules of the type $C_1 \sqcap C_2 \sqsubseteq C_3$. The harmless property also ensures that each instance query (IQ) can be rewritten as a 2RPQ, as we show in Section 5. Building on this and using results from (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014), we show in Section 6 that CQs over harmless $\mathcal{ELHI}^{\ell in}$ TBoxes can be rewritten as UC2RPQs, thus avoiding a polynomial blow-up.

Before defining harmless $\mathcal{ELHI}^{\ell in}$ TBoxes, denoted by $\mathcal{ELHI}^{\ell in}_h$, we introduce some auxiliary definitions.

Definition 4. Let R and R' be role names appearing in an $\mathcal{ELHI}^{\ell in}$ TBox $\mathcal{T}$ that is in normal form. If there exist role names $R_0, \ldots, R_n$ in $\mathcal{T}$ such that *(i)* $R = R_0$, $R' = R_n$ and *(ii)* either $R = R'$ or, for $1 \leqslant i \leqslant n$, either $R_{i-1} \sqsubseteq R_i \in \mathcal{T}$ or $R_{i-1} \sqsubseteq R_i^- \in \mathcal{T}$, then:

1. if the number of inverse roles R_i^- is even, we write $R \rightharpoonup_\mathcal{T} R'$;
2. if the number is odd, we write $R \rightharpoonup_\mathcal{T} R'^-$.

The syntactic property defined above is equivalent to the semantic property of role inclusion with respect to an $\mathcal{ELHI}^{\ell in}$ TBox. This is stated in the following proposition.

Proposition 1. *Given two role names R, R' appearing in an $\mathcal{ELHI}^{\ell in}$ TBox $\mathcal{T}$ in normal form, we have:*

1. *$\mathcal{T} \models R \sqsubseteq R'$ if and only if $R \rightharpoonup_\mathcal{T} R'$, and*
2. *$\mathcal{T} \models R \sqsubseteq R'^-$ if and only if $R \rightharpoonup_\mathcal{T} R'^-$.*

The proof of the above proposition follows from the observation that the only way to infer a role inclusion in an $\mathcal{ELHI}^{\ell in}$ TBox in normal form is through the closure of role inclusion axioms of the form $R_1 \sqsubseteq R_2$ or $R_1 \sqsubseteq R_2^-$.

We now define the *harmless* condition for two given role names appearing in an $\mathcal{ELHI}^{\ell in}$ TBox in normal form:

Definition 5. Let R_1 and R_2 be two, not necessarily distinct, role names appearing in an $\mathcal{ELHI}^{\ell in}$ TBox $\mathcal{T}$ in normal form. We say that R_1 and R_2 are *mutually harmless* roles with respect to $\mathcal{T}$ if there is no role name R_3 in $\mathcal{T}$ (which may be equal to R_1 or R_2) such that $R_3 \rightharpoonup_\mathcal{T} R_1$ and $R_3 \rightharpoonup_\mathcal{T} R_2^-$.

Example 6. First note that *teaches* and *taughtBy* in Example 5 are not mutually harmless because $teaches \sqsubseteq taughtBy^-$ is in $\mathcal{T}$. Now consider a modification to the KB $\mathcal{K}$ in Example 5, whereby we (1) add the axiom $teaches \sqsubseteq contributesTo$, and (2) replace *teaches* by *contributesTo* in the fourth

axiom, giving a TBox comprising the following axioms:

$$\begin{aligned} \mathit{Teacher} &\sqsubseteq \exists \mathit{teaches}.\top \\ \mathit{teaches} &\sqsubseteq \mathit{taughtBy}^- \\ \exists \mathit{taughtBy}.\mathit{Professor} &\sqsubseteq \mathit{Course} \\ \mathit{teaches} &\sqsubseteq \mathit{contributesTo} \\ \exists \mathit{contributesTo}.\mathit{Course} &\sqsubseteq \mathit{Person} \end{aligned}$$

Notice that the role names *taughtBy* and *contributesTo* are not mutually harmless, since $\mathit{teaches} \sqsubseteq \mathit{taughtBy}^-$ and $\mathit{teaches} \sqsubseteq \mathit{contributesTo}$ are axioms in the TBox. With the ABox from Example 5, *alice* would still be added to the interpretation of *Person*, i.e., the TBox would encode the axiom $\mathit{Teacher} \sqcap \mathit{Professor} \sqsubseteq \mathit{Person}$. This would continue to be the case even if, instead of the axiom $\exists \mathit{contributesTo}.\mathit{Course} \sqsubseteq \mathit{Person}$, we had the axioms $\mathit{Course} \sqsubseteq \mathit{LearningActivity}$ and $\exists \mathit{contributesTo}.\mathit{LearningActivity} \sqsubseteq \mathit{Person}$.

In order to prevent the encoding of conjunction on the LHS of axioms as demonstrated in the previous example, we need also to determine subsumption between concepts in an $\mathcal{ELHI}^{\ell in}$ TBox $\mathcal{T}$. We use the *digraph representation of* $\mathcal{T}$ introduced by Lembo, Santarelli, and Savo (2013). Note that their method applies to OWL 2 QL TBoxes, which include additional constructs such as attributes and value domains, but crucially do not allow qualified existential quantification on the LHS of axioms. We limit the graph construction below to constructs in $\mathcal{ELHI}^{\ell in}$, excluding qualified existential quantification.

Definition 6. Let $\mathcal{T}$ be an $\mathcal{ELHI}^{\ell in}$ TBox in normal form. We construct a directed graph $G_\mathcal{T} = (N_\mathcal{T}, E_\mathcal{T})$, where $N_\mathcal{T}$ is a set of nodes and $E_\mathcal{T} \subseteq N_\mathcal{T} \times N_\mathcal{T}$ is a set of edges, as follows. For each concept name A in $\mathcal{T}$, there is a node $A \in N_\mathcal{T}$. For each role name R in $\mathcal{T}$, there are nodes $\exists R$ and $\exists R^-$ in $N_\mathcal{T}$. The edges $E_\mathcal{T}$ are defined as follows:

1. If $A \sqsubseteq B \in \mathcal{T}$, for concept names A and B, then $(A, B) \in E_\mathcal{T}$.
2. If $A \sqsubseteq \exists R.\top \in \mathcal{T}$, for concept name A and role name R, then $(A, \exists R) \in E_\mathcal{T}$.
3. If $R \sqsubseteq S \in \mathcal{T}$, for role names R and S, then $(\exists R, \exists S) \in E_\mathcal{T}$ and $(\exists R^-, \exists S^-) \in E_\mathcal{T}$.
4. If $R \sqsubseteq S^- \in \mathcal{T}$, for role names R and S, then $(\exists R, \exists S^-) \in E_\mathcal{T}$ and $(\exists R^-, \exists S) \in E_\mathcal{T}$.
5. If $\exists R.\top \sqsubseteq A \in \mathcal{T}$, for role name R and concept name A, then $(\exists R, A) \in E_\mathcal{T}$.

We write $A \Rightarrow_\mathcal{T} B$ if there is a path from A to B in $G_\mathcal{T}$.

Example 7. If we replaced *Professor* and *Course* by $\top$ in the two axioms using qualified existential quantification in the previous example, then $G_\mathcal{T}$ would contain a path from *Teacher* to *Person* via $\exists \mathit{teaches}$ and $\exists \mathit{contributesTo}$, i.e., $\mathit{Teacher} \Rightarrow_\mathcal{T} \mathit{Person}$. As we prove shortly, this implies that $\mathcal{T} \models \mathit{Teacher} \sqsubseteq \mathit{Person}$.

We are now ready to define the class of *harmless* $\mathcal{ELHI}^{\ell in}$ TBoxes:

Definition 7. Let $\mathcal{T}$ be an $\mathcal{ELHI}^{\ell in}$ TBox in normal form, with $A_1, A_2, A_3 \in \mathsf{A}$, and $R_1, R_2 \in \mathsf{R}$. We say that $\mathcal{T}$ is *harmless* if, whenever $\exists R_2.A_2$ appears on the left-hand side of an axiom in $\mathcal{T}$ as well as there being either an axiom $\exists R_1.\top \sqsubseteq A_1$ or an axiom $\exists R_1.A_3 \sqsubseteq A_1$ in $\mathcal{T}$ such that $A_1 \Rightarrow_{\mathcal{T}} A_2$, then we have that R_1 and R_2 are mutually harmless roles with respect to $\mathcal{T}$. The language of all harmless $\mathcal{ELHI}^{\ell in}$ TBoxes is denoted by $\mathcal{ELHI}_h^{\ell in}$.

Example 8. The TBox in Example 5 is not harmless because it contains the axioms $\exists taughtBy.Professor \sqsubseteq Course$ and $\exists teaches.Course \sqsubseteq Person$ (where *Course* plays the role of both A_1 and A_2 from the above definition), where *teaches* and *taughtBy* are not mutually harmless.

The TBox in Example 6 (modified to include $Course \sqsubseteq LearningActivity$ and $\exists contributesTo.LearningActivity \sqsubseteq Person$ instead of $\exists contributesTo.Course \sqsubseteq Person$) is not harmless since it contains the axioms $\exists taughtBy.Professor \sqsubseteq Course$, $\exists contributesTo.LearningActivity \sqsubseteq Person$ and $Course \sqsubseteq LearningActivity$ (so $Course \Rightarrow_{\mathcal{T}} LearningActivity$), and *taughtBy* and *contributesTo* are not mutually harmless.

Removing the axiom $teaches \sqsubseteq taughtBy^-$ from the TBoxes of Examples 5 and 6 would make each of them harmless.

For an $\mathcal{ELHI}^{\ell in}$ TBox $\mathcal{T}$ which is not harmless, it is possible that an axiom such as $A \sqsubseteq B$ can be inferred from $\mathcal{T}$ without it being the case that $A \Rightarrow_{\mathcal{T}} B$ (i.e., by relying on axioms which use qualified existential quantification). We prove below that, for harmless $\mathcal{ELHI}^{\ell in}$, the only way we can have $\mathcal{T} \models A \sqsubseteq B$ is if $A \Rightarrow_{\mathcal{T}} B$.

Proposition 2. *If $\mathcal{T}$ is an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form, with $A, B \in \mathsf{A}$, then $\mathcal{T} \models A \sqsubseteq B$ if and only if $A \Rightarrow_{\mathcal{T}} B$.*

Proof. The "if" direction follows from Theorem 2 in (Lembo et al., 2013). The "only if" direction also follows from (Lembo et al., 2013), except in a case where qualified existential quantification is used. Consider a set of sound and complete inference (or classification) rules for $\mathcal{ELHI}$, such as those given in (Baader et al., 2005) and extended in (Kazakov, 2009) or (Vu, 2008). There is only one rule, given as rule (1) below, in the set given in (Kazakov, 2009) which involves qualified existential quantification. Let $\mathcal{T}$ be a harmless $\mathcal{ELHI}^{\ell in}$ TBox and assume that the first use of rule (1) derives the axiom $A \sqsubseteq B$, where rule (1) is as follows:

$$\frac{A \sqsubseteq \exists R.C \quad C \sqsubseteq D \quad \exists R.D \sqsubseteq B}{A \sqsubseteq B} \tag{1}$$

for some role name R and some concept names C and D. Each of $A \sqsubseteq \exists R.C$ and $C \sqsubseteq D$ might be derived from $\mathcal{T}$, but $\exists R.D \sqsubseteq B \in \mathcal{T}$, since no axioms with qualified existential quantification on the LHS can be derived using the rules in (Kazakov, 2009). The axiom $A \sqsubseteq \exists R.C$ might have been derived from one such as $E \sqsubseteq \exists S.C$, for concept name E and role name S, in $\mathcal{T}$, by repeated application of rules using the facts that $\mathcal{T} \models E \sqsubseteq A$ and $\mathcal{T} \models S \sqsubseteq R$. So, by Proposition 1, we have $S \rightharpoonup_{\mathcal{T}} R$. Because it is the first invocation of rule (1), both $E \sqsubseteq A$ and $C \sqsubseteq D$ must have been derived without using axioms involving qualified existential quantification; therefore we have $E \Rightarrow_{\mathcal{T}} A$ and $C \Rightarrow_{\mathcal{T}} D$.

When converting $E \sqsubseteq \exists S.C$ into our normal form, we get the axioms

$$\begin{aligned} E &\sqsubseteq \exists U.\top \\ U &\sqsubseteq S \\ U &\sqsubseteq V^- \\ \exists V.\top &\sqsubseteq C \end{aligned}$$

for new role names U and V. However, $\mathcal{T}$ is not harmless since we have $\exists R.D$ on the LHS of an axiom, $\exists V.\top \sqsubseteq C$, $C \Rightarrow_{\mathcal{T}} D$, and R and V not mutually harmless (since $U \rightharpoonup_{\mathcal{T}} R$ and $U \rightharpoonup_{\mathcal{T}} V^-$), a contradiction. We conclude that an inclusion such as $A \sqsubseteq B$ cannot be inferred using axioms involving qualified existential quantification, and hence $\mathcal{T} \models A \sqsubseteq B$ if and only if $A \Rightarrow_{\mathcal{T}} B$. □

We note that each DL-Lite$_{\mathcal{R}}$ KB is also an $\mathcal{ELHI}_h^{\ell in}$ KB, since complex concepts of the form $\exists R.D$ are forbidden on the LHS of DL-Lite$_{\mathcal{R}}$ TBoxes. Also, each $\mathcal{ELH}^{\ell in}$ KB is an $\mathcal{ELHI}_h^{\ell in}$ KB, since inverse roles are not included in $\mathcal{ELH}^{\ell in}$ and therefore roles are always harmless. Thus, $\mathcal{ELHI}_h^{\ell in}$ is a generalisation of both DL-Lite$_{\mathcal{R}}$ and $\mathcal{ELH}^{\ell in}$.

In the next section, we show that an $\mathcal{ELHI}_h^{\ell in}$ TBox cannot simulate a conjunction of concepts on the left-hand side of a concept inclusion axiom. We also show that, given an $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$, each IQ over $\mathcal{T}$ can be rewritten into a 2RPQ.

5. Rewriting Instance Queries into 2RPQs under $\mathcal{ELHI}_h^{\ell in}$

As shown in Example 3, it is not possible to generate a first-order query as a perfect rewriting if we allow qualified existential quantification on the left-hand side of concept inclusion axioms, even in the case where the input query is an instance query (IQ). In this section, we present a technique that uses the expressive power of NFAs in order to rewrite IQs into 2RPQs under $\mathcal{ELHI}_h^{\ell in}$ TBoxes.

We first describe in Section 5.1 a query rewriting procedure modified from that in (Calvanese et al., 2007). The original procedure was designed to rewrite a given CQ into a union of CQs with respect to a DL-Lite$_{\mathcal{R}}$ TBox. When modified to take account of $\mathcal{ELHI}_h^{\ell in}$ TBoxes, the procedure may not terminate. To address this issue, we propose in Section 5.2 a novel algorithm, which makes use of NFAs, that is able to rewrite IQs into 2RPQs under $\mathcal{ELHI}_h^{\ell in}$ TBoxes by encoding the possibly infinitely many steps of the above rewriting procedure. In Section 6, we extend the treatment to show that CQs can be rewritten into UC2RPQs.

5.1 Rewriting Instance Queries to CQs for $\mathcal{ELHI}_h^{\ell in}$

The procedure we present in this subsection takes as input an IQ and an $\mathcal{ELHI}_h^{\ell in}$ TBox, and produces as output a (possibly infinite) set of CQs. This set of CQs is then interpreted as a *union of conjunctive queries* which can be evaluated over the ABox. As stated above, the procedure is a modification of one from (Calvanese et al., 2007).

Recall that, in an IQ, the single atom in the body of the query is of the form $\alpha(t,t')$, where t,t' are terms (either variables from V or individual names from I), and $\alpha \in (\mathsf{P} \cup \mathsf{A})$, where P is the set of role names and their inverses, and A is the set of concept names.

We also recall some terminology relating to CQs from (Calvanese et al., 2007). A term of an atom in a CQ is said to be *bound* if it corresponds to (i) an answer variable, (ii) a *shared variable*, that is, a variable occurring at least twice in the query body, or (iii) a *constant*, that is, an element in

Procedure Rewrite($q, \mathcal{T}$)

input : Instance query q, TBox $\mathcal{T}$.
output: Set of conjunctive queries Q.

```
1  Q := {q};
2  repeat
3      Q' := Q ;
4      foreach qr ∈ Q' do
5          foreach axiom I ∈ T do
6              if I is applicable to atom g in qr then
7                  qr' := replace g in qr by gr(g, I) ;
8                  Q := Q ∪ {qr'}
9  until Q' = Q;
10 return Q
```

I. Conversely, a term of an atom in a query is *unbound* if it corresponds to a non-shared existentially quantified variable. As is customary, we adopt the symbol '_' to represent an unbound term[1].

Procedure `Rewrite` repeatedly applies rewriting rules, based on the axioms in the TBox given as input, to the atoms of CQs generated as a result of the process. So we need to define when an axiom I is applicable to an atom (used on line 6), as well as the result of applying the rewriting rule corresponding to I (used on line 7). These definitions are modified from those in (Calvanese et al., 2007).

Definition 8. An axiom I is *applicable to an atom* $A(x_1, x_2)$ for $A \in \mathsf{A}$ if the RHS of I is A. An axiom I is *applicable to an atom* $R(x_1, x_2)$ for $R \in \mathsf{R}$ if either (1) the RHS of I is $\exists R.\top$ and $x_2 = \_$; or (2) the RHS of I is either R or R^-.

The rewriting rules listed below are those of (Calvanese et al., 2007), except that we add rule (c) in order to deal with concept inclusion axioms where qualified existential quantification appears on the left-hand side (which are disallowed by DL-Lite$_\mathcal{R}$). Let I be an inclusion axiom that is applicable to an atom g. The set of atoms obtained from g by applying I, denoted by $gr(g, I)$, is defined as follows:

(a) If $g = A_2(x_1, \_)$ and $I = A_1 \sqsubseteq A_2$, then $gr(g, I) = \{A_1(x_1, \_)\}$;

(b) If $g = A(x_1, \_)$ and $I = \exists R.\top \sqsubseteq A$, then $gr(g, I) = \{R(x_1, \_)\}$;

(c) If $g = A_1(x_1, \_)$ and $I = \exists R.A_2 \sqsubseteq A_1$, then $gr(g, I) = \{R(x_1, z), A_2(z, \_)\}$, where z is a fresh variable;

(d) If $g = R(x_1, \_)$ and $I = A_1 \sqsubseteq \exists R.\top$ then $gr(g, I) = \{A_1(x_1, \_)\}$;

(e) If $g = R_2(x_1, x_2)$ and $I = R_1 \sqsubseteq R_2$, then $gr(g, I) = \{R_1(x_1, x_2)\}$;

(f) If $g = R_2(x_1, x_2)$ and $I = R_1 \sqsubseteq R_2^-$, then $gr(g, I) = \{R_1(x_2, x_1)\}$.

1. The underscore symbol '_' is commonly used in logic programming, where it is named "don't care". In the presence of multiple occurrences of "don't care" symbols in a formula, such symbols are to be considered as *distinct* existentially quantified variables.

We prove in Theorem 2 below that the output produced by Procedure `Rewrite`, which we denote by Rewrite$(q,\mathcal{T})$, generates the perfect rewriting of q with respect to $\mathcal{T}$. The original procedure (Calvanese et al., 2007) takes as input a CQ and includes a reduction step within the outer **foreach** loop (starting on line 4) in order to remove redundant atoms (an atom is *redundant* in query q if its removal from q results in a query equivalent to q). Our version of the procedure takes as input an IQ rather than a CQ. We also prove below (in Lemma 3) that each CQ output by the procedure is minimal (i.e., contains no redundant atoms); hence, no reduction step is necessary in our procedure.

Example 9. Consider the TBox in Example 5 but leaving out the axiom $teaches \sqsubseteq taughtBy^-$, that is the TBox becomes

$$\begin{aligned} Teacher &\sqsubseteq \exists teaches.\top \\ \exists taughtBy.Professor &\sqsubseteq Course \\ \exists teaches.Course &\sqsubseteq Person \end{aligned}$$

which is harmless.

Let **q** be the following concept IQ

$$q(x) \leftarrow Person(x,\_).$$

Using rule (c) of the Rewrite procedure, we can apply $\exists teaches.Course \sqsubseteq Person$ to **q** to get

$$q(x) \leftarrow teaches(x,z_1), Course(z_1,\_).$$

We can then apply $\exists taughtBy.Professor \sqsubseteq Course$ to the above query, yielding

$$q(x) \leftarrow teaches(x_1,z_1), taughtBy(z_1,z_2), Professor(z_2,\_).$$

The union of the above three CQs is a perfect rewriting for the given query. With the ABox

$$\begin{aligned} \{ \quad & Teacher(alice),\ taughtBy(CS101,bob),\ Professor(bob), \\ & teaches(carol,CS101),\ teaches(dave,CS201),\ Course(CS201)\ \} \end{aligned}$$

we correctly receive the answers *dave*, because he teaches a course ($CS201$), and *carol*, because she teaches something ($CS101$) which is taught by a professor (*bob*). We do not receive *alice* as an answer because, although she teaches something, we do not know that what she teaches is a course.

Note that the rewriting rule associated with axiom $Teacher \sqsubseteq \exists teaches.\top$ (i.e., rule (d)) cannot be applied at any stage, since the second argument of *teaches* (i.e., z_1) is bound whenever *teaches* appears in a query above. We will show in Lemma 4 below that, when the TBox is harmless, no *shared* variable (such as z_1 above) is mapped to a labelled null when a match is found for a CQ generated from an initial IQ; thus, requiring that the second argument of R be unbound in rule (d) is correct.

Example 10. Now consider the full TBox in Example 5, that is, including the axiom $teaches \sqsubseteq taughtBy^-$. Recall that this TBox is *not* harmless. Starting from the same concept IQ as in Example 9, we obtain the same rewritten queries as in that example plus the following, by applying $teaches \sqsubseteq taughtBy^-$ to the last rewritten query:

$$q(x_1) \leftarrow teaches(x_1,z_1), teaches(z_2,z_1), Professor(z_2,\_).$$

Note that the rewriting rule associated with axiom $Teacher \sqsubseteq \exists teaches.\top$ (i.e., rule (d)) still cannot be applied at any stage. With ABox $\{Teacher(alice), Professor(alice)\}$, none of the four queries returns any answers (because there are no *Person* or *teaches* assertions in the ABox), so the rewriting does not encode the axiom $Teacher \sqcap Professor \sqsubseteq Person$ which is logically implied by the TBox (cf. Example 5). As a result, the rewriting is not correct, but we require Procedure `Rewrite` to produce a correct rewriting only when it is given a TBox which *is* harmless.

As can be seen in the above examples, Procedure `Rewrite` produces CQs of a restricted form when given a concept IQ. We call this subclass of CQs *two-way simple path conjunctive queries*.

Definition 9. A *two-way simple path conjunctive query* (2SPCQ) is a CQ of one of the following forms:

1. $q(x) \leftarrow A(x, \_)$,
2. $q(x) \leftarrow \tau_{R_1}(x, y_1), \tau_{R_2}(y_1, y_2), \ldots, \tau_{R_n}(y_{n-1}, \_)$, or
3. $q(x) \leftarrow \tau_{R_1}(x, y_1), \tau_{R_2}(y_1, y_2), \ldots, \tau_{R_n}(y_{n-1}, y_n), A(y_n, \_)$,

where:

- $n \geqslant 1$;
- $A \in \mathsf{A}$ and $R_1, \ldots, R_n \in \mathsf{R}$.
- $\tau_R(x, y)$ is either $R(x, y)$ or $R(y, x)$;
- each variable (x, $\_$ and y_i, $1 \leqslant i \leqslant n$) is distinct.

A 2SPCQ $head(\mathbf{q}) \leftarrow Z_1(x_0, x_1), \ldots, Z_n(x_{n-1}, x_n)$ is equivalent to a 2RPQ of the form $head(\mathbf{q}) \leftarrow Y_1 \cdots Y_n(x_0, x_n)$, where (1) $Y_i = Z_i$ if $Z_i(x_{i-1}, x_i)$ or $Z_i(x, \_)$ appears in the 2SPCQ, or (2) $Y_i = Z_i^-$ if $Z_i(x_i, x_{i-1})$ appears in the 2SPCQ. We define $path(\mathbf{q})$ to be $Y_1 \cdots Y_n$. For example, if $\mathbf{q}$ is $q(x) \leftarrow P(x, y_1), T(y_2, y_1), B(y_2, \_)$, where P and T are role names and B is a concept name, then $path(\mathbf{q})$ is PT^-B, and if $\mathbf{q}$ is $q(x) \leftarrow P(x, y_1), T(\_, y_1)$, then $path(\mathbf{q})$ is PT^-. Throughout the paper we will use either the 2RPQ form or the CQ form of a 2SPCQ, whichever is more natural in the given context.

The following two lemmas show that, when Procedure `Rewrite` is given a concept IQ, each query in its output is a 2SPCQ, and when it is given a role IQ, each query in its output is a role IQ.

Lemma 1. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form, and $\mathbf{q}$ be a concept IQ. If $\mathbf{q}_{rew} \in$ Rewrite$(\mathbf{q}, \mathcal{T})$, then $\mathbf{q}_{rew}$ is a 2SPCQ.*

Proof. Let $\mathbf{q}$ be a concept IQ of the form $q(x) \leftarrow A(x, \_)$. The proof is by induction on the number of rewriting steps needed to produce $\mathbf{q}_{rew}$. We denote by $Q^{[i]}$ the set of the queries produced after the i-th iteration of the repeat loop (starting on line 2) in Procedure `Rewrite`.

BASE STEP. $Q^{[1]}$ contains $\mathbf{q}$, which is a 2SPCQ of form (1), along with the queries obtained by the first rewriting step. The only rewriting rules which are applicable to $\mathbf{q}$ are (a), (b) and (c). Rule (a) generates a 2SPCQ of form (1), (b) a 2SPCQ of form (2), and (c) a 2SPCQ of form (3).

INDUCTIVE STEP. We assume that each query in $Q^{[i]}$, for some $i \geqslant 1$, is a 2SPCQ, and consider a query $q \in Q^{[i+1]}$. Query q has been generated by applying some rewriting rule to a query $q' \in Q^{[i]}$. We know that q' is a 2SPCQ, so consider each of the rewriting rules (a)–(f) in turn. To apply (a), (b)

or (c), q' must be of 2SPCQ form (1) or (3). By applying (a), q is of the same form as q'. If (b) is applied, q is of form (2) in both cases. If (c) is applied, then q is of form (3) in both cases. For (d) to be applied, q' must be of form (2) because no arguments are unbound in role atoms in a 2SPCQ of form (3). Furthermore, the role atom to be replaced must be the rightmost in q', because that is the only atom whose second argument is unbound. Hence, q is of form (3) after applying (d). If (e) or (f) is applied to q', then q' must be of form (2) or (3). In each case, q is of the same form as q'. □

Lemma 2. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form, and $\mathbf{q}$ be a role IQ. If $\mathbf{q}_{rew} \in \mathsf{Rewrite}(\mathbf{q}, \mathcal{T})$, then $\mathbf{q}_{rew}$ is a role IQ.*

Proof. When Procedure `Rewrite` is given a role IQ, i.e., a query of the form $q(x,y) \leftarrow R(x,y)$ or $q(x,y) \leftarrow R(y,x)$, where R is a role name, only rewriting rules (e) and (f) apply. As a result, it should be clear that each query in the output of the procedure is also a role IQ. □

Lemma 3. *Given $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$ and IQ $\mathbf{q}$, each CQ in $\mathsf{Rewrite}(\mathbf{q}, \mathcal{T})$ is minimal.*

Proof. If $\mathbf{q}$ is a role IQ, minimality of each query in the output follows from Lemma 2.

So assume that $\mathbf{q}$ is a concept IQ. Hence, by Lemma 1, each query in the output is a 2SPCQ. Clearly $\mathbf{q}$ itself and any other query of 2SPCQ form (1) in the output is minimal. Therefore we need to show that any 2SPCQ of form (2) or (3) in the output is minimal. The atom $A(y_n, \_)$ in a 2SPCQ of form (3) cannot be redundant. In fact, since each atom in a 2SPCQ of form (3) is connected by a chain of variables to the output variable x, and by another chain to the variable y_n which appears in an atom whose predicate name is a concept name while all other predicate names in the query are role names, we conclude that no atom in a 2SPCQ of form (3) can be redundant.

Now assume that q is an 2SPCQ of form (2) in $\mathsf{Rewrite}(\mathbf{q}, \mathcal{T})$. If there are any redundant atoms in q, then they must include the last atom, and the last two atoms in q are either $R(y_{n-1}, y_n), R(\_, y_n)$ or $R(y_n, y_{n-1}), R(y_n, \_)$, for some role name R. Consider the first case (the second case is similar). Atom $R(\_, y_n)$ must have been produced by replacing $S(y_n, \_)$ for some role name S. This must mean that we have $S \rightharpoonup_{\mathcal{T}} R^-$, so R and S are not mutually harmless. Atom $S(y_n, \_)$ must have arisen through replacing $A(y_n, \_)$, for some concept name A such that $\exists S.\top \sqsubseteq A \in \mathcal{T}$. Atom $A(y_n, \_)$ must have arisen through some number of applications of rewriting rule (a), starting from $B(y_n, \_)$, for some concept name B. So we have that $A \Rightarrow_{\mathcal{T}} B$. The pair of atoms $R(y_{n-1}, y_n), B(y_n, \_)$ must have come about through applying rule (c) to some $C(y_{n-1}, \_)$, with $\exists R.B \sqsubseteq C \in \mathcal{T}$. But now we have $\exists S.\top \sqsubseteq A$ and $\exists R.B \sqsubseteq C$ in $\mathcal{T}$, with $A \Rightarrow_{\mathcal{T}} B$ and S and R not mutually harmless; hence, $\mathcal{T}$ is not harmless, a contradiction. We conclude that q is minimal. □

Before proving the following lemma and theorem, we introduce the notion of a chase graph for an assertion in a canonical model.

Definition 10. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB, where $\mathcal{T}$ is an $\mathcal{ELHI}^{\ell in}$ TBox (not necessarily harmless). Let $\mathcal{J}_\mathcal{K}$ be the canonical model of $\mathcal{K}$, $A(a)$ a concept assertion in $\mathcal{J}_\mathcal{K}$, and $R(b,c)$ a role assertion in $\mathcal{J}_\mathcal{K}$, where each of a, b and c is either an individual or a labelled null. A *chase graph* $G_\mathcal{K} = (V, E)$ *for* $A(a)$, respectively $R(b,c)$, is a directed graph showing how $A(a)$, respectively $R(b,c)$, can be generated from $\mathcal{A}$ and $\mathcal{T}$ using the chase rules of Definition 3. Let α be an axiom used in the chase, which is applied to either one or two assertions, say x and y, and generates an assertion z. Then there is a directed edge in E from each of the nodes representing x and y to the node representing

z. Nodes representing assertions from $\mathcal{A}$ used in the chase are source nodes in $G_\mathcal{K}$, while the node representing $A(a)$, respectively $R(b,c)$, in $G_\mathcal{K}$ is the single sink node, called the *root* of $G_\mathcal{K}$. We assume that $G_\mathcal{K}$ is minimal in the sense that there are no edges present that are unnecessary for the generation of $A(a)$, respectively $R(b,c)$. This also implies that $G_\mathcal{K}$ is acyclic. The *height* of $G_\mathcal{K}$ is the length of the longest path from any source node to the root.

Example 11. Consider the TBox $\mathcal{T}$ comprising the axioms

$$\begin{aligned} \mathit{Teacher} &\sqsubseteq \exists \mathit{teaches}.\top \\ \mathit{teaches} &\sqsubseteq \mathit{taughtBy}^- \\ \exists \mathit{taughtBy}.\top &\sqsubseteq \mathit{Course} \end{aligned}$$

This is a harmless simplification of the TBox in Example 5. Let *alice* be an individual and d be a labelled null. Assume that the ABox $\mathcal{A}$ contains the assertion $\mathit{Teacher}(\mathit{alice})$. The chase graph for $\mathit{Course}(d)$ would be a simple path, with $\mathit{Teacher}(\mathit{alice})$ as the single source node, followed by $\mathit{teaches}(\mathit{alice},d)$ (by applying $\mathit{Teacher} \sqsubseteq \exists \mathit{teaches}.\top$), followed by $\mathit{taughtBy}(d,\mathit{alice})$ (by applying $\mathit{teaches} \sqsubseteq \mathit{taughtBy}^-$), and followed by $\mathit{Course}(d)$ as the root (by applying $\exists \mathit{taughtBy}.\top \sqsubseteq \mathit{Course}$).

When considering the correctness of Procedure `Rewrite`, we can observe that the rewriting rules of Definition 8 are essentially mirror images of the chase rules in Definition 3, except that rewriting rule (d) requires that the second argument of the role atom be unbound (as pointed out in Example 9), whereas chase rule (vii) can generate a labelled null which can be shared. By "shared" we mean the following.

Definition 11. Let $G_\mathcal{K} = (V,E)$ be a chase graph for the assertion $A(a)$ represented by the root of $G_\mathcal{K}$. Let $u \in V$ represent an assertion generated by applying chase rule (vii) in which labelled null w is generated. If there is more than one path from u to the root of $G_\mathcal{K}$, we say that w is *shared*.

For KB $\mathcal{K} = (\mathcal{T},\mathcal{A})$, the following lemma shows that if $\mathcal{T}$ *is* harmless, then no shared labelled null appears in a chase for any concept assertion in $\mathcal{J}_\mathcal{K}$. (Note that the labelled null appearing in the chase graph in Example 11 is not shared.)

Lemma 4. *Let $\mathcal{K} = (\mathcal{T},\mathcal{A})$ be a KB, where $\mathcal{T}$ is an $\mathcal{ELHI}^{\ell in}$ TBox in normal form. If $\mathcal{T}$ is harmless, then no shared labelled null appears in a chase graph for any concept assertion in $\mathcal{J}_\mathcal{K}$.*

Proof. We prove the contrapositive. Let $G_\mathcal{K} = (V,E)$ be a chase graph for the assertion $A(a)$ in $\mathcal{J}_\mathcal{K}$. Assume that labelled null w is shared in $G_\mathcal{K}$. Chase rule (vii) must have been used to generate w. Let the assertion in which w first appears, which must be of the form $R(b,w)$ for some role name R and individual or labelled null b, be represented by node $u \in V$ (see Figure 1). Since w is shared, there must be (at least) two paths p_1 and p_2 from u to the root of $G_\mathcal{K}$. Paths p_1 and p_2 must also "converge" at some node because there is a single root node in $G_\mathcal{K}$. Let $z \in V$ be the node of smallest distance from u at which p_1 and p_2 meet. Node z must represent an assertion resulting from applying chase rule (vi)(b), since that is the only chase rule which applies to a pair of assertions (nodes). Assume that the axiom used when applying the rule was $\exists S.B \sqsubseteq C$. Hence, z represents either (i) the assertion $C(b)$, or (ii) the assertion $C(w)$, and z has predecessors x and y in $G_\mathcal{K}$ representing assertions (i) $S(b,w)$ and $B(w)$, or (ii) $S(w,b)$ and $B(b)$, respectively.

Next, we show that $\mathcal{T}$ is not harmless. We will consider only case (i) above; case (ii) can be proved similarly. Recall that $R(b,w)$ is represented by $u \in V$. Let $S(b,w)$ be represented by x and

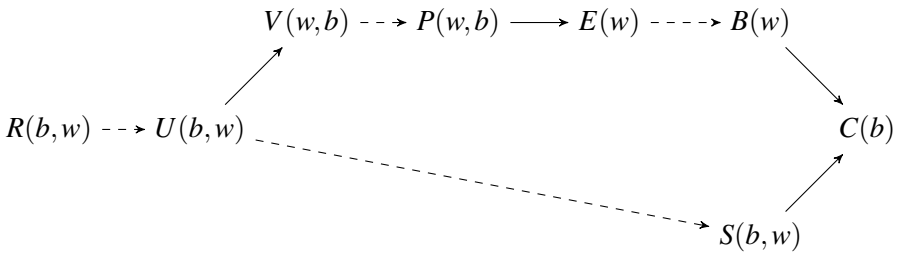

Figure 1: Fragment of the chase graph used in the proof of Lemma 4, where dashed arrows represent paths, possibly of length zero.

$B(w)$ be represented by y. Since there is a path from u to x, we have that $R \rightharpoonup_{\mathcal{T}} S$. Now consider the path from u to y. In order to generate $B(w)$ from $R(b,w)$, an axiom of the form $\exists P.\top \sqsubseteq E$ or $\exists P.F \sqsubseteq E$, for some role name P and concept names E and F, must have been applied during the chase, where it must be the case that $R \rightharpoonup_{\mathcal{T}} P^-$ and $E \Rightarrow_{\mathcal{T}} B$. The relevant fragment of the chase graph is shown in Figure 1. So we have $\exists S.B$ on the LHS of an axiom in $\mathcal{T}$, as well as either $\exists P.\top \sqsubseteq E$ or $\exists P.F \sqsubseteq E$ with $E \Rightarrow_{\mathcal{T}} B$, but S and P are not mutually harmless; hence $\mathcal{T}$ is not harmless. □

We are finally ready to prove the correctness of Procedure `Rewrite` in the following lemma and theorem. For the subsequent proofs and particularly those in Section 6, it is helpful to consider Procedure `Rewrite` being applied to an atom, rather than a query. This change is not significant since each IQ is a query containing a single atom in its body. The only difference is that all variables are now considered to be unbound. Given an assertion $A(a)$ or $R(b,c)$, we call $A(x)$ or $R(y,z)$, respectively, the atom *associated with* the assertion, where x, y and z are arbitrary variables.

Lemma 5. *Let $\mathcal{K} = (\mathcal{T},\mathcal{A})$ be a KB, where $\mathcal{T}$ is an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form. Let a, b and c be individuals or labelled nulls, A be a concept name and R be a role name. There is an assertion $A(a)$ or $R(b,c)$ in $\mathcal{J}_\mathcal{K}$ if and only if there is an atom $A(x)$ or $R(y,z)$, respectively, which Procedure* `Rewrite` *rewrites to a set of atoms which can be matched to assertions in $\mathcal{A}$, with x being mapped to a, or y and z being mapped to b and c, respectively.*

Proof. (If) Assume there is an atom $A(x)$ or $R(y,z)$, respectively, which Procedure `Rewrite` rewrites to a set S of atoms which can be matched using homomorphism π to assertions in $\mathcal{A}$, with $\pi(x) = a$ or $\pi(y) = b$ and $\pi(z) = c$, respectively. The proof proceeds by induction on the number of rule applications taken by Procedure `Rewrite` to produce S.

The base case of zero rule applications is obvious. So assume that for every concept atom $A(x)$ or role atom $R(y,z)$ requiring up to i rule applications to rewrite it to a set of atoms matching assertions in $\mathcal{A}$ using a homomorphism π, it is the case that $A(\pi(x)) \in \mathcal{J}_\mathcal{K}$ or $R(\pi(y),\pi(z)) \in \mathcal{J}_\mathcal{K}$. Now consider the case of a concept or role atom requiring $i+1$ applications.

Assume first that it is the concept atom $A(x)$ which requires $i+1$ rule applications. Let π be the homomorphism matching atoms in S to assertions in $\mathcal{A}$. The first rule applied must be (a), (b) or (c). Consider (a), and assume the axiom applied is $B \sqsubseteq A$, for some concept name B. So $A(x)$ is rewritten to $B(x)$ by the procedure. $B(x)$ requires i rule applications for its rewriting so, by the

inductive hypothesis, $B(\pi(x)) \in \mathcal{J}_\mathcal{K}$. Note that $\pi(x)$ could be a labelled null. Since the axiom $B \sqsubseteq A$ is in $\mathcal{T}$, clearly we have $A(\pi(x)) \in \mathcal{J}_\mathcal{K}$.

Case (b) is similar, except that $A(x)$ is rewritten to $R(x,y)$, for some role name R.

In case (c), $A(x)$ is rewritten to $R(x,y)$ and $B(y)$ according to axiom $\exists R.B \sqsubseteq A$, for some role name R and concept name B. Note that variable y is shared, so rule (d) cannot be applied to $R(x,y)$ subsequently. This means that y is retained in applications of rules applied to role atoms. Similarly, rules applied to concept atoms retain the variable, so y appears in all subsequent sets of atoms. Hence, π is defined on y and must be some individual in $\mathcal{A}$. Both $R(x,y)$ and $B(y)$ are rewritten using at most i rule applications so, by the inductive hypothesis, $R(\pi(x),\pi(y)) \in \mathcal{J}_\mathcal{K}$ and $B(\pi(y)) \in \mathcal{J}_\mathcal{K}$. Applying the axiom $\exists R.B \sqsubseteq A$ gives us that $A(\pi(x)) \in \mathcal{J}_\mathcal{K}$, as required.

Now assume that it is the role atom $R(y,z)$ which requires $i+1$ rule applications. Note that z is unbound so that rule (d) can be applied. In this case, there is an axiom $A \sqsubseteq \exists R.\top$ in $\mathcal{T}$ and $R(y,z)$ is rewritten to $A(y)$, and π is not defined on z. By the inductive hypothesis, $A(\pi(y)) \in \mathcal{J}_\mathcal{K}$. The axiom $A \sqsubseteq \exists R.\top$ ensures that $R(\pi(y),c) \in \mathcal{J}_\mathcal{K}$, where c is a labelled null. The other cases of rules (e) and (f) are straightforward.

(Only if) Assume there is an assertion $A(a)$ or $R(b,c)$ in $\mathcal{J}_\mathcal{K}$. The proof proceeds by induction on the height of the chase graph $G_\mathcal{K}$ of smallest height for $A(a)$ or $R(b,c)$, respectively. If $G_\mathcal{K}$ is of height zero, then $A(a)$ or $R(b,c)$ is in $\mathcal{A}$, and clearly $A(x)$ or $R(y,z)$, respectively, will match the assertion.

Assume that for all chase graphs of height at most i for assertions $A(a)$ or $R(b,c)$, there is a rewriting of $A(x)$ or $R(y,z)$, respectively, into a set S of atoms and a homomorphism π which maps each atom in S to an assertion in $\mathcal{A}$. Now let $A(a)$ be an assertion whose smallest chase graph $G_\mathcal{K}$ is of height $i+1$. Assertion $A(a)$ must have been produced by applying rule (iii) or rule (vi). If rule (iii) was used by applying the axiom $B \sqsubseteq A$, for some concept name B, then there must be a chase graph of smallest height i for the assertion $B(a)$. By the inductive hypothesis, the atom $B(x)$ is rewritten to a set of atoms which can match assertions in $\mathcal{A}$ using a homomorphism π such that $\pi(x) = a$. Procedure `Rewrite` can rewrite $A(x)$ to $B(x)$ using $B \sqsubseteq A$, and homomorphism π still provides the matching, giving the required result.

Now assume that rule (vi)(a) was used to produce $A(a)$ by applying axiom $\exists R.\top \sqsubseteq A$, for some role name R. Thus, there is a chase graph of height i for producing $R(a,b)$, for some individual or labelled null b. The result follows using similar reasoning to the above case.

Next assume that rule (vi)(b) was used to produce $A(a)$ by applying axiom $\exists R.B \sqsubseteq A$, for some role name R and concept name B. Thus, there are chase graphs of height at most i for producing each of $R(a,b)$ and $B(b)$. Note that, by Lemma 4, b cannot be labelled null. There are homomorphisms π_1 and π_2 mapping the rewritten atoms of $R(y,z)$ and $B(z)$, respectively, to assertions in $\mathcal{A}$ such that $\pi_1(z) = \pi_2(z) = b$. Hence π_1 and π_2 can be combined into a homomorphism mapping the rewritten atoms of $A(y)$ to assertions in $\mathcal{A}$.

We next turn to the case of a role assertion $R(b,c)$. This must have been produced by applying rules (iv), (v) or (vii). The cases of rules (iv) and (v) are straightforward, so let us consider the case of rule (vii). Assume rule (vii) used the axiom $A \sqsubseteq \exists R.\top$ to produce $R(b,c)$. Hence, there is a chase graph of height i for $A(b)$, and c is a labelled null. By the inductive hypothesis, $A(x)$ can be rewritten to a set of atoms which are matched to assertions in $\mathcal{A}$ using a homomorphism π such that $\pi(x) = b$. Now consider the atom $R(x,y)$. Because y is not bound, $R(x,y)$ can be rewritten to $A(x)$, and π still provides a homomorphism, as required. □

Theorem 2. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form and $\mathbf{q}$ a (concept or role) IQ over $\mathcal{T}$. Let QR be the set of CQs returned by* Rewrite$(\mathbf{q},\mathcal{T})$. *Then for each ABox $\mathcal{A}$, $\mathbf{q}^{(\mathcal{T},\mathcal{A})} = \bigcup_{qr\in QR} qr^{(\varnothing,\mathcal{A})}$.*

Proof. Let $\mathbf{q}$ be either the role IQ $q(x,y) \leftarrow R(x,y)$ (the case of $q(x,y) \leftarrow R(y,x)$ is analogous), where R is a role name in $\mathcal{K}$, or be the concept IQ $q(x) \leftarrow A(x)$, where A is a concept name in $\mathcal{K}$, and QR be the set of CQs returned by Rewrite$(q,\mathcal{T})$. Lemma 5 shows that the mappings for the variables x and y when evaluating queries in QR on $\mathcal{A}$ correspond exactly to the individuals or labelled nulls appearing in the respective assertions for R or A in the canonical model for $(\mathcal{T},\mathcal{A})$. Clearly, the result also holds when restricted to individuals, i.e., certain answers. □

Theorem 2 leads to the following corollary, which shows that a conjunction of the form $C_1 \sqcap C_2$ cannot be encoded on the left-hand side of a concept inclusion axiom in $\mathcal{T}$.

Corollary 1. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form, with $C_1, C_2 \in \mathsf{A}$ such that $\mathcal{T} \not\models C_1 \sqsubseteq C_2$ and $\mathcal{T} \not\models C_2 \sqsubseteq C_1$. Then $\mathcal{T}$ cannot encode a concept of the form $C_1 \sqcap C_2$ on the LHS of a concept inclusion axiom.*

Proof. Suppose that $\mathcal{T}$ encodes the axiom $C_1 \sqcap C_2 \sqsubseteq C_3$, for $C_3 \in \mathsf{A}$. Now consider the concept IQ $\mathbf{q}$ given by $q(x) \leftarrow C_3(x)$. Since Theorem 2 shows that Procedure `Rewrite` generates the perfect rewriting of $\mathbf{q}$, $\mathcal{T} \not\models C_1 \sqsubseteq C_2$ and $\mathcal{T} \not\models C_2 \sqsubseteq C_1$, Rewrite$(\mathbf{q},\mathcal{T})$ must contain the CQ $q(x) \leftarrow C_1(x), C_2(x)$. But this contradicts Lemma 1 which states that each query in Rewrite$(\mathbf{q},\mathcal{T})$ is a 2SPCQ. □

A further corollary is that, in the case of role IQs, the CQ rewriting can be computed in polynomial time by a simple check on sequences of role inclusions in the TBox.

Corollary 2. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form, $\mathbf{q}_R$ a role IQ of the form $q(x,y) \leftarrow R(x,y)$ and $\mathbf{q}_{R^-}$ a role IQ of the form $q(x,y) \leftarrow R(y,x)$. Let $R_\mathcal{T}$ and $R_\mathcal{T}^-$ be the sets of roles such that $R' \in R_\mathcal{T}$ if and only if $R' \rightharpoonup_\mathcal{T} R$ and $R'^- \in R_\mathcal{T}^-$ if and only if $R' \rightharpoonup_\mathcal{T} R^-$. Then for every ABox $\mathcal{A}$, it holds that*

$$\mathbf{q}_R^{(\mathcal{T},\mathcal{A})} = \bigcup_{P\in(R_\mathcal{T}\cup R_\mathcal{T}^-)} \mathbf{q}_P^{(\varnothing,\mathcal{A})} \quad \textit{and} \quad \mathbf{q}_{R^-}^{(\mathcal{T},\mathcal{A})} = \bigcup_{P\in(R_\mathcal{T}\cup R_\mathcal{T}^-)} \mathbf{q}_{P^-}^{(\varnothing,\mathcal{A})}.$$

Proof. We know from Proposition 1 that, given two role names R, R' appearing in an $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$ in normal form, we have that $\mathcal{T} \models R \sqsubseteq R'$ if and only if $R \rightharpoonup_\mathcal{T} R'$, and $\mathcal{T} \models R \sqsubseteq R'^-$ if and only if $R \rightharpoonup_\mathcal{T} R'^-$. Then the claim follows immediately from Theorem 2. □

Even though Theorem 2 shows that Procedure `Rewrite` is correct, for concept IQs the procedure may not terminate, i.e., the set QR of rewritten queries may be infinite. In the next subsection, we show how a finite set of rewritten queries can be produced if we rewrite concept IQs to 2RPQs rather than CQs.

5.2 NFA Rewriting of Concept Instance Queries for $\mathcal{ELHI}_h^{\ell in}$

In this section, we show how to encode rewritings for concept IQs under an $\mathcal{ELHI}_h^{\ell in}$ TBox by means of a non-deterministic finite-state automaton (NFA); intuitively, the automaton is able to encode the potentially infinite sequences of rewriting steps executed by Procedure `Rewrite`. Given an $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$ and a concept IQ $\mathbf{q}$ using concept name A in its body, we present next the

construction of an NFA $\mathsf{NFA}_{A,\mathcal{T}}$ based on $\mathcal{T}$ and A. Each symbol in the alphabet of $\mathsf{NFA}_{A,\mathcal{T}}$ is a concept name, a role name or the inverse of a role name. Since the sets of concept names and role names are disjoint, we can interpret a sequence accepted by the NFA as a complex concept by inserting existential quantifiers before the role names and inverse role names.

Recall from Lemma 1 that every query produced by Procedure `Rewrite` is a 2SPCQ. Theorem 3 below proves that the language accepted by $\mathsf{NFA}_{A,\mathcal{T}}$ is exactly $\{path(\mathbf{q}_{rew}) \mid \mathbf{q}_{rew} \in QR\}$, where QR is the set of queries returned by Procedure `Rewrite` for concept IQ $\mathbf{q}$.

Definition 12. Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form, Σ be the alphabet $\mathsf{P} \cup \mathsf{A}$, and $A \in \mathsf{A}$ be a concept name. The *NFA-rewriting* of A with respect to $\mathcal{T}$, denoted $\mathsf{NFA}_{A,\mathcal{T}}$, is the NFA $(Q, \Sigma, \delta, S_A, F)$ defined as follows:

(1) states S_A, SF_A and $S_\top$ are in Q, SF_A and $S_\top$ are in F, and transition (S_A, A, SF_A) is in δ; S_A is the initial state;

(2) for each $B \in \mathsf{A}$ that appears in at least one concept inclusion axiom of $\mathcal{T}$, states S_B and SF_B are in Q, SF_B is in F, and transition (S_B, B, SF_B) is in δ;

(3) for each concept inclusion axiom $\rho \in \mathcal{T}$:

(3.1) if ρ is of the form $B \sqsubseteq C$, where $B, C \in \mathsf{A}$, the transition (S_C, ε, S_B) is in δ;

(3.2) if ρ is of the form $B \sqsubseteq \exists R.\top$, where $B \in \mathsf{A}$ and $R \in \mathsf{R}$, for each transition $(S_X, R, S_\top) \in \delta$, for some $X \in \mathsf{A}$, the transition (S_X, ε, S_B) is in δ;

(3.3) if ρ is of the form $\exists R.\top \sqsubseteq B$, where $B \in \mathsf{A}$ and $R \in \mathsf{R}$, the transition $(S_B, R, S_\top)$ is in δ;

(3.4) if ρ is the form $\exists R.D \sqsubseteq C$, where $C, D \in \mathsf{A}$ and $R \in \mathsf{R}$, the transition (S_C, R, S_D) is in δ;

(4.1) for each role inclusion axiom $T \sqsubseteq S \in \mathcal{T}$ and each transition of the form $(S_C, S, S_B) \in \delta$ or $(S_C, S^-, S_B) \in \delta$ (where S_B could be $S_\top$), the transition (S_C, T, S_B) or (S_C, T^-, S_B) is in δ, respectively.

(4.2) for each role inclusion axiom $T \sqsubseteq S^- \in \mathcal{T}$ and each transition of the form $(S_C, S, S_B) \in \delta$ or $(S_C, S^-, S_B) \in \delta$ (where S_B could be $S_\top$), the transition (S_C, T^-, S_B) or (S_C, T, S_B) is in δ, respectively.

(5) there are no other states in Q or transitions in δ.

Example 12. Consider the TBox $\mathcal{T}$ defined by the following inclusion axioms in normal form:

$$\begin{array}{rcl} \exists P.\top & \sqsubseteq & A \\ \exists P.\top & \sqsubseteq & B \end{array} \qquad \begin{array}{rcl} \exists T.B & \sqsubseteq & C \\ \exists S.A & \sqsubseteq & A \\ \exists R.C & \sqsubseteq & D \end{array} \qquad \begin{array}{rcl} D & \sqsubseteq & \exists P.\top \\ V & \sqsubseteq & T^- \end{array}$$

where P, R, S, T, V are role names and A, B, C and D are concept names. Consider now the concept IQ $\mathbf{q} = q(x) \leftarrow A(x, \_)$. It is easy to see that $\mathsf{Rewrite}(\mathbf{q}, \mathcal{T})$ runs indefinitely, because rewriting rule (c) can be applied to the atom $A(x, \_)$ ad infinitum.

Let us consider the NFA rewriting of A with respect to $\mathcal{T}$. We construct $\mathsf{NFA}_{A,\mathcal{T}}$ (shown in Figure 2) as follows: by (2) in Definition 12 we have the transitions

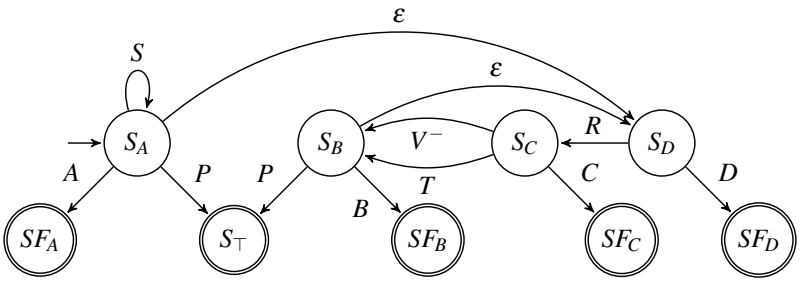

Figure 2: NFA for Example 12.

$(S_A,A,SF_A),(S_B,B,SF_B),(S_C,C,SF_C)$ and (S_D,D,SF_D); by (3.3) and the inclusion axioms $\exists P.\top \sqsubseteq A$ and $\exists P.\top \sqsubseteq B$, we have the transitions $(S_A,P,S_\top)$ and $(S_B,P,S_\top)$; by (3.2) and the inclusion axiom $D \sqsubseteq \exists P.\top$, we have the transitions (S_A,ε,S_D) and (S_B,ε,S_D); by (3.4) and the inclusion axioms $\exists R.C \sqsubseteq D$, $\exists T.B \sqsubseteq C$ and $\exists S.A \sqsubseteq A$, we have the transitions (S_D,R,S_C), (S_C,T,S_B) and (S_A,S,S_A); finally, by (4.2) and the inclusion axiom $V \sqsubseteq T^-$ we have the transition (S_C,V^-,S_B).

The language accepted by $\mathsf{NFA}_{A,\mathcal{T}}$ can be described by the regular expression

$$S^*(\,(A|P|D)\mid(((R(T|V^-))^*(P|B|D|RC))))),$$

where a sequence such as SRV^-B corresponds to the complex concept $\exists S.\exists R.\exists V^-.B$.

It can be verified that each of the infinitely many queries in $\mathsf{Rewrite}(\mathbf{q},\mathcal{T})$ is of the form $q(x) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(x,y)$. For example, some rewritings of $\mathbf{q}$ are:

$$\begin{array}{l} q(x) \leftarrow P(x,y) \\ q(x) \leftarrow S(x,z_1),A(z_1,y) \\ q(x) \leftarrow S(x,z_1),S(z_1,z_2),P(z_2,y) \\ q(x) \leftarrow R(x,z_1),T(z_1,z_2),R(z_2,z_3),C(z_3,y) \\ q(x) \leftarrow R(x,z_1),V(z_2,z_1),R(z_2,z_3),C(z_3,y) \end{array}$$

It is easy to verify that each of these output queries is a 2SPCQ and that each path is in $L(\mathsf{NFA}_{A,\mathcal{T}})$.

We now prove our main theorem for concept instance queries, which states that the set of sequences produced by $\mathsf{NFA}_{A,\mathcal{T}}$ is the same as the set of paths of the queries produced by Procedure `Rewrite` for the query $q(x) \leftarrow A(x,\_)$.

Theorem 3. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form, and $\mathbf{q}$ be the query $q(x) \leftarrow A(x,\_)$, with $A \in \mathsf{A}$. Then $L(\mathsf{NFA}_{A,\mathcal{T}}) = \{path(\mathbf{q}_{rew}) \mid \mathbf{q}_{rew} \in \mathsf{Rewrite}(\mathbf{q},\mathcal{T})\}$.*

Proof. Let QR denote the set of rewritten queries returned by $\mathsf{Rewrite}(\mathbf{q},\mathcal{T})$. We prove that (A) for each sequence $w \in L(\mathsf{NFA}_{A,\mathcal{T}})$, there is a query $\mathbf{q}_{rew} \in QR$ such that $path(\mathbf{q}_{rew}) = w$, and (B) for each query $\mathbf{q}_{rew} \in QR$, $path(\mathbf{q}_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$.

(A) Let w be a sequence in $L(\mathsf{NFA}_{A,\mathcal{T}})$. The proof proceeds by induction on the length $|w|$ of w.

BASE STEP. Let $|w| = 1$. There are three possibilities for the sequence of transitions from S_A yielding w.

1. There is a transition (S_A,A,SF_A) and $w = A$. Clearly, the path of the original query $\mathbf{q}$ is equal to A.

2. There is a transition $(S_A, R, S_\top)$ or $(S_A, R^-, S_\top)$ and $w = R$ or $w = R^-$, respectively, for some role name R. In the case of $(S_A, R, S_\top)$, the transition could have been added by step (3.3) in Definition 12, in which case the axiom $\exists R.\top \sqsubseteq A$ is in $\mathcal{T}$, and $\mathbf{q}$ would have been rewritten to a query $\mathbf{q}_{rew}$ for which $path(\mathbf{q}_{rew}) = R$. Alternatively, the transition could have been added as a result of a number of applications of steps (4.1) and (4.2) to some original transition $(S_A, T, S_\top)$, for role name T. This is also the case if the transition is $(S_A, R^-, S_\top)$. Steps (4.1) and (4.2) mimic the application of rewriting rules (e) and (f), hence there will be a query $\mathbf{q}_{rew} \in QR$ for which $path(\mathbf{q}_{rew})$ is R or R^-, respectively.

3. There is a sequence s of ε-transitions from state S_A to a state S_B, for some $B \in \mathsf{A}$, followed by the transition (S_B, B, SF_B) (since ε-transitions enter only states associated with concept names, not state $S_\top$ nor any final state). So $w = B$. Let one of the transitions in s be $t = (S_C, \varepsilon, S_D)$, for $C, D \in \mathsf{A}$. If t were added to the NFA by step (3.1), then $D \sqsubseteq C \in \mathcal{T}$. If t were added by step (3.2), then there must also be a transition $(S_C, R, S_\top)$ (added by step (3.3)), for some $R \in \mathsf{R}$, and axioms $D \sqsubseteq \exists R.\top$ and $\exists R.\top \sqsubseteq C$ are in $\mathcal{T}$. Then the sequence s corresponds to applying rewriting rules (a), (b) and (d), some number of times, to $\mathbf{q}$, resulting in a query $\mathbf{q}_{rew}$ such that $path(\mathbf{q}_{rew}) = B$.

INDUCTIVE STEP. Now assume that the result holds for all sequences of length k, for some $k \geqslant 1$, and consider a sequence w of length $k+1$. In the sequence s of transitions that leads to acceptance of w, there must be some transition whose label is a role name or its inverse (since that is the only way to generate a sequence of length greater than one). Let $t = (S_B, R, S_C)$ be the last such transition in s. Transition t may be followed by a sequence of ε-transitions leading to state S_D, say, followed by the final transition (S_D, D, SF_D) or $(S_D, P, S_\top)$ for some role name or its inverse P. So $w = uRD$ or $w = uRP$, where $|u| = k-1$. The NFA also has the transition $p = (S_B, B, SF_B)$. Following the sequence of transitions s up to but not including t followed by transition p leads to acceptance of the sequence uB of length k. By the inductive hypothesis, there is a query $\mathbf{q}_{rew} \in QR$ with $path(\mathbf{q}_{rew}) = uB$. Transition t must have been added by step (3.4) in Definition 12 as a result of axiom $\exists R.C \sqsubseteq B$ being in $\mathcal{T}$. Hence, Procedure `Rewrite` rewrites $\mathbf{q}_{rew}$ using rule (c) to a query $\mathbf{q}'_{rew} \in QR$ where $path(\mathbf{q}'_{rew}) = uRC$. The ε-transitions followed after transition t correspond to applying rewrites to $\mathbf{q}'_{rew}$, replacing C in $path(\mathbf{q}'_{rew})$ finally by D or P, as in case (3) above, yielding a query $\mathbf{q}''_{rew} \in QR$ with $path(\mathbf{q}''_{rew})$ equal to uRD or uRP, as required.

(B) The proof is by induction on the number of rewriting steps of Procedure `Rewrite` needed to produce $\mathbf{q}_{rew}$. As before, we denote by $Q^{[i]}$ the set of queries produced after the i-th iteration of the repeat loop in Procedure `Rewrite`.

BASE STEP. $Q^{[0]} = \{\mathbf{q}\}$, where $\mathbf{q}$ is $q(x) \leftarrow A(x, \_)$. Step (1) in Definition 12 sets S_A to be the initial state, and step (2) adds the transition (S_A, A, SF_A); therefore, $A \in L(\mathsf{NFA}_{A,\mathcal{T}})$ and the claim follows.

INDUCTIVE STEP. Suppose that for each $\mathbf{q}'_{rew} \in Q^{[i]}$, for some $i \geqslant 0$, we have that $path(\mathbf{q}'_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$, and let $\mathbf{q}_{rew} \in Q^{[i+1]}$. From Lemma 1, we know that the body of each query is a 2SPCQ. Query $\mathbf{q}_{rew}$ is produced by applying a rewriting rule to a query $\mathbf{q}'_{rew} \in Q^{[i]}$, based on one of the following forms of axioms in $\mathcal{T}$:

- $B \sqsubseteq A$: so $body(\mathbf{q}'_{rew})$ must contain an atom of the form $A(x_1, x_2)$. By applying rule (a) to $body(\mathbf{q}'_{rew})$, we obtain the same set of atoms in $body(\mathbf{q}_{rew})$ except with atom $B(x_1, x_2)$ instead of atom $A(x_1, x_2)$. When $path(\mathbf{q}'_{rew})$ is accepted by the NFA, the last transition must be

(S_A, A, SF_A) (since $\mathbf{q}'_{rew}$ is a 2SPCQ). Steps (3.1) and (1) in Definition 12 ensure that transitions (S_A, ε, S_B) and (S_B, B, SF_B) are also present. Hence, $path(\mathbf{q}_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$.

- $B \sqsubseteq \exists R.\top$: so $body(\mathbf{q}'_{rew})$ contains an atom of the form $R(x_1, \_)$. Since the second argument of R is unbound, R must be the last symbol in $path(\mathbf{q}'_{rew})$. Hence the last transition in the acceptance of $path(\mathbf{q}'_{rew})$ by the NFA must be $(S_X, R, S_\top)$, for some state S_X. By applying rule (d) to $body(\mathbf{q}'_{rew})$, we obtain the same set of atoms in $body(\mathbf{q}_{rew})$ except with atom $B(x_1, \_)$ instead of atom $R(x_1, \_)$. Then $path(\mathbf{q}_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$, since steps (3.2) and (1), respectively, ensure that transitions (S_X, ε, S_B) and (S_B, B, SF_B) are also present.

- $\exists R.\top \sqsubseteq A$: so $body(\mathbf{q}'_{rew})$ contains an atom of the form $A(x_1, x_2)$ (once again as the last atom). Hence the last transition in the acceptance of $path(\mathbf{q}'_{rew})$ by the NFA must be (S_A, A, SF_A). Applying rule (b) to $\mathbf{q}'_{rew}$, we obtain the same set of atoms in $body(\mathbf{q}_{rew})$ except with atom $R(x_1, \_)$ instead of atom $A(x_1, x_2)$. Then $path(\mathbf{q}_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$, since step (3.3) ensures that the transition $(S_A, R, S_\top)$ is present.

- $\exists R.B \sqsubseteq A$: so $body(\mathbf{q}'_{rew})$ contains an atom of the form $A(x_1, x_2)$ (once again as the last atom). Again we have that the last transition in the acceptance of $path(\mathbf{q}'_{rew})$ by the NFA must be (S_A, A, SF_A). Applying rule (c) to $\mathbf{q}'_{rew}$, we obtain the same set of atoms in $body(\mathbf{q}_{rew})$ except with the pair of atoms $R(x_0, x_1), B(x_1, x_2)$ instead of $A(x_1, x_2)$. Then $path(\mathbf{q}'_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$, since steps (3.4) and (1), respectively, ensure that transitions (S_A, R, S_B) and (S_B, B, SF_B) are present.

- $R_1 \sqsubseteq R_2$: so $body(\mathbf{q}'_{rew})$ contains an atom of the form $R_2(x_k, x_{k+1})$ or $R_2(x_{k+1}, x_k)$, for some $k \geqslant 1$. The acceptance of $path(\mathbf{q}'_{rew})$ by the NFA must traverse a transition (S_B, R_2, S_C) or (S_B, R_2^-, S_C), respectively, for some states S_B and S_C. Applying rule (e) we obtain $body(\mathbf{q}_{rew})$ from $body(\mathbf{q}'_{rew})$ by replacing $R_2(x_k, x_{k+1})$ by $R_1(x_k, x_{k+1})$ (respectively $R_2(x_{k+1}, x_k)$ by $R_1(x_{k+1}, x_k)$). Step (4.1) ensures that the transition (S_B, R_1, S_C) (respectively (S_B, R_1^-, S_C)) is present, hence $path(\mathbf{q}'_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$.

- $R_1 \sqsubseteq R_2^-$: so $body(\mathbf{q}'_{rew})$ contains an atom of the form $R_2(x_k, x_{k+1})$ or $R_2(x_{k+1}, x_k)$, for some $k \geqslant 1$. The acceptance of $path(\mathbf{q}'_{rew})$ by the NFA must traverse a transition (S_B, R_2, S_C) or (S_B, R_2^-, S_C), respectively, for some states S_B and S_C. Applying rule (f) we obtain $body(\mathbf{q}_{rew})$ from $body(\mathbf{q}'_{rew})$ by replacing $R_2(x_k, x_{k+1})$ by $R_1(x_{k+1}, x_k)$ (respectively $R_2(x_{k+1}, x_k)$ by $R_1(x_k, x_{k+1})$). Step (4.2) ensures that the transition (S_B, R_1^-, S_C) (respectively (S_B, R_1, S_C)) is present, hence $path(\mathbf{q}'_{rew}) \in L(\mathsf{NFA}_{A,\mathcal{T}})$.

□

As suggested above, we can interpret the paths of rewritten queries as complex concepts. Below, for complex concept B, we use the notation $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$ to mean that the sequence corresponding to the complex concept B is in the language denoted by the NFA. The following corollary relates concept inclusions involving complex concepts on the LHS which are entailed by a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ to satisfaction of queries produced by the NFA-rewriting in the base model $\mathcal{I}_\mathcal{A}$. This result will be used in the next section.

Corollary 3. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox, A be a concept name, and B be an $\mathcal{ELHI}_h^{\ell in}$ complex concept (not necessarily in normal form). The following are equivalent:*

1. $\mathcal{T} \models B \sqsubseteq A$,
2. $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, *and*
3. *for each ABox* $\mathcal{A}$ *and individual* $a \in \mathsf{ind}(\mathcal{A})$ *such that* $\mathcal{I}_\mathcal{A} \models B(a)$, *it is the case that* $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(a,\_)$, *where here we use* $\mathsf{NFA}_{A,\mathcal{T}}$ *as shorthand for the regular expression denoted by* $\mathsf{NFA}_{A,\mathcal{T}}$.

Proof. We prove that (1) if and only if (3), and (2) if and only if (3).

From Theorems 2 and 3 we have that $q(x) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(x,\_)$ is a perfect rewriting of $q(x) \leftarrow A(x,\_)$ with respect to $\mathcal{T}$. Therefore, for each ABox $\mathcal{A}$ and for each individual a, it is the case that $(\mathcal{T},\mathcal{A}) \models q(a) \leftarrow A(a,\_)$ if and only if $\mathcal{I}_A \models q(a) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(a,\_)$.

(1) $\Rightarrow$ (3) If $\mathcal{T} \models B \sqsubseteq A$, then for each ABox $\mathcal{A}$ and for each individual a we have that $\mathcal{I}_\mathcal{A} \models B(a)$ implies $(\mathcal{T},\mathcal{A}) \models q(a) \leftarrow A(a,\_)$ and therefore $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(a,\_)$.

(3) $\Rightarrow$ (1) If for each ABox $\mathcal{A}$ and each individual $a \in \mathsf{ind}(\mathcal{A})$ such that $\mathcal{I}_\mathcal{A} \models B(a)$, we have that $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(a,\_)$, this implies that $\mathcal{T} \models B \sqsubseteq A$.

(2) $\Rightarrow$ (3) If $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, then B corresponds to a complex concept. Let $\mathcal{A}$ be an ABox and a be an individual in $\mathsf{ind}(\mathcal{A})$ such that $\mathcal{I}_\mathcal{A} \models B(a)$. Then clearly $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(a,\_)$ since $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$.

(3) $\Rightarrow$ (2) Assume that, for each ABox $\mathcal{A}$ and each individual $a \in \mathsf{ind}(\mathcal{A})$ such that $\mathcal{I}_\mathcal{A} \models B(a)$, we have that $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow \mathsf{NFA}_{A,\mathcal{T}}(a,\_)$. Each sequence in $L(\mathsf{NFA}_{A,\mathcal{T}})$ corresponds to a distinct complex concept, so it must be the case that $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$. □

6. Rewriting CQs into UC2RPQs under $\mathcal{ELHI}_h^{\ell in}$

In this section, we build on the rewriting approaches for concept and role IQs developed in the previous section in order to rewrite CQs expressed with respect to an $\mathcal{ELHI}_h^{\ell in}$ TBox into UC2RPQs. The approach we take follows that of (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014), who investigate the problem of rewriting CQs under DL-Lite$_\mathcal{R}$ (or QL) and divide the problem into two parts:

Firstly, they deal with those TBox axioms that do not use existential quantification on the right-hand side, i.e., axioms that do not produce any labelled nulls when expanded. They term the set of such axioms the *flat* part of the TBox. Secondly, they present a *tree-witness* approach in order to generate rewritings for CQs under full QL. Tree witnesses capture those assertions that may produce labelled nulls.

In our case, the flat part of the TBox allows axioms with qualified existential quantification on the left-hand side of concept inclusions, so we make use of the NFA-based rewriting defined in Section 5. We describe our method, which rewrites a CQ to a C2RPQ, in Section 6.1. In Section 6.2, we then show how the rewriting of the flat part of the TBox can be combined with the tree-rewriting approach of (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014) to generate the perfect rewriting of a CQ into a UC2RPQ.

6.1 Rewriting for Flat $\mathcal{ELHI}_h^{\ell in}$

We first show how to rewrite CQs into C2RPQs under the special case of *flat* $\mathcal{ELHI}_h^{\ell in}$ TBoxes, i.e., those that do not contain existential quantifiers on the right-hand side of concept inclusions. In other words, a flat $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form can only contain concept and role inclusions of

the form $A_1 \sqsubseteq A_2$, $\exists R.D \sqsubseteq A$, $R_1 \sqsubseteq R_2$ or $R_1 \sqsubseteq R_2^-$, for concept names A, A_1, A_2, role names R_1, R_2, and D a concept name or $\top$.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB, with $\mathcal{T}$ a flat $\mathcal{ELHI}_h^{\ell in}$ TBox, $\mathcal{J}_\mathcal{K}$ be the canonical model for $\mathcal{K}$, $\mathbf{q}$ be a conjunctive query, and $\mathbf{a}$ a tuple of individuals. Then $\mathcal{K} \models \mathbf{q}(\mathbf{a})$ if and only if $\mathbf{q}(\mathbf{a})$ is true in $\mathcal{J}_\mathcal{K}$. Since $\mathcal{T}$ is flat, $\mathcal{J}_\mathcal{K}$ contains no labelled nulls, and so, from the definition of $\mathcal{J}_\mathcal{K}$ (Definition 3), Theorem 2 and Corollary 3, we have that:

- $\mathcal{J}_\mathcal{K} \models A(a)$ if and only if $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow B(a, \_)$ and $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, for some concept or role name B,
- $\mathcal{J}_\mathcal{K} \models P(a,b)$ if and only if (i) $\mathcal{I}_\mathcal{A} \models R(a,b)$ and $\mathcal{T} \models R \sqsubseteq P$, or (ii) $\mathcal{I}_\mathcal{A} \models R(b,a)$ and $\mathcal{T} \models R \sqsubseteq P^-$, for some role name R.

Following from this observation, we are now able to define a C2RPQ $\mathbf{q}_{\mathcal{T}\text{-}ext}$ such that, for any CQ $\mathbf{q}$ and any flat $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$, $\mathbf{q}_{\mathcal{T}\text{-}ext}$ is the perfect rewriting of $\mathbf{q}$ with respect to $\mathcal{T}$. (Here, the subscript $\mathcal{T}$-*ext* follows the notation of (Kontchakov & Zakharyaschev, 2014) and indicates the *extension* of the atoms of $\mathbf{q}$ according to the TBox $\mathcal{T}$.)

Definition 13. Given a CQ $\mathbf{q}$ and an $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$, we construct a C2RPQ $\mathbf{q}_{\mathcal{T}\text{-}ext}$ by replacing each atom $A(u_1,u_2)$ in $\mathbf{q}$, where A is a concept name, by $A_{\mathcal{T}\text{-}ext}(u_1,u_2)$, and each atom $P(u_1,u_2)$ in $\mathbf{q}$, where P is a role name, by $P_{\mathcal{T}\text{-}ext}(u_1,u_2)$. Formula $A_{\mathcal{T}\text{-}ext}(u_1,u_2)$ is defined as follows:

$$A_{\mathcal{T}\text{-}ext}(u_1,u_2) = \alpha(u_1,u_2),$$

where α is a regular expression denoting $L(\mathsf{NFA}_{A,\mathcal{T}})$. For $P_{\mathcal{T}\text{-}ext}(u_1,u_2)$, we first define

$$\mathcal{R} = \{R \mid \mathcal{T} \models R \sqsubseteq P\} \cup \{R^- \mid \mathcal{T} \models R \sqsubseteq P^-\}.$$

Now if $\mathcal{R} = \{R_1, \ldots, R_n\}$, where each R_i is a role name or inverse role name, then

$$P_{\mathcal{T}\text{-}ext}(u_1,u_2) = (R_1 \mid \cdots \mid R_n)(u_1,u_2).$$

Example 13. Consider the flat $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$ comprising the axioms

$$\begin{aligned} \exists \mathit{hasRelative}.\mathit{Person} &\sqsubseteq \mathit{Person} \\ \mathit{hasParent} &\sqsubseteq \mathit{hasRelative} \end{aligned}$$

and the CQ $\mathbf{q}$ given by $q(x,y) \leftarrow \mathit{Person}(x,z), \mathit{hasRelative}(x,y)$. Using Definition 13, we construct $\mathbf{q}_{\mathcal{T}\text{-}ext}$ as follows:

$$q(x,y) \leftarrow ((\mathit{hasRelative} | \mathit{hasParent})^* \mathit{Person})(x,z), (\mathit{hasRelative} | \mathit{hasParent})(x,y)$$

Proposition 3. *For each $\mathcal{ELHI}_h^{\ell in}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, concept name A, role name P, and individual names a and b we have:*

- $\mathcal{J}_\mathcal{K} \models A(a)$ *if and only if* $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow A_{\mathcal{T}\text{-}ext}(a, \_)$,
- $\mathcal{J}_\mathcal{K} \models P(a,b)$ *if and only if* $\mathcal{I}_\mathcal{A} \models q(a) \leftarrow P_{\mathcal{T}\text{-}ext}(a,b)$.

Proof. From Corollary 3, we have that $q() \leftarrow A_{\mathcal{T}\text{-}ext}(a, \_)$ is the perfect rewriting of $q() \leftarrow A(a)$ with respect to $\mathcal{T}$. The fact that $\mathcal{J}_\mathcal{K} \models P(a,b)$ if and only if $q() \leftarrow P_{\mathcal{T}\text{-}ext}(a,b)$ follows from the observation that the only way to infer $P(a,b)$, where a and b are individuals in $\mathcal{A}$, using an $\mathcal{ELHI}_h^{\ell in}$ TBox is through the closure of role inclusion axioms of the form $R_1 \sqsubseteq R_2$ and $R_1 \sqsubseteq R_2^-$. The claim follows. □

The next proposition shows that, for any CQ $\mathbf{q}$ and any flat $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$, $\mathbf{q}_{\mathcal{T}\text{-}ext}$ is a C2RPQ rewriting of $\mathbf{q}$ with respect to $\mathcal{T}$.

Proposition 4. *For any CQ* $\mathbf{q}$ *and any flat* $\mathcal{ELHI}_h^{\ell in}$ *TBox* $\mathcal{T}$, $\mathbf{q}_{\mathcal{T}\text{-}ext}$ *is the C2RPQ rewriting of* $\mathbf{q}$ *with respect to* $\mathcal{T}$.

Proof. Since $\mathcal{T}$ is flat, no axioms contain existential quantifiers on the right-hand side, so no labelled nulls appear during the chase procedure. Therefore we can construct the perfect rewriting of $\mathbf{q}$ by splitting $\mathbf{q}$ into its atoms, generating the perfect rewriting of the single atomic queries and taking the conjunction of the resulting set of atoms. More specifically, we can substitute every atom $A(z_1, z_2)$ in $\mathbf{q}$ with $A_{\mathcal{T}\text{-}ext}(z_1, z_2)$ and every atom $P(z_1, z_2)$ in $\mathbf{q}$ with $P_{\mathcal{T}\text{-}ext}(z_1, z_2)$ which gives rise to $\mathbf{q}_{\mathcal{T}\text{-}ext}$. Since each $A_{\mathcal{T}\text{-}ext}(z_1, z_2)$ and each $P_{\mathcal{T}\text{-}ext}(z_1, z_2)$ is a 2RPQ, $\mathbf{q}_{\mathcal{T}\text{-}ext}$ is a C2RPQ. □

6.2 Rewriting for Full $\mathcal{ELHI}_h^{\ell in}$

In this section, we show how the *tree-witness* approach of (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014) can be applied to generate rewritings for full $\mathcal{ELHI}_h^{\ell in}$ TBoxes. Tree witnesses capture those assertions in a KB canonical model that involve labelled nulls. Since the axioms that lead to the creation of labelled nulls are essentially the same in *QL* (as used in (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014)) and $\mathcal{ELHI}_h^{\ell in}$, it turns out that the structure of the tree witnesses is the same for both languages.

Since we will be comparing our results closely to those regarding *QL* in (Kontchakov & Zakharyaschev, 2014), we now introduce *QL* and relate it to $\mathcal{ELHI}_h^{\ell in}$. Concept and role inclusions in *QL* are of the form

$$B \sqsubseteq C \quad \text{and} \quad R_1 \sqsubseteq R_2$$

where R_1 and R_2 are roles (role names or their inverses) and B and C are concepts defined by the following grammar:

$$\begin{array}{lcl} B & ::= & A \mid \exists R.\top \\ C & ::= & A \mid \exists R.\top \mid \exists R.C \end{array}$$

where A is a concept name and R is a role. As we see above, unlike $\mathcal{ELHI}_h^{\ell in}$, *QL* allows only *unqualified* existential quantification on the LHS of concept inclusion axioms. The normal form for *QL* used in (Kontchakov & Zakharyaschev, 2014), which we will refer to here as *KZ normal form*, requires that each concept inclusion is of the form

$$A' \sqsubseteq A, \quad \exists R.\top \sqsubseteq A \quad \text{or} \quad A \sqsubseteq \exists R.D$$

where R is a role, A and A' are concept names and D is either a concept name or $\top$.

We see that KZ normal form allows for axioms of the form $A \sqsubseteq \exists R.D$, as well as use of inverse role names on the LHS of axioms. In contrast, our normal form for $\mathcal{ELHI}_h^{\ell in}$ — presented in Section 3 — introduces new role names and axioms to ensure that only unqualified existential

quantification appears on the RHS of concept inclusion axioms, and that inverse role names appear only on the RHS of role inclusion axioms.

Let $\mathcal{T}$ be a *QL* TBox in KZ normal form. For each axiom of the form $A \sqsubseteq \exists R.D \in \mathcal{T}$, Kontchakov and Zakharyaschev introduce the symbol $w_{\exists R.D}$, representing a *witness* to the application of rule *(vii)* to the axiom during the chase. These witnesses are used in the construction of labelled nulls needed in the canonical model of $\mathcal{T}$ and a given ABox. In order to model how an individual in the ABox gives rise to a witness, and how one witness gives rise to another, they define a *generating relation* $\rightsquigarrow_{\mathcal{T},\mathcal{A}}$ on the set of witnesses together with $\mathsf{ind}(\mathcal{A})$ by setting:

1. $a \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R.D}$ if $a \in \mathsf{ind}(\mathcal{A})$, $\mathcal{I}_\mathcal{A} \models B(a)$ and $\mathcal{T} \models B \sqsubseteq \exists R.D$
2. $w_{\exists S.B} \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R.D}$ if $\mathcal{T} \models \exists S^-.\top \sqsubseteq \exists R.D$ or $\mathcal{T} \models B \sqsubseteq \exists R.D$.

Henceforth we will assume that *QL* TBoxes are also in our normal form, defined in Section 3, and we will point out the modifications to the definitions from (Kontchakov & Zakharyaschev, 2014) that are required as a result. If we assume that a *QL* TBox is in our normal form, the definitions of the generating relation become:

1. $a \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R}$ if $a \in \mathsf{ind}(\mathcal{A})$, $\mathcal{I}_\mathcal{A} \models B(a)$ and $\mathcal{T} \models B \sqsubseteq \exists R.\top$
2. $w_{\exists S} \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R}$ if $\mathcal{T} \models \exists S^-.\top \sqsubseteq \exists R.\top$.

Note that we have abbreviated a witness of the form $w_{\exists R.\top}$ to $w_{\exists R}$. We can drop the condition $\mathcal{T} \models B \sqsubseteq \exists R.D$ from the original case (2) because, when converting to our normal form, new role names representing $\exists S.B$ and $\exists R.D$, say S_B and R_D respectively, will be introduced, and in the converted TBox $\mathcal{T}'$, we will have $\mathcal{T}' \models \exists S_B^-.\top \sqsubseteq B$ and $\mathcal{T}' \models \exists R_D^-.\top \sqsubseteq D$. So if we had $\mathcal{T} \models B \sqsubseteq \exists R.D$ in the original TBox, we will have $\mathcal{T}' \models \exists S_B^-.\top \sqsubseteq \exists R_D.\top$ in the converted TBox, which is covered by the new case (2).

Kontchakov and Zakharyaschev then compose individual witnesses into paths that are used to represent labelled nulls. A *path* σ on the generating relation $\rightsquigarrow_{\mathcal{T},\mathcal{A}}$ is a finite concatenation $aw_{\exists R_1} \dots w_{\exists R_n}$, $n \geqslant 0$, such that $a \in \mathsf{ind}(\mathcal{A})$ and, if $n > 0$, then $a \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R_1}$ and $w_{\exists R_i} \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R_{i+1}}$, for $1 \leqslant i < n$. Thus, a path of the form $\sigma w_{\exists R}$ denotes the fresh labelled null introduced by applying *(vii)* to some $A \sqsubseteq \exists R.\top$ on (the individual or labelled null represented by) σ. These paths are then used to construct their canonical model, denoted by $\mathcal{C}_{\mathcal{T},\mathcal{A}}$, for the KB $\mathcal{K} = (\mathcal{T},\mathcal{A})$ (in what follows, we will differentiate between $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ and the canonical model $\mathcal{J}_\mathcal{K}$ constructed using our technique — see Definitions 15 and 16).

Because $\mathcal{ELHI}_h^{\ell in}$ allows for qualified existential quantification on the LHS of axioms, we cannot use precisely the same approach as Kontchakov and Zakharyaschev, as demonstrated in the following example.

Example 14. Let $\mathcal{K} = (\mathcal{T},\mathcal{A})$ be a $\mathcal{ELHI}_h^{\ell in}$ KB, with $\mathcal{T}$ comprising the axioms

$$\begin{aligned} A &\sqsubseteq \exists R.\top \\ R &\sqsubseteq S^- \\ \exists S.B &\sqsubseteq C \\ C &\sqsubseteq \exists U.\top \end{aligned}$$

and ABox $\mathcal{A}$ containing only the assertions $A(a)$, $A(b)$ and $B(a)$. Then the labelled nulls $aw_{\exists R}$ and $bw_{\exists R}$ are generated when applying the first axiom to $A(a)$ and $A(b)$, respectively. Hence, $(a, aw_{\exists R})$

and $(b, bw_{\exists R})$ are in $R^{\mathcal{J}_\mathcal{K}}$. The second axiom adds $(aw_{\exists R}, a)$ and $(bw_{\exists R}, b)$ to $S^{\mathcal{J}_\mathcal{K}}$. Because of qualified existential quantification in the third axiom, we have to check that the second component of S also occurs in B. Hence, $aw_{\exists R}$ can be added to $C^{\mathcal{J}_\mathcal{K}}$ by the third axiom but $bw_{\exists R}$ cannot, because $B(a)$ is in $\mathcal{A}$ but $B(b)$ is not.

Now consider the axiom $C \sqsubseteq \exists U.\top$. With $aw_{\exists R}$ being in $C^{\mathcal{J}_\mathcal{K}}$, this results in $(aw_{\exists R}, aw_{\exists R}w_{\exists U})$ being added to $U^{\mathcal{J}_\mathcal{K}}$. But the same is not true for $bw_{\exists R}$, so rule (2) from the (modified) generating relation is not correct in our setting because it is not always the case that $w_{\exists R} \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists U}$.

As shown in the above example, we cannot use a generating relation to produce paths representing labelled nulls in $\mathcal{ELHI}_h^{\ell in}$ KBs. Instead, we will generate the paths directly, and call them w-paths, for "witness paths".

Definition 14. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ELHI}_h^{\ell in}$ KB. A *w-path* σ for $\mathcal{K}$ is a sequence of one of the following forms:

1. $aw_{\exists R}$, where $a \in \mathsf{ind}(\mathcal{A})$, $A \sqsubseteq \exists R.\top \in \mathcal{T}$, and for some $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, $\mathcal{I}_\mathcal{A} \models B(a)$, or
2. $\sigma w_{\exists S}$, where $\sigma = a \cdots w_{\exists R}$ is a w-path such that $|\sigma| \geqslant 2$, $A \sqsubseteq \exists S.\top \in \mathcal{T}$, and either
   (a) $R^- \in L(\mathsf{NFA}_{A,\mathcal{T}})$, or
   (b) $|\sigma| = 2$, and, for some concept name B, $R^-B \in L(\mathsf{NFA}_{A,\mathcal{T}})$ and $\mathcal{I}_\mathcal{A} \models B(a)$.

Rule (2)(a) in the above definition captures the case of unqualified existential quantification on the LHS of axioms, while rule (2)(b) captures the case of qualified existential quantification, as demonstrated in the following example.

Example 15. Consider again the TBox $\mathcal{T}$ and ABox $\mathcal{A}$ in Example 14. The w-paths $aw_{\exists R}$ and $bw_{\exists R}$ are produced by rule (1) in Definition 14 because $L(\mathsf{NFA}_{A,\mathcal{T}}) = \{A\}$ and $\{A(a), A(b)\} \subseteq \mathcal{A}$.

Now consider w-paths produced by adding the witness $w_{\exists U}$ using the axiom $C \sqsubseteq \exists U.\top$. $L(\mathsf{NFA}_{C,\mathcal{T}}) = \{C, SB, R^-B\}$. Since there are no assertions in $\mathcal{A}$ for any of the (complex) concepts corresponding to the sequences in $L(\mathsf{NFA}_{C,\mathcal{T}})$, rule (1) produces nothing. Rule (2)(a) does not apply because $R^- \notin L(\mathsf{NFA}_{C,\mathcal{T}})$. Rule (2)(b) applies to both $aw_{\exists R}$ and $bw_{\exists R}$ because they are each of length two and $R^-B \in L(\mathsf{NFA}_{C,\mathcal{T}})$. However, only the w-path $aw_{\exists R}w_{\exists U}$ is generated because $\mathcal{I}_\mathcal{A} \models B(a)$ while $\mathcal{I}_\mathcal{A} \not\models B(b)$.

The reason that the sequences produced by rule (2)(b) are limited to length three is as follows. Assume we add the following three axioms to those of $\mathcal{T}$ in Example 14:

$$\begin{aligned} U &\sqsubseteq V^- \\ \exists V.D &\sqsubseteq E \\ E &\sqsubseteq \exists W.\top \end{aligned}$$

Starting from the w-path $aw_{\exists R}w_{\exists U}$, it seems we should be able to generate the w-path $aw_{\exists R}w_{\exists U}w_{\exists W}$ in certain circumstances. That would require $(aw_{\exists R}w_{\exists U}, aw_{\exists R})$ being in $V^{\mathcal{J}_\mathcal{K}}$ and $aw_{\exists R}$ being in $D^{\mathcal{J}_\mathcal{K}}$. However, this is impossible because Lemma 4 proved that no shared labelled nulls can appear in any chase graph for an $\mathcal{ELHI}_h^{\ell in}$ KB. So the maximum length of any w-path in which qualified existential quantification is used in the final step of generation is 3.

The following proposition shows that each w-path corresponds to a labelled null introduced during the chase procedure according to Definition 3 and vice versa.

Proposition 5. *Let $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ be an $\mathcal{ELHI}_h^{\ell in}$ KB, with $\mathcal{T}$ in normal form, and $A \sqsubseteq \exists R.\top \in \mathcal{T}$. A labelled null is introduced by the chase procedure using rule* (vii) *on $A \sqsubseteq \exists R.\top$ if and only if there is a w-path $\sigma w_{\exists R}$.*

Proof. (Only if) Assume that a labelled null is introduced by the chase procedure using rule *(vii)* on $A \sqsubseteq \exists R.\top$. The proof proceeds by induction on the number of applications of rule *(vii)*. If it is the first application, then no labelled nulls have been generated yet and so the rule must be applied to some assertion $A(a)$ such that $\mathcal{T} \models B \sqsubseteq A$ and $\mathcal{I}_\mathcal{A} \models B(a)$, where $a \in \mathsf{ind}(\mathcal{A})$. From Corollary 3, we know that $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, so rule (1) in Definition 14 applies and we have the w-path $aw_{\exists R}$.

Assume the result holds for i or fewer applications and let this be application number $i+1$. If the rule is applied to an assertion involving an individual, this is similar to the base case. So assume that the rule is applied to $A(d)$, where d is a labelled null. Labelled null d must have been produced by an application of rule *(vii)* to an axiom of the form $B \sqsubseteq \exists S.\top$. By the inductive hypothesis, there is a w-path $\sigma w_{\exists S}$ corresponding to labelled null d. For $\mathcal{J}_\mathcal{K} \models A(d)$, we must have either (i) $\mathcal{T} \models \exists S^- \sqsubseteq A$ or (ii) $\mathcal{T} \models \exists S^-.C \sqsubseteq A$, for some concept C such that $\mathcal{I}_\mathcal{A} \models C(c)$, for some $c \in \mathsf{ind}(\mathcal{A})$. Note that c cannot be a labelled null since, if it were, it would be shared, contradicting Lemma 4.

In case (i), we have $S^- \in L(\mathsf{NFA}_{A,\mathcal{T}})$ by Corollary 3. Hence, by rule 2(a) in Definition 14, the w-path $\sigma w_{\exists S} w_{\exists R}$ is generated, corresponding to the labelled null introduced by applying rule *(vii)* to $A \sqsubseteq \exists R.\top$.

In case (ii), we have $S^- C \in L(\mathsf{NFA}_{A,\mathcal{T}})$ by Corollary 3. Hence, rule 2(b) in Definition 14 applies and the w-path $\sigma w_{\exists S} w_{\exists R}$ is once again generated, corresponding to the labelled null introduced by applying rule *(vii)* to $A \sqsubseteq \exists R.\top$.

(If) Conversely, assume that there is a w-path $\sigma w_{\exists R}$. The proof is by induction on the length of the w-path. Assume σ is of length 1, i.e., σ is some $a \in \mathsf{ind}(\mathcal{A})$. Hence, $aw_{\exists R}$ must have been generated by rule (1) in Definition 14. So we know that $\mathcal{I}_\mathcal{A} \models B(a)$ and $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, for some B. By Corollary 3, we have $\mathcal{T} \models B \sqsubseteq A$; hence, rule *(vii)* will be applied to $A \sqsubseteq \exists R.\top$ and $A(a)$, generating a labelled null.

Assume the result holds for w-paths of length i, for some $i \geqslant 2$, and consider a w-path $\sigma w_{\exists R}$ of length $i+1$. Now rule (2) in Definition 14 applies. By the inductive hypothesis a labelled null d corresponding to σ is generated. Let $\sigma = a \cdots w_{\exists S}$. Assume first that the w-path was generated by rule (2)(a). So we have that $S^- \in L(\mathsf{NFA}_{A,\mathcal{T}})$. By Corollary 3, we know that $\mathcal{T} \models \exists S^- \sqsubseteq A$; so $\mathcal{J}_\mathcal{K} \models A(d)$ and a labelled null will be generated by applying rule *(vii)* of the chase procedure to $A \sqsubseteq \exists R.\top$ and $A(d)$.

Now assume that the w-path was generated by rule (2)(b). Therefore, $\sigma = aw_{\exists S}$ and $S^- B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, for some concept name B such that $\mathcal{I}_\mathcal{A} \models B(a)$. Recall that the labelled null d corresponds to σ. By Corollary 3, we know that $\mathcal{T} \models \exists S^- B \sqsubseteq A$; hence $\mathcal{I}_\mathcal{K} \models S(a,d)$ and therefore $\mathcal{I}_\mathcal{K} \models A(d)$. We conclude that a labelled null will be generated by applying rule *(vii)* of the chase procedure to $A \sqsubseteq \exists R.\top$ and $A(d)$. □

Given KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we now use the labelled nulls represented by w-paths for $\mathcal{K}$ to construct our canonical model for $\mathcal{K}$.

Definition 15. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ELHI}_h^{\ell in}$ KB, and let $\Delta^\mathcal{K}$ denote the set of all w-paths. The *canonical model* $\mathcal{J}_\mathcal{K}$ is defined as follows (note that B below denotes a complex concept, while P

denotes a role name):

$$
\begin{aligned}
a^{\mathcal{J}_\mathcal{K}} &= a, \text{ for } a \in \mathsf{ind}(\mathcal{A}) \\
A^{\mathcal{J}_\mathcal{K}} &= \{a \in \mathsf{ind}(\mathcal{A}) \mid B \in L(\mathsf{NFA}_{A,\mathcal{T}}) \text{ and } \mathcal{I}_\mathcal{A} \models B(a)\} \cup \\
&\quad \{\sigma w_{\exists R} \mid \sigma w_{\exists R} \in \Delta^\mathcal{K} \text{ and } R^- \in L(\mathsf{NFA}_{A,\mathcal{T}})\} \cup \\
&\quad \{a w_{\exists R} \mid a w_{\exists R} \in \Delta^\mathcal{K} \text{ and } R^- B \in L(\mathsf{NFA}_{A,\mathcal{T}}) \text{ and } \mathcal{I}_\mathcal{A} \models B(a)\}, \text{ for concept name } A \\
P^{\mathcal{J}_\mathcal{K}} &= \{(a,b) \mid \mathcal{I}_\mathcal{A} \models R(a,b) \text{ and } \mathcal{T} \models R \sqsubseteq P\} \cup \\
&\quad \{(b,a) \mid \mathcal{I}_\mathcal{A} \models R(a,b) \text{ and } \mathcal{T} \models R \sqsubseteq P^-\} \cup \\
&\quad \{(\sigma, \sigma w_{\exists R}) \mid \sigma w_{\exists R} \in \Delta^\mathcal{K} \text{ and } \mathcal{T} \models R \sqsubseteq P\} \cup \\
&\quad \{(\sigma w_{\exists R}, \sigma) \mid \sigma w_{\exists R} \in \Delta^\mathcal{K} \text{ and } \mathcal{T} \models R \sqsubseteq P^-\}, \text{ for role name } P.
\end{aligned}
$$

In the following theorem, we show that $\mathcal{J}_\mathcal{K}$ is indeed a canonical model for $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

Theorem 4. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ELHI}_h^{\ell in}$ KB, with $\mathcal{T}$ in normal form. The model $\mathcal{J}_\mathcal{K}$ of Definition 15 is isomorphic to the model produced by the chase of $\mathcal{K}$ (Definition 3).*

Proof. Corollary 3 proves that $\mathcal{T} \models B \sqsubseteq A$ if and only if $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$. Hence, an individual a is added to A^{I_k}, for some k, by rule *(iii)* or rule *(vi)* in Definition 3 if and only if $\mathcal{I}_\mathcal{A} \models B(a)$ and $\mathcal{T} \models B \sqsubseteq A$ which matches the first set in the definition of $A^{\mathcal{J}_\mathcal{K}}$ in Definition 15.

Proposition 5 proves that a labelled null d' is introduced by the chase using rule *(vii)* if and only if there is a w-path $\sigma w_{\exists R}$, where role name R is existentially qualified on the RHS of an axiom. Hence, the labelled nulls of the chase and the w-paths of Definition 14 are in one-to-one correspondence. Rule (2) in Definition 14 captures the two circumstances in which a w-path can appear in the interpretation of a concept name A. These circumstances match the definitions of the two sets involving w-paths in the definition of $A^{\mathcal{J}_\mathcal{K}}$ in Definition 15.

For a role name P, a pair (d,d') is added to $P^{\mathcal{I}_k}$, for some k, by one of rules *(iv)*, *(v)* or *(vii)*. If d and d' are both individuals, then it must be the case that, for some role name R, $\mathcal{I}_\mathcal{A} \models R(d,d')$ and $\mathcal{T} \models R \sqsubseteq P$, or $\mathcal{I}_\mathcal{A} \models R(d',d)$ and $\mathcal{T} \models R \sqsubseteq P^-$. These two cases match the definitions of the first two sets in the definition of $P^{\mathcal{J}_\mathcal{K}}$ in Definition 15.

Now assume that at least one of d and d' is a labelled null. If only one of d or d' is a labelled null (while the other is an individual), then let j ($j < k$) be the chase iteration at which the labelled null was introduced by rule *(vii)*. If both d and d' are labelled nulls, then let j ($j < k$) be the greater of the two chase iterations in which d and d' were introduced by rule *(vii)*. Assume in all cases that rule *(vii)* was applied to an axiom with $\exists R.\top$ on the RHS, for some role name R. There are two cases to consider. (1) If d is an individual and d' a labelled null or both are labelled nulls and d' was introduced at iteration j, then (d,d') must have been added to $R^{\mathcal{I}_j}$ by the chase. (2) If, instead, d' is an individual and d a labelled null or both are labelled nulls and d was introduced at iteration j, then (d',d) must have been added to $R^{\mathcal{I}_j}$ by the chase. In order for the pair (d,d') to be added to $P^{\mathcal{I}_k}$, it must be that case (1) applies and $\mathcal{T} \models R \sqsubseteq P$, or that case (2) applies and $\mathcal{T} \models R \sqsubseteq P^-$. From Proposition 5, we know that in case (1) there exists a w-path or individual σ corresponding to d and a w-path $\sigma w_{\exists R}$ corresponding to d', or in case (2) there exists a w-path or individual σ corresponding to d' and a w-path $\sigma w_{\exists R}$ corresponding to d. Hence, analogous pairs to (d,d') or (d',d) are added to $P^{\mathcal{J}_\mathcal{K}}$ by the last two sets in the union of sets for $P^{\mathcal{J}_\mathcal{K}}$ in Definition 15.

We conclude that $\mathcal{J}_\mathcal{K}$ is isomorphic to the model produced by the chase, and is hence a canonical model for $\mathcal{K}$. □

Below we prove that, if $\mathcal{K}$ is a *QL* KB, then the canonical model $\mathcal{J}_\mathcal{K}$ defined above is identical to the canonical model $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ based on $\rightsquigarrow_{\mathcal{T},\mathcal{A}}$-paths defined in (Kontchakov & Zakharyaschev, 2014). Since we assume that $\mathcal{K}$ is expressed in the normal form of Section 3, we give next the definition of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ modified to take account of our normal form.

Definition 16. Let $\mathcal{K} = (\mathcal{T},\mathcal{A})$ be a *QL* KB in our normal form, and let $\Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ denote the set of all $\rightsquigarrow_{\mathcal{T},\mathcal{A}}$-paths. The *KZ canonical model* $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ is defined as follows:

$$\begin{array}{rcl}
a^{\mathcal{C}_{\mathcal{T},\mathcal{A}}} & = & a, \text{ for } a \in \mathsf{ind}(\mathcal{A}) \\
A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}} & = & \{a \in \mathsf{ind}(\mathcal{A}) \mid \mathcal{T} \models B \sqsubseteq A \text{ and } \mathcal{I}_\mathcal{A} \models B(a)\} \cup \\
 & & \{\sigma w_{\exists R} \mid \mathcal{T} \models \exists R^- \sqsubseteq A\}, \text{ for concept name } A \\
P^{\mathcal{C}_{\mathcal{T},\mathcal{A}}} & = & \{(a,b) \mid \mathcal{I}_\mathcal{A} \models R(a,b) \text{ and } \mathcal{T} \models R \sqsubseteq P\} \cup \\
 & & \{(b,a) \mid \mathcal{I}_\mathcal{A} \models R(a,b) \text{ and } \mathcal{T} \models R \sqsubseteq P^-\} \cup \\
 & & \{(\sigma w_{\exists R}, \sigma) \mid \mathsf{tail}(\sigma) \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R}, \mathcal{T} \models R \sqsubseteq P^-\} \cup \\
 & & \{(\sigma, \sigma w_{\exists R}) \mid \mathsf{tail}(\sigma) \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R}, \mathcal{T} \models R \sqsubseteq P\}, \text{ for role name } P.
\end{array}$$

Proposition 6. *If $\mathcal{K} = (\mathcal{T},\mathcal{A})$ is a QL KB in our normal form, then*

1. $a^{\mathcal{J}_\mathcal{K}} = a^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$*, for each* $a \in \mathsf{ind}(\mathcal{A})$*,*
2. $\Delta^\mathcal{K} = \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$*,*
3. $A^{\mathcal{J}_\mathcal{K}} = A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$*, for each concept name A, and*
4. $P^{\mathcal{J}_\mathcal{K}} = P^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$*, for each role name P.*

Proof. The fact that (1) is true is trivial.

For (2), note that, since $\mathcal{T}$ is a *QL* TBox, only unqualified existential quantification occurs on the LHS of any axiom in $\mathcal{T}$. Hence, for each concept name A, each sequence in $L(\mathsf{NFA}_{A,\mathcal{T}})$ is of length one, which means that rule (2)(b) in Definition 14 is not used in constructing w-paths. Clearly, for individual a, $a \in \Delta^\mathcal{K}$ if and only if $a \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. So we next prove that $\sigma w_{\exists R} \in \Delta^\mathcal{K}$ if and only if $\sigma w_{\exists R} \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$.

(Only if) Assume that $\sigma w_{\exists R} \in \Delta^\mathcal{K}$. The proof proceeds by induction on the length of $\sigma w_{\exists R}$. The base case corresponds to σ being a, for some $a \in \mathsf{ind}(\mathcal{A})$. From rule (1) in Definition 14, we have that $A \sqsubseteq \exists R.\top \in \mathcal{T}$, and for some $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$, $\mathcal{I}_\mathcal{A} \models B(a)$. Corollary 3 tells us that $\mathcal{T} \models B \sqsubseteq A$, and hence $\mathcal{T} \models B \sqsubseteq \exists R.\top$. From case (1) of the generating relation, we have $a \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists R}$, and therefore $\sigma w_{\exists R} \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$.

Assume that for each path σ of length i, for some $i \geqslant 2$, $\sigma \in \Delta^\mathcal{K}$ implies that $\sigma \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, and let $\sigma w_{\exists S} \in \Delta^\mathcal{K}$. So $\sigma w_{\exists S}$ must have been generated by rule (2)(a), where $\sigma = a \cdots w_{\exists R}$, $A \sqsubseteq \exists S.\top \in \mathcal{T}$, and $R^- \in L(\mathsf{NFA}_{A,\mathcal{T}})$. Hence, by Corollary 3, we have that $\mathcal{T} \models R^- \sqsubseteq \exists S.\top$. Rule (2) of the generating relation gives us that $w_{\exists R} \rightsquigarrow_{\mathcal{T},\mathcal{A}} w_{\exists S}$. By the inductive hypothesis, $\sigma = a \cdots w_{\exists R}$ is in $\Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$; hence, $\sigma w_{\exists S} \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$.

(If) Now assume that $\sigma w_{\exists R} \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. The proof again proceeds by induction on the length of $\sigma w_{\exists R}$. The base case corresponds to σ being a, for some $a \in \mathsf{ind}(\mathcal{A})$. This path must have been generated using case (1) of the generating relation; hence, for some concept B, we have $\mathcal{I}_\mathcal{A} \models B(a)$ and $\mathcal{T} \models B \sqsubseteq \exists R.\top$. Therefore, there must be some concept name A (possibly equal to B) such that

$A \sqsubseteq \exists R.\top \in \mathcal{T}$ and $\mathcal{T} \models B \sqsubseteq A$. By Corollary 3, we know that $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$. Hence, $aw_{\exists R}$ is generated by rule (1) of Definition 14.

Assume that for each path σ of length i, for some $i \geqslant 2$, $\sigma \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ implies that $\sigma \in \Delta^{\mathcal{K}}$, and let $\sigma w_{\exists S} \in \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. So $\sigma = a \cdots w_{\exists R}$, for some role name R, and $\sigma w_{\exists S}$ must have been generated by applying rule (2) of the generating relation, so we know that $\mathcal{T} \models \exists R^{-}.\top \sqsubseteq \exists S.\top$. Therefore, there must be some concept name A such that $A \sqsubseteq \exists S.\top \in \mathcal{T}$ and $\mathcal{T} \models R^{-} \sqsubseteq A$. By Corollary 3, we know that $R^{-} \in L(\mathsf{NFA}_{A,\mathcal{T}})$. By the inductive hypothesis, σ is in $\Delta^{\mathcal{K}}$; hence, $\sigma w_{\exists S}$ is generated by rule(2)(a) of Definition 14.

Now consider claim (3), that $A^{\mathcal{J}_{\mathcal{K}}} = A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, for each concept name A. Let A be a concept name A and $a \in \mathsf{ind}(\mathcal{A})$. Considering the sets of individuals specified for $A^{\mathcal{J}_{\mathcal{K}}}$ and $A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ in Definitions 15 and 16, respectively, Corollary 3 tells us that $\mathcal{T} \models B \sqsubseteq A$ if and only if $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$. Hence, $a \in A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ if and only if $a \in A^{\mathcal{J}_{\mathcal{K}}}$.

The set of paths in $A^{\mathcal{J}_{\mathcal{K}}}$ is constructed using only the first of the two sets in Definition 15 relating to paths, since, as noted earlier in the proof, each sequence in $L(\mathsf{NFA}_{A,\mathcal{T}})$ is of length one. Once again, Corollary 3 tells us that $\mathcal{T} \models R^{-} \sqsubseteq A$ if and only if $R^{-} \in L(\mathsf{NFA}_{A,\mathcal{T}})$. This, along with the fact that $\Delta^{\mathcal{K}} = \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, allows us to conclude that, for path σ, $\sigma \in A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ if and only if $\sigma \in A^{\mathcal{J}_{\mathcal{K}}}$

Claim (4), namely $P^{\mathcal{J}_{\mathcal{K}}} = P^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, follows from the respective definitions and the fact that $\Delta^{\mathcal{K}} = \Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. □

Given the above proposition and also for easier comparison with their work, we will now switch to using the notation of (Kontchakov & Zakharyaschev, 2014) for our canonical model of $\mathcal{K} = (\mathcal{T},\mathcal{A})$, where $\mathcal{T}$ is an $\mathcal{ELHI}_h^{lin}$ TBox; that is, we use $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ rather than $\mathcal{J}_{\mathcal{K}}$, even though our canonical model is still defined as in Definition 15.

Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{lin}$ TBox and $A \sqsubseteq \exists R.\top$ an axiom in $\mathcal{T}$. For an arbitrary individual name a, we define the *$\exists R$-generated $\mathcal{T}$-tree on a*, denoted by $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$, as the restriction of the canonical model of the KB $(\mathcal{T},\{\exists R(a)\})$ to the domain consisting of a and the labelled nulls with prefix $aw_{\exists R}$ (Kontchakov & Zakharyaschev, 2014). Three such trees are shown by the light grey areas with dashed outlines in Figure 3. Each tree is rooted at an individual (a or b in the figure), with the remaining nodes in each tree being labelled nulls. Nodes are labelled with concept names, where known. The (directed) edges are labelled with role names. There is an edge labelled P from node u to node v if (u,v) is in the canonical model, as defined in Definitions 15 or 16.

Example 16. Consider the *QL* TBox $\mathcal{T}$ with the following axioms (adapted from (Kontchakov & Zakharyaschev, 2014)):

$$\begin{array}{rcl} A & \sqsubseteq & \exists R.\top \\ R & \sqsubseteq & U^{-} \\ \exists U.\top & \sqsubseteq & D \end{array} \qquad \begin{array}{rcl} D & \sqsubseteq & \exists P_1.\top \\ D & \sqsubseteq & \exists P_2.\top \\ B & \sqsubseteq & \exists S.\top \end{array}$$

Let the ABox $\mathcal{A}$ contain $A(a)$, $A(b)$, $B(b)$, $P_1(a,b)$ and $P_2(b,c)$. The canonical model of $(\mathcal{T},\mathcal{A})$ is shown in Figure 3. The individual a in this canonical model has a single tree $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$. This tree has an edge from a to $aw_{\exists R}$ labelled R since $(a, aw_{\exists R})$ is in the extension of R in the canonical model. The edge in the reverse direction is labelled U, because of the second axiom above. The third axiom results in $aw_{\exists R}$ being in the extension of D, so the node $aw_{\exists R}$ is labelled D. The labelled nulls $aw_{\exists R}w_{\exists P_1}$ and $aw_{\exists R}w_{\exists P_2}$ are generated by the two axioms with D on the LHS.

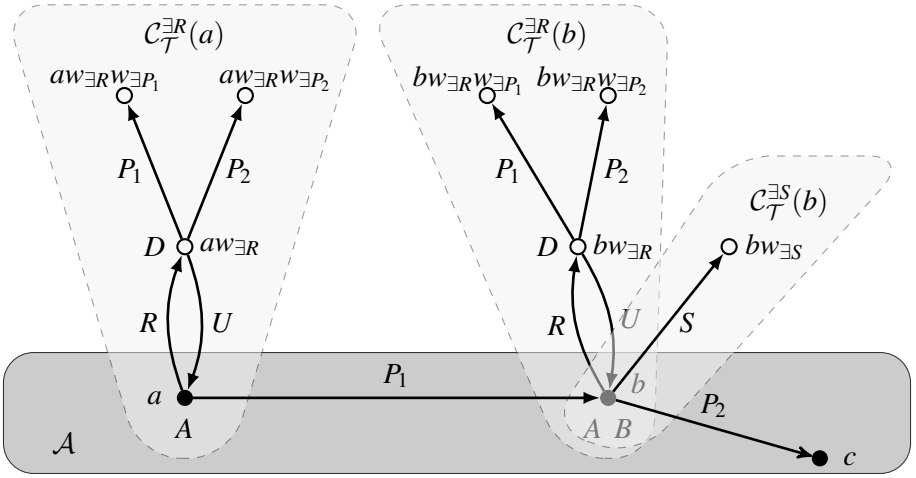

Figure 3: The canonical model $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ from Example 16.

The individual b has two trees, $\mathcal{C}_{\mathcal{T}}^{\exists R}(b)$ and $\mathcal{C}_{\mathcal{T}}^{\exists S}(b)$, which intersect only at their common root b. The second tree results from $B(b)$ being in $\mathcal{A}$ and the axiom $B \sqsubseteq \exists S.\top$.

The "join" of all the $\exists R$-generated $\mathcal{T}$-trees on a, for each $\exists R.\top$ appearing on the RHS of an axiom and each individual a, is called the *anonymous part* of the canonical model (Kontchakov & Zakharyaschev, 2014). The tree-witness rewriting approach finds all ways in which atoms in a CQ can be mapped to the anonymous part of the canonical model; hence, we need to compare the trees generated by an $\mathcal{ELHI}_h^{\ell in}$ TBox to those generated by a *QL* TBox. We present an example below before characterising the relationship. In what follows, given an $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$, we call the result of replacing each axiom of the form $\exists R.A \sqsubseteq B$ by $\exists R.\top \sqsubseteq B$ the *QL* TBox *corresponding to* $\mathcal{T}$, denoted $\mathcal{T}_{QL}$.

Example 17. Let $\mathcal{T}$ be the $\mathcal{ELHI}_h^{\ell in}$ TBox used in Example 15, comprising the axioms:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl} A & \sqsubseteq & \exists R.\top & C & \sqsubseteq & \exists U.\top & & & \\ R & \sqsubseteq & S^- & U & \sqsubseteq & V^- & E & \sqsubseteq & \exists W.\top \\ \exists S.B & \sqsubseteq & C & \exists V.D & \sqsubseteq & E & & & \end{array}$$

Assume, as before, that ABox $\mathcal{A}$ contains only the assertions $A(a)$, $A(b)$ and $B(a)$. TBox $\mathcal{T}_{QL}$ is identical to $\mathcal{T}$, except that B and D are replaced by $\top$ in the third and sixth axioms, respectively. The trees $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$ and $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(b)$ are shown in Figure 4. Removing the lighter nodes, labels and edges yields the trees $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ and $\mathcal{C}_{\mathcal{T}}^{\exists R}(b)$. In other words, removing the node labelled $aw_{\exists R}w_{\exists U}w_{\exists W}$, the edge labelled W and the label E from the node labelled $aw_{\exists R}w_{\exists U}$ in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$ yields $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$; removing the nodes labelled $bw_{\exists R}w_{\exists U}w_{\exists W}$ and $bw_{\exists R}w_{\exists U}$, the edges labelled W, U and V, and the label C from the node labelled $bw_{\exists R}$ in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(b)$ yields $\mathcal{C}_{\mathcal{T}}^{\exists R}(b)$.

For $aw_{\exists R}$ to be labelled with C in $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ requires that $(\mathcal{T},\mathcal{A}) \models B(a)$; hence, $bw_{\exists R}$ cannot be labelled with C in $\mathcal{C}_{\mathcal{T}}^{\exists R}(b)$. The same restriction does not apply in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$ and $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(b)$; hence, both $aw_{\exists R}$ and $bw_{\exists R}$ are labelled with C. For $aw_{\exists R}w_{\exists U}$ to be labelled with E in $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ would require that

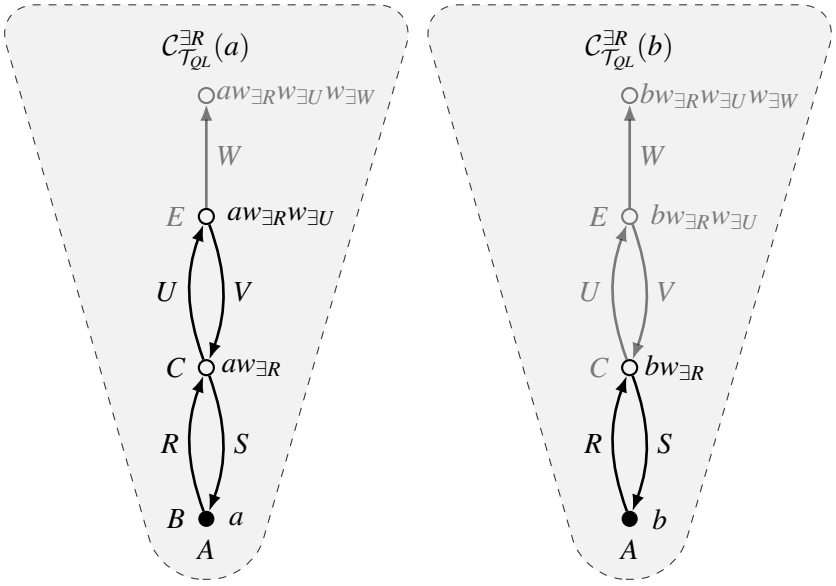

Figure 4: Trees $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$ and $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(b)$.

$(\mathcal{T},\mathcal{A}) \models D(aw_{\exists R})$. As pointed out in Example 15, this cannot be the case for an $\mathcal{ELHI}^{\ell in}_h$ TBox because $aw_{\exists R}$ would then be a shared labelled null.

In the following proposition, we use the notion of a $\mathcal{T}$-tree being a sub-tree of a $\mathcal{T}_{QL}$-tree. We say that tree $\mathcal{C}^{\exists R}_{\mathcal{T}}(a)$ is a *sub-tree* of tree $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$, denoted $\mathcal{C}^{\exists R}_{\mathcal{T}}(a) \subseteq \mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$, if each node and edge in $\mathcal{C}^{\exists R}_{\mathcal{T}}(a)$ appears in $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$, and each label on each node u and each edge (v,w) in $\mathcal{C}^{\exists R}_{\mathcal{T}}(a)$ appears on node u and edge (v,w), respectively, in $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$.

Proposition 7. *Let $(\mathcal{T},\mathcal{A})$ be an $\mathcal{ELHI}^{\ell in}_h$ KB, with $\mathcal{T}$ in normal form, and $\mathcal{T}_{QL}$ be the QL TBox corresponding to $\mathcal{T}$. For each role name R and individual a, $\mathcal{C}^{\exists R}_{\mathcal{T}}(a) \subseteq \mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$.*

Proof. Let $\mathcal{K} = (\mathcal{T},\mathcal{A})$ be an $\mathcal{ELHI}^{\ell in}_h$ KB and $\mathcal{T}_{QL}$ be the *QL* TBox corresponding to $\mathcal{T}$. Let $\mathcal{C}^{\exists R}_{\mathcal{T}}(a)$ and $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$ be the $\exists R$-generated $\mathcal{T}$-tree on a and the $\exists R$-generated $\mathcal{T}_{QL}$-trees on a, respectively, for role name R and individual a. The proof is by contradiction, considering (i) node labels, (ii) nodes, and (iii) edges.

(i) Assume that node u appears in both $\mathcal{C}^{\exists R}_{\mathcal{T}}(a)$ and $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$, and has label A in $\mathcal{C}^{\exists R}_{\mathcal{T}}(a)$ but not in $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$. If u is the root, then, by Corollary 3, it is labelled with each B such that $\mathcal{T} \models B \sqsubseteq \exists R.\top$. Clearly, $\mathcal{T} \models B \sqsubseteq \exists R.\top$ if and only if $\mathcal{T}_{QL} \models B \sqsubseteq \exists R.\top$, so u cannot be the root.

Node u must therefore be of the form $\sigma w_{\exists S}$, with an edge to u labelled with S from the node σ. This means that $\sigma w_{\exists R} \in A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. From Definition 15 we have either (a) $R^- \in L(\mathsf{NFA}_{A,\mathcal{T}})$, or (b) $R^-\rho \in L(\mathsf{NFA}_{A,\mathcal{T}})$ and $\mathcal{I}_{\mathcal{A}} \models \rho(a)$, for some non-empty ρ. Corollary 3 specifies that either (a) $(\mathcal{T},\mathcal{A}) \models \exists R^-.\top \sqsubseteq A$, or (b) $(\mathcal{T},\mathcal{A}) \models \exists R^-.\rho \sqsubseteq A$. For $\mathcal{T}_{QL}$, only case (a) applies. Definition 16 specifies that $aw_{\exists R} \in A^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, so $aw_{\exists R}$ will be labelled with A in $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$, a contradiction.

(ii) Assume that node u in $\mathcal{C}^{\exists R}_{\mathcal{T}}(a)$ is a node of smallest distance from the root which does not also appear in $\mathcal{C}^{\exists R}_{\mathcal{T}_{QL}}(a)$. Clearly, u cannot be the root node, so it must be of the form $\sigma w_{\exists S}$, with an

edge to u labelled with S from the node σ. The edge $(\sigma, \sigma w_{\exists S})$ can only be present if $A \sqsubseteq \exists S.\top \in \mathcal{T}$, for some A and node σ is labelled with A in $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$. By assumption, node σ appears in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$. From (i), we know it must be labelled with A. Axiom $A \sqsubseteq \exists S.\top$ is in $\mathcal{T}_{QL}$, so u must be in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$, a contradiction.

(iii) Assume that edge (u,v) in $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ is an edge of smallest distance from the root which does not also appear in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$. If (u,v) is $(\sigma, \sigma w_{\exists S})$ and labelled with S, we have already established in (ii) that v and the edge must be in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$. So assume that (u,v) is labelled with $U \neq S$. Since $(\sigma, \sigma w_{\exists S}) \in S^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ and $(\sigma, \sigma w_{\exists S}) \in U^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, it must be the case that $\mathcal{T} \models S \sqsubseteq U$ (from Definition 15). But then we also have that $\mathcal{T}_{QL} \models S \sqsubseteq U$ and $(\sigma, \sigma w_{\exists S}) \in S^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ (from Definition 15). So (u,v) labelled with U must be in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$. The only possibility left is that (u,v) is $(\sigma w_{\exists S}, \sigma)$ and labelled with some V. Hence, $\mathcal{T} \models S \sqsubseteq V^-$ (from Definition 15). But then we also have that $\mathcal{T}_{QL} \models S \sqsubseteq V^-$ and $(\sigma w_{\exists S}, \sigma) \in U^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ (from Definition 15). So (u,v) labelled with V must be in $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R}(a)$, a contradiction. □

The fact that each tree in the anonymous part of the canonical model of an $\mathcal{ELHI}_h^{lin}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is a sub-tree of a tree in the anonymous part of the canonical model of the QL KB $\mathcal{K}_{QL} = (\mathcal{T}_{QL}, \mathcal{A})$ means that we can apply the tree-witness rewriting approach of (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014) to our setting with almost no modification. The only modification is to take into account that atoms in a CQ need to be rewritten using the technique in Section 6.1. Nevertheless, for completeness, we present a brief overview of the tree-witness rewriting approach below, adapted from (Kontchakov & Zakharyaschev, 2014). The tree-witness rewriting is applied to a CQ and an $\mathcal{ELHI}_h^{lin}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{A}$ is assumed to be *H-complete* with respect to $\mathcal{T}$ (see the definition below). This is not a limitation because the tree-witness rewriting can itself be rewritten using the results of Section 6.1 so that it is correct for an arbitrary ABox, as shown in Proposition 8 below.

Definition 17 (H-completeness, cf. (Kontchakov & Zakharyaschev, 2014))**.** Let $\mathcal{T}$ be a (not necessarily flat) $\mathcal{ELHI}_h^{lin}$ TBox. An ABox $\mathcal{A}$ is said to be *H-complete with respect to* $\mathcal{T}$ if, for each concept name A and each role name P, we have:

- $A(a) \in \mathcal{A}$ if $\mathcal{I}_{\mathcal{A}} \models B(a)$ and $\mathcal{T} \models B \sqsubseteq A$, for (complex) concept B.
- $P(a,b) \in \mathcal{A}$ if either (i) $\mathcal{I}_{\mathcal{A}} \models R(a,b)$ and $\mathcal{T} \models R \sqsubseteq P$, or (ii) $\mathcal{I}_{\mathcal{A}} \models R(b,a)$ and $\mathcal{T} \models R \sqsubseteq P^-$, for some R.

Given an arbitrary ABox $\mathcal{A}$, its *H-completion with respect to* $\mathcal{T}$, denoted $\mathcal{A}_H$, is given by initially adding the assertions in $\mathcal{A}$ to $\mathcal{A}_H$ and then adding to $\mathcal{A}_H$ all assertions satisfying either of the conditions above.

Recall that, given a CQ $\mathbf{q}$ of arity n, we say that a C2RPQ $\mathbf{p}$ is a *perfect rewriting* of $\mathbf{q}$ with respect to an $\mathcal{ELHI}_h^{lin}$ TBox $\mathcal{T}$, if, for any ABox $\mathcal{A}$ and any n-tuple $\mathbf{a}$ from $\mathsf{ind}(A)$, the following holds:

$$(\mathcal{T}, \mathcal{A}) \models \mathbf{q}(\mathbf{a}) \text{ if and only if } \mathcal{I}_{\mathcal{A}} \models \mathbf{p}(\mathbf{a}).$$

When the above formula holds only if $\mathcal{A}$ is H-complete with respect to an $\mathcal{ELHI}_h^{lin}$ TBox $\mathcal{T}$, then we say that $\mathbf{p}$ is a *perfect rewriting* of $\mathbf{q}$ and $\mathcal{T}$ *over H-complete ABoxes*, cf. (Kontchakov & Zakharyaschev, 2014). Kontchakov and Zakharyaschev observe that, if an ABox A is H-complete

with respect to $\mathcal{T}$, then the ABox part of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$, i.e. the part that does not contain labelled nulls, coincides with $\mathcal{I}_\mathcal{A}$. Thus, if $\mathcal{T}$ is flat then $\mathbf{q}$ itself is clearly the perfect rewriting of $\mathbf{q}$ and T over H-complete ABoxes.

They also state the following proposition, slightly reworded here. We prove it below for $\mathcal{ELHI}_h^{\ell in}$ because our definition of $\mathcal{T}$-*ext* is different to theirs.

Proposition 8. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox and $\mathbf{q}$ be a CQ. If $\mathbf{p}$ is the perfect rewriting of $\mathbf{q}$ and $\mathcal{T}$ over H-complete ABoxes, then $\mathbf{p}_{\mathcal{T}\text{-}ext}$ is the perfect rewriting of $\mathbf{q}$ with respect to $\mathcal{T}$.*

Proof. Let $\mathbf{q}$ be a CQ and $\mathbf{p}$ be the perfect rewriting of $\mathbf{q}$ and $\mathcal{T}$ over H-complete ABoxes. Recall that $\mathbf{p}_{\mathcal{T}\text{-}ext}$ is formed by replacing each atom $A(u_1,u_2)$ in $\mathbf{p}$, where A is a concept name, by $A_{\mathcal{T}\text{-}ext}(u_1,u_2)$, and each atom $P(u_1,u_2)$ in $\mathbf{p}$, where P is a role name, by $P_{\mathcal{T}\text{-}ext}(u_1,u_2)$. Let $\mathcal{A}$ be an arbitrary ABox, with $\mathcal{A}_H$ its H-completion.

Assume atom $A(u_1,u_2)$ in $\mathbf{p}$, where A is a concept name, matches assertion $A(a) \in \mathcal{A}_H$. Then either $A(a) \in \mathcal{A}$ or $\mathcal{I}_\mathcal{A} \models B(a)$ and $\mathcal{T} \models B \sqsubseteq A$, for (complex) concept B. Corollary 3 shows that $\mathcal{T} \models B \sqsubseteq A$ if and only if $B \in L(\mathsf{NFA}_{A,\mathcal{T}})$. Recall that $A_{\mathcal{T}\text{-}ext}(u_1,u_2) = \alpha(u_1,u_2)$, where α is a regular expression denoting $L(\mathsf{NFA}_{A,\mathcal{T}})$. Hence, $A_{\mathcal{T}\text{-}ext}(u_1,u_2)$ in $\mathbf{p}_{\mathcal{T}\text{-}ext}$ will match assertion $A(a) \in \mathcal{A}$.

Now assume that $P(u_1,u_2)$ in $\mathbf{p}$, for P a role name, matches assertion $P(a,b) \in \mathcal{A}_H$. Then either $P(a,b) \in \mathcal{A}$ or (i) $\mathcal{I}_\mathcal{A} \models R(a,b)$ and $\mathcal{T} \models R \sqsubseteq P$, or (ii) $\mathcal{I}_\mathcal{A} \models R(b,a)$ and $\mathcal{T} \models R \sqsubseteq P^-$, for some R. Recall that $P_{\mathcal{T}\text{-}ext}(u_1,u_2) = (R_1 \mid \cdots \mid R_n)(u_1,u_2)$, where each R_i is a role name such that $\mathcal{T} \models R_i \sqsubseteq P$ or an inverse role name such that $\mathcal{T} \models R_i \sqsubseteq P^-$. Hence, $P_{\mathcal{T}\text{-}ext}(u_1,u_2)$ in $\mathbf{p}_{\mathcal{T}\text{-}ext}$ will match assertion $P(a,b) \in \mathcal{A}$. □

Consider a CQ $\mathbf{q}$ and a $\mathcal{ELHI}_h^{\ell in}$ knowledge base $(\mathcal{T},\mathcal{A})$. Let $\mathbf{a}$ be a tuple of individuals from $\mathsf{ind}(A)$ and h be a homomorphism from $\mathbf{q}(\mathbf{a})$ to $\mathcal{C}_{\mathcal{T},\mathcal{A}}$. A subset of atoms in $\mathbf{q}$ are mapped by h to the ABox part of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$, with the remainder mapped to trees in the anonymous part of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$. The tree-witness rewriting of $\mathbf{q}$ and $\mathcal{T}$ considers all possible ways in which the atoms of $\mathbf{q}$ can be mapped to $\mathcal{C}_{\mathcal{T},\mathcal{A}}$. As a result, it produces a union of CQs, where each CQ in the rewriting has atoms which are guaranteed to match the anonymous part of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ removed.

Example 18. Consider the $\mathcal{ELHI}_h^{\ell in}$ (but not *QL*) TBox $\mathcal{T}$ with the axioms

$$\begin{aligned} \mathit{Instructor} &\sqsubseteq \exists \mathit{teaches}.\top \\ \mathit{teaches} &\sqsubseteq \mathit{taughtBy}^- \\ \exists \mathit{taughtBy}.\mathit{Prof} &\sqsubseteq \mathit{Course} \\ \mathit{Course} &\sqsubseteq \exists \mathit{hasTitle}.\top \end{aligned}$$

and the CQ

$$q(x) \leftarrow \mathit{teaches}(x,y), \mathit{hasTitle}(y,z).$$

In what follows, and particularly in Figure 5, we will use the abbreviations I for $\mathit{Instructor}$, T for $\mathit{teaches}$, TB for $\mathit{taughtBy}$, P for Prof, C for Course and HT for $\mathit{hasTitle}$.

Let $\mathcal{A}$ be an ABox such that $a,b \in \mathsf{ind}(\mathcal{A})$, where a is an $\mathit{Instructor}$ and b is a Course, i.e., we have $a \in I^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ and $b \in C^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. Then $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ must also contain the trees $\mathcal{C}_\mathcal{T}^{\exists T}(a)$ and $\mathcal{C}_\mathcal{T}^{\exists HT}(b)$. If a is also a Prof, then the two trees are as shown in Figure 5.

Any homomorphism h from the atoms of $q(x)$ to $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ must map the answer variable x to $\mathsf{ind}(\mathcal{A})$ in order for $h(x)$ to be a certain answer. Assuming that there are individuals c, d and e in $\mathsf{ind}(\mathcal{A})$

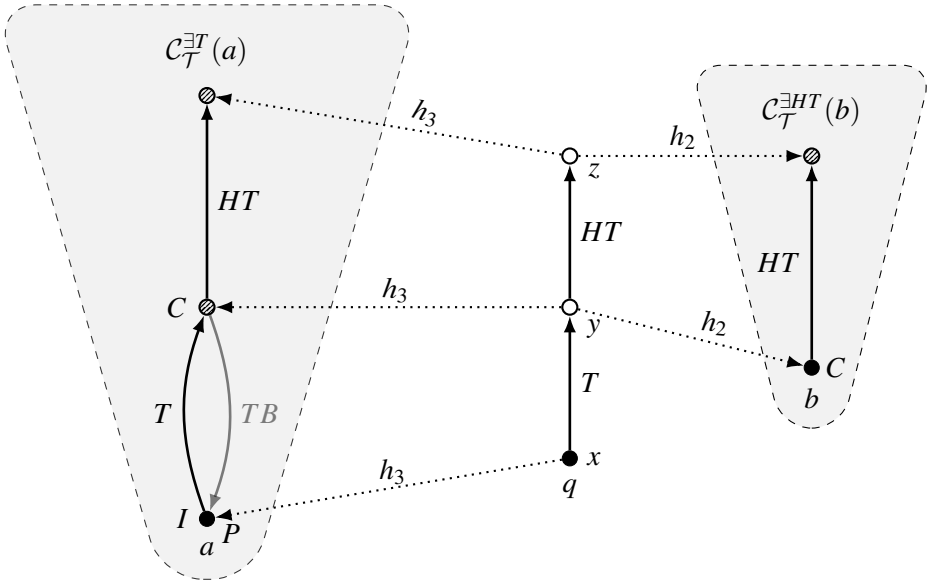

Figure 5: Homomorphisms from subsets of q to $\mathcal{C}_{\mathcal{T}}^{\exists T}(a)$ and $\mathcal{C}_{\mathcal{T}}^{\exists HT}(b)$.

such that c teaches d and d has title e, then there is a homomorphism h_1 which maps x, y and z to c, d and e, respectively, leaving the original query unchanged. A second homomorphism, shown as h_2 in Figure 5, maps x and y to $\mathsf{ind}(\mathcal{A})$, and if $h_2(y) = b$ is in $C^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, maps the last atom of $q(x)$, namely $HT(y,z)$, to $\mathcal{C}_{\mathcal{T}}^{\exists HT}(b)$. The third homomorphism, shown as h_3 in Figure 5, maps x to an individual a which is in both $I^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$ and $P^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$. As a result, both atoms of $q(x)$ can be mapped to $\mathcal{C}_{\mathcal{T}}^{\exists T}(a)$. These three homomorphisms give rise to the tree-witness rewriting of $q(x)$ and $\mathcal{T}$ over H-complete ABoxes as the union of the following three conjunctive queries:

$$q_{1_{tw}}(x) \leftarrow teaches(x,y), hasTitle(y,z)$$
$$q_{2_{tw}}(x) \leftarrow teaches(x,y), Course(y)$$
$$q_{3_{tw}}(x) \leftarrow Teacher(x), Prof(x)$$

Each tree witness is closely related to a homomorphism such as those in the above example, in that it results in some subset of the atoms in a query being able to be mapped to a tree in the anonymous part of the canonical model. We now include the formal definition of tree witnesses because they are needed in the proofs of the complexity results in Section 7. We refer the reader who is interested in the details of how tree witnesses are used to generate the tree-witness rewriting to (Kikot et al., 2012; Kontchakov & Zakharyaschev, 2014).

Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form and q a CQ with at least one existentially quantified variable in its body. Consider a pair $\mathbf{t} = (\mathbf{t}_r, \mathbf{t}_i)$ of disjoint sets of variables appearing in q, where

- $\mathbf{t}_i$ is non-empty and contains only existentially quantified variables, and
- $\mathbf{t}_r$ can be empty or can contain answer variables and existentially quantified variables.

Let

$$q_{\mathbf{t}} = \{S(\boldsymbol{z}) \mid S(\boldsymbol{z}) \text{ is an atom in the body of } q,\ \boldsymbol{z} \subseteq \mathbf{t}_r \cup \mathbf{t}_i \text{ and } \boldsymbol{z} \not\subseteq \mathbf{t}_r\}.$$

Then $\mathbf{t}$ is a *tree witness* for q and $\mathcal{T}$ generated by $\exists R.\top$ if the following two conditions are satisfied:

(a) there exists a homomorphism h from $q_{\mathbf{t}}$ to $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$, for some a, such that $\mathbf{t}_r = \{z \mid h(z) = a\}$ and $\mathbf{t}_i$ contains the remaining variables in $q_{\mathbf{t}}$, and

(b) $q_{\mathbf{t}}$ is a minimal subset of q such that, for any $y \in \mathbf{t}_i$, every atom in q containing y belongs to $q_{\mathbf{t}}$.

The terms in $\mathbf{t}_r$ (if any) are called the *roots* of $\mathbf{t}$ and the (existentially quantified) variables in $\mathbf{t}_i$ the *interior* of $\mathbf{t}$, cf. (Kontchakov & Zakharyaschev, 2014).

Example 19. There are two tree witnesses for the query q and TBox $\mathcal{T}$ of Example 18, $\mathbf{t}^1 = (\mathbf{t}_r^1, \mathbf{t}_i^1)$ and $\mathbf{t}^2 = (\mathbf{t}_r^2, \mathbf{t}_i^2)$, with $\mathbf{t}_r^1 = \{y\}$, $\mathbf{t}_i^1 = \{z\}$, $\mathbf{t}_r^2 = \{x\}$ and $\mathbf{t}_i^2 = \{y,z\}$. The corresponding sets of atoms of q for each are $q_{\mathbf{t}^1} = \{hasTitle(y,z)\}$ and $q_{\mathbf{t}^2} = \{teaches(x,y), hasTitle(y,z)\}$. Tree witness $\mathbf{t}^1$ is generated by $\exists hasTitle.\top$ since homomorphism h_1 in Figure 5 maps $\mathbf{t}_r^1 = \{y\}$ to the root of $\mathcal{C}_{\mathcal{T}}^{\exists HT}(b)$. Tree witness $\mathbf{t}^2$ is generated by $\exists teaches.\top$ since homomorphism h_2 maps $\mathbf{t}_r^2 = \{x\}$ to the root of $\mathcal{C}_{\mathcal{T}}^{\exists T}(a)$.

Given a CQ $\mathbf{q}$ and $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$, the union of CQs produced by the tree-witness rewriting of $\mathbf{q}$ and $\mathcal{T}$ over H-complete ABoxes is denoted by $\mathbf{q}_{tw}$.

Proposition 9. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form and $\mathbf{q}$ a CQ. For any H-complete ABox $\mathcal{A}$ and any tuple $\mathbf{a}$ of individuals from $\mathsf{ind}(\mathcal{A})$, we have $\mathcal{C}_{\mathcal{T},\mathcal{A}} \models \mathbf{q}(\mathbf{a})$ if and only if $\mathcal{I}_{\mathcal{A}} \models \mathbf{q}_{tw}(\mathbf{a})$.*

Proof. Let $\mathcal{T}_{QL}$ be the *QL* TBox corresponding to $\mathcal{T}$. Proposition 27 in (Kikot et al., 2012) shows that $\mathcal{C}_{\mathcal{T}_{QL},\mathcal{A}} \models \mathbf{q}(\mathbf{a})$ if and only if $\mathcal{I}_{\mathcal{A}} \models \mathbf{q}_{tw}(\mathbf{a})$, for any H-complete ABox $\mathcal{A}$ and any tuple $\mathbf{a}$ of individuals from $\mathsf{ind}(\mathcal{A})$. Since $\mathcal{A}$ is H-complete, the proof involves only the structure of the anonymous part of $\mathcal{C}_{\mathcal{T}_{QL},\mathcal{A}}$, which is equal to the "join" of the trees $\mathcal{C}_{\mathcal{T}_{QL}}^{\exists R(a)}$ for each individual a and $\exists R$ in $\mathcal{T}_{QL}$ (Kontchakov & Zakharyaschev, 2014). Proposition 7 shows that tree in $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ is a sub-tree of a tree in $\mathcal{C}_{\mathcal{T}_{QL},\mathcal{A}}$. Therefore, the proof in (Kikot et al., 2012) carries through for $\mathcal{ELHI}_h^{\ell in}$ TBox $\mathcal{T}$. □

Let $\mathbf{q}$ be a CQ and $\mathbf{q}_{tw}$ be the tree-witness rewriting of $\mathbf{q}$. We denote by $\mathbf{q}_{tw_{\mathcal{T}\text{-}ext}}$ the result of replacing each CQ $\mathbf{p}$ in $\mathbf{q}_{tw}$ by the C2RPQ $\mathbf{p}_{\mathcal{T}\text{-}ext}$ (see Definition 13). Thus, $\mathbf{q}_{tw_{\mathcal{T}\text{-}ext}}$ is a UC2RPQ.

Theorem 5. *Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox in normal form and $\mathbf{q}$ a CQ. For any ABox $\mathcal{A}$ and any tuple $\mathbf{a}$ of individuals from $\mathsf{ind}(\mathcal{A})$, we have $\mathcal{C}_{\mathcal{T},\mathcal{A}} \models \mathbf{q}(\mathbf{a})$ if and only if $\mathcal{I}_{\mathcal{A}} \models \mathbf{q}_{tw\mathcal{T}\text{-}ext}(\mathbf{a})$.*

Proof. The proof follows directly from Propositions 8 and 9. □

7. Complexity Analysis

In this section, we establish results on the computational complexity of the problem of query answering for $\mathcal{ELHI}_h^{\ell in}$ knowledge bases. For CQ answering, we show that the problem is NLOGSPACE-complete with respect to data complexity and is NP-complete with respect to combined complexity; for IQ answering, we show that the problem is NLOGSPACE-complete with respect to data complexity and is in PTIME with respect to combined complexity. We present our complexity results in terms of query answering problems (as is common practice (Calvanese et al., 2013)), although technically the results refer to the decision versions of the problems.

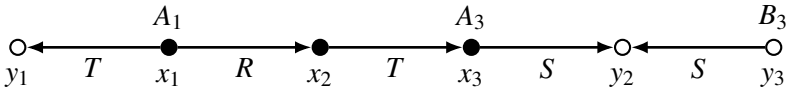

Figure 6: Conjunctive query as a labelled directed multigraph, with answer variables filled in black.

Here we view a CQ as a labelled directed multigraph $\langle N, E, \psi, \ell_N, \ell_E \rangle$, with N a set of nodes, E a set of edges, $\psi : E \to N \times N$ an *incidence function* assigning to each edge an ordered pair of nodes, $\ell_N : N \to \mathsf{A}$ a function assigning labels to nodes from the set of concept names, and $\ell_E : E \to \mathsf{R}$ a function assigning labels to edges from the set of role names. For a given CQ q, the *graph of* q is composed as follows:

1. the set of nodes N is the set of terms in q;
2. for each atom in the body of q of the form $A(x)$, there is a label assignment $x \to A \in \ell_N$; and
3. for each atom of the form $R(x_1, x_2)$, there is an edge $(x_1, x_2) \in E$ and a label assignment $(x_1, x_2) \to R \in \ell_E$.

For example, the graph of the query

$$q(x_1, x_2, x_3) \leftarrow A_1(x_1), A_3(x_3), B_3(y_3), T(x_1, y_1), R(x_1, x_2), T(x_2, x_3), S(x_3, y_2), S(y_3, y_2)$$

is illustrated in Figure 6, where node identifiers are shown below each node and label assignments for nodes are shown above each node. Also, in this section we adopt the notion of a *polytree*, which is simply a directed graph with the property that ignoring the directions on edges and then merging multiple edges between nodes yields an undirected graph with no cycles (this is a generalisation of the definition in (Dasgupta, 1999)).

Definition 18. Given a CQ q and a set of terms $\boldsymbol{t}$ in q, we say that q is *polytree-transformable* with respect to $\boldsymbol{t}$ if there is a homomorphism h from the terms of q to terms of q, such that:

1. for each $t \in \boldsymbol{t}$, $h(t) = \mathsf{root}_h$, where root_h denotes what we term the *root of* h;
2. for each term t in q such that $t \notin \boldsymbol{t}$, we have that $h(t) \neq \mathsf{root}_h$; and
3. the graph of $h(q)$, i.e., the query resulting from applying h to each term in q, is a polytree.

Example 20. Consider the CQ q

$$q(x) \leftarrow R(x, y_1), R(x, y_2), U(y_2, x), S(y_1, y_3), T(y_1, y_4), T(y_2, y_4), R(y_5, y_2)$$

whose graph is not a polytree (see Figure 7). Query q is polytree-transformable with respect to $\{x, y_5\}$ via the homomorphism $h = \{x \to x, y_1 \to y_2, y_3 \to y_3, y_4 \to y_4, y_5 \to x\}$ where $\mathsf{root}_h = x$. The transformation results in the query $h(q)$:

$$h(q)(x) \leftarrow R(x, y_2), U(y_2, x), S(y_2, y_3), T(y_2, y_4).$$

Figure 7 shows the graphs of q and $h(q)$.

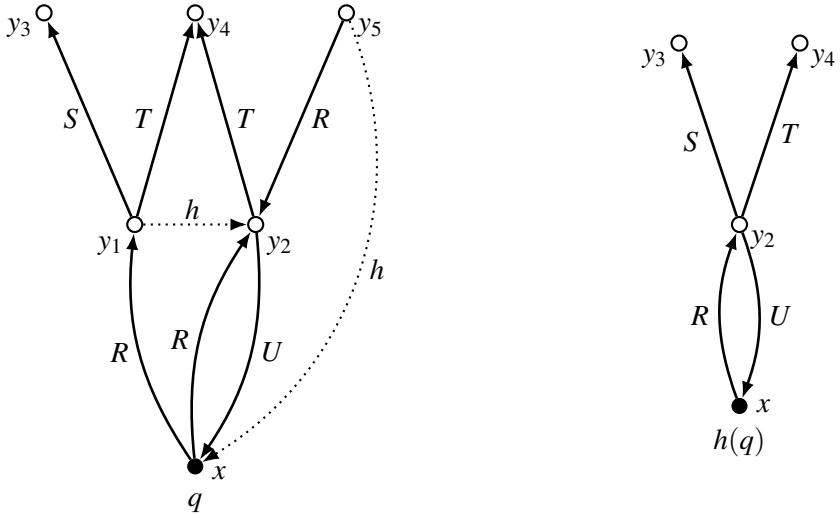

Figure 7: The graphs of q and $h(q)$ of Example 20

Definition 19. Consider a query q such that the graph of q is a polytree, a term root in q, a constant a and a TBox $\mathcal{T}$. We say that q *tree-maps* $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ *on* root if there is a homomorphism h from the atoms of q to $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ such that h is an injective function and $h(t) = a$ only if $t = \mathsf{root}$.

Proposition 10. *Consider a CQ q, a set of terms $\mathbf{t}$ in q, a TBox $\mathcal{T}$ and a pair $\mathbf{t} = (\mathbf{t}_r, \mathbf{t}_i)$ of disjoint sets of terms in q, where $\mathbf{t}_i$ is non-empty and contains only existentially quantified variables, and $\mathbf{t}_r$ contains the remaining terms of q. Then, there exists a homomorphism h from the atoms of q to $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$, for some a, such that $\mathbf{t}_r = \{z \mid h(z) = a\}$ if and only if q is polytree-transformable with respect to $\mathbf{t}_r$ via a homomorphism $\overline{h}$, and $\overline{h}(q)$ tree-maps $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ on $\mathsf{root}_{\overline{h}}$.*

Proof. (Only if) From the definition of h, we know that $h(z) = a$ if and only if $z \in \mathbf{t}_r$, and therefore conditions (1) and (2) in Definition 18 are satisfied. From Theorem 4 we know that, for every ABox $\mathcal{A}$, each labelled null in $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ is also a w-path σ. From Definition 15, it is clear that each edge in $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ must be from a w-path σ to a w-path $\sigma w_{\exists S}$, for some role name S, or vice versa. Hence, the only cycles that can exist in $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ must be between pairs of adjacent nodes. It follows that $h(q)$ is a polytree and condition (3) in Definition 18 is also satisfied. Now, we know that q is polytree-transformable with respect to $\mathbf{t}_r$ via $h(q)$ with $\mathsf{root}_h = a$. It follows that $h(q)$ tree-maps $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ on a via the identity function i, since h is a homomorphism from the atoms of q to $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ and $i(t) = a$ only if $t = \mathsf{root}_h = a$. Also, the identity function i is injective by definition, so the claim follows.

(If) We know that q is polytree-transformable with respect to $\mathbf{t}_r$ via a homomorphism $\overline{h}$, and $\overline{h}(q)$ tree-maps $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$ on $\mathsf{root}_{\overline{h}}$ via some homomorphism h^*. Now, set $h = \overline{h} \circ h^*$. Then, h is a homomorphism from the atoms of q to $\mathcal{C}_{\mathcal{T}}^{\exists R}(a)$. From Definition 19 we know $h^*(t) = a$ only if $t = \mathsf{root}_{\overline{h}}$. Also, from Definition 19 we know that *i)* for each $t_r \in \mathbf{t}_r$, $\overline{h}(t_r) = \mathsf{root}_{\overline{h}}$, and that *ii)* for each term in q, t_i, such that $t_i \notin \mathbf{t}_r$, we have that $\overline{h}(t_i) \neq \mathsf{root}_h$. Thus, $\mathbf{t}_r = \{z \mid h(z) = a\}$ and the claim follows. □

Definition 20. Let $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox, a an arbitrary individual, and $\exists R.\top$ a complex concept appearing on the RHS of an axiom in $\mathcal{T}$. We define the *length* of a labelled null (i.e., w-path)

appearing in $\mathcal{C}_{\mathcal{T}}^{\exists R(a)}$ as follows: the length of $aw_{\exists R}$ is one, while the length of $aw_{\exists R}w_{\exists T_1}\cdots w_{\exists T_i}$, $i \geqslant 1$, is $i+1$. We then define $\mathcal{C}_{\mathcal{T}_n}^{\exists R(a)}$ as the maximum subset of $\mathcal{C}_{\mathcal{T}}^{\exists R(a)}$ such that each null in $\mathcal{C}_{\mathcal{T}_n}^{\exists R(a)}$ is of length at most n.

The above definition is used in the following lemma to limit the size of trees to which a tree map from query q can be found.

Lemma 6. *Let q be a polytree query with root root, $\mathcal{T}$ be an $\mathcal{ELHI}_h^{\ell in}$ TBox and a be an individual. If the maximum length of any path in the graph of q starting from root is n, then q tree-maps $\mathcal{C}_{\mathcal{T}}^{\exists R(a)}$ on root only if q tree-maps $\mathcal{C}_{\mathcal{T}_n}^{\exists R(a)}$ on root.*

Proof. If q tree-maps $\mathcal{C}_{\mathcal{T}}^{\exists R(a)}$ on root then there is an injective homomorphism h from the atoms of q to $\mathcal{C}_{\mathcal{T}}^{\exists R(a)}$ such that $h(t) = a$ only if $t = \mathsf{root}$. Any path in the graph of q must be mapped by h to a path of the same length in $\mathcal{C}_{\mathcal{T}}^{\exists R(a)}$ since h is injective. We know that the maximum length of any path in the graph of q starting from root is n; hence h maps this to a path of maximum length n starting from a in $\mathcal{C}_{\mathcal{T}}^{\exists R(a)}$. We conclude that q tree-maps $\mathcal{C}_{\mathcal{T}_n}^{\exists R(a)}$ on root. □

The following theorem resolves the complexity of answering IQs and CQs with respect to data complexity.

Theorem 6. *Answering IQs and CQs on $\mathcal{ELHI}_h^{\ell in}$ knowledge bases is* NLOGSPACE*-complete with respect to data complexity.*

Proof. It is known that the problem of answering IQs in $\mathcal{ELH}^{\ell in}$ is NLOGSPACE-hard with respect to data complexity (Calvanese et al., 2013), so the same holds for $\mathcal{ELHI}_h^{\ell in}$, which is a proper extension of $\mathcal{ELH}^{\ell in}$. Membership in NLOGSPACE for IQ answering follows from the fact that we can rewrite instance queries to 2RPQs, and answering 2RPQs is in NLOGSPACE (Barceló Baeza, 2013).

For CQs, the upper bound follows from the tree-witness rewriting algorithm which generates a perfect rewriting of $\mathbf{q}$ for $\mathcal{T}$ as a UC2RPQ query, and the fact that the problem of UC2RPQ answering is NLOGSPACE-complete with respect to data complexity (Barceló Baeza, 2013). For both IQs and CQs, the rewriting algorithm relies solely on the query and the TBox. Since both the query and the TBox are considered fixed in the definition of data complexity, producing the rewriting is done in constant time; therefore, the rewriting algorithm does not use more than logarithmic space. □

In terms of combined complexity, the following theorem classifies the problem of answering CQs, while Theorem 8 deals with the problem of answering IQs.

Theorem 7. *Answering CQs on $\mathcal{ELHI}_h^{\ell in}$ knowledge bases is* NP*-complete with respect to combined complexity.*

Proof. For the upper bound for CQs, we present the following non-deterministic version of the rewriting algorithm for a given CQ $\mathbf{q}$ and TBox $\mathcal{T}$:

1. guess a tree witness $q_{\mathbf{t}} = \{S(\mathbf{z}) \mid S(\mathbf{z})$ is an atom in the body of q, $\mathbf{z} \subseteq \mathbf{t}_r \cup \mathbf{t}_i$ and $\mathbf{z} \not\subseteq \mathbf{t}_r\}$;
2. guess a homomorphism $\overline{h}$ from the atoms of $q_{\mathbf{t}}$ to the atoms of $q_{\mathbf{t}}$;

3. check that each term $t \in \mathbf{t}_r$ maps to the same term, i.e., $\mathsf{root}_{\overline{h}}$;
4. check if the graph of $\overline{h}(q_{\mathbf{t}})$ is a polytree via a graph traversal (in polynomial time with respect to the size of $q_{\mathbf{t}}$);
5. guess a role name R and generate $\mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$, where $a = \mathsf{root}_{\overline{h}}$ if $\mathsf{root}_{\overline{h}}$ is a individual name (or for an arbitrary individual a otherwise), and m is the length of the longest path in $\overline{h}(q_{\mathbf{t}})$ starting from $\mathsf{root}_{\overline{h}}$; the cost of generating $\mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$ is bounded by $m \times |\mathcal{T}|$;
6. check if $\overline{h}(q_{\mathbf{t}})$ tree-maps $\mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$ on $\mathsf{root}_{\overline{h}}$:
   (a) guess an injective function h from the terms in $\overline{h}(q_{\mathbf{t}})$ to the terms in $\mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$;
   (b) check if $h(\overline{h}(q_{\mathbf{t}})) \subseteq \mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$;
7. check if $q_{\mathbf{t}}$ is a minimal subset of $\mathbf{q}$ such that, for any $y \in \mathbf{t}_i$, every atom in $\mathbf{q}$ containing y belongs to $q_{\mathbf{t}}$;
8. rewrite $\mathbf{q}$ to $\mathbf{q}'$ accordingly and generate $\mathbf{q}'_{\mathcal{T}\text{-}ext}$;
9. check if $\mathbf{q}'_{\mathcal{T}\text{-}ext}$ is true when evaluated on the ABox.

Since $\mathbf{q}'_{\mathcal{T}\text{-}ext}$ is a C2RPQ and the data complexity of answering C2RPQs over a plain database is in NLOGSPACE, the problem of answering $\mathbf{q}$ is in NP. This NP bound is optimal, since the combined complexity of CQ answering is already NP-hard for DL-Lite$_{\mathcal{R}}$ (Calvanese et al., 2007). □

Theorem 8. *Answering IQs on $\mathcal{ELHI}_h^{\ell in}$ knowledge bases is in* PTIME *with respect to combined complexity.*

Proof. For a concept instance query we can build the NFA in polynomial time, and then answer the path query resulting from the rewriting over the ABox. For a role instance query, the rewriting can be constructed in linear time and the resulting query is a 2RPQ.

We also need to consider the cost of satisfiability of NIs, which is done (as before) by checking the knowledge base against a set of Boolean CQs of linear size with respect to the TBox. By definition, these Boolean CQs contain at most two atoms and two variables. Thus, we can generate the rewriting of each BCQ in polynomial time by substituting the non-deterministic guesses of the algorithm presented in the proof of Theorem 7 with the following deterministic steps:

1. generate all the possible tree-witnesses $q_{\mathbf{t}}$, of which are at most three;
2. generate the homomorphisms $\overline{h}$ from the atoms of $q_{\mathbf{t}}$ to the atoms of $q_{\mathbf{t}}$; the number of homomorphisms is bounded by 2^2;
3. generate the set of all $\mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$, whose size is bounded by $|\mathcal{T}|$; note that m is at most 1;
4. check if $\overline{h}(q_{\mathbf{t}})$ tree-maps $\mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$ on $\mathsf{root}_{\overline{h}}$; by definition, this is possible only if the term in $\overline{h}(q_{\mathbf{t}})$ that is not $\mathsf{root}_{\overline{h}}$ is mapped via h to a term in $\mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$ that is not a, and then $h(\overline{h}(q_{\mathbf{t}})) \subseteq \mathcal{C}_{\mathcal{T}_m}^{\exists R(a)}$; since m is at most 1, then the number of injective functions h is bounded by $|\mathcal{T}|$.

At this point, we only need to check each rewritten query against the ABox alone, thus not using more than logarithmic space for each query, since they are Boolean C2RPQs. □

8. Conclusions

In this paper we have introduced the ontology language $\mathcal{ELHI}_h^{\ell in}$ (harmless linear $\mathcal{ELH}$) which generalises both of the ontology languages DL-Lite$_\mathcal{R}$ and linear $\mathcal{ELH}$. We have shown that our language allows for both qualified existential quantification on the left-hand sides of axioms as well as inverse roles, but only if their interaction is deemed harmless. We have shown that instance queries (queries with a single atom in their body) are rewritable with respect to $\mathcal{ELHI}_h^{\ell in}$ knowledge bases using 2RPQs as the target language. The query rewriting algorithm makes use of non-deterministic finite-state automata. Following on from that, we proposed a query rewriting algorithm for answering conjunctive queries under $\mathcal{ELHI}_h^{\ell in}$ knowledge bases, with UC2RPQs as the target language. This algorithm utilises the tree-witness rewriting of (Kontchakov & Zakharyaschev, 2014) along with the above NFA-based rewriting technique. Since UC2RPQs can be straightforwardly expressed in SPARQL 1.1 by means of property paths, our approach is directly applicable to real-world querying settings.

In terms of computational complexity, we have proved that CQ answering with respect to $\mathcal{ELHI}_h^{\ell in}$ ontologies is in NLOGSPACE in terms of data complexity and in NP in terms of combined complexity; we have shown that these bounds are tight. In addition, we have shown that instance query answering with respect to $\mathcal{ELHI}_h^{\ell in}$ ontologies is NLOGSPACE-complete with respect to data complexity and in PTIME with respect to combined complexity. Our contribution in this paper is therefore an ontology formalism that combines highly tractable query answering with greater expressive power than the ontology languages on which it is based. DL-Lite$_\mathcal{R}$ is used in real-world application domains such as energy, healthcare, government, education and innovation, transport and infrastructure (Xiao, Ding, Cogrel, & Calvanese, 2019); therefore our new language $\mathcal{ELHI}_h^{\ell in}$ will allow knowledge of greater expressivity to be modelled in these, and possibly other, domains while retaining tractable query answering.

Future work includes an empirical evaluation of our rewriting algorithms on real-world data sets, and investigation of other ontology languages that may lie within the scope of tractability of CQ answering. In particular, it would be interesting to investigate more general ways of introducing inverse roles into our language.

Acknowledgments

The authors wish to thank Roman Kontchakov and Michael Zakharyaschev for their advice, and the reviewers for their very detailed and constructive comments.

The work reported in this article was done while Mirko Dimartino and Andrea Calì were at Birkbeck, University of London.

References

Artale, A., Calvanese, D., Kontchakov, R., & Zakharyaschev, M. (2009). The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, *36*(1), 1–69.

Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th International Joint Conference on Artificial Intelligence*, pp. 364–369.

Baader, F., Horrocks, I., Lutz, C., & Sattler, U. (2017). *An Introduction to Description Logic*. Cambridge University Press.

Baader, F., & Nutt, W. (2007). Basic description logics. In *Description Logic Handbook, Second Edition*, pp. 47–104.

Barceló Baeza, P. (2013). Querying graph databases. In *Proc. of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 175–188.

Berry, G., & Sethi, R. (1986). From regular expressions to deterministic automata. *Theoretical Computer Science*, *48*, 117–126.

Bienvenu, M., Hansen, P., Lutz, C., & Wolter, F. (2016). First order-rewritability and containment of conjunctive queries in Horn description logics. In *Proc. of the 25th International Joint Conference on Artificial Intelligence*, pp. 965–971.

Bienvenu, M., Lutz, C., & Wolter, F. (2013). First-order rewritability of atomic queries in Horn description logics. In *Proc. of the 23rd International Joint Conference on Artificial Intelligence*, pp. 754–760.

Bienvenu, M., Ortiz, M., & Simkus, M. (2015). Regular path queries in lightweight description logics: Complexity and algorithms. *Journal of Artificial Intelligence Research*, *53*, 315–374.

Calì, A., Gottlob, G., & Lukasiewicz, T. (2012). A general datalog-based framework for tractable query answering over ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, *14*, 57–83.

Calì, A., Lembo, D., & Rosati, R. (2003). Query rewriting and answering under constraints in data integration systems. In *Proc. of the 18th International Joint Conference on Artificial Intelligence*, pp. 16–21.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The DL-lite family. *Journal of Automated Reasoning*, *39*(3), 385–429.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2013). Data complexity of query answering in description logics. *Artificial Intelligence*, *195*, 335–360.

Calvanese, D., De Giacomo, G., Lenzerini, M., & Vardi, M. Y. (2000). View-based query processing for regular path queries with inverse. In *Proc. of the 19th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems*, pp. 58–66.

Dasgupta, S. (1999). Learning polytrees. In *Proc. of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 134–141. Morgan Kaufmann Publishers Inc.

Dimartino, M. M., Calì, A., Poulovassilis, A., & Wood, P. T. (2016). Query rewriting under linear $\mathcal{EL}$ knowledge bases. In *Proc. of the 10th International Conference on Web Reasoning and Rule Systems*, pp. 61–76.

Gottlob, G., Orsi, G., & Pieris, A. (2011). Ontological queries: Rewriting and optimization. In *Proc. of the 27th IEEE International Conference on Data Engineering*, pp. 2–13.

Hansen, P., Lutz, C., Seylan, I., & Wolter, F. (2014). Query rewriting under $\mathcal{EL}$ TBoxes: Efficient algorithms. In *Proc. of the 27th International Workshop on Description Logics*, pp. 197–208.

Hansen, P., Lutz, C., Seylan, I., & Wolter, F. (2015). Efficient query rewriting in the description logic $\mathcal{EL}$ and beyond. In *Proc. of the 24th International Joint Conference on Artificial Intelligence*, pp. 3034–3040.

Harris, S., & Seaborne, A. (2013). SPARQL 1.1 Query Language, W3C Recommendation 21 March 2013. Available at `https://www.w3.org/TR/sparql11-query/`.

Kazakov, Y. (2009). Consequence-driven reasoning for Horn ontologies. In *Proc. of the 21st International Joint Conference on Artificial Intelligence*, pp. 2040–2045.

Kikot, S., Kontchakov, R., & Zakharyaschev, M. (2012). Conjunctive query answering with OWL 2 QL. In *Proc. of the 13th International Conference on Knowledge Representation and Reasoning*, pp. 275–285.

Kontchakov, R., & Zakharyaschev, M. (2014). An introduction to description logics and query rewriting. In *Reasoning Web. Reasoning on the Web in the Big Data Era*, pp. 195–244. Springer.

Lembo, D., Santarelli, V., & Savo, D. F. (2013). Graph-based ontology classification in OWL 2 QL. In *Proc. of the 10th Extended Semantic Web Conference*, pp. 320–334. Springer Berlin Heidelberg.

Lenzerini, M. (2002). Data integration: a theoretical perspective. In *Proc. of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 233–246, New York, NY, USA. ACM.

Lutz, C., & Sabellek, L. (2022). A complete classification of the complexity and rewritability of ontology-mediated queries based on the description logic $\mathcal{EL}$. *Artificial Intelligence*, *308*.

Pérez-Urbina, H., Horrocks, I., & Motik, B. (2009). Efficient query answering for OWL 2. In *Proc. of the 8th International Conference on the Semantic Web*, pp. 489–504. Springer Berlin Heidelberg.

Pérez-Urbina, H., Motik, B., & Horrocks, I. (2008). Rewriting conjunctive queries over description logic knowledge bases. In *Proc. of the Third International Workshop on Semantics in Data and Knowledge Bases*, pp. 199–214. Springer.

Pérez-Urbina, H., Motik, B., & Horrocks, I. (2010). Tractable query answering and rewriting under description logic constraints. *Journal of Applied Logic*, *8*(2), 186–209.

Rosati, R. (2007). On conjunctive query answering in $\mathcal{EL}$. In *Proc. 20th International Workshop on Description Logics*, pp. 451–458.

Schaerf, A. (1993). On the complexity of the instance checking problem in concept languages with existential quantification. *Journal of Intelligent Information Systems*, *2*(3), 265–278.

Trivela, D., Stoilos, G., Chortaras, A., & Stamou, G. (2015). Optimising resolution-based rewriting algorithms for OWL ontologies. *Journal of Web Semantics*, *33*, 30–49.

Vu, Q. H. (2008). Subsumption in the description logic $\mathcal{ELHI}f_{\nabla^+}$ with respect to general TBoxes. Master's thesis, TU Dresden.

Xiao, G., Ding, L., Cogrel, B., & Calvanese, D. (2019). Virtual Knowledge Graphs: An Overview of Systems and Use Cases. *Data Intelligence*, *1*(3), 201–223.