On the Parameterised Complexity of Tree-Shaped Ontology-Mediated Queries in *OWL 2 QL*

M. Bienvenu[1], S. Kikot[2], R. Kontchakov[2], V. Ryzhikov[3], and M. Zakharyaschev[2]

[1] CNRS & University of Montpellier, France (`meghyn@lirmm.fr`)
[2] Birkbeck, University of London, UK (`{kikot, roman, michael}@dcs.bbk.ac.uk`)
[3] Free University of Bozen-Bolzano, Italy (`ryzhikov@inf.unibz.it`)

Abstract. We discuss the parameterised complexity of answering tree-shaped ontology-mediated queries (OMQs) in *OWL 2 QL* under various restrictions on their ontologies and conjunctive queries (CQs). In particular, we construct an ontology $\mathcal{T}$ such that answering OMQs $(\mathcal{T}, \boldsymbol{q})$ with tree-shaped CQs $\boldsymbol{q}$ is W[1]-hard if the number of leaves in $\boldsymbol{q}$ is regarded as the parameter. The number of leaves has previously been identified as an important characteristic of CQs as bounding it leads to tractable OMQ answering. Our result shows that treating it as a parameter does not make the problem fixed-parameter tractable, even for a fixed ontology.

1 Introduction

Our concern here is the computational complexity of answering ontology-mediated queries (OMQs, for short) $(\mathcal{T}, \boldsymbol{q})$, where $\mathcal{T}$ is an *OWL 2 QL* ontology (aka *DL-Lite*$_{\mathcal{R}}$ [7] or *DL-Lite*$^{\mathcal{H}}_{core}$ TBox [2]) and $\boldsymbol{q}$ a *tree-shaped* Boolean conjunctive query (CQ). The classical OMQ answering problem TREEOMQ

Instance: $\mathcal{T}$, $\boldsymbol{q}$ and an ABox $\mathcal{A}$,
Problem: decide whether $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$

is known to be NP-complete [15]. On the other hand, the restriction of TREEOMQ to tree-shaped CQs with at most $\boldsymbol{\ell}$ leaves, for some fixed number $\boldsymbol{\ell} \geq 2$—we denote this problem by TREEOMQ[*leaves* $\leq \boldsymbol{\ell}$]—is LOGCFL-complete [5]. If we further bound the existential depth of $\mathcal{T}$ by some fixed $\boldsymbol{d} \geq 0$, then the resulting problem TREEOMQ[*leaves* $\leq \boldsymbol{\ell}$, *depth* $\leq \boldsymbol{d}$] turns out to be NL-complete [5].

Results of this kind prompt a few additional interesting questions. For example, what is the *parameterised* complexity of the following problem *depth*-TREEOMQ?

Instance: $\mathcal{T}$ of finite depth, $\boldsymbol{q}$ and $\mathcal{A}$,
Parameter: the depth of $\mathcal{T}$,
Problem: decide whether $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$.

What is the parameterised complexity of a similar problem *leaves*-TREEOMQ, which takes the number of leaves in $\boldsymbol{q}$ as the parameter? Or what if we fix an ontology $\mathcal{T}$ in TREEOMQ and consider the problem TREEOMQ[$\mathcal{T}$]? Note that this problem reflects a typical ontology-based data access scenario, where the users are provided with a fixed ontology designed by a domain expert. Furthermore, we can consider the leaf-parameterisation *leaves*-TREEOMQ[$\mathcal{T}$] of TREEOMQ[$\mathcal{T}$].

In this paper, we summarise and discuss our recent results answering some of the questions above and presented in [4]. We also prove a new theorem by constructing an ontology $\mathcal{T}_\Box$ (of infinite depth) for which the problem *leaves*-TREEOMQ$[\mathcal{T}_\Box]$ is $W[1]$-hard.

2 Preliminaries

To make the ontology axioms in Section 3 more compact and readable, we use the syntax of first-order rather than description logic. Thus, an *OWL 2 QL ontology*, $\mathcal{T}$, is a finite set of sentences of the form

$$\begin{array}{ll}
\forall x\,(\tau(x) \to \tau'(x)), & \forall x\,(\tau(x) \land \tau'(x) \to \bot),\\
\forall xy\,(\varrho(x,y) \to \varrho'(x,y)), & \forall xy\,(\varrho(x,y) \land \varrho'(x,y) \to \bot),\\
\forall x\,\varrho(x,x), & \forall x\,(\varrho(x,x) \to \bot),
\end{array}$$

where $\tau(x)$ and $\varrho(x,y)$ are defined, using unary predicates A and binary predicates P, by the grammars

$$\begin{array}{rcl}
\tau(x) & ::= & \top \;\mid\; A(x) \;\mid\; \exists y\,\varrho(x,y),\\
\varrho(x,y) & ::= & \top \;\mid\; P(x,y) \;\mid\; P(y,x).
\end{array}$$

When writing ontology axioms, we omit the universal quantifiers. Denote by $\boldsymbol{R}_\mathcal{T}$ the set of binary predicates P occurring in $\mathcal{T}$ and their inverses P^-, assuming that $P^{--} = P$. An *ABox*, $\mathcal{A}$, is a finite set of unary or binary ground atoms. We denote by $\mathsf{ind}(\mathcal{A})$ the set of individual constants in $\mathcal{A}$.

A *conjunctive query* (CQ) $\boldsymbol{q}(\boldsymbol{x})$ is a formula of the form $\exists \boldsymbol{y}\,\varphi(\boldsymbol{x},\boldsymbol{y})$, where φ is a conjunction of (unary $A(u)$ or binary $P(u,v)$) atoms $S(\boldsymbol{z})$ all of whose variables are among $\boldsymbol{x} \cup \boldsymbol{y}$. We assume, without loss of generality, that CQs contain no constants. We often regard a CQ as the set of its atoms.With every CQ $\boldsymbol{q}$, we associate its *Gaifman graph* $\mathcal{G}$ whose vertices are the variables of $\boldsymbol{q}$ and whose edges are the pairs $\{u,v\}$ such that $P(u,v) \in \boldsymbol{q}$, for some P (note that the atoms $P(u,u)$ do not add any edges to $\mathcal{G}$). We call $\boldsymbol{q}$ *connected* if $\mathcal{G}$ is connected; $\boldsymbol{q}$ is *tree-shaped* if $\mathcal{G}$ is a tree, and *linear* if $\mathcal{G}$ is a tree with two leaves.

An *ontology-mediated query* (OMQ) is a pair $\boldsymbol{Q}(\boldsymbol{x}) = (\mathcal{T}, \boldsymbol{q}(\boldsymbol{x}))$, where $\mathcal{T}$ is an ontology and $\boldsymbol{q}(\boldsymbol{x})$ a CQ. A tuple $\boldsymbol{a}$ in $\mathsf{ind}(\mathcal{A})$ is a *certain answer* to $(\mathcal{T}, \boldsymbol{q})$ over an ABox $\mathcal{A}$ if $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a})$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$; in this case we write $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}(\boldsymbol{a})$. For a *Boolean* $\boldsymbol{q}$, in which case $\boldsymbol{x} = \emptyset$, a certain answer to $\boldsymbol{Q}$ over $\mathcal{A}$ is 'yes' if $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$ and 'no' otherwise.

Every consistent *knowledge base* (KB) $(\mathcal{T}, \mathcal{A})$ has a *canonical model* (or *chase* in database theory) [1] $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ with the property that $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}(\boldsymbol{a})$ iff $\mathcal{C}_{\mathcal{T},\mathcal{A}} \models \boldsymbol{q}(\boldsymbol{a})$, for all CQs $\boldsymbol{q}(\boldsymbol{x})$ and $\boldsymbol{a}$ in $\mathsf{ind}(\mathcal{A})$. In our constructions, we use the following definition of $\mathcal{C}_{\mathcal{T},\mathcal{A}}$, where without loss of generality we assume that $\mathcal{T}$ contains no binary predicates P such that $\mathcal{T} \models \forall xy\, P(x,y)$. The domain, $\Delta^{\mathcal{C}_{\mathcal{T},\mathcal{A}}}$, consists of $\mathsf{ind}(\mathcal{A})$ and the *witnesses* (or *labelled nulls*) of the form $w = a\varrho_1 \dots \varrho_n$, for $n \geq 1$, such that

- $a \in \mathsf{ind}(\mathcal{A})$ and $\mathcal{T}, \mathcal{A} \models \exists y\, \varrho_1(a,y)$;

– $\mathcal{T} \not\models \varrho_i(x, x)$, for $1 \leq i \leq n$;
– $\mathcal{T} \models \exists x \varrho_i(x, y) \to \exists z \varrho_{i+1}(y, z)$ but $\mathcal{T} \not\models \varrho_i(x, y) \to \varrho_{i+1}(y, x)$, for $1 \leq i < n$.

We denote by $W_\mathcal{T}$ the set consisting of the empty word ε and all non-empty words $\varrho_1 \dots \varrho_n \in \boldsymbol{R}_\mathcal{T}^+$ satisfying the last two conditions. Every $a \in \mathsf{ind}(\mathcal{A})$ is interpreted in $\mathcal{C}_{\mathcal{T},\mathcal{A}}$ by itself, and unary and binary predicates are interpreted as follows:

– $\mathcal{C}_{\mathcal{T},\mathcal{A}} \models A(u)$ iff either $u \in \mathsf{ind}(\mathcal{A})$ and $\mathcal{T}, \mathcal{A} \models A(u)$, or $u = w\varrho$ and we have $\mathcal{T} \models \exists y\, \varrho(y, x) \to A(x)$;
– $\mathcal{C}_{\mathcal{T},\mathcal{A}} \models P(u, v)$ iff one of the following three conditions holds: (*i*) $u, v \in \mathsf{ind}(\mathcal{A})$ and $\mathcal{T}, \mathcal{A} \models P(u, v)$; (*ii*) $u = v$ and $\mathcal{T} \models P(x, x)$; (*iii*) $\mathcal{T} \models \varrho(x, y) \to P(x, y)$ and either $v = u\varrho$ or $u = v\varrho^-$.

We say that $\mathcal{T}$ is *of depth* $0 \leq \boldsymbol{d} < \infty$ if $\boldsymbol{d}$ is the maximum length of the words in $W_\mathcal{T}$, and *of depth* ∞ if $W_\mathcal{T}$ is infinite. (Note that the depth of $\mathcal{T}$ is computable in NL; cf. [12, 6] for related results on chase termination for tgds.)

We consider various parameterisations and restrictions of the decision problem TREEOMQ defined above. As *parameters*, we can take the following[1]:

query:	a tree-shaped CQ $\boldsymbol{q}$,
leaves:	the number of leaves in the Gaifman graph of a tree-shaped CQs $\boldsymbol{q}$,
ontology:	an *OWL 2 QL* ontology $\mathcal{T}$,
depth:	the depth of $\mathcal{T}$.

Restrictions can take the forms

leaves $\leq \boldsymbol{\ell}$:	we consider only CQs with $\leq \boldsymbol{\ell}$ leaves, for some $\boldsymbol{\ell} \geq 2$,
depth $\leq \boldsymbol{d}$:	we consider only ontologies of depth $\leq \boldsymbol{d}$, for some $\boldsymbol{d} \geq 0$,
query $= \boldsymbol{q}$:	we fix the tree-shaped CQ $\boldsymbol{q}$,
ontology $= \mathcal{T}$:	we fix the ontology $\mathcal{T}$.

The decision problems we are interested in look as follows:

$$\textit{parameter}\text{-}\textsc{TreeOMQ}[\textit{restriction}_1, \dots, \textit{restriction}_n],$$

where *parameter* is one of the parameters above or blank, and each *restriction*$_i$, if any, is one of the restrictions above. To simplify notation, instead of [*ontology* $= \mathcal{T}$] we write $[\mathcal{T}]$ and similarly for [*query* $= \boldsymbol{q}$]. For example, *leaves*-TREEOMQ$[\mathcal{T}]$ is the problem

Instance:	$\boldsymbol{q}$ and $\mathcal{A}$,
Parameter:	the number of leaves in $\boldsymbol{q}$,
Problem:	decide whether $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$.

We now construct an ontology $\mathcal{T}$ for which this problem is $W[1]$-hard; for any results from parameterised complexity theory we use below, consult, e.g., [11].

[1] Following Downey [9], our parameters are not necessarily numerical.

3 *leaves*-TREEOMQ$[\mathcal{T}]$ can be $W[1]$-hard

Theorem 1. *There is an ontology $\mathcal{T}_\Box$ such that leaves-*TREEOMQ$[\mathcal{T}_\Box]$ *is $W[1]$-hard.*

Proof. The proof is by reduction of the $W[1]$-hard problem *SquareTiling* [11], which is defined as follows:

Instance:	a set $\mathfrak{T}$ of tile types painted in colours from a set $\mathfrak{C}$, a positive integer k,
Parameter:	k,
Problem:	decide whether $\mathfrak{T}$ tiles a $k \times k$-grid.

Suppose $\mathfrak{C} = \{0, \dots, n\}$, for $n \geq 1$, and $\mathfrak{T} = \{\mathcal{S}_1, \dots, \mathcal{S}_m\}$. Denote by $\mathit{right}(t)$, $\mathit{left}(t)$, $\mathit{top}(t)$ and $\mathit{bottom}(t)$ the right, left, top and bottom colour of $\mathcal{S}_t$, respectively. Given a binary predicate name R, we denote by R^i a sequence of i-many predicates R. We represent each colour $c \leq n$ by the following two sequences of binary predicates:

$$\mathtt{enc}_c = P^{3n-c}F^c \qquad \text{of length } 3n,$$

$$\mathtt{cne}_c = N^{2c}MN^{2(n-c)} \qquad \text{of length } 2n+1.$$

Examples of $\mathtt{enc}_c$ and $\mathtt{cne}_c$, for $n = 4$ and $c = 3$, are shown in Figs. 2a and 2d, respectively. Each tile $\mathcal{S}_t$ is represented by the following sequence of binary predicates:

$$\mathtt{tile}_t = B\,\mathtt{cne}_{\mathit{right}(t)}\,\mathtt{enc}_{\mathit{left}(t)}\,\mathtt{cne}_{\mathit{top}(t)}\,\mathtt{enc}_{\mathit{bottom}(t)}E \qquad \text{of length } 10n+4.$$

Since the length of $\mathtt{enc}_c$ does not depend on c, we use $|\mathtt{enc}|$ to denote the length of some (any) $\mathtt{enc}_c$, and similarly for $|\mathtt{cne}|$ and $|\mathtt{tile}|$. We shall also require the following sequences

$$\begin{aligned}
\mathtt{segm} &= \mathtt{tile}_1\,\mathtt{tile}_2 \dots \mathtt{tile}_m\,S && \text{of length } m \cdot |\mathtt{tile}| + 1,\\
\mathtt{spring} &= (XYI)^n && \text{of length } |\mathtt{enc}|,\\
\mathtt{probelt} &= I^{|\mathtt{cne}|+|\mathtt{enc}|+2}\,\mathtt{spring}\,I^{|\mathtt{tile}|+1}I^{2n}M^- && \text{of length } 2 \cdot |\mathtt{tile}| + 1,\\
\mathtt{probedn} &= I^2\,\mathtt{spring}\,I^{(|\mathtt{tile}|+1)k}I^{2n}M^- && \text{of length } 5n+3+(|\mathtt{tile}|+1)k.
\end{aligned}$$

The first sequence will be called a *segment* and the second a *spring*.

The Boolean CQ $\boldsymbol{q}$ (see Fig. 1) is now defined by taking the following set of atoms (assuming that all variables are existentially quantified):

$$\{A(x_0)\} \cup \mathtt{segm}(x_0, x_{1,1}) \cup \bigcup_{\substack{2\leq i\leq k\\1\leq j\leq k}} \mathtt{segm}(x_{i-1,j}, x_{i,j}) \cup \bigcup_{2\leq j\leq k} \mathtt{segm}(x_{k,j-1}, x_{1,j}) \cup$$
$$\bigcup_{\substack{2\leq i\leq k\\1\leq j\leq k}} \mathtt{probelt}(x_{i,j}, y_{i,j}) \cup \bigcup_{\substack{1\leq i\leq k\\2\leq j\leq k}} \mathtt{probedn}(x_{i,j}, z_{i,j}),$$

where $R_1 \dots R_l(x, y)$ stands for $\{R_1(x, x_1), R_2(x_1, x_2), \dots, R_l(x_{l-1}, y)\}$ with fresh variables $x_1, \dots, x_{l-1}$. It can be seen that $\boldsymbol{q}$ is a tree-shaped CQ with $2(k-1)k$ leaves.

Let $\mathcal{T}_\Box$ be an ontology with the following axioms:

$$A(x) \to \exists y\,\big(BI(x,y) \land \mathit{Right}(y)\big), \qquad A(x) \to \exists u\,\mathit{Sink}(x,u),$$

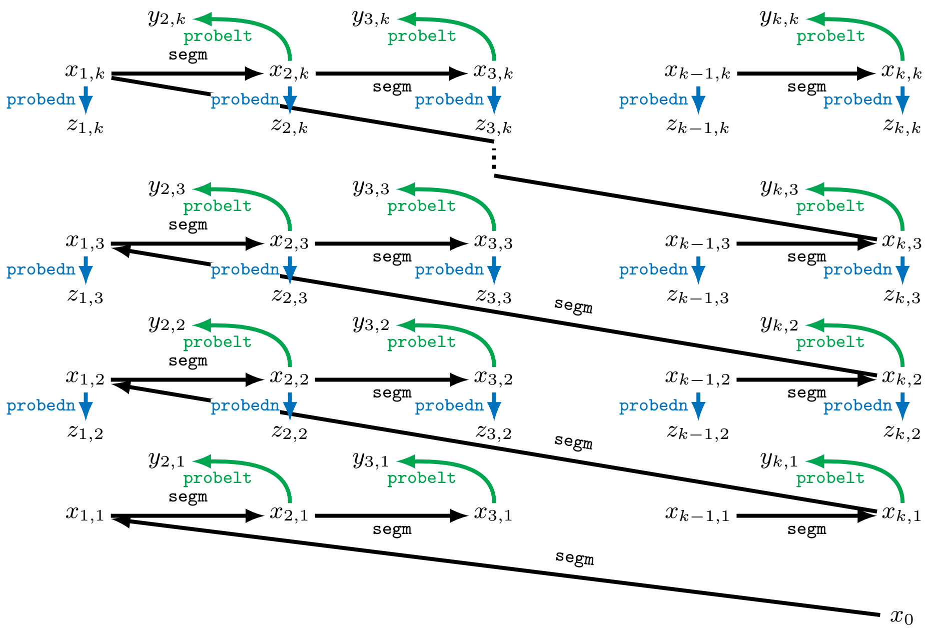

Fig. 1. The structure of the CQ $\boldsymbol{q}$.

$$
\begin{aligned}
\mathit{Right}(x) &\to \exists y\,\big(\mathit{NI}(x,y) \land \mathit{Right}(y)\big), & \mathit{Right}(x) &\to \exists z\,\big(\mathit{MI}(x,z) \land \mathit{Right}'(z)\big),\\
\mathit{Right}'(x) &\to \exists y\,\big(\mathit{NI}(x,y) \land \mathit{Right}'(y)\big), & \mathit{Right}'(x) &\to \mathit{Left}(x),\\
\mathit{Left}(x) &\to \exists y\,\big(\mathit{PI}(x,y) \land \mathit{Left}(y)\big), & \mathit{Left}(x) &\to \exists z\,\big(\mathit{FI}(x,z) \land \mathit{Left}'(z)\big),\\
\mathit{Left}'(x) &\to \exists y\,\big(\mathit{FI}(x,y) \land \mathit{Left}'(y)\big), & \mathit{Left}'(x) &\to \mathit{Top}(x),\\
\mathit{Top}(x) &\to \exists y\,\big(\mathit{NI}(x,y) \land \mathit{Top}(y)\big), & \mathit{Top}(x) &\to \exists z\,\big(\mathit{MI}(x,z) \land \mathit{Top}'(z)\big),\\
\mathit{Top}'(x) &\to \exists y\,\big(\mathit{NI}(x,y) \land \mathit{Top}'(y)\big), & \mathit{Top}'(x) &\to \mathit{Bot}(x),\\
\mathit{Bot}(x) &\to \exists y\,\big(\mathit{PI}(x,y) \land \mathit{Bot}(y)\big), & \mathit{Bot}(x) &\to \exists z\,\big(\mathit{FI}(x,z) \land \mathit{Bot}'(z)\big),\\
\mathit{Bot}'(x) &\to \exists y\,\big(\mathit{FI}(x,y) \land \mathit{Bot}'(y)\big), & \mathit{Bot}'(x) &\to \exists z\,\big(\mathit{EI}(x,z) \land A_2(z)\big),\\
A_2(z) &\to \exists x\,\big(\mathit{SI}(z,x) \land A(x)\big), & A_2(z) &\to \exists u\,\mathit{Sink}(z,u),\\
\mathit{Left}'(x) &\to \exists y\,X\bar{Y}(x,y), & \mathit{Bot}'(x) &\to \exists y\,X\bar{Y}(x,y),\\
P(x,y) &\to X(y,x), & P(x,y) &\to Y(y,x),\\
X\bar{Y}(x,y) &\to X(x,y), & X\bar{Y}(x,y) &\to Y(y,x),
\end{aligned}
$$

where $C(x) \to \exists y\,\big(Q(x,y) \land D(y)\big)$ abbreviates three axioms

$$C(x) \to \exists y\,Q_D(x,y), \qquad Q_D(x,y) \to Q(x,y) \quad\text{and}\quad Q_D(x,y) \to D(y).$$

In addition, $\mathcal{T}_\Box$ contains the axioms

– $QI(x,y) \to Q(x,y)$ and $QI(x,y) \to I(y,x)$, for all predicates of the form QI;

– $\mathit{Sink}(x, y) \to Q(x, y)$ and $\mathit{Sink}(x, y) \to Q(y, x)$, for all binary predicates Q except I and S.

A path in the canonical model $\mathcal{C}_{\mathcal{T}_\Box, \{A(a)\}}$ where $\boldsymbol{q}$ can be homomorphically mapped is shown in Fig. 2 sandwiched between $\mathsf{segm}(x_0, x_{1,1}) \cup \mathsf{segm}(x_{1,1}, x_{2,1})$ on the left and bottom and $\mathsf{probelt}(x_{2,1}, y_{2,1})$ on the right and top (most predicate names are omitted).

We show that $\mathcal{T}_\Box, \{A(a)\} \models \boldsymbol{q}$ iff $\mathfrak{T}$ tiles a $k \times k$-grid. Here, we only prove ($\Rightarrow$) and leave the converse direction to the reader. For two sequences $w = R_1 \dots R_l$ and $w' = R'_1 \dots R'_l$ of binary predicate names, we write $w \sqsubseteq w'$ if $\mathcal{T}_\Box \models R_i(x, y) \to R'_i(x, y)$, for all i $(1 \le i \le l)$.

Let h be a homomorphism from $\boldsymbol{q}$ to $\mathcal{C} = \mathcal{C}_{\mathcal{T}_\Box, \{A(a)\}}$ such that $h(x_0) = v_0$ (cf. the definition of labelled nulls in Section 2). Then $v_0 \in A^{\mathcal{C}}$ and $h(x_{1,1}) = v_{1,1} \in A^{\mathcal{C}}$ with $v_{1,1}$ of the form $v_0 w_{1,1} S$, for some $w_{1,1}$ that begins with B but does not contain S. Since $(v_0, v_0 \mathit{Sink}) \notin S^{\mathcal{C}}$ and $|\mathtt{tile}|$ is even, it follows that there is a unique tile $t_{1,1}$ such that

– $|w_{1,1}| = |\mathtt{tile}|$ and $w_{1,1} \sqsubseteq \mathtt{tile}_{t_{1,1}}$,
– the subquery of $\mathtt{segm}(x_0, x_{1,1})$ for the sequence $\mathtt{tile}_1 \dots \mathtt{tile}_{t_{1,1}-1}$ is mapped to the Sink arrow at v_0 (i.e., forwards and backwards between v_0 and $v_0 \mathit{Sink}$);
– the subquery for the sequence $\mathtt{tile}_{t_{1,1}+1} \dots \mathtt{tile}_m$ is mapped to a Sink arrow at $v_0 w_{1,1}$ (see Fig. 2).

Consider now any subquery $\mathtt{segm}(x_{i-1,1}, x_{i,1})$, for $2 \le i \le k$. By the same argument, we obtain $h(x_{i,1}) = v_{i,1}$, for $v_{i,1} = v_{i-1,1} w_{i,1} S$, and $w_{i,1} \sqsubseteq \mathtt{tile}_{t_{i,1}}$, for a unique $1 \le t_{i,1} \le m$. Next, $h(x_{1,2}) = v_{1,2}$ for $v_{1,2} = v_{k,1} w_{1,2} S$ with $w_{1,2} \sqsubseteq \mathtt{tile}_{t_{1,2}}$ and, eventually, every subquery $\mathtt{segm}(x_{i-1,j}, x_{i,j})$, for $1 \le i \le k$ and $1 \le j \le k$, is mapped in such a way that

$$h(x_{i,j}) = \begin{cases} v_{i-1,j} w_{i,j} S, & \text{if } i \ge 2, \\ v_{k,j-1} w_{i,j} S, & \text{if } i = 1, j \ge 2, \\ v_0 w_{1,1} S, & \text{if } i = 1, j = 1. \end{cases} \quad \text{with} \quad w_{i,j} \sqsubseteq \mathtt{tile}_{t_{i,j}}, \text{ for a unique } t_{i,j},$$

We prove now that the tiles $\mathcal{S}_{t_{i,j}}$ placed at (i, j) of the $k \times k$-grid form a tiling.

First, we show that $\mathit{left}(t_{i,j}) = \mathit{right}(t_{i-1,j})$, for $2 \le i \le k$ and $1 \le j \le k$. As we observed above, $h(x_{i,j})$ is of the form $v\, w_{i-1,j}\, S\, w_{i,j}\, S$. Consider the subquery $\mathtt{probelt}(x_{i,j}, y_{i,j})$ and recall that

$$w_{i,j} \sqsubseteq B\, \mathtt{cne}_{\mathit{right}(t_{i,j})}\, \mathtt{enc}_{\mathit{left}(t_{i,j})}\, \mathtt{cne}_{\mathit{top}(t_{i,j})}\, \mathtt{enc}_{\mathit{bottom}(t_{i,j})}\, E.$$

By the structure of $\mathcal{T}_\Box$, the subqueries $I^{|\mathsf{enc}|+|\mathsf{cne}|+2}(x_{i,j}, v_{i,j})$ and $\mathtt{spring}(v_{i,j}, u_{i,j})$ of $\mathtt{probelt}(x_{i,j}, y_{i,j})$ are mapped by h in such a way (see Fig. 2) that

$$h(v_{i,j}) \sqsubseteq v\, w_{i-1,j}\, S\, B\, \mathtt{cne}_{\mathit{right}(t_{i,j})}\, \mathtt{enc}_{\mathit{left}(t_{i,j})},$$
$$h(u_{i,j}) \sqsubseteq v\, w_{i-1,j}\, S\, B\, \mathtt{cne}_{\mathit{right}(t_{i,j})}\, P^{2c}, \qquad \text{for } c = \mathit{left}(t_{i,j}).$$

On the other hand, the last element in $\mathtt{probelt}$ is M^-, and so we must have

$$h(y_{i,j}) \sqsubseteq v\, B\, N^{2c'}, \qquad \text{for } c' = \mathit{right}(t_{i-1,j}),$$

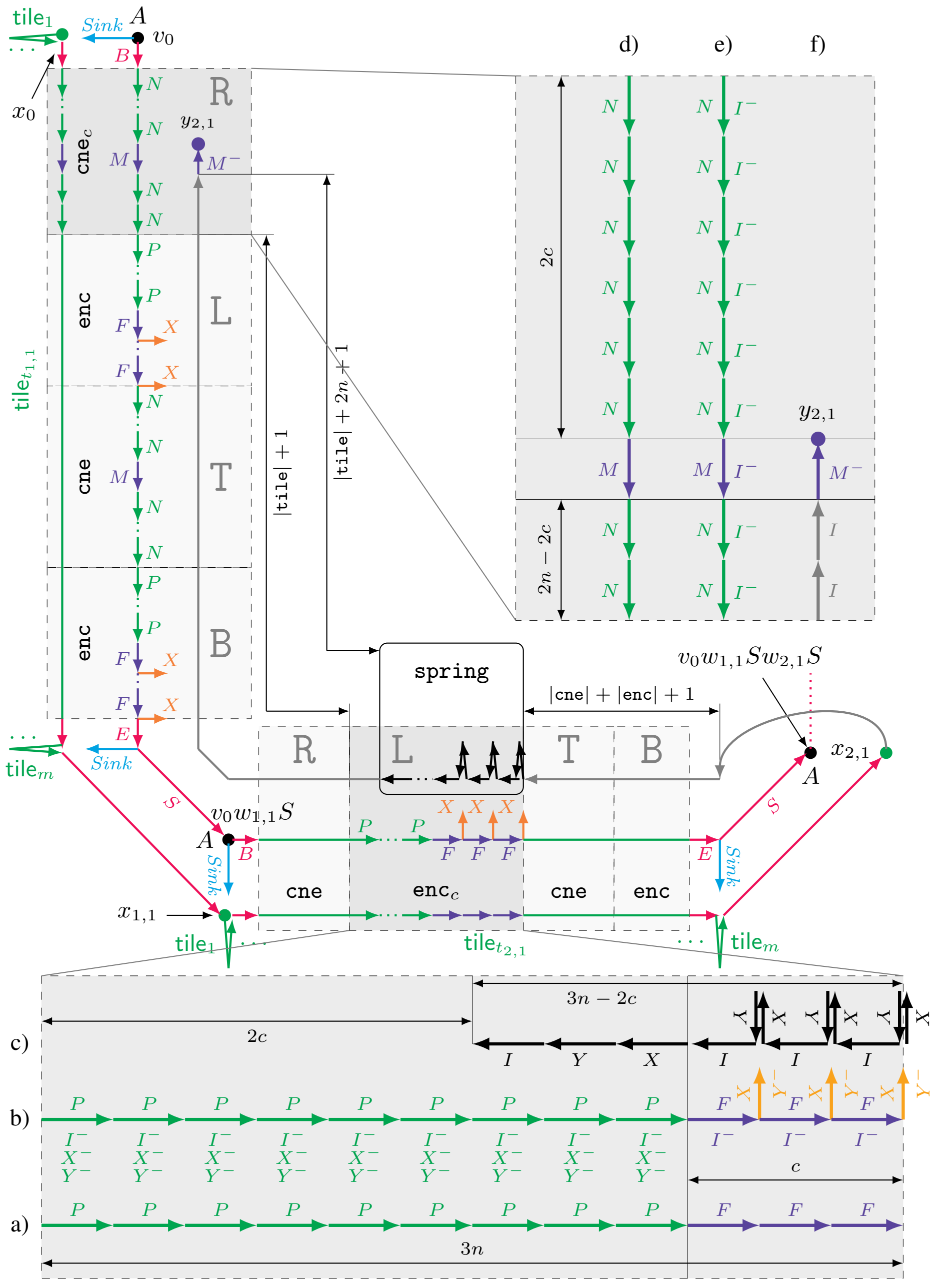

Fig. 2. Matching the first two segments, $\texttt{segm}(x_0, x_{1,1})$ and $\texttt{segm}(x_{1,1}, x_{2,1})$, of $\boldsymbol{q}$ and $\texttt{probelt}(x_{2,1}, y_{2,1})$ in the canonical model $\mathcal{C}$, and the magnified fragments for $\texttt{enc}_c$ and $\texttt{cne}_c$ with $n = 4$ and $c = 3$: a) subsequence $\texttt{enc}_c$ of $\texttt{tile}_t$ in $\boldsymbol{q}$; b) a path in $\mathcal{C}$ where $\texttt{enc}_c$ is mapped; c) matching the $\texttt{spring}$ subquery of $\texttt{probelt}$ in $\mathcal{C}$; d) subsequence $\texttt{cne}_c$ of $\texttt{tile}_t$ in $\boldsymbol{q}$; e) a path in $\mathcal{C}$ where $\texttt{cne}_c$ is mapped; f) mapping variable $y_{2,1}$ of the subquery $\texttt{probelt}(x_{2,1}, y_{2,1})$ in $\mathcal{C}$.

which is only possible if $\mathit{right}(t_{i-1,j}) = \mathit{left}(t_{i,j})$; see Fig. 2.

That $\mathit{down}(t_{i,j}) = \mathit{up}(t_{i,j-1})$, for $2 \leq j \leq k$ and $1 \leq i \leq k$, is proved similarly by considering the mapping of the subquery $\mathtt{probedn}(x_{i,j}, z_{i,j})$.

4 Complexity Landscape

In this concluding section, we summarise what is known about the complexity of the decision problems introduced in Section 2. In the table below, parameters are listed horizontally, while restrictions on ontologies and CQs vertically in the first two columns, where '—' means no restriction or parameter, FPT† indicates that FTP follows from tractability, while in the grey areas, the problems are trivial.

restrictions on		parameter				
$\mathcal{T}$	$\boldsymbol{q}$	— (combined complexity)	query	leaves	ontology	depth
—	—	NP-complete	FPT	W[1]-hard	paraNP see $(*)$	W[2]-hard
	fixed	NL-complete			FPT†	FPT†
	leaves $\leq \boldsymbol{\ell}$	LogCFL-complete	FPT†		FPT†	FPT†
fixed	—	in NP NP-hard for some $\mathcal{T}$ $(*)$	FPT	W[1]-hard for some $\mathcal{T}$		
	fixed	in AC^0				
	leaves $\leq \boldsymbol{\ell}$	in LogCFL LogCFL-hard for some $\mathcal{T}$, $\boldsymbol{\ell} = 2$	FPT†			
depth $\leq \boldsymbol{d}$	—	LogCFL-complete	FPT†	FPT†	FPT†	
	fixed	NL-complete			FPT†	
	leaves $\leq \boldsymbol{\ell}$	NL-complete	FPT†		FPT†	

Recall first that the basic problem TreeOMQ of answering tree-shaped OMQs is NP-complete [15], while standard evaluation of tree-shaped CQs—that is, TreeOMQ$[\emptyset]$—is LogCFL-complete [13]. On the other hand, TreeOMQ$[\mathcal{T}, \boldsymbol{q}]$ is in AC^0, for any $\mathcal{T}$ and $\boldsymbol{q}$ [7, 2], which matches TreeOMQ$[\emptyset, \boldsymbol{q}]$. Observe also that NL-completeness of TreeOMQ$[\boldsymbol{q}]$ matches the complexity of reasoning in *OWL 2 QL* [7, 2]. The parameterised problem *query*-TreeOMQ is fixed-parameter tractable [16, Theorem 21], which means that TreeOMQ can be solved in time $f(|\boldsymbol{q}|) \cdot \mathit{poly}(|\mathcal{T}|, |\mathcal{A}|)$, for some computable f (here $|\cdot|$ is the size of an encoding of $\cdot$).

The intractability result for TreeOMQ has recently been refined in [4, Theorem 20], which constructed an ontology $\mathcal{T}_\dagger$ (of infinite depth) such that TreeOMQ$[\mathcal{T}_\dagger]$ is NP-complete. It follows that—unless P = NP—no algorithm can solve TreeOMQ in time $\mathit{poly}(|\boldsymbol{q}|, |\mathcal{A}|)^{f(|\mathcal{T}|)}$, for any computable function f. Such a complexity bound would usually be regarded as an indication that, in practice, TreeOMQ could be solved efficiently for small $\mathcal{T}$. Thus, *ontology*-TreeOMQ (actually, with any parameter determined by the ontology) cannot be FPT.

Tractability can be restored by restricting the number of leaves in $\boldsymbol{q}$ and/or the depth of $\mathcal{T}$: as shown in [5], both TREEOMQ[*leaves* $\leq \boldsymbol{\ell}$] and TREEOMQ[*depth* $\leq \boldsymbol{d}$] are LOGCFL-complete, while TREEOMQ[*leaves* $\leq \boldsymbol{\ell}$, *depth* $\leq \boldsymbol{d}$] is NL-complete. (In fact, for ontologies of bounded depth, tree-shaped CQs can be generalised to CQs of bounded treewidth.) Moreover, [4] presented an ontology $\mathcal{T}_\ddagger$ such that the problem TREEOMQ[$\mathcal{T}_\ddagger$, *leaves* ≤ 2] is LOGCFL-complete. Yet, *leaves*-TREEOMQ turns out to be W[1]-hard and *depth*-TREEOMQ W[2]-hard [4]. In Section 3, we have sharpened the former result by constructing $\mathcal{T}_\Box$ (of infinite depth) such that *leaves*-TREEOMQ[$\mathcal{T}_\Box$] is W[1]-hard.

In fact, answering OMQs $(\mathcal{T}, \boldsymbol{q})$ with $\boldsymbol{q}$ having at most $\boldsymbol{\ell}$ leaves can be done in time $\mathit{poly}(|\mathcal{T}|, |\boldsymbol{q}|^{\boldsymbol{\ell}}, |\mathcal{A}|^{\boldsymbol{\ell}})$ [4]. The W[1]-hardness results mean, however, that this upper bound cannot be improved—unless $W[1] = \text{FPT}$—to $f(\boldsymbol{\ell}) \cdot \mathit{poly}(|\boldsymbol{q}|, |\mathcal{T}|, |\mathcal{A}|)$ or even to $f(\boldsymbol{\ell}) \cdot \mathit{poly}(|\boldsymbol{q}|, |\mathcal{A}|)^{g(|\mathcal{T}|)}$, as Theorem 1 suggests. Answering OMQs $(\mathcal{T}, \boldsymbol{q})$ with $\mathcal{T}$ of depth $\boldsymbol{d} < \infty$ can be done in time $\mathit{poly}(|\mathcal{T}|^{\boldsymbol{d}}, |\boldsymbol{q}|, |\mathcal{A}|)$ [4]. However—unless $W[2] = \text{FPT}$—it is impossible to improve this to $f(\boldsymbol{d}) \cdot \mathit{poly}(|\boldsymbol{q}|, |\mathcal{T}|, |\mathcal{A}|)$. Thus, we can expect reasonable efficiency for ontologies of small depth, but no general scalability. This does not appear to be a serious restriction as our experience shows that ontologies used for real-world ontology-based data access are of depth at most 5 (see also [8]). On another positive note, the tractable problems TREEOMQ[*leaves* $\leq \boldsymbol{\ell}$], TREEOMQ[*depth* $\leq \boldsymbol{d}$] and TREEOMQ[*leaves* $\leq \boldsymbol{\ell}$, *depth* $\leq \boldsymbol{d}$] can be solved using theoretically optimal resources (LOGCFL or NL) by means of OMQ rewriting to nonrecursive datalog queries; for details and experiments, consult [4].

A challenging open problem is to classify *OWL 2 QL* ontologies $\mathcal{T}$ according to the combined complexity of answering OMQs $(\mathcal{T}, \boldsymbol{q})$ with tree-shaped or arbitrary CQs $\boldsymbol{q}$. In particular, are there interesting conditions on $\mathcal{T}$ that ensure tractability of TREEOMQ[$\mathcal{T}$]? Is it the case that, for every $\mathcal{T}$, the problem TREEOMQ[$\mathcal{T}$] is either in P or NP-complete? In a more general setting, dichotomies of this kind have been considered in [3, 14, 10].

Acknowledgements. This work was supported by the UK EPSRC grant EP/M012670/ 'iTract: Islands of Tractability in Ontology-Based Data Access'.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. Journal of Artificial Intelligence Research (JAIR) 36, 1–69 (2009)
3. Bienvenu, M., ten Cate, B., Lutz, C., Wolter, F.: Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. ACM Transactions on Database Systems 39(4), 33:1–44 (2014)
4. Bienvenu, M., Kikot, S., Kontchakov, R., Podolskii, V.V., Ryzhikov, V., Zakharyaschev, M.: The complexity of ontology-based data access with OWL 2 QL and bounded treewidth queries. In: Proc. of the 36th ACM Symposium on Principles of Database Systems, PODS 2017, pp. 201–216. ACM (2017)
5. Bienvenu, M., Kikot, S., Podolskii, V.V.: Tree-like queries in OWL 2 QL: succinctness and complexity results. In: Proc. of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015. pp. 317–328. IEEE Computer Society (2015)

6. Calautti, M., Gottlob, G., Pieris, A.: Chase termination for guarded existential rules. In: Proc. of the 34th ACM Symposium on Principles of Database Systems, PODS 2015. pp. 91–103. ACM (2015)
7. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the DL-Lite family. Journal of Automated Reasoning 39(3), 385–429 (2007)
8. Cuenca Grau, B., Horrocks, I., Krötzsch, M., Kupke, C., Magka, D., Motik, B., Wang, Z.: Acyclicity notions for existential rules and their application to query answering in ontologies. Journal of Artificial Intelligence Research (JAIR) 47, 741–808 (2013)
9. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Springer (2013)
10. Feier, C., Kuusisto, A., Lutz, C.: Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics. CoRR abs/1701.02231 (2017).
11. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series, Springer (2006)
12. Gogacz, T., Marcinkowski, J.: All-instances termination of chase is undecidable. In: Proc. of the 41st Int. Colloquium Automata, Languages, and Programming (ICALP 2014), Part II. Lecture Notes in Computer Science, vol. 8573, pp. 293–304. Springer (2014)
13. Gottlob, G., Leone, N., Scarcello, F.: The complexity of acyclic conjunctive queries. Journal of the ACM 48(3), 431–498 (2001)
14. Hernich, A., Lutz, C., Ozaki, A., Wolter, F.: Schema.org as a description logic. In: Proc. of the 28th Int. Workshop on Description Logics (DL 2015). CEUR Workshop Proceedings, vol. 1350. CEUR-WS (2015),
15. Kikot, S., Kontchakov, R., Zakharyaschev, M.: On (in)tractability of OBDA with OWL 2 QL. In: Proc. of the 24th Int. Workshop on Description Logics (DL 2011). CEUR Workshop Proceedings, vol. 745, pp. 224–234. CEUR-WS (2011)
16. Kikot, S., Kontchakov, R., Podolskii, V.V., Zakharyaschev, M.: On the succinctness of query rewriting over shallow ontologies. In: Proc. of the Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS'14. pp. 57:1–57:10. ACM (2014)