Conjunctive Query Answering with OWL 2 QL

Stanislav Kikot, Roman Kontchakov and Michael Zakharyaschev
Department of Computer Science and Information Systems
Birkbeck, University of London, U.K.
{kikot,roman,michael}@dcs.bbk.ac.uk

Abstract
We present a novel rewriting technique for conjunctive query answering over OWL 2 QL ontologies. In general, the obtained rewritings are not necessarily correct and can be of exponential size in the length of the query. We argue, however, that in most, if not all, practical cases the rewritings are correct and of polynomial size. Moreover, we prove some sufficient conditions, imposed on queries and ontologies, that guarantee correctness and succinctness. We also support our claim by experimental results.

Introduction
OWL 2 QL, one of three profiles of the Web Ontology Language OWL 2, was designed with the aim of supporting ontology-based data access (OBDA). The key idea is that data, ‘stored in a standard relational database management system (RDBMS), can be queried through an OWL 2 QL ontology via a simple rewriting mechanism, i.e., by rewriting the query into an SQL query that is then answered by the RDBMS, without any changes to the data’ ([www.w3.org/TR/owl2-profiles]). The rewritability property ensures, in particular, that the data complexity of answering queries over OWL 2 QL ontologies matches the complexity of database query answering, which is in AC0.

It has been observed, however, that the available ‘rewriting mechanisms’ for OWL 2 QL (Calvanese et al. 2007a; Pérez-Urbina, Motik, and Horrocks 2009; Rosati and Almatelli 2010; Chortaras, Trivela, and Stamou 2011; Gottlob, Orsi, and Pieris 2011) are actually not so ‘simple.’ In fact, the rewritten queries are often too long to be executed by modern RDBMSs, and the question whether ‘short’ rewritings exist has attracted considerable attention over the last two years. For example, Kikot, Kontchakov, and Zakharyaschev (2011) showed that no polynomial algorithm can construct a ‘pure rewriting’ of a conjunctive query (CQ) q over an OWL 2 QL ontology T. Here by a pure rewriting we mean an any first-order (FO) rewriting with the same signature (predicates and constants) as q and T, possibly with equality. On the other hand, Gottlob and Schwentick (2011) gave a polynomial-time (‘impure’) rewriting using additional constants and predicates. Optimised (but still exponential) pure rewritings to nonrecursive Datalog were suggested by Rosati and Almatelli (2010) and Gottlob, Orsi, and Pieris (2011). The combined approach of Kontchakov et al. (2010) was developed for OWL 2 QL without role inclusions; it uses a simple polynomial rewriting over the data expanded by applying the ontology axioms and introducing a small number of new individuals.

The diversity of approaches to query rewriting prompts another question: what is the type/shape/size of rewritings we should aim at to make OBDA with OWL 2 QL efficient? When trying to answer this question, we should bear in mind that (i) the OBDA paradigm relies on the proven efficiency of RDBMSs, but (ii) database query answering is not tractable in the size of queries (PSpace-complete for FO queries and NP-complete for CQs). High efficiency of RDBMSs in practice appears to indicate only that answering real-world queries over real-world databases turns out to be tractable. As rewritings can turn a standard query to something ‘out of this world,’ a first rule of thumb could be as follows: the rewritten query should look similar to the original one. In this respect, as Gottlob and Schwentick (2011) remark, their polynomial rewriting is of rather ‘theoretical nature’ (it uses the extra constants, predicates and existentially quantified variables to encode, by making nondeterministic guesses, a relevant part of the chase, aka the canonical model; see the discussion in Conclusions for more details).

The aim of this paper is to investigate what causes exponentially long pure rewritings of CQs over OWL 2 QL ontologies and to check experimentally whether those ‘bad guys’ occur in real-world queries and ontologies. As a result of this analysis, we suggest some short (polynomial-size) rewritings that cover most, if not all, practical cases.

We think of a CQ as a labelled directed multigraph. For example, the query ‘find x1, x2, x3 for which

\[ \exists y_1, y_2, y_3 \ (A_1(x_1) \land A_3(x_3) \land B_3(y_3) \land T(x_1, y_1) \land R(x_1, x_2) \land T(x_2, x_3) \land S(x_3, y_2) \land S(y_3, y_2)) \]

holds’ can be represented as the graph

\[ y_1 \rightarrow A_1 \rightarrow x_1 \rightarrow R \rightarrow x_2 \rightarrow T \rightarrow x_3 \rightarrow S \rightarrow y_2 \rightarrow S \rightarrow y_3 \]

\[ B_3 \]

Copyright © 2012, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
To answer a CQ \(q(\vec{x})\) over data \(A\) and ontology \(T\), we seek homomorphisms from \(q\) to the structure, called the canonical model and obtained by expanding \(A\) (extensional data) with knowledge in \(T\) (intensional data). Thus, we have to find possible cuts of \(q\) into a number of pieces: some of them—say, of type (A)—are mapped to individuals in \(A\), while the others—of type (B)—may only have some of their terms (‘roots’) in \(A\), whereas the remaining part is implied by the knowledge in \(T\). For example, the query above can be cut, for a suitable \(T\), into 3 pieces of which only the middle one is of type (A), and the right piece has two roots \(x_3, y_3\):

If \(q(\vec{x})\) does not have existentially quantified variables then the whole \(q\) forms the only possible cut of type (A), because the answer variables \(\vec{x}\) must be mapped to individuals in \(A\). If \(T\) does not contain role inclusion axioms then, as shown by Kontchakov et al. (2010; 2011), every root determines a unique piece of type (B). However, in general, \(q\) may have exponentially many pieces of type (B). One can encode the intuition above as a pure positive existential rewriting \(q_{\epsilon}\), which is, roughly speaking, a disjunction over all possible cuts of \(q\), and so is exponential in \(|q|\) in the worst case.

A slightly different approach to checking possible cuts of \(q\) is to consider, for every edge in \(q\), whether it belongs to an (A) piece, or generates a (B) piece itself, or lies inside the (B) piece generated by some other edge. This ‘local’ view gives another rewriting, a conjunction of disjunctions \(q_{\epsilon}\), which may still be exponential as the same edge may generate exponentially many distinct (B) pieces. (Two (B) pieces are different if their domains or roots are different.) Moreover, the new rewriting is not necessarily correct because some (B) pieces may not be realised together in the canonical model: we call such pieces conflicting.

We analyse—both theoretically and experimentally—conditions under which the number of (B) pieces is polynomial and they are not in conflict with each other. In particular, we develop techniques to ensure that rewritings are not affected by conflicting pieces. For example, one simple—but impure—approach involves a single fresh constant to represent all intensional objects in the canonical model (labelled nulls in the chase), but no extra variables.

We show that the number of (B) pieces generated by one edge in the query \(q\) is largely determined by a sophisticated interaction between role inclusions and inverse roles in the ontology \(T\), which can produce canonical models with very complex intensional parts. We give a sufficient condition (on \(q\) and \(T\)) which guarantees that one edge in \(q\) can generate at most one piece of type (B) (though the number of ways this piece can be matched in the canonical model can still be exponential). This leads to our shortest pure rewriting, \(q_{\epsilon}\), which can be constructed in polynomial time, but is correct only if \(q\) and \(T\) satisfy the sufficient condition, which can also be checked in polynomial time. Trivial examples where this condition holds are ontologies without inverse roles, ontologies without role inclusions and qualified existential quantification, or those without positive occurrences of existential quantifiers. Our experiments with a number of standard OWL 2 QL ontologies and queries demonstrate that the sufficient conditions always apply, the queries normally contain very few pieces of type (B), if any, and moreover, these pieces are never in conflict. Thus, in practice the rewritings \(q_{\epsilon}\) and \(q_{\epsilon}\) are both short and correct.

Omitted proofs can be found in the full version of the paper at www.dcs.bbk.ac.uk/~kikot

### OWL 2 QL

The language of OWL 2 QL contains individual names \(a_i\), concept names \(A_i\), and role names \(P_i\) (\(i \geq 1\)). Roles \(R\), basic concepts \(B\), and concepts \(C\) are defined by the grammar:

\[
\begin{align*}
R &::= P_i | P_i^-, \\
B &::= \bot | A_i | \exists R, \\
C &::= B | \exists R.B
\end{align*}
\]

(Note that concepts of the form \(\exists R.B\) can only occur in the right-hand side of concept inclusions in OWL 2 QL. An inclusion \(B' \subseteq \exists R.B\) can be regarded as an abbreviation for three inclusions: \(B' \subseteq \exists R.B, \exists R.B \subseteq B, \text{ and } R_B \subseteq R\), where \(R_B\) is a fresh role name.) An ABox, an is a finite set of assertions of the form \(A_i(a_i)\) and \(P_j(a_i, a_j)\). \(T\) and \(\mathcal{A}\) together constitute the knowledge base (KB) \(K = (T, \mathcal{A})\). The semantics for OWL 2 QL is defined in the usual way based on interpretations \(I = (\mathcal{X}, \mathcal{E})\); consult (Baader et al. 2003) for details. The set of individual names in \(\mathcal{A}\) will be denoted by ind(\(\mathcal{A}\)). For concepts or roles \(E_1\) and \(E_2\), we write \(E_1 \subseteq E_2\) if \(T \models E_1 \subseteq E_2\); and we set \([E] = \{E' \mid E \subseteq E' \land E' \subseteq \mathcal{T} E\}\).

A conjunctive query (CQ) \(q(\vec{x})\) is a first-order formula \(\exists \vec{y}. \varphi(\vec{x}, \vec{y})\), where \(\varphi\) is constructed, using \(\land\), from the form \(A_k(t_1)\) and \(P_k(t_1, t_2)\), where each \(t_i\) is a term (an individual or a variable from \(\vec{x}\) or \(\vec{y}\)). Variables in \(\vec{x}\) are called answer variables, and those in \(\vec{y}\) bound variables. A tuple \(\vec{a} \subseteq \text{ind}(\mathcal{A})\) is a certain answer to \(q(\vec{x})\) over \(\mathcal{K} = (T, \mathcal{A})\), if \(I \models q(\vec{a})\) for all models \(I\) of \(\mathcal{K}\); in this case we write \(\mathcal{K} \models q(\vec{a})\). To simplify notation, we will often identify \(q\) with the set of its atoms and use \(P^-(t, t) \in q\) as a synonym of \(P(t, t') \in q\); term(\(q\)) is the set of terms in \(q\). We call \(q\) tree-shaped if its primal graph (term(\(q\)), \(\{\{t, t\} \mid R(t, t') \in q\}\)) is a tree.

**Remark 1** Although the official OWL 2 QL contains the concept \(\top\) (for the whole domain), we do not consider it here as \(\top\) makes OBDA with OWL 2 QL domain dependent (Abiteboul, Hull, and Vianu 1995): take, for example, the query \(A(x)\) over the ontology \(\{x \subseteq A\}\). (But we shall use \(\top\) as an auxiliary symbol as it is convenient to regard \(\exists R\) as an abbreviation for \(\exists R.\top\)). To simplify presentation, we omit data properties and (ir)reflexivity constraints for roles.

Query answering over OWL 2 QL KBs is based on the fact that, for any consistent KB \(\mathcal{K} = (T, \mathcal{A})\), there is an interpretation \(\mathcal{C}_\mathcal{K}\) such that, for all CQs \(q(\vec{x})\) and \(\vec{a} \subseteq \text{ind}(\mathcal{A})\),
we have $\mathcal{K} \models q[\bar{a}]$ iff $C_{\mathcal{K}} \models q[\bar{a}]$. The interpretation $C_{\mathcal{K}}$, called the canonical model of $\mathcal{K}$, can be constructed as follows. For each pair $[R],[B]$ with $\exists R.B$ in $\mathcal{T}$ (recall that $\exists R.T$ is another way of writing $\exists R$), we introduce a fresh symbol $w_{[R,B]}$ and call it the witness for $\exists R.B$. We write $\mathcal{K} \models C(w_{[R,B]})$ if $\exists R^- \sqsubseteq C$ or $B \sqsubseteq C$. Define a generating relation, $\rightarrow$, on the set of these witnesses together with $\mathrm{ind}(A)$ by taking:

$\rightarrow$ a $\leadsto$ $w_{[R,B]}$ if $a \in \mathrm{ind}(A)$, $\exists R.B(a)$ and there is no $b \in \mathrm{ind}(A)$ with $\mathcal{K} \models R(a,b) \land B(b)$;

$\rightarrow$ $w_{[R',B']} \leadsto w_{[R,B]}$ if $u \leadsto w_{[R',B']}$ for some $u$, $\exists R$ and $[B]$ are $\sqsubseteq_T$-minimal such that $\mathcal{K} \models \exists R.B(u)$ and it is not the case that $R' \sqsubseteq R$ and $\mathcal{K} \models B(u)$.

If $a \leadsto w_{[R_i,B_i]} \leadsto \cdots \leadsto w_{[R_n,B_n]}$, $n \geq 0$, then we say that $a$ generates the path $w_{[R_1,B_1]} \cdot \cdots \cdot w_{[R_n,B_n]}$. Denote by $\text{path}_C(a)$ the set of paths generated by $a$, and by $\text{tail}(\pi)$ the last element in $\pi \in \text{path}_C(a)$. $C_{\mathcal{K}}$ is defined by taking:

$$\Delta^{C_{\mathcal{K}}} = \bigcup_{a \in \mathrm{ind}(A)} \text{path}_C(a), \quad a^{C_{\mathcal{K}}} = a, \text{ for } a \in \mathrm{ind}(A),$$

$$A^{C_{\mathcal{K}}} = \{ \pi \in \Delta^{C_{\mathcal{K}}} \mid \mathcal{K} \models A(\text{tail}(\pi)) \},$$

$$P^{C_{\mathcal{K}}} = \{ (a,b) \in \mathrm{ind}(A) \times \mathrm{ind}(A) \mid \mathcal{K} \models P(a,b) \} \cup \{ (\pi \cdot w_{[R,B]}), \pi \cdot \text{tail}(\pi) \leadsto w_{[R,B]} \mid R \sqsubseteq_T P \} \cup \{ (\pi, w_{[R,B]}), \pi \leadsto w_{[R,B]} \mid R \sqsubseteq_T P^- \}.$$
a homomorphism \( f \) (with domain \( \text{dom} \ f \)) from the query
\[
q_f = \{ (s, s') \in q \mid s, s' \in \text{dom} f \} \cup \\
\{ A(s) \in q \mid s \in \text{dom} f \setminus f^{-1}(r_f) \}
\]
to the RB-subtree of \( C_T \), for some \( \exists R.B \), with root \( r_f = a_{RB} \) such that the following conditions hold:

(i) \( \text{dom} \ f \) is the smallest set containing \( t, t' \) and such that if \( s \in \text{dom} f \setminus f^{-1}(r_f) \) and \( S(s, s') \in q \) then \( s' \in \text{dom} f \),

(ii) if \( f(t) = r_f \) and if \( s \in \text{dom} f \setminus f^{-1}(r_f) \) then \( s \) is bound.

Note that this notion of tree witness is different from the one used for the description logic DL-Lite\(^\text{ext}_{\text{horn}} \) by Kontchakov et al. (2010), where the structure of the canonical models ensured uniqueness of a tree witness if it existed. In Example 3, there are two tree witnesses, \( f \) and \( q \), for \( (x, y) \). As we shall see below, there may be exponentially many of them.

Returning back to case (B), we can say now that there must exist a tree witness \( f \) for \( (z, z') \) such that \( a(z) \) satisfies all \( A(s) \in q \) with \( s \in f^{-1}(r_f) \). Case (B') is symmetric, and in case (O) there must exist \( R(t, t') \in q \) for which (B) holds and \( P(z, z') \) is ‘covered’ by a tree witness for \( (t, t') \) (as we assumed that \( a(t) \in \text{ind}(A) \), for some \( t \in \text{term}(q) \)).

This analysis suggests the following idea. Given a CQ \( q \) and KB \( K = (T, \mathcal{A}) \), we guess pairs of adjacent terms \((t, t')\) in \( q \) that will be mapped to edges of the tree part of \( C_K \) starting from \( \text{ind}(A) \) (as in case (B) above) and, for each such pair \((t, t')\), we guess a tree witness for \((t, t')\). The part of the query that is not covered by the chosen tree witnesses will be mapped to \( \text{ind}(A) \) (as in case (A)). The query representing these guesses is then evaluated over \( I_A \). If \( q \) has no answer variables, then we also have to take account of the case where the whole \( q \) is mapped into the tree part of \( C_K \). Unfortunately, this idea cannot be implemented in a straightforward way, as shown by the following example.

**Example 4** Let \( K = (\mathcal{T}, \{ A(a) \}) \), where \( \mathcal{T} = \{ A \sqsubseteq \exists R, A \sqsubseteq \exists R^- \} \). Consider the query
\[
q(x_1, x_4) = \{ R(x_1, y_2), R(y_3, y_2), R(y_3, x_4) \}
\]
shown in the picture below alongside \( C_K \).

We obviously have a tree witness \( f \) for \((x_1, y_2)\) such that \( \text{dom} f = \{ x_1, y_2 \} \) and \( f^{-1}(r_f) = \{ x_1, y_2 \} \), and also a tree witness \( g \) for \((x_4, y_3)\) with \( \text{dom} g = \{ x_4, y_3 \} \) and \( g^{-1}(r_g) = \{ x_4, y_2 \} \). Although these tree witnesses cover the whole query \( q \), they are only ‘realised’ in \( C_K \) under conflicting maps: \( f \) sends \( x_1, y_2 \) to \( a \) and \( y_2 \) to \( a \cdot w[R \leftarrow \gamma] \), while \( g \) sends \( x_4, y_2 \) to \( a \) and \( y_3 \) to \( a \cdot w[R^- \leftarrow \gamma] \); in fact, \( K \not\models q(a, a) \).

This example motivates the following definitions. Let \( \Theta \) be the set of tree witnesses for \( q \) and \( T \). We say that \( f, g \in \Theta \) are compatible if \( \text{dom} f \cap \text{dom} g \subseteq f^{-1}(r_f) \cap g^{-1}(r_g) \).

If \( f \) and \( g \) are incompatible and neither \( \text{dom} f \subseteq \text{dom} g \) nor \( \text{dom} g \subseteq \text{dom} f \), then we call \( f \) and \( g \) conflicting. In Example 4, \( \text{dom} f \cap \text{dom} g = \{ y_2, y_3 \} \), and so \( f \) and \( g \) are conflicting.

We are now in a position to formulate a rewriting based on the idea discussed above. Given a tree witness \( f \in \Theta \) for \((t, t')\) with \( r_f = a_{RB} \), we first define a tree-witness formula \( \text{tw}_f \) for \( f \) by taking
\[
\text{tw}_f = \text{ext}_{\exists R.B}(t) \land \bigwedge_{s \in \text{dom} f \setminus f^{-1}(r_f)} (s = t) \land \bigwedge_{A(s) \in q} \text{ext}_A(s).
\]

A (possibly empty) subset \( \Xi \subseteq \Theta \) is called consistent if all pairs of tree witnesses in \( \Xi \) are compatible. Now, assuming that \( q \) is a list of all bound variables in \( q(x) \), we set
\[
q_e(x) = \text{detached}_q \lor \bigvee_{\Xi \subseteq \Theta} \text{detached}_q \lor \bigvee_{\Xi \subseteq \Theta} (\text{tw}_f \land \bigwedge_{A(s) \in q} \text{ext}_A(s) \land \bigwedge_{\Xi \subseteq \Theta} \text{ext}_f(s, s'))
\]
where

\( - \text{detached}_q = \bot \) if \( q \) has at least one answer variable; otherwise \( \text{detached}_q \) is a disjunction of the sentences \( \exists x \text{ext}_{\exists R.B}(x) \) such that there is a homomorphism from \( q \) to the RB-subtree of \( C_T \).

Clearly, \( q_e \) is a positive existential formula built from the atoms occurring in \( q \) and \( T \) together with equality unifying certain terms; it contains the same variables and constants as the original CQ \( q \). Moreover, if we consider the predicates \( \text{ext}_f \) for concepts and roles as primitive, then \( q_e \) is a union of conjunctive queries (UCQ), where each subquery can be thought of as the result of folding the respective pieces of \( q \) into the tree witness formulas.

**Theorem 5** For every ABox \( A \) and every \( a \in \text{ind}(A) \), we have \( (T, A) \models q[a] \) if \( I_A \models q_e[a] \).

We illustrate the rewriting \( q_e \) by a simple example.

**Example 6** Suppose that \( q(x) = \{ R_i(x, y_i) \mid i \leq n \} \) and \( T = \{ A_i \sqsubseteq \exists R_i \mid i \leq n \} \). Each pair \((x, y_i)\) gives rise to one tree witness \( f_i \) with \( \text{tw}_{f_i} = A_i(x) \lor \exists y R_i(x, y) \) and \( q_e = \bigvee_{N \subseteq [0, n]} \exists y (A_i(x) \land \bigwedge_{j \in N} R_j(x, y_j)) \).

As all other known pure rewritings for OWL 2 QL, \( q_e \) is of exponential size: \( O((n_{R \cdot Q} + 1)^{|q|} \cdot |T| \cdot |q|^2) \), where \( n_{R \cdot Q} \) is the maximum number of distinct tree witness formulas \( \text{tw}_f \) for tree witnesses containing a pair \((t, t')\) of adjacent terms in \( q \). Two recent results may help to shed some light on whether this exponential blowup is unavoidable in OWL 2 QL.

One of them shows that no polynomial-time algorithm can construct pure rewritings for CQs over OWL 2 QL ontologies, unless \( P = \text{NP} \) (Kikot, Kontchakov, and Zakharyaschev 2011, Theorem 2). The idea of the proof is as follows. First, we encode any CNF \( \chi = \bigwedge_{i=1}^m D_i \) over propositional variables \( p_1, \ldots, p_n \) as an OWL 2 QL TBox,
\(T_k\), containing the axioms, for \(1 \leq i \leq n, 1 \leq j \leq m\) and \(k = 0, 1,\)
\[A_{i-1} \subseteq P^{-}X_{i}^{k}, \quad X_{i}^{k} \subseteq A_i, \quad C_j \subseteq P.C_j, \quad X_{i}^{0} \subseteq P.C_j \quad \text{if} \quad p_i \in D_j, \quad X_{i}^{1} \subseteq P.C_j \quad \text{if} \quad p_i \in D_j,\]
and consider the CQ
\[q(y_0) = A_0(y_0) \land \bigwedge_{i=1}^{n} P(y_i, y_{i-1}) \land A_n(y_n) \land \bigwedge_{j=1}^{m} (P(y_j, z_j^0) \land \bigwedge_{n=1}^{n} P(z_{j-1}^n, z_j^0) \land C_j(z_j)).\]

Now, suppose \(q'(y_0)\) is a rewriting of \(q(y_0)\) and \(T_k\) that does not use any constants. Consider the ABox \(A = \{ A_0(a) \}\). It is not hard to see that \((I_A, A) \models q'[a] \iff \chi\) is satisfiable. On the other hand, checking whether \(I_A \models q'[a]\) can be done in polynomial time in \(|q'|\) because the domain of \(I_A\) is a singleton. Thus, constructing the rewriting \(q'\) must be at least as hard as deciding satisfiability of \(\chi\). (See the Conclusions for a further discussion and results.)

The argument above does not go through if databases are assumed to have two special constants, say \(0\) and \(1\), which can be employed in rewrites \(q'\). Indeed, as shown by Gottlob and Schwentick (2011), using \(0\) and \(1\) and fresh predicates of arity \(O(\log(|q| \cdot |T|))\), one can construct a nonrecursive Datalog rewriting \(q'\) for any given CQ \(q\) and OWL 2 QL ontology \(T\) in polynomial time. Intuitively, \(q'\) uses the extra resources to encode a part of the canonical model, which is enough to provide all certain answers to \(q\), to guess a map from \(q\) into this part and then check whether it is a homomorphism. As argued in the introduction, RDBMSs do not appear to be best suitable for tasks of that sort, although thorough experiments are required to confirm or refute this claim.

Very often, however, there are other ways to construct polynomial rewrites. Let us assume for a moment that the following condition holds:

**Example 8** The exponential query \(q_e\) in Example 6 reduces to polynomial \(q_e = \exists y \bigwedge_{i \leq n} \left( \bigwedge_{t \text{ and } t' \text{ adjacent}} R_i(x, y_i) \lor \text{tw}_{f_i} \right)\).

A good illuminative example of a CQ with exponentially many tree witness formulas is \(q(y_0)\) over the TBox \(T_k\) constructed above for a CNF \(\chi\). One can show that the pairs \((z_i^1, z_{i-1}^0)\) give rise to exponentially different tree witness formulas \(\text{tw}_f\) for a suitable \(\chi\).

To illustrate, consider the CQ \(q(y_0)\) for \(n = m = 2\) and suppose that \(\chi\) is such that the \(P^{-}X_{1}^{1}\)-subtree of \(CT_k\) contains the fragment as shown in the picture below.

We can construct a tree witness \(f\) for \((z_1^1, z_0^1)\) by taking \(f(z_1^1) = r_0, f(z_0^1) = r_1, f(y_2) = r_2\), after which we have three different options for defining \(f(z_2^1)\) and \(f(z_2^0)\): go back to \(r_0\), go to \(r_1\) and then take the \(C_2\)-branch, or take the \(C_2\)-branch starting from \(r_2\). The last two tree witnesses give the same tree witness formula, which is different from that given by the first tree witness. Imagine now some large \(n\) and \(m\). It is this fact that makes it ‘hard’ to construct a rewriting for \(q(y_0)\) and \(T_k\).

The TBox \(T_k\) and CQ \(q(y_0)\) involve a very complex interplay between role inclusions (or concepts of the form \(\exists R.C\)) and inverse roles, which appears to be rather artificial compared to how roles are used in real-world ontologies. The experiments to be reported later on in the paper demonstrate that real-world ontologies and queries generate very few tree witnesses, which are never in conflict, and so the rewriting \(q_e\) is both short and correct.

In the next section we analyse conflicting tree witnesses in more detail.

**Conflicting Tree Witnesses**

One simple way to tackle the problem of conflicting tree witnesses, identified in Example 4, would be to introduce a fresh constant symbol, say \(\nu\), representing all the non-ABox elements of the canonical models. We then use the following variant of the tree witness formula \(\text{tw}_f\) for \((t, t')\):

\[\text{tw}_f' = \text{tw}_f \land \bigwedge_{s \in \text{dom } f(t, t')}(s = \nu).\]

We denote the resulting ‘impure’ rewriting by \(q_e'\). Given an ABox \(A\), denote by \(I_A\) the interpretation \(I_A\) extended with a new domain element (interpreting) \(\nu\). Thus, \(\nu\) is not involved in the interpretation of any predicate in \(I_A\), and so, of any predicate \(\text{ext}_E\).
Example 9 In the context of Example 4, \( tw_q \) would contain the conjuncts \( ext_\exists R_...T(x_1) \) and \( (y_2 = x_1) \), which cannot be satisfied in \( I_A \) at the same time as the conjunct \( (y_2 = \nu) \) of \( tw_q' \). Thus, \( I_A \notmodels tw_q \).

The following result shows that \( q^+ \) is a correct rewriting over this interpretation.

**Theorem 10** For any ABox \( A \) and any \( d \subseteq \text{ind}(A) \), we have \((T,A) \models q[d] \iff I_A \models q^+ [d] \).

The rewriting \( q^+ \) can be viewed as a step in the direction of the combined approach (Lutz, Toman, and Wolter 2009; Kontchakov et al., 2010), though without expanding the ABox by applying the TBox axioms. Roughly, in both cases the RDBMS has to guess whether a bound variable is mapped to \( \text{ind}(A) \) or to the tree part of the canonical model.

There is another way to 'suppress conflicts' in \( q^+ \) for the majority of practical cases, while keeping the resulting rewriting 'pure.' For a CQ \( q \), let \( q^+ \) be the rewriting obtained from \( q^+ \) by replacing every \( tw_f \) in it with the following formula \( tw_f^+ \):

\[
\text{(41)} \quad tw_f^+ = tw_f \land \bigwedge_{f,g \text{ conflicting}} (q_{\phi[f]}^+)^+, \\
\]

where \( q_{\phi[f]}^+ \) denotes the restriction of the query \( q \) to the set \( \text{dom} q \setminus \text{dom} f \setminus f^{-1}(r_f) \) (i.e., all terms in \( q \) that are not non-root terms in \( f \)) and \((q_{\phi[f]}^+) ^+ \), in turn, is the rewriting (as defined above) of the query \( q_{\phi[f]}^+ \) in which all the variables in \( f^{-1}(r_f) \cup g^{-1}(r_g) \) are regarded to be answer variables.

Example 11 In the context of Example 4, the rewriting \( q^+ \) is constructed in two steps as follows: first, we obtain a representation of \( q^+ \) as the following formula:

\[
\exists y_2, y_3 \left( (R(x_1, y_2) \lor tw_f^+) \land (R(y_3, y_2) \lor tw_g^+) \land (R(y_3, x_4) \lor tw_g^+) \right), \\
\]

where

\[
\text{tw}_f^+ = \text{ext}_\exists R_\exists R(x_1) \land (x_1 = y_3) \land (R(y_3, x_4)), \\
\text{tw}_g^+ = \text{ext}_\exists R_\exists R(x_4) \land (x_4 = y_2) \land (R(x_1, y_2)).
\]

with \( R(y_3, x_4) \) and \( R(x_1, y_2) \) being two new queries, which have only answer variables. Then, the rewritings of these two queries coincide with the queries themselves: \( R(y_3, x_4) \) and \( R(x_1, y_2) \), respectively. So, for example, the modified tree witness formula \( tw_f^+ \) for \( f \) contains the conjunct \( R(y_3, x_4) \), which cannot be satisfied in the interpretation \( I_A \) with \( A = \{ A(a) \} \).

**Theorem 12** Suppose that \( q \) and \( T \) satisfy the condition (conf) for every \( f \in \Theta \), if there are \( g, h \in \Theta \) such that the pairs \( f, g \) and \( f, h \) are both conflicting, then either \( \text{dom} q \subseteq \text{dom} f \) or \( \text{dom} q \supseteq \text{dom} g \) or \( \text{dom} q \subseteq \text{dom} h \).

Then, for every ABox \( A \) and every \( d \subseteq \text{ind}(A) \), we have \((T,A) \models q[d] \iff I_A \models q^+ [d] \).

We do not know any cases where \( q^+ \) is not correct. It is to be noted, however, that the rewriting \( q^+ \) can be of exponential size even if every pair \((t, t') \) in \( q \) gives rise to at most one tree witness.

**Example 13** Given a word \( R_1 \ldots R_m \), over roles, define the CQ

\[
q^{R_1 \ldots R_m}(x_0, x_m) = \{ R_i(x_{i-1}, x_i) \mid 1 \leq i \leq m \}.
\]

Let \( \sigma_n \) be the following sequence of words of roles:

\[
\sigma_1 = T_1 T_1, \quad \sigma_{n+1} = S_n T_{n+1} T_{n+1} S_n \sigma_n S_n, \text{ for } n \geq 1.
\]

Now, consider the CQs \( q_n = q^{\sigma_n}(x_0, x_n) \) and TBoxes \( T_n = \{ A_n \subseteq \exists R_0 A_n, A_{n+1} \subseteq \exists S_0 A_n \} \). It is not hard to see that \( q_n \) contains conflicting tree witnesses \( f \) and \( g \) as shown in the picture below, the query \( q_{n-1} \) is a subquery of \( q_n^+ \), and so \( (q_{n-1})^+ \) occurs in \( (q_n)^+ \) four times. On the other hand, there is a constant \( c \) such that \( |q_n| = |q_{n-1}| + c \).

**Sufficient Conditions for Polynomial Rewriting**

The rewriting \( q^+ \) is exponentially long if there are exponentially many tree witness formulas \( tw_f \) for some pair \((t, t')\) of adjacent terms in \( q \). Each \( tw_f \) is determined by the RB-subtree of \( C_T \) (containing the range of \( f \)), the domain \( dom f \) of \( f \) and the set \( f^{-1}(r_f) \); the terms in this set will be called the roots of \( f \). Now we show that, for a large class of CQs \( q \) and OWL 2 QL TBoxes \( T \), all tree witnesses for the same pair \((t, t')\) in \( q \) have the same domain and roots. As the number of distinct RB-subtrees of \( C_T \) is polynomial in the size of \( T \), the rewriting \( q^+ \) in this case is also polynomial; moreover, we show that it can be constructed in polynomial time in \(|q| \) and \(|T| \). Roughly, Theorem 19 to be proved below demonstrates that this can be done if condition (conf) holds and there are no roles \( T, S \) and a tree witness \( f \) for which \( f(q) \) and \( C_T \) simultaneously contain fragments of the form

\[
\text{conf}_{f,q} \quad \text{if } (f,q) \text{ modulo the equivalence relation } \sim \text{ of } f(s) = f(s'), \text{ Roles } T \text{ and } S \text{ are called adjacent in } f(q) \text{ if } (T(s_1, s_2), S(s_2, s_3) \in f(q), \text{ for some } s_i \in \text{term } f(q) \text{ such that } s_2 \in f^{-1}(r_f).
\]

We say that a role \( S \) is forward in \( T \) if \( u \sim v \) for all \((u, v) \in S \). If neither \( S \) nor its inverse \( S^{-1} \) is forward then \( S \) is said to be a twisty role in \( T \).

A tree witness \( f \) is called perfect if, for every pair \( T, S \) of adjacent roles in \( f(q) \) such that \( S \) is twisty in \( T \), we have
(perf) \( C_T \neq inv(T,S) \land suc(T,S) \),

where

\[
inv(T,S) = \exists x, y \ (T(x,y) \land S(y,x)), \\
suc(T,S) = \exists x, y, z \ (T(x,y) \land S(y,z) \land (x \neq z)).
\]

**Example 14** Consider \( T_1 = \{ A \subseteq \exists T, T \subseteq S^- \} \) and \( q(x) = \{ T(x,y), S(y,z) \} \). We have \( C_{T_1} \models inv(T,S) \) and \( C_{T_1} \models inv(R,R^-) \), for every role \( R \). Both \( T \) and \( S^- \) are forward roles in \( T_1 \). The homomorphism \( f \) shown below is a perfect tree witness for \((x,y)\):

Consider now \( T_2 = \{ A \subseteq \exists T, T \subseteq S^- \subseteq 3\} \). As \( C_{T_2} \) contains the following fragment

\[
\begin{array}{c}
S^-
\end{array}
\begin{array}{c}
T
\end{array}
\begin{array}{c}
A'
\end{array}
\]

\( S \) is a twisty role in \( T_2 \), and \( C_{T_2} \models inv(T,S) \land suc(T,S) \).

Thus, there is no perfect tree witness for \((x,y)\) in \( q \) and \( T_2 \), though there are two ‘imperfect’ tree witnesses.

It turns out that if all tree witnesses \( f \) for a pair of terms in \( q \) are perfect then all of them have the same domain and roots, and so \( i \) the tree-witness formulas \( Tw_f \) occur in the disjunctions for the same edges \((t,t')\) and \( ii \) the \( Tw_f \) may differ only in their \( ext_{R,R'}(t) \) conjuncts. In the following example, exponentially many different tree witnesses give rise to the same tree witness formula.

**Example 15** Let \( q(x) = \{ S(x,y), R(y,z_i) \mid 1 \leq i \leq n \} \) and \( T = \{ A \subseteq \exists S, \exists S^- \subseteq R \exists B_1, \exists S^- \subseteq R \exists B_2 \} \). There are \( 2^n \) (perfect) tree witnesses for \((x,y)\), each \( z_i \) can be mapped either to a \( B_1 \) or a \( B_2 \)-point in \( C_T \). All these tree witnesses have the same domain (all terms in \( q \)) and only one root \( x \). They define the same tree-witness formula \( ext_{3S}(x) \).

The notion of universal tree witness we are about to introduce will serve as a compact representation for all such similar tree witnesses.

For a pair \((t,t')\) of adjacent terms in \( q \), a tree-shaped CQ \( f \) is called a universal tree witness for \((t,t')\) in \( q \) and \( T \) if there is a partial homomorphism \( f \) from \( q \) onto \( f \) such that, for every tree witness \( g \) for \((t,t')\) in \( q \) and \( T \),

- \( \text{dom } q = \text{term } (f) \), and
- there exists a forward homomorphism \( f' \) from \( f \) to \( C_T \) such that \( g(s) = f'(f(s)) \), for every \( s \in \text{term } (f) \),

where a forward homomorphism with understand a homomorphism that preserves the distance from the root (recall that both \( f \) and \( C_T \) are tree-shaped).

**Lemma 16** (i) If all tree witnesses for a pair \((t,t')\) of terms in \( q \) and \( T \) are perfect, then there is a unique (up to isomorphism) universal tree witness for \((t,t')\) in \( q \) and \( T \).

(ii) There is a polynomial-time algorithm which, given \( q \), \( T \) and a pair \((t,t')\) of adjacent terms in \( q \), checks whether all the tree witnesses for \((t,t')\) in \( q \) and \( T \) are perfect, and if this is the case, returns a universal tree witness for \((t,t')\) in \( q \) and \( T \).

As a consequence of this lemma we obtain that, if all tree witnesses for \((t,t')\) are perfect, then all of them share the same domain and roots.

**Example 17** In Example 15, a universal tree witness for \((x,y)\) coincides with the whole query \( q(x) \).

In the proof of Lemma 16, we construct a finite sequence \( f_i \) of tree-shaped CQs \( f_i \) such that the final \( f_n \) is a universal tree witness for \((t,t')\). We begin by taking all the atoms with terms \( t, t' \) and then try to satisfy the tree witness condition \((ti)\) by adding atoms with new terms to the current \( f_i \). Condition \((perf)\), which is being checked during the construction, allows us to decide whether we have to add a new term to \( f_i \) or reuse an existing one. To illustrate, we give one more example.

**Example 18** Consider \( T = \{ A \subseteq \exists S.C, C \subseteq \exists T^- \}, \exists S^- \subseteq \exists R \}, R \subseteq \exists R' \}, B \subseteq \exists T'^- \}, T' \subseteq T, T'^ \subseteq S \} \) and \( q(x_0) = \{ S(x_0,y_1), R(y_1,y_2), R(y_3,y_4), T(y_4,y_5) \} \) in the picture above. A universal tree witness for \((x_0,y_1)\) must contain \( S(x_0,y_1) \). As the role \( R \) in \( R' (y_1,y_2) \) is forward, the universal tree witness must contain this atom too. The role \( R \) in \( R' (y_2,y_3) \) is also forward, so we add this atom and identify \( y_1 \) with \( y_3 \). This gives three approximations of the universal tree witness we are constructing:

\[
\begin{array}{c}
f_1
\end{array}
\begin{array}{c}
f_2
\end{array}
\begin{array}{c}
f_3
\end{array}
\]

In \( T(y_4,y_5) \), the role \( T \) is twisty in \( T \). The canonical model \( C_T \) suggests two alternative ways of treating \( T(y_4,y_5) \): either to add it as it is, or to identify \( y_4 \) with root \( x_0 \). Thus, no universal tree witness for \((x_0,y_1)\) exists. This is also signalled by the fact that condition \((perf)\) fails: by identifying \( y_1 \) and \( y_3 \), we make \( S \) and \( T \) adjacent, while both \( suc(S,T) \) and \( inv(S,T) \) hold in \( C_T \). Note that \( S \) and \( T \) were not adjacent in the original query \( q \); that is why \((perf)\) is checked for the image \( f(q) \) of every tree witness \( f \). As the number of tree witnesses can be exponential, this complication might suggest that \((perf)\) cannot be checked efficiently. The proof
of Lemma 16 shows that this check can be done in polynomial time without constructing all tree witnesses.

Strictly speaking, a universal tree witness is not a tree witness in the sense of our original definition, but rather a convenient structure representing all tree witnesses for \((t, t')\), even if there are exponentially many of them. Universal tree witnesses can be used to further simplify the rewriting \(q_c\).

Namely, all formulas \(tw_f\) for \((t, t')\) are merged into a single tree-witness formula \(tw_f\) for the universal tree witness \(f\) for \((t, t')\), which is defined by taking:

\[
    tw_f = \bigvee_{\exists H, B \text{ such that there is a homomorphism } h: f \mapsto C_T} \bigwedge_{s \text{ is root of } f} \left( \bigwedge_{A(s) \in q} A(s) \right).
\]

As a universal tree witness is unique for each pair \((t, t')\) of adjacent terms, let us denote it by \(q_{t,t'}\). We are now in a position to define our polynomial rewriting \(q_p\) for \(q\) and \(T\):

\[
    q_p(x) = \text{detached } q \lor \\
    \exists y \bigwedge_{\{t, t'\} \subseteq \{s, s'\}} \left( \bigwedge_{A(s) \in q} A(s) \land \bigwedge_{R(t, t') \in q} R(t, t') \land \bigvee_{t, t' \in \text{term}(A(s, s'))} tw_{q(s, s')} \right).
\]

**Theorem 19** If all tree witnesses for \(q\) and \(T\) are perfect and condition (conf) is satisfied then, for any ABox \(A\) and any \(a \subseteq \text{ind}(A)\), we have \((T, A) \models q[a]\) iff \(L_A \models q_p[a]\).

Moreover, \(q_p\) is constructed in time polynomial in \(|q|, |T|\).

Theorem 19 can be substantially sharpened by taking account of the ‘types’ of terms in \(q\).

**Example 20** Consider again the TBox \(T_2\) from Example 14 and the CQ \(q_1(x) = \{T(x, y), S(y, z), A'(x)\}\). This time we have only one tree witness for \((x, y)\). It is not perfect; yet, it is type-perfect because the ‘typed’ suc formula \(\exists x, y, z (T(x, y) \land S(y, z) \land A'(x) \land (x \neq z))\) does not hold in \(C_T\).

If an ontology \(T\) does not contain any twisty roles, then all tree witnesses in any CQ \(q\) over \(T\) are clearly perfect. On the other hand, all examples of conflicting tree witnesses given above involve twisty roles. The following theorem shows that this is no accident:

**Theorem 21** Suppose that \(q\) is a CQ and \(T\) an OWL 2QL ontology without twisty roles. Then there are no conflicting tree witnesses for \(q\) and \(T\). Thus, the rewriting \(q_p\) for \(q\) and \(T\) is correct and can be constructed in polynomial time.

It may be worth noting that OWL 2 EL (Baader, Brandt, and Lutz 2005; 2008) ontologies satisfy this sufficient condition, and so a polynomial rewriting similar to \(q_p\) can also be used for conjunctive query answering over such ontologies that the ABoxes are complete (or saturated) with respect to the ontologies.

**Experiments**

To understand how the rewriting \(q_p\) looks like in real-world practice, we have run experiments with three known OWL 2 QL ontologies—Adolena (A), University (U) and Stockexchange (S)—using the same conjunctive queries as in (Pérez-Urbina, Motik, and Horrocks 2009; Rosati and Almatelli 2010; Gottlob, Ors, and Pieris 2011) and two new ones (Q6 and Q7); the ontologies and queries are available at www.dcs.bbk.ac.uk/~roman/query-rewriting. The results of the experiments are collected in the table above.

The most important conclusion we can draw from these experiments is that, in practice, queries and ontologies generate very few tree witnesses, if any; they are never in conflict with each other, the sufficient condition of Theorem 19 is satisfied, and so the rewritings \(q_c\) and \(q_p\) are both short and correct.

To describe how the rewritten queries look like and explain the figures in the table, assume first that a given CQ \(q\) contains no tree witnesses with respect to a given ontology \(T\). In this case, \(q_p\) is obtained from \(q\) by replacing each unary atom \(A(s)\) with \(\text{ext}_{A}(s)\) and each binary atom \(R(s, s')\) with \(\text{ext}_{R}(s, s')\). Roughly, the \(\text{ext}_E\) contain those concepts/roles that are located under \(E\) in the classification of \(T\) by an ontology reasoner. It will be convenient for us to represent the \(\text{ext}_E\) as nonrecursive Datalog programs. For example, Adolena gives the rules:

\[
    \text{ext}_{\text{affects}}(x, y) \gets \text{affects}(x, y).
\]

\[
    \text{ext}_{\text{isAffectedBy}}(x, y) \gets \text{isAffectedBy}(y, x).
\]

\[
    \text{ext}_\text{Device}(x) \gets \text{Device}(x).
\]

The column ‘ext rules’ in the table refers to the number of such Datalog rules for the given CQ and ontology.

Consider now a query containing tree witnesses, for instance the following query Q4 over Adolena:

\[
    q(x) = \exists y \left( \text{Device}(x) \land \text{assistsWith}(x, y) \land \text{PhysicalAbility}(x, y) \right).
\]

This CQ contains a tree witness \(f\) with root

\[
    f(x) = \text{a} \text{assistsWith} \text{MovementAbility}.
\]
In this case, for each pair \((s, s')\) of adjacent terms in \(q\), we introduce a fresh predicate, say \(\text{edge}_{s,s'}\), and define it by means of a nonrecursive Datalog program such as

\[
\text{edge}_{x,y}(x, y) :\leftarrow \text{ext}_{\text{Device}}(x), \text{ext}_{\text{assistsWith}}(x, y), \text{ext}_{\text{PhysicalAbility}}(x, y).
\]

\[
\text{edge}_{x,y}(x, y) :\leftarrow \text{ext}_{\text{Device}}(x), \text{ext}_{\text{assistsWith}}(x, y), \text{ext}_{\text{MovementAbility}}(x).
\]

where the first rule corresponds to choosing both \(x\) and \(y\) among the ABox individuals and the second rule comes from a single universal tree witness for \((x, y)\). Then we replace by \(\text{edge}_{s,s'}\) all the atoms in the CQ that are ‘covered’ by the terms \(s\) and \(s'\). In our running example, we obtain

\[
q(x) := \text{edge}_{x,y}(x, y).
\]

The column ‘non-ext rules’ in the table refers to the number of such rules (one rule for the whole \(q\) plus the rules defining the \(\text{edge}_{s,s'}\)); ‘tw’ gives the number of tree witnesses in the CQ, and ‘uw’ the number of universal tree witnesses.

In theory, a correct rewriting of a CQ \(q\) and an OWL 2 QL ontology \(T\) can be exponential only if \(q\) and \(T\) give rise to exponentially many tree witnesses, in which case the canonical model \(\mathcal{C}_T\) must be extremely complex. Our experiments indicate that, in practice, the contribution of tree witnesses does not look essential at all, especially in comparison with the contribution of the definitions of \(\text{ext}_E\), which reflects the depth and width of the concept and role hierarchies in \(T\) rather than the complexity of \(\mathcal{C}_T\). Note also that the same predicates \(\text{ext}_E\) are used in all queries, which makes these predicates an ideal target for optimisations. The ways to minimise the influence of these rules depend on how we store the data.

There are two main approaches to storing data in OBDA. Suppose first that an ABox \(A\) is stored in a local database and the system has a certain degree of control over the data. In this case, one can saturate \(A\) with the intensional data that is implied by the TBox axioms (more precisely, construct the ABox part of the canonical model). Having done so, we do not need the \(\text{ext}_E\) predicates any more and can replace them with the corresponding \(E\). The ABox saturation (more precisely, a finite encoding of the whole canonical model) was suggested in the combined approach (Lutz, Toman, and Wolter 2009; Kontchakov et al. 2010). However, the downside of the ABox saturation is a significant increase of the storage space required for the data (and a slowdown of updates). One solution to this problem was found by Rodriguez Muro and Calvanese (2011a; 2011b). In a nutshell, the idea is to build a ‘semantic index’ by assigning numerical identifiers to the concept names, used in the ontology, in such a way that all subclasses of a given concept are associated with an interval (or a few intervals) of numbers. The semantic index allows one to encode the definitions of any of the \(\text{ext}_E\) predicates as a single query that selects all instances with concept identifiers falling into the respective intervals. In this case, the classical database indexing techniques are employed to ensure efficiency of these interval queries.

In the other typical OBDA scenario, ABoxes do not come as sets of triples stored in a single database. Instead, the sets of individuals that belong to concepts and roles are defined by means of queries (mappings) to a number of (relational) data sources (Lenzerini 2002; Calvanese et al. 2007b). Consider, for instance, the rules for the role \(\text{affects}\) above. One can clearly expect mappings to be defined in such a way that \(\text{affects}(a, b) \in A\) iff \(\text{inAffectedBy}(b, a) \in A\) in every ABox \(A\). This suggests that, in fact, there is no need for two separate rules, so that the predicate \(\text{ext}_{\text{affects}}\) can be eliminated altogether (replaced by \(\text{affects}(x, y)\), which halves the number of CQs produced). It is also quite feasible that information about all devices is stored in a single database relation and mappings for each of the 26 subclasses of the concept \(\text{Device}\) select appropriate devices from the same database relation. In such a case, every ABox \(A\) defined by these mappings will be complete for all subclasses \(A\) of \(\text{Device}\) in the sense that \(A(a) \in A\) iff \((T, A) \models A(a)\). Therefore, with this information at hand, one can replace the 26 rules defining \(\text{ext}_{\text{Device}}(x)\) with just a single rule

\[
\text{ext}_{\text{Device}}(x) :\leftarrow \text{Device}(x), \text{or even eliminate the predicate } \text{ext}_{\text{Device}}(x)\text{ altogether.}
\]

**Conclusions**

In this paper, we considered pure (positive existential) rewritings of conjunctive queries over OWL 2 QL ontologies and analysed why such rewritings can be lengthy. We showed that the length of a rewriting is related to the number of tree witnesses in the query, which reflect how various parts of the query can be homomorphically mapped to the tree (‘intensional’) part of the canonical model. Thus, a rewriting can be lengthy if the original query is sufficiently long and the intensional part of the canonical model for the ontology is sufficiently complex. We proved that by restricting the interaction between inverse roles and role inclusion axioms in ontologies and queries, we can guarantee transparent polynomial rewritings. Moreover, we also demonstrated that real-world ontologies and queries contain very few tree witnesses, satisfy the above mentioned restrictions, and so enjoy polynomial rewritings.

**Remark 22** When the final version of this paper was ready for submission, we obtained some new results that shed more light on the size of pure rewritings. Below is a brief summary of these results; for details consult the preliminary report (Kikot, Kontchakov, Podolskii and Zakharyaschev 2012).

1. An exponential blow-up is unavoidable for pure positive existential rewritings (PE) and pure nonrecursive Datalog (NDL) rewritings; pure FO-rewritings can blow-up super-polynomially unless \(\text{NP} \subseteq \text{P/poly}\).
2. Pure NDL-rewritings are in general exponentially more succinct than pure PE-rewritings.
3. Pure FO-rewritings can be superpolynomially more succinct than pure PE-rewritings.
4. Impure PE-rewritings can always be made polynomial, and so they are exponentially more succinct than pure PE-rewritings.

(1)–(3) are proved by first establishing connections between pure rewritings for CQs over OWL 2 QL ontologies and circuits for monotone Boolean functions, and then using known...
lower bounds and separation results for the circuit complexity of such functions as $\text{CLIQUE}(n, k)$ ‘a graph with $n$ nodes contains a $k$-clique’ and $\text{MATCHING}(2n)$ ‘a bipartite graph with $n$ vertices in each part has a perfect matching’ (Razborov 1985; Borodin, von zur Gathen, and Hopcroft 1982; Raz and Wigderson 1992; Raz and McKenzie 1997).

The polynomial PE-rewriting in (4) is similar to the NDL-rewriting of Gottlob and Schwentick (2011): using two extra constants, $\_1$ and (polynomially-many) new existentially quantified variables, one can encode a relevant part of the canonical model of $T$ in the rewritten query. The difference between the resulting impure PE-rewritings and the exponential-size pure PE-rewritings is of the same kind as the difference between deterministic and nondeterministic Boolean circuits. As shown by Razborov (1985), no polynomial-size deterministic monotone circuit can compute $\text{CLIQUE}(n, k)$; however, it can be computed by a polynomial-size nondeterministic circuit (or a QBF), where the existentially quantified variables guess $k$ vertices and the circuit checks whether they form a $k$-clique in the given graph. In the polynomial impure PE-rewriting (4), the extra constants, variables and $\_1$ are used to nondeterministically guess a part of the canonical model into which the query can be mapped. (We conjecture that there does not exist an impure polynomial-size rewriting if the number of new existentially quantified variables is bounded.)

Acknowledgments

The work on this paper was supported by the U.K. EPSRC grant EP/H05099X/1.

We are grateful to G. Gottlob, G. Orsi, R. Rosati and D. Tsarkov who helped us to run the experiments.

References


Appendix: Proofs of Theorems 5, 7, 10, 12

Recall that certain answers to a CQ \( q(\bar{x}) \) over an OWL 2 QL KB \( K = (T, A) \) boils down to finding homomorphisms \( a \) from \( q \)—understood as a directed graph or an interpretation—to \( C_K \) under which \( a(x) \in \text{ind}(A) \) for each \( x \in \bar{x} \).

Definition 23 Let \( K = (T, A) \) and let \( a \) be a tree witness for \((t, t')\) in a CQ \( q \) over \( K \). An \( a \in \text{ind}(A) \), denote by \( \overline{C_K} \) the sub-interpretation of \( C_K \) with domain \( \text{path}_K(a) \). We say that a map \( a : \text{term}(q) \rightarrow \Delta^{C_K} \) realises \( q \) at \( a \in \text{ind}(A) \) if the following conditions hold:
- \( a^{-1}(r_g) = \{a\} \);
- \( a^{-1}(\text{dom } g \setminus \{r_g\}) \subseteq \Delta^{C_K} \setminus \{a\} \);
- the restriction of \( a \) to \( \text{dom } g \) is a homomorphism from \( q|_{\text{dom } g} \) to \( \overline{C_K} \),

where

\[ q|_{\text{dom } g} = \{S(s, s') \in q \mid s, s' \in \text{dom } g\} \cup \{A(s) \in q \mid s \in \text{dom } g\} \]

We will regard such an \( a \) as an assignment for the variables in \( q \) in the interpretation \( C_K \).

Proposition 24 (i) Suppose that \( a \) is a homomorphism from \( q \) to \( C_K \) and \( g \) is a tree witness in \( q \). If \( a \) realises \( g \) at \( a \), then \( \mathcal{I}_A \models_b tw_g \), where \( b(t) = \text{head}(a(t)) \).

(ii) Let \( g \) be a tree witness for \((t, t')\) in \( q \). If \( \mathcal{I}_A \models_b tw_g \) for \( b : \text{term}(q) \rightarrow \text{add}(A) \) then there is a \( a : \text{term}(q) \rightarrow \Delta^{C_K} \) realising \( q \) at \( b(t) \) and such that

\[ a(t) \neq b(t) \quad \text{iff} \quad t \in \text{dom } g \setminus g^{-1}(r_g). \tag{1} \]

Proof (i) is checked by direct inspection. (ii) On \( \text{dom } g \), we define \( a \) to be the composition of \( g \) and the natural embedding of the RB-subtree of \( C_T \), containing the image of \( g \), into \( C_K^{(b)} \) (this embedding exists because \( C_K \models tw_g(b(t)) \)).

For \( s \in \text{term}(q) \setminus \text{dom } g \), we set \( a(s) = b(s) \). It is readily checked now that \( a \) is as required.

Proposition 25 For all concept names \( A \), role names \( P \) and individual names \( a \) and \( b \),

\[ C_K \models A(a) \iff \mathcal{I}_A \models \text{ext}_A(a), \]
\[ C_K \models P(a, b) \iff \mathcal{I}_A \models \text{ext}_P(a, b). \]

Proof Follows immediately from the definition of the formulas \( \text{ext}_A \) and \( \text{ext}_P \).

Proposition 26 Let \( C_K \models a \cdot q \) for an assignment \( a \) and \( s \in \text{term}(q) \), and let \( C_K \models \det_q \).

Then there is a tree witness \( g \) for some \((t, t')\) in \( q \), which is realised by \( a \) at \( a(t) \) and such that \( s \in \text{dom } g \setminus g^{-1}(r_g) \).

Proof Let \( a(s) = a \cdot w_{\text{RB}} \cdot \sigma \) for some \( \sigma \), and let \( \overline{C_K}(s) \) be the restriction of \( C_K \) to set elements of \( \Delta^{C_K} \) of the form \( a \cdot w_{\text{RB}} \cdot \sigma \). Note that the RB-subtree of \( C_T \) coincides with \( \overline{C_K}(s) \) except perhaps the interpretation of some concepts at \( a \). Take some adjacent \( t, t' \in \text{term}(q) \) with \( a(t) = a \) and \( a(t') = a w_{\text{RB}} \).

Such \( t \) and \( t' \) must exist because \( C_K \not\models \det_q \) and \( q \) is connected. Let \( D \) be the smallest set containing \( t, t' \) and such that if \( r \in D \setminus a^{-1}(a) \) and \( R(r, r') \in q \) then \( r' \in D \). Now we define \( g \) to be the restriction of \( a \) to \( D \). It is readily seen that \( g \) is a tree witness in \( q \) for \((t, t')\).

In what follows we make use of the following abbreviation:

\[ E_q(t, t') = \bigwedge_{A(s) \in q} \text{ext}_A(s) \wedge \bigwedge_{R(t, t') \in q} \text{ext}_R(t, t'). \]

Proposition 27 Let \( b \) be an assignment for \( q \) in \( \mathcal{I}_A \) and \( \Xi \) a consistent set of tree witnesses in \( q \) such that \( \mathcal{I}_A \models_C tw_f \), for all \( f \in \Xi \). Then there is an assignment \( a \) in \( C_K \) such that, for every \( g \in \Xi \), the restriction of \( a \) to \( \text{dom } g \) is a homomorphism from \( q|_{\text{dom } g} \) to \( C_K \) and \( a(r) = b(r) \), for every root \( r \) of \( g \).

If, in addition, for every pair \((t, t')\) of adjacent terms in \( q \), which do not belong both to the same \( \text{dom } g \) for any \( g \in \Xi \), we have \( \mathcal{I}_A \models_C E_q(t, t') \), then \( a \) is a homomorphism from \( q \) to \( C_K \).

Proof For every tree witness \( g \in \Xi \) for \((t, t')\) in \( q \), we use Proposition 24 (ii) and obtain a homomorphism \( a_g \) from \( q|_{\text{dom } g} \) to \( C_K \) such that (i) holds and \( a_g(r) = b(r) \), for every root \( r \) of \( g \).

Since \( \Xi \) is consistent, for every term \( s \), there is at most one \( q \) such that \( a_g(s) \) is defined and \( a_g(s) \neq b(s) \). Thus we can set, for \( s \in \text{term}(q) \),

\[ a(s) = \begin{cases} \text{if } a_g(s) \text{ is defined, for some } g \in \Xi; \\ b(s), \text{ otherwise.} \end{cases} \]

It follows that, if the second condition of the proposition holds, then \( C_K \models a \cdot q \).

Proof of Theorem 5 Let \( K = (T, A) \) be an OWL 2 QL and \( q(\bar{x}) \) a CQ over \( K \). We have to show that \( C_K \models q(\bar{x}) \) iff \( \mathcal{I}_A \models q(\bar{x}) \).

(\( \Rightarrow \)) Suppose that \( C_K \models q(\bar{x}) \). Then there is a homomorphism \( a \) from \( q \) to \( C_K \) such that \( a(x) \in \text{ind}(A) \) for \( x \in \bar{x} \). Denote by \( \text{im}(a) \) the image of \( a \) and consider two cases.

Case 1: \( \text{im}(a) \cap \text{ind}(A) = \emptyset \), Then \( \mathcal{I}_A \models \det_q \).

Case 2: \( \text{im}(a) \cap \text{ind}(A) \neq \emptyset \). Without loss of generality, we may assume that \( C_K \not\models \det_q \). Set \( b(t) = \text{head}(a(t)) \), for all \( t \in \text{term}(q) \). We show now that all atoms in \( q \) are true in \( \mathcal{I}_A \) under the assignment \( b \).

Take the set \( \Xi \) of all tree witnesses in \( q \) realised by \( a \), and let \( \Xi \) be a subset of \( \Xi' \) such that, for every \( f' \in \Xi' \), there is precisely one \( f \in \Xi \) with \( \text{dom } f = \text{dom } f' \). We show first that \( \Xi \) is consistent. Indeed, suppose that distinct \( h, g \in \Xi \) are incompatible. Then \( \text{dom } g \neq \text{dom } h \), so, without loss of generality, we may assume that there is some \( t \in \text{dom } g \setminus \text{dom } h \). Consider some term \( s \in \text{dom } g \cap \text{dom } h \) such that \( s \notin g^{-1}(r_g) \cap h^{-1}(r_h) \). If \( s \in g^{-1}(r_g) \) then \( a(s) \in \text{ind}(A) \) because \( a \) realises \( g \); on the other hand, as \( a \) realises \( h \) and \( s \notin h^{-1}(r_h) \), we must have \( a(s) \notin \text{ind}(A) \), which is
a contradiction. Take a path in dom $g$ from $t$ to $s$ containing no roots of $g$ save possibly $t$. This path must contain a root of $h$ because $s \in \text{dom} \ h$ while $t \not\in \text{dom} \ h$. Denote it by $r$. But then we again obtain that $a(r) \not\in \text{ind}(A)$ as $r \in \text{dom} \ g$ is not a root of $g$, while $a(r) \in \text{ind}(A)$ as $r$ is a root of $h$. Thus, $\Xi$ is a consistent set of tree witnesses.

By Proposition 24, $\mathcal{I}_A \models \mathcal{I}_A \models b \bigwedge_{f \in \Xi} \text{tw}_f$. By Proposition 26, if $a(s) \not\in \text{ind}(A)$, then there exists $g \in \Xi$ such that $s \in \text{dom} \ g \ \& \ g^{-1}(r_g)$. It follows that, for all pairs $(t, t')$ of adjacent terms in $q$ that do not belong to the same dom $g$, for any $g \in \Xi$, we have $a(t) = b(t)$ and $a(t') = b(t')$, whence, by Proposition 25, $\mathcal{I}_A \models b \text{ext}_A(t) \ \& \ \mathcal{I}_A \models b \text{ext}_A(t', t''),$ for all $A(t) \in q$ and $P(t, t') \in q$. Thus, we obtain $\mathcal{I}_A \models b \ q_c$. (\Leftarrow) Suppose now that $\mathcal{I}_A \models q_c[a]$. Our aim is to show that $\mathcal{C}_\mathcal{K} \models q_c[a]$. 

Case 1: $\mathcal{I}_A \models \text{det}_q$. Then there are $R$, $B$, and $a$ such that $\mathcal{I}_A \models \text{ext}_{R_B}(a)$ and there is a homomorphism $b$ from term($q$) to the $RB$-subtree of $C_T$ such that $w_{[R_B]} \not\in \text{im}(b)$. As $\mathcal{I}_A \models \text{ext}_{R_B}(a)$, $C_K$ contains (an isomorphic copy of) the $RB$-subtree of $C_T$, so $b$ gives a homomorphism from $q$ to $C_K$, from which $\mathcal{I}_A \models b \ q_c$. 

Case 2: $\mathcal{I}_A \not\models \text{det}_q$. Then there is a map $b$ from $q$ to $\mathcal{I}_A$ such that $b(x) \in \text{ind}(A)$, for all $x \in \bar{x}$. We also have a consistent set of $\Xi$ of tree witnesses such that

$$\mathcal{I}_A \models b \bigwedge_{f \in \Xi} \text{tw}_f \ \& \ \bigwedge_{(t, t') \text{ adjacent} \ \ldots} E_q(t, t'),$$

Then Proposition 27 provides us with an assignment $a$ such that $\mathcal{C}_\mathcal{K} \models a \ q_c$. \hfill \Box

We now prove Theorem 7 on the rewriting $q_c$. It will be convenient to represent it as

$$q_c(\bar{x}) = \text{det}_q \ \vee \ \exists y \bigwedge_{(t, t') \text{ adjacent}} \left[ E_q(t, t') \ \vee \ \bigvee_{f \in \Theta} \text{tw}_f \right].$$

Proof of Theorem 7

Let $\mathcal{K} = (T, A)$ be an OWL 2 QL and $q(\bar{x})$ a CQ over $\mathcal{K}$. We have to show that if (conf) holds then $\mathcal{C}_\mathcal{K} \models q[a]$ if $\mathcal{I}_A \models q_c[a]$. 

(\Rightarrow) Suppose that $\mathcal{C}_\mathcal{K} \models q[a]$. Without loss of generality we may assume that $\mathcal{C}_K \not\models \text{det}_q$ (if this is not the case, we can use Case 1 in the proof of Theorem 5). Take a homomorphism $a$ from $q$ to $C_K$ such that $a(x) \in \text{ind}(A)$ for $x \in \bar{x}$ and, after all, set $b(t) = \text{head}(a(t))$, for all $t \in \text{term}(q)$. Suppose that $t, t'$ are adjacent in $q$. Our task is to show that

$$\mathcal{I}_A \models b \ E_q(t, t') \ \vee \ \bigvee_{f \in \Theta} \text{tw}_f.$$

Two cases are possible: 

Case 1: $a(t), a(t') \in \text{ind}(A)$. Then $b(t) = a(t), b(t') = a(t)$ and, by Proposition 25, we obtain $\mathcal{I}_A \models b \ E_q(t, t')$. 

Case 2: $a(t), a(t') \not\in \text{ind}(A)$. Then $t' \in \Delta_C^{\text{tw}} \setminus \{a\}$, for some $a \in \text{ind}(A)$, and $a(t) \in \Delta_C^{\text{tw}}$. Now recall that $\mathcal{C}_K$ is (isomorphic to) one of the $RB$-trees of $C_T$, and so there is a tree witness $f \in \Theta$ with $t, t' \in \text{dom} \ f$. By Proposition 24 (i), we then obtain $\mathcal{I}_A \models b \ \text{tw}_f$, which yields (2).

(\Leftarrow) Suppose that $\mathcal{I}_A \models b \ q_c$. Without loss of generality we again assume that $\mathcal{C}_K \not\models \text{det}_q$. Let

$$\Xi' = \{ f \ | \ \mathcal{I}_A \models b \ \text{tw}_f \ \& \ \mathcal{I}_A \not\models b \ E_q(t, t') \}.$$ 

Let $\Xi$ be the subset of $\Xi'$ obtained by removing those tree witnesses that are not maximal w.r.t. domain inclusion. Condition (conf) guarantees that $\Xi$ is consistent. So, by Proposition 27, there is an assignment $a$ such that $\mathcal{C}_\mathcal{K} \models a \ q_c$. \hfill \Box

Recall that

$$q'_{c} = \text{det}_q \ \vee \ \exists y \bigwedge_{(t, t') \text{ adjacent}} \left[ E_q(t, t') \ \vee \ \bigvee_{f \in \Theta} \text{tw}_f \right].$$

To prove Theorem 10, we require two propositions. The first of them follows immediately from the fact that $\nu$ does not belong to the extension of any predicate in $\mathcal{I}_A'$.

Proposition 28 $\mathcal{I}_A' \models \forall x \ (\text{ext}_{R_B}(x) \iff (x \not= \nu))$, for all $R$ and $B$.

The second one is proved in the same way as Proposition 24:

Proposition 29 Let $g$ be a tree witness for $(t, t')$ in $q$. If $\mathcal{I}_A' \models b \ \text{tw}_f$ for $b : \text{term}(q) \to \text{ind}(A)$ and then there exists a map $a : \text{term}(q) \to C_K$ realising $b$ at $(t, t')$ such that

$$a(t) \not= b(t) \ \iff \ t \in \text{dom} \ g \ \& \ \text{tw}_f \not\subseteq \text{dom} \ g \ \& \ \text{tw}_f.$$ 

(3)

Proof of Theorem 10

The direction (\Rightarrow) is proved in the same way as in Theorem 7 using Proposition 24.

(\Leftarrow) Suppose that $\mathcal{I}_A' \models b \ q'_c$. Without loss of generality we assume that $\mathcal{C}_K \not\models \text{det}_q$. Let

$$\Xi' = \{ f \ | \ \mathcal{I}_A' \models b \ \text{tw}_f \ \& \ \mathcal{I}_A' \not\models b \ E_q(t, t') \}$$

and let $\Xi$ be the subset of $\Xi'$ obtained by removing those tree witnesses that are not maximal w.r.t. set-theoretic inclusion between their domains.

We show that $\Xi$ is consistent. Since $\Xi$ is an antichain w.r.t. domain inclusion, it is sufficient to show that $\Xi$ does not contain conflicting tree witnesses. Suppose that $f$ and $g$ are distinct conflicting tree witnesses in $\Xi$. Then there must exist a term $t$ such that $t \in \text{dom} \ f \ \& \ \text{dom} \ g$, $f(t) = r_f$ and $g(t) \not= r_g$. By Proposition 28, $\mathcal{I}_A' \models b \ \text{tw}_f$ implies $b(t) \not= \nu$, while $\mathcal{I}_A' \models b \ \text{tw}_g$, by Proposition 29, implies $b(t) = \nu$, which is impossible.

To complete the proof, it remains to make use of Proposition 27. \hfill \Box

The following two propositions are required in the proof of Theorem 12.

Proposition 30 Suppose that condition (conf) holds for $q$ and $K$, $\mathcal{C}_K \not\models \text{det}_q$ and that $f, g$ are conflicting tree witnesses in $q$. Assume also that $\mathcal{C}_K \models a \ q$. If $g$ realises $q$ then $a(s) \in \text{ind}(A)$, for all terms $s \in g^{-1}(r_g) \cup f^{-1}(r_f)$. 

Proof of Proposition 30

Let $\mathcal{I}_A' \models b \ q'_c$. Without loss of generality we assume that $\mathcal{C}_K \not\models \text{det}_q$. Let

$$\Xi' = \{ f \ | \ \mathcal{I}_A' \models b \ \text{tw}_f \ \& \ \mathcal{I}_A' \not\models b \ E_q(t, t') \}.$$
Proof The case $s \in g^{-1}(r_p)$ is trivial, so we consider the case when $s \in f^{-1}(r_f)$ and suppose that $a(s) \notin \text{ind}(A)$. Now we apply Proposition 26 to the term $s$ and obtain a tree witness $h$ conflicting with $f$ and such that $s \in \text{dom} \ h$ and $h(s) \neq r_h$. Condition (conf1), applied to $f$, says that this can only be the case if $\text{dom} \ h \subseteq \text{dom} \ g$ (which cannot be since $s \notin \text{dom} \ g$) or if $\text{dom} \ g \subseteq \text{dom} \ h$, contrary to the fact that both $g$ and $h$ are realised. 

As a consequence, we obtain the following variant of the Proposition 2.1 (i):

**Proposition 31** Suppose Theorem 12 holds for all $q'$ such that $|q'| < |q|$. Let $a$ be a homomorphism from $q$ to $C_K$. If $a$ realises $g$ at $o$, then $\mathcal{I}_A \models^b tw_g^+$, where $b(t) = \text{head}(a(t))$.

**Proof of Theorem 12** The proof proceeds by induction on the size of the query $q$. Now, assume that our theorem holds for all CQs $q'$ with $|q'| < |q|$.

($\Rightarrow$) As soon as Proposition 31 holds, the proof is the same as the proof of the Theorem 10.

($\Leftarrow$) Suppose that $\mathcal{I}_A \models q^+_{p}[a]$. Then there exists $b$ such that $\mathcal{I}_A \models^b q^+_{p}$. Without loss of generality we again assume that $C_K \neq \text{det}_q$. Let

$$\Xi' = \{f \in \Theta \mid \mathcal{I}_A \models^b tw_f \text{ and } \mathcal{I}_A \not\models^b E_q(t,t')\}$$

and let $\Xi$ be the subset of $\Xi'$ obtained by removing those tree witnesses that are not maximal w.r.t. domain inclusion. Since (conf1) holds, the set $\Xi$ is of the form

$$\Xi = \{(f_1,g_1), \ldots, (f_n,g_n), h_1, \ldots, h_m\},$$

where every $f_i$ conflicts only with $g_i$ (and the other way round), while all the $h_j$ are consistent with all the $f_i$ and $g_i$.

First, we apply IH to the queries $q_{g_i \setminus f_i}$ for all $i \leq n$ and obtain an assignment $a_i$ such that the restriction of $a_i$ to $\text{dom} \ q_{g_i \setminus f_i}$ is a homomorphism from $q$ to $C_K$. Then we apply Proposition 24 (ii) to the following consistent set of tree witnesses

$$\Xi'' = \{f_1, \ldots, f_n, h_1, \ldots, h_m\} = \{e_1, \ldots, e_{n+m}\}$$

and obtain partial homomorphisms $a''_i$, for $i \leq n+m$ defined on $e_i$ and realising $e_i$.

Now we set

$$a(t) = \begin{cases} 
  a''_i(t), & \text{if } t \in \text{dom} \ e_i, i \leq n+m; \\
  a_i(t), & \text{if } t \in \text{dom} \ q_{g_i \setminus f_i}; \\
  t, & \text{otherwise.}
\end{cases}$$

Since $e_i$ and $q_{g_i \setminus f_i}$ are compatible, $a$ is well defined. It is easy to see that $a$ is a homomorphism from $q$ to $C_K$. 

**Appendix: Proof of Theorem 19**

In the proof below, we require a notion of an ‘abstract’ tree witness. A usual tree witness, as defined above, is a partial homomorphism $f$ from a CQ $q$ to $C_T$. An abstract tree witness, to be defined below, is a partial homomorphism $f$ from $q$ to some other query $f$. Any usual tree witness may be considered as an abstract tree witness, where $f = f(q)$. All tree witnesses in this section will be abstract in this sense.

It will be convenient for us to consider a CQ $q$ as an interpretation over the domain term($q$), where some elements are identified as ‘quantified.’ By a subredct of $q$ we understand a pair $(f, f)$, where $f$ is a CQ and $f$ a partial map from $q$ onto $f$ satisfying the following conditions:

- $f$ is a homomorphism from $f$ onto $f$ (regarded as interpretations);
- $f(a) = a$, for all $a \in \text{ind}(A)$;
- if $f(t)$ is quantified in $f$ then $t$ is quantified in $q$;
- if $A(t') \in f$ then there exists $A(t) \in q$ such that $f(t) = t'$;
- if $P(s', t') \in f$ then there exists $P(s, t) \in q$ such that $f(s) = s'$ and $f(t) = t'$.

If, in addition, (the primal graph of) $f$ is connected and acyclic then $(f, f)$ is called a tree subredct of $q$. To simplify notation, we prefer to denote subreductions $(f, f)$ simply by $f$, bearing in mind that $f$ is uniquely (up to isomorphism) defined by $q$ and $f$, i.e., $f = f(q)$.

We say that a role $S$ is a forward role in $T$ if $(u, v) \in S^{C_T}$ implies $u \rightarrow v$, for any $u, v \in \Delta^{C_T}$; if $(u, v) \in S^{C_T}$ implies $v \rightarrow u$, for any $u \in \Delta^{C_T}$, then $S$ is called a backward role in $T$. If $S$ is neither forward nor backward then we say that $S$ is a twisted role in $T$.

We say that a subredct $f$ respects twisted roles in $T$ if, for any $T, S \in \text{role}(T)$, such that $S$ is twisted

- if $C_T \models \neg \text{suc}(T, S)$, then $f(q) \models \neg \text{suc}(T, S)$,
- if $C_T \models \neg \text{inv}(T, S)$, then $f(q) \models \neg \text{inv}(T, S)$.

We also say that a tree subredct $f$ of $q$ respects directed roles in $T$ for some $r \in \text{term}(f(q))$ if, for all $R(t, s) \in f(q)$,

- $\phi^f(q)(r, t) = \phi^f(q)(r, s) + 1$ whenever $R$ is a forward role in $T$,
- $\phi^f(q)(r, t) = \phi^f(q)(r, s) - 1$ whenever $R$ is a backward role in $T$,

where $\phi^f(q)(u, v)$ is the distance between $u$ and $v$ in the graph $f(q)$.

Let $q$ be a connected CQ and $R(t, t') \in q$. An (abstract) tree witness for $q$, $R(t, t')$ and $T$ is a tree subredct $f$ of $q$ such that the following conditions hold:

- (10) $f$ respects twisted roles in $T$;
- (11) $f$ respects directed roles in $T$ for $f(t)$;
- (12) $\text{dom} \ f$ is a minimal subset of $\text{term}(q)$ for which
  - $t, t' \in \text{dom} \ f$;
  - if $s \in \text{dom} \ f$ and $f(s) \neq f(t)$ then $s' \in \text{dom} \ f$;
- (13) $s \in \text{dom} \ f$ and $f(s) \neq f(t)$ then $s$ is quantified in $q$. If $f$ satisfies both (10) and (11), we say that $f$ respects roles. A tree witness $f$ is perfect if whenever $T(t_1, t_2), S(t_2, t_3) \in f(q)$, for some terms $t_i$, and $S$ is a twisted role in $T$, then $C_T \models \neg \text{inv}(T, S) \land \text{suc}(T, S)$. We say that $f$ is suitable for $T$ if there is a homomorphism from $f$ to $C_T$. 


Lemma 32 Let $g$ and $h$ be suitable perfect tree witnesses for $R(t, t') \in q$ and $T$. Then

1. $\text{dom } g = \text{dom } h$;
2. there exists a perfect tree witness $f$ for $R(t, t') \in q$ and $T$ such that $\text{dom } f = \text{dom } g = \text{dom } h$ and both $q$ and $h$ are homomorphic images of $f(q)$.

Proof We construct inductively a sequence of sets of terms $D_0 \subseteq D_1 \subseteq \cdots \subseteq D_n$, a sequence of tree subreducts $f_i$ of $q$ over $D_i$ and homomorphisms $\alpha_i : f_i(q) \rightarrow g_i(q)$ and $\beta_i : f_i(q) \rightarrow h_i(q)$ such that $\text{dom } f_i = \text{dom } g = \text{dom } h$ and the following diagram commutes:

\[
\begin{array}{ccc}
  g_i(q) & \xrightarrow{g_i} & f_i(q) \\
  \alpha_i & & \beta_i \\
  f_i(q) & \xrightarrow{f_i} & h_i(q)
\end{array}
\]

Here $h_i|_{D_i}$ and $g_i|_{D_i}$ are the restrictions of $h$ and $g$ onto $D_i$, respectively. For each $i$, we denote the distance from $h_i(u)$ to the root $h_i(t)$ in $h_i(q)$ by $\|h_i(u)\|$, and similarly for $g_i(u)$ and $f_i(u)$. Recall that $f_i = f_i(q)$ is a tree with root $f_i(t)$; we write $u \sim f_i(q) v$ to say that $v$ is a child of $u$ in this tree.

Step 0: Let $D_0 = \{t, t'\}$ and let $f_0$ be the identity map on $D_0$; and set $\alpha_0 = g_0$ and $\beta_0 = h_0$. Clearly, $f_0(q)$ is a tree subreduct with root $f_0(t)$ and the diagram commutes for $i = 0$. $\alpha_0$ is a homomorphism, since if $f_0(q) \models S(t, t')$, then $q \models S(t, t')$, hence $g_0(q) \models S(t, t')$ because $g$ is a homomorphism, and a similar argument proves that $\beta_0$ preserves unary atoms. Similarly, $\beta_0$ is a homomorphism too. Note that since $g$ is suitable, $R$ cannot be a backward role, whence $\|g_0(t)\| = \|f_0(t)\| = \|h_0(t)\| = 0$ and $\|g_0(t')\| = \|f_0(t')\| = \|h_0(t')\| = 1$, and so $\|g_0(s)\| = \|f_0(s)\| = \|h_0(s)\|$, for every $s \in D_0$.

Step $n + 1$: Let $S(s, s') \in q$ be such that $s \in D_n$, $\|g_0(s)\| \neq 0$ and $s' \notin D_n$. If there is no such atom then the construction terminates. Otherwise, we set $D_{n+1} = D_n \cup \{s'\}$. The definition of $f_{n+1}$ depends on whether $S$ is a forward, backward or twisty role in $T$.

Case 1: Suppose that $S$ is a forward role in $T$.

Case 1.1: If there is a twisty role $T(f_n(s''), f_n(s)) \in f_n(q)$ such that $f_n(s)$ is an immediate predecessor of $f_n(s'')$ on the path from the root $f_n(t)$ in $f_n(q)$ and

\[
C_T \models \text{inv}(T, S),
\]

then define $f_{n+1} = f_n \cup \{s' \mapsto f_n(s'')\}$ and $\alpha_{n+1} = \alpha_n$.

First, let us show that $\alpha_{n+1}$ is a homomorphism. Note that the roles $T$ and $S$ are incident in $g(q)$, since $g(q) \models T(g(s'), g(s))$ (as, by IH, $\alpha$ is a homomorphism) and $g(q) \models S(g(s), g(s'))$ (as $g$ is a homomorphism). Therefore, (4) implies

\[
C_T \models \neg \text{suc}(T, S),
\]

and we conclude that $g(s') = g(s'')$ (otherwise we have $g(q) \models \text{suc}(T, S)$ which implies $C_T \models \text{suc}(T, S)$, contrary to (5)). Thus, $g(q) \models S(g(s), g(s'))$. A similar argument shows that $\beta_{n+1}$ is also a homomorphism.

Let us show next that $f_{n+1}(q)$ respects roles. First, suppose that $f_{n+1}(q) \models \text{inv}(T', S')$. Then, since $\alpha_{n+1}$ is a homomorphism, we have $g(q) \models \text{inv}(T', S')$, from which $C_T \models \text{inv}(T', S')$. Now, suppose $f_{n+1}(q) \models \text{suc}(T', S')$, where at least one of $T'$ and $S'$ is twisty. In other words: $f_{n+1}(q) \models T'(x, y) \land S'(y, z) \land (x \neq z)$. The only case to consider is when $y = f_n(s)$, $z = f_n(s'')$, and $S' = S$ (in all other cases $C_T \models \text{suc}(T', S')$ by IH). Two cases are possible now:

Case 1.1.1. Let $x \sim f_n(q) f_n(s)$. This case is simple, as $\alpha_{n+1}$ preserves $\sim$, so $x \sim f_n(q) f_n(s) \sim f_n(q) f_n(s'')$ implies $\alpha_{n+1}(x) \sim g(q) g(q) g(q) g(s'')$, hence $\alpha_{n+1}(x) \neq g(s'')$, and so (since $\alpha_{n+1}$ is a homomorphism) we have $g(q) \models \text{suc}(T', S)$, which yields $C_T \models \text{suc}(T', S)$.

Case 1.1.2. Suppose that $f_n(q) \sim f_n(q) x$. Since $S$ is forward, we conclude that $T'$ is twisty. If $\alpha_{n+1}(x) \neq g(s'') = g(s')$, then we have $q \models \text{suc}(T', S)$, and $C_T \models \text{suc}(T', S)$, so assume that $\alpha_{n+1}(x) = g(s')$. So we have $g(q) \models S(g(s), g(s')) \land T'(g(s'), g(s)) \land T'(g(s'), g(s))$. Thus $g(q) \models \text{inv}(T', T')$, from which

\[
C_T \models \text{inv}(T', T').
\]

On the other hand, $f_{n+1}(q) \models T'(x, y) \land S'(y, z) \land (x \neq z)$ and $z = f_n(s'')$ imply $f_n(q) \models \text{suc}(T', T')$, hence by IH $C_T \models \text{suc}(T', T')$, which together with the fact that $T'$ is twisty contradicts (6). This means that the case $\alpha_{n+1}(x) = g(s'')$ is impossible.

Case 1.2. There is no $T(f_n(s''), f_n(s)) \in f_n(q)$ with the properties as in Case 1.1. Then define $f_{n+1} = f_n \cup \{s' \mapsto u\}$, where $u$ is a fresh quantified variable ($s'$ must be a quantified variable by (3)), and set $\alpha_{n+1} = \alpha_n \cup \{ u \mapsto g(s') \}$ and $\beta_{n+1} = \beta_n \cup \{ u \mapsto h(s') \}$. $\alpha_{n+1}$ is a homomorphism since $g(q) \models S(g(s), g(s'))$.

Let us show that $f_{n+1}(q)$ respects roles. That $\text{inv}$ is respected follows from the fact that $\alpha_{n+1}$ is a homomorphism. Suppose that $f_{n+1}(q) \models \text{suc}(T', S')$, where at least one of $T'$ and $S'$ is twisty. In other words, we have $f_{n+1}(q) \models T'(x, y) \land S'(y, z) \land (x \neq z)$. Again, the only case to consider is when $y = f_n(s)$, $z = f_n(s')$, and $S' = S$ (in all other cases $C_T \models \text{suc}(T', S')$ by IH). Again, two cases are possible:

Case 1.2.1. Suppose that $x \sim f_n(q) f_n(s)$. In this case we argue similarly to Case 1.1.1.

Case 1.2.2. Suppose that $f_n(q) \sim f_n(q) x$. Since $S$ is forward, we conclude that $T'$ is twisty. Again, if $\alpha_{n+1}(x) \neq f_n(s) = g(s')$, then $g(q) \models \text{suc}(T', S)$ and $C_T \models \text{suc}(T', S)$, so assume that $\alpha_{n+1}(x) = g(s')$. In this case we have $g(q) \models S(g(s), g(s')) \land T'(g(s'), g(s)) \land T'(g(s'), g(s))$. Thus $g(q) \models \text{inv}(T', T')$, from which

\[
C_T \models \text{inv}(T', T').
\]

But (7) together with $f_{n+1}(q) \sim f_{n+1}(q) x$ and the fact that $T'$ is twisty exactly coincide with the condition of the choice of Case 1.1, as we have chosen Case 1.2. This means that the case $\alpha_{n+1}(x) = g(s')$ is impossible.

In both Cases 1.1 and 1.2, $|f_{n+1}(s'')| = |f_{n+1}(s')| = |f_{n+1}(s)| + 1$, $|f_{n+1}(s'')| = |f_{n+1}(s')| = |f_{n+1}(s)| + 1$, and similarly for $h_{n+1}$. 
Case 2: If $S$ is a backward role in $T$, we define $f_{n+1} = f_n \cup \{s^i \to u\}$, where $u$ is the immediate successor of $f_n(s)$ on the path from the root $f_n(t)$ and set $\alpha_{n+1} = \alpha_n$ and $\beta_{n+1} = \beta_n$ (as $u$ is already in their domains). Clearly, $||f_{n+1}(s^i)|| = ||f_n(s)|| - 1$ and, as $g$ respects the backward direction of $S$, $||g_n(s^i)|| = ||g_n(s)|| - 1 = ||f_{n+1}(s^i)||$. Similarly for $h_{n+1}$.

Let us prove that $\alpha_{n+1}$ is a homomorphism. Indeed, $g(q) \models S(g(s), g(s'))$ and the fact that $S$ is backward and $g$ is suitable implies $g(s') \sim g(q), g(s)$, therefore $g(q) \models S(g(s), \alpha_{n+1}(u))$.

Let us prove that $f_{n+1}(q)$ respects roles. Again, respecting $\text{inv}$ and $\text{suc}$ for points $x$ and $z$ from different levels we obtain automatically from the fact that $\alpha_{n+1}$ is a homomorphism, and respecting $\text{suc}$ for the atoms different from $S(f(s), u)$ we obtain by IH. So the only case to consider is when $f_{n+1}(q) \models S(f(s), u) \land T(u, z) \land (f_n(s) \neq z)$ for some twisty role $T$ and a point $z$ such that $u \sim f_{n+1}(z)$. For respecting this instance of $\text{suc}$ it is sufficient to prove that $\alpha_{n+1}(f(s)) \neq \alpha_{n+1}(z)$. But since $f_n(q)$ is connected, there is a role $S'$ such that $f_n(q) \models S'(f(s), u)$, and hence, in case if $\alpha_{n+1}(f(s)) = \alpha_{n+1}(z)$ we obtain a contradiction to $f_{n+1}(q) \models \text{suc}(S', T)$, together with the fact that by IH $f_n$ respects roles.

Case 3: If $S$ is a twisty role in $T$, denote by $f_n(r)$, for some $r \in D_n$, the immediate predecessor of $f_n(s)$ on the path from the root $f_n(t)$ (recall that $||f_n(s)|| \neq 0$). By construction, there is $T(f_n(r), f_n(s)) \in f_n(q)$, for some role $T$. Clearly, we cannot have $C_T \models \neg \text{inv}(T, S) \land \neg \text{suc}(T, S)$ and, as $g$ is perfect, $C_T \models \neg \text{inv}(T, S) \land \text{suc}(T, S)$. Consider the remaining two cases.

Case 3.1: Suppose $C_T \models \neg \text{inv}(T, S) \land \text{suc}(T, S)$. Let

$$\Gamma = \{ T'(f(s), z) \mid T'(f(s), z) \in f_n(q), f(s) \sim f_n(q) z, C_T \models \neg \text{suc}(S', T') \}. $$

Let $\Gamma = \{ T_1'(f(s), z_1), \ldots , T_m'(f(s), z_m) \}$ where $m \geq 0$. Then we have $g(s') \models \alpha_{n+1}(z_1) = \cdots = \alpha_{n+1}(z_m)$. Indeed, if, for some $z_i$, $g(s') \neq \alpha_{n+1}(z_i)$, then $g(q) \models \text{suc}(S', T')$ while from the definition of $\Gamma$ and the fact that $g(q)$ respects roles it follows that $g(q) \models \neg \text{suc}(S', T')$, which is a contradiction.

Now take a new variable $u$ and define

$$f'_{n+1} = \{ (s'' \to f_n(s'''') \mid \forall i f_n(s'''') \neq z_i) \cup (s'' \to u) \mid s'' = s' \text{ or } f_n(s''') = z_i \text{ for some } i \} $$

and

$$\alpha_{n+1} = \{ (y \to \alpha_n(y) \mid \forall i y \neq z_i) \cup (u \to g(s')) \}. $$

(In other words, we merge all $z_i$ and $u$ into one point.)

But it still may happen that, in $f_n(q)$, $z_1$ has a $T_1$-child and $z_2$ has a $T_2$-child such that $C_T \models \neg \text{suc}(T_1, T_2)$ while after merging $z_1$ with $z_2$ we obtain $f'_{n+1}(q) \models \text{suc}(T_1, T_2)$. In order to eliminate this ‘defect’ of $f'_{n+1}$, we must do some additional merging.

Suppose that $a \in f'_{n+1}(q)$. On the set of all $\sim f'_{n+1}(q)$-successors of $a$ define the following relation $\equiv$ by setting

$$x \equiv y \text{ if } f'_{n+1}(q) \models T_1(a, x) \land T_2(a, y) \text{ and } C_T \models \neg \text{suc}(T_1, T_2).$$

Let $\equiv$ be the reflexive and transitive closure of $\equiv'$.

Let $\text{merge}_g(f'_{n+1})$ denote the tree subredut of $q$ obtained by gluing in $f'_{n+1}$ the children of $a$ that are equivalent w.r.t. $\equiv$. Note that since $x \equiv y$ implies $g(x) = g(y)$, both $x \equiv y$ and $y \equiv x$ imply that $g(z) = g(x) = g(y)$, hence $\alpha_{n+1}$ is well defined on $\text{merge}_g(f'_{n+1})(q)$.

Clearly, if the subtrees of $f'_{n}(q)$ generated by $a$ and $b$ do not intersect (which, for example, is the case when $a$ and $b$ belong to the same ‘level’), then $\text{merge}_g(f'_{n+1}(q)) = \text{merge}_g(f'_{n+1}(q))$ and can be denoted by $\text{merge}_g(f'_{n+1}(q))$, and therefore we have a well defined operation $\text{merge}_g(f'_{n+1})$ for a finite set $W$ of points of $f'_{n+1}$ with non-intersecting subtrees. By $\text{merge}_g(f'_{n+1})$ we denote $\text{merge}_g(f'_{n+1})$ where $W$ is the set of all points of $f'_{n+1}(q)$ of the level $l$.

Now let $l_2$ be the depth of the tree $f'_{n+1}(q)$, and let the level of $f'_{n+1}(q)$ in $f_{n+1}(q)$ be $l_1$. We set $f_{n+1} = \text{merge}_{l_2}^{l_1}$, where $\text{merge}_{l_2}^{l_1} = f'_{n+1}$, and $\text{merge}_{l_2}^{l_1} = \text{merge}_{l_2}^{l_1}(\text{merge}_{l_2}^{l_1})$.

We claim that $f_{n+1}$ respects roles. In fact, $f_{n+1}$ respects $\text{inv}$ and $\text{suc}$ for the points from different levels because it always does so, and it respects $\text{suc}$ for the points of different levels because from the definition of merge it follows that $\text{merge}_{l_2}^{l_1} = f'_{n+1}$ respects $\text{suc}$ for the points of all levels below $l$, and so $f_{n+1} = \text{merge}_{l_2}^{l_1} \text{suc}$ for all points.

Case 3.2: Suppose $C_T \models \text{inv}(T, S) \land \neg \text{suc}(T, S)$. As $\alpha_n$ is a homomorphism, we have $T(\alpha_n (f_n(r)), \alpha_n (f_n(s))) = g_n(q)$ and, since the diagram commutes, $T(g_n(r), g_n(s)) = g_n(q)$. As $g$ respects the roles in $T$, $g(q) \models \neg \text{suc}(T, S)$, and so we must have $g(s') = g(r)$. So, we set $f_{n+1} = f_n \cup \{ s' \to f_n(r) \}$ and $\alpha_{n+1} = \alpha_n$. Clearly, $||f'_{n+1}(s')|| = ||f_n(s)|| - 1$ and $||g_n(s')|| = ||g_n(s)||$. Then $||f'_{n+1}(s')|| - 1 = ||f_n(s)|| - 1 = ||f_{n+1}(s)||$. The same argument applies to $h_n$ and $h_{n+1}$.

Let us prove that $\alpha_{n+1}$ is a homomorphism. Similarly to Case 2, we conclude that $g(s') = g(r)$ (otherwise $g_{n+1}(q) \models \text{suc}(T, S)$, and so $C_T \models \text{suc}(T, S)$, which is a contradiction).

However, $f'_{n+1}$ may not respect roles, so we have to perform the same operation of ‘merging’ as in Case 3.1, i.e., define $f_{n+1} = \text{merge}_{l_2}^{l_1}$, where $\text{merge}_{l_2}^{l_1} = f'_{n+1}$, and $\text{merge}_{l_2}^{l_1} = \text{merge}_{l_2}^{l_1}(\text{merge}_{l_2}^{l_1})$, where $l_2$ is the depth of $f'_{n+1}(q)$, and $l_1$ is the level of $f'_{n+1}(s)$ in $f_{n+1}(q)$.

The proof that $f_{n+1}$ respects roles is the same as in Case 3.1.

It should be clear that in all cases the above diagram commutes for $i = n + 1$, and that $f_{n+1}$ respects roles.

Now suppose that the process stopped after $k$ steps. Then $\text{dom} g \subseteq \text{dom} h$, and by swapping $g$ and $h$, we also obtain that $\text{dom} h \subseteq \text{dom} g$. It follows that $g(q)$ and $h(q)$ are homomorphic images of the tree subredut $f_k(q)$, which satisfies (2) and (3).

Lemma 16 is obtained by a multiple application of the lemma above: a polynomial-time algorithm constructing a universal tree witness can be extracted from the proof.
Appendix: Proof of Theorem 21

Suppose that tree witnesses $f$ and $g$ are conflicting. This means that we can find terms $\alpha \in f^{-1}(r_f)$, $\beta \in \text{dom } f \setminus \text{dom } g$, $\delta \in (\text{dom } f \cap \text{dom } g) \setminus (f^{-1}(r_f) \cap g^{-1}(r_g))$, $\eta \in \text{dom } g \setminus \text{dom } f$, $\theta \in g^{-1}(r_g)$ and a path $\Gamma$ in $\gamma$ coming through all of them.

Since $\text{dom } f$ is connected, we may assume that its part containing $\alpha$, $\beta$ and $\delta$ belongs to $\text{dom } f$, and that its part from $\delta$ to $\theta$ belongs to $\text{dom } g$. Moreover, it follows from definition of a tree witness that $\text{dom } f \setminus f^{-1}(r_f)$ is connected, and so we may assume that the part of this path from $\beta$ to $\delta$ belongs to $\text{dom } f \setminus f^{-1}(r_f)$.

Since $\beta \notin \text{dom } g$ and $\delta \in \text{dom } g$, there exists a point $\gamma \in g^{-1}(r)$ on the part of this path lying between $\beta$ and $\delta$. Similarly, there exists a point $\zeta \in f^{-1}(r)$ on the part of this path lying between $\delta$ and $\eta$. If there are different possibilities to choose $\gamma$ and $\zeta$, then $\gamma$ and $\zeta$ are supposed to be nearest to $\delta$ (and may be even equal to $\delta$), in particular, this guarantees that the part of $\Gamma$ between $\delta$ and $\zeta$ is in $\text{dom } f$ and the part between $\gamma$ and $\delta$ is in $\text{dom } g$. Note that

$$\gamma \notin f^{-1}(r_f). \quad (8)$$

Let $\Gamma_1$ be the part of $\Gamma$ lying between $\alpha$ and $\zeta$. Let $\Gamma_1 = t_0 \ldots t_n$. Note that every $t_i$ is in $\text{dom } f$, hence we assign to $\Gamma_1$ a word $\sigma = \sigma_1 \ldots \sigma_n$ over the alphabet $\{(,)\}$ as follows:

$$\sigma_i = \begin{cases} (, \text{ if } f(t_i) \text{ is further from } r_f \text{ than } f(t_{i-1}) \text{ in } C_T; \\ ), \text{ if } f(t_i) \text{ is closer to } r_f \text{ than } f(t_{i-1}) \text{ in } C_T. \end{cases}$$

Clearly, $\sigma$ is a balanced word of parentheses. We denote by $\sigma'$ the suffix of $\sigma$ corresponding to the part of the path between $\gamma$ and $\zeta$.

Similarly, let $\rho$ be the word over the alphabet $\{(,)\}$ corresponding to the part of the path between $\gamma$ and $\theta$ generated by $g$, that is:

$$\rho_i = \begin{cases} (, \text{ if } g(t_i) \text{ is further from } r_g \text{ than } g(t_{i-1}) \text{ in } C_T; \\ ), \text{ if } g(t_i) \text{ is closer to } r_g \text{ than } g(t_{i-1}) \text{ in } C_T. \end{cases}$$

Let $\rho'$ be the prefix of $\rho$ corresponding to the part of the path between $\gamma$ and $\zeta$.

It is sufficient to show that $\rho' \neq \sigma'$ (if this is the case then, for some $j$, $\rho_j' \neq \sigma_j'$ and any role connecting $t_j \ldots t_{j-1}$ and $t_j'$ with a corresponding $j'$ is a twisty role.)

To this end, note that $\sigma'$ has strictly more closing parentheses than opening brackets (not less because $\sigma'$ is a suffix of a balanced parentheses word; strictly more because of (8)). On the other hand, $\rho'$ has not less opening brackets than closing brackets. This two inequalities together contradict $\rho' = \sigma'$.

Notice also that the length of $\sigma'$ and $\rho'$ is positive, since if it is equal to zero, then $\gamma = \delta = \zeta$ and, in particular, $\delta \in f^{-1}(r_f) \cap g^{-1}(r_g)$, contrary to the choice of $\delta$. 