

How to Approximate Ontology-Mediated Queries

Anneke Haga¹, Carsten Lutz¹, Leif Sabellek¹, Frank Wolter²

¹Department of Computer Science, University of Bremen, Germany

²Department of Computer Science, University of Liverpool, UK

{anneke,clu,sabellek}@uni-bremen.de, wolter@liverpool.ac.uk

Abstract

We introduce and study several notions of approximation for ontology-mediated queries based on the description logics \mathcal{ALC} and \mathcal{ALCI} . Our approximations are of two kinds: we may (1) replace the ontology with one formulated in a tractable ontology language such as \mathcal{ELI} or certain TGDs and (2) replace the database with one from a tractable class such as the class of databases whose treewidth is bounded by a constant. We determine the computational complexity and the relative completeness of the resulting approximations. (Almost) all of them reduce the data complexity from CONP -complete to PTIME , in some cases even to fixed-parameter tractable and to linear time. While approximations of kind (1) also reduce the combined complexity, this tends to not be the case for approximations of kind (2). In some cases, the combined complexity even increases.

1 Introduction

Ontology-mediated querying enriches database queries by an ontology, in this way providing domain knowledge and extending the language available for formulating queries. For ontologies written in popular expressive description logics (DLs) such as \mathcal{ALC} and \mathcal{ALCI} , however, the complexity of ontology-mediated querying is prohibitively high, CONP -complete in data complexity (Schaerf 1993) and EXPTIME - resp. 2EXPTIME -complete in combined complexity (Lutz 2008). As a consequence, practical implementations often resort to approximating the answers to ontology mediated queries (OMQs) (Tserendorj et al. 2008; Thomas, Pan, and Ren 2010; Zhou et al. 2015), mostly using rather pragmatic approaches. The aim of this paper is to carry out a systematic study of OMQ approximation from a theoretical angle, introducing several principled notions of approximation and clarifying their computational complexity and relative completeness. In particular, we aim to find approximations that reduce the data complexity to PTIME or even to fixed-parameter tractability (FPT). Preferably, they should additionally reduce the combined complexity.

We mainly consider approximation from below, that is, approximations that are sound, but incomplete. While we also present some first results on approximation from above, whenever we speak of approximation without further qualification we mean approximation from below. An OMQ is a triple $Q = (\mathcal{O}, \Sigma, q)$ where \mathcal{O} is an ontology, q an actual

query such as a conjunctive query (CQ), and Σ a signature for the databases \mathcal{D} that Q is evaluated on. Our starting point is the observation that this gives us three points of attack for approximation: we can relax the ontology \mathcal{O} , the query q , and the database \mathcal{D} . However, relaxing the query is not useful for attaining PTIME data complexity as in the DLs mentioned above, ontology-mediated querying is CONP -hard already for atomic queries (AQs), that is, for very simple CQs of the form $A(x)$. We are thus left with the ontology and the database.

For ontology relaxing approximation, we choose an ontology language \mathcal{L} that admits ontology-mediated querying in PTIME in data complexity. To define approximate answers to OMQ $Q = (\mathcal{O}, \Sigma, q)$, we then consider all \mathcal{L} -ontologies \mathcal{O}' with $\mathcal{O} \models \mathcal{O}'$ (to guarantee soundness), replace \mathcal{O} with \mathcal{O}' , and take the union of the answers to the resulting OMQs Q' . Equivalently, we can use a single \mathcal{O}' , namely the unique *logically strongest* \mathcal{L} -ontology with $\mathcal{O} \models \mathcal{O}'$. Such \mathcal{O}' will typically be infinite (Bötcher, Lutz, and Wolter 2019; Haga et al. 2020), but it turns out that it never has to be materialized by an algorithm that computes approximate answers; the ontologies \mathcal{O}' in fact only serve the purpose of defining the semantics of approximation. As choices for \mathcal{L} , we consider Horn description logics such as \mathcal{ELI} and sets of restricted tuple-generating dependencies (TGDs), also known as existential rules and Datalog^\pm (Gottlob, Morak, and Pieris 2015). A related (but stronger) notion of OMQ approximation was proposed in (Haga et al. 2020).

For database relaxing approximation, we choose a class of databases \mathcal{D} that admits ontology-mediated querying in PTIME in data complexity. To define approximate answers to OMQ Q on input database \mathcal{D} , we then consider all databases $\mathcal{D}' \in \mathcal{D}$ such that there is a homomorphism from \mathcal{D}' to \mathcal{D} (to guarantee soundness) and take the union of the answers to Q on all such \mathcal{D}' . As choices for \mathcal{D} , we consider databases of bounded treewidth and databases that are proper trees. Equivalently, we can use the unraveling of \mathcal{D} into a structure of bounded treewidth, resp. into a tree. Such an unraveling may be infinite, but again this is unproblematic as the unraveling never has to be materialized by an algorithm that computes approximate answers.

Both of these approaches to approximation can also be used to define approximation from above. For ontology strengthening approximation, one requires $\mathcal{O}' \models \mathcal{O}$ and

for database strengthening approximation, one requires that there is a homomorphism from \mathcal{D} to \mathcal{D}' , rather than the other way around. In both cases, one then takes the intersection of the answers rather than the union. The resulting approximations are complete, but unsound. Note that approximation from above is particularly useful in combination with approximation from below (Tserendorj et al. 2008; Zhou et al. 2015). If both approximations produce the same answers, one has actually succeeded to compute the ‘real’, non-approximate answers.

An OMQ language is a pair $(\mathcal{L}, \mathcal{Q})$ with \mathcal{L} an ontology language and \mathcal{Q} a query language. We consider the approximation of OMQ languages $(\mathcal{L}, \mathcal{Q})$ where $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ and $\mathcal{Q} \in \{\text{UCQ}, \text{CQ}, \text{AQ}, \text{bELIQ}\}$ with UCQ denoting unions of CQs and bELIQ denoting the class of unary CQs that correspond to \mathcal{ELI} -concepts (ELIQs) and of Boolean CQs $\exists x q(x)$ with $q(x)$ an ELIQ. The exact problem that we consider is (approximate) *OMQ evaluation*, meaning to decide, given an OMQ Q , a database \mathcal{D} , and a tuple \bar{a} of constants from \mathcal{D} , whether \bar{a} is an (approximate) answer to Q on \mathcal{D} . We give an overview of our results.

We start in Section 4 with ontology relaxing approximation, choosing for \mathcal{L} the description logic \mathcal{ELI}_{\perp}^u , that is, the extension of \mathcal{ELI} with the universal role and bottom (logical falsity). We then prove that \mathcal{ELI}_{\perp}^u -ontology relaxing OMQ evaluation is in PTIME in data complexity and EXPTIME-complete in combined complexity for all OMQ languages $(\mathcal{L}, \mathcal{Q})$ mentioned above. In contrast, non-approximate OMQ evaluation is 2EXPTIME-complete in $(\mathcal{ALCI}, \mathcal{Q})$ for $\mathcal{Q} \in \{\text{CQ}, \text{UCQ}\}$. If we consider more restricted classes of queries, the complexity improves further. In fact, \mathcal{ELI}_{\perp}^u -ontology relaxing OMQ evaluation is fixed-parameter tractable (FPT) with single exponential overall running time if \mathcal{Q} is the class of all CQs or UCQs whose treewidth is bounded by a constant; here and in what follows, the parameter is the size of the OMQ. For $\mathcal{Q} = \text{bELIQ}$, we even obtain linear time in data complexity.

In Section 5, we consider tree-database relaxing approximation. These (almost) deliver the same answers as \mathcal{ELI}_{\perp}^u -ontology relaxing approximation in $(\mathcal{ALCI}, \text{bELIQs})$, but are incomparable for more expressive query languages. They turn out to be less well-behaved regarding combined complexity, being 2EXPTIME-complete in $(\mathcal{ALCI}, \mathcal{Q})$ for $\mathcal{Q} \in \{\text{UCQ}, \text{CQ}\}$; this in fact even holds for (U)CQs of bounded treewidth. If $\mathcal{L} = \mathcal{ALC}$ or $\mathcal{Q} \in \{\text{AQ}, \text{bELIQ}\}$, then they are only EXPTIME-complete, as in the non-approximate case. On the other hand, tree-database relaxing approximation enjoys a slight advantage in data complexity over \mathcal{ELI}_{\perp}^u -ontology relaxing approximation, namely linear time for *all* OMQ languages $(\mathcal{Q}, \mathcal{L})$ considered in this paper. We also prove the surprising result that tree-database relaxing OMQ evaluation is EXPSpace-hard in $(\mathcal{ALC}, \text{CQ})$ and 2EXPTIME-complete in $(\mathcal{ALC}, \text{UCQ})$. This means that it is *harder* in combined complexity than non-approximate OMQ evaluation (which is only EXPTIME-complete in these two cases), while it is easier in data complexity.

In Section 6, we revisit ontology relaxing approximation, replacing \mathcal{ELI}_{\perp}^u with frontier-one TGDs whose rule bodies and heads are of bounded treewidth. Recall that a TGD is

frontier-one if body and head share at most a single variable (Baget et al. 2009). For rule heads, we do not only require bounded treewidth, but the existence of a tree decomposition in which the bags overlap in at most a *single* element. The resulting approximations are significantly more complete than \mathcal{ELI}_{\perp}^u -ontology relaxing approximations, but enjoy the same favourable computational properties regarding both data and combined complexity except that we do not attain linear time. We also observe that by increasing the treewidth of the rule bodies and heads, we obtain infinite hierarchies of increasingly complete approximations.

In Section 7, we generalize tree-database relaxing approximation into btw-database relaxing approximation, replacing tree databases with databases of bounded treewidth. These are strictly more complete than TGD-ontology relaxing approximations. They enjoy the same computational properties as tree-database relaxing approximations both regarding data and combined complexity except that we do not attain linear time.

Finally, in Section 8 we consider approximation from above. For database strengthening approximation, the results are negative: we show CONP-completeness in data complexity even if the original OMQ is from $(\mathcal{EL}, \text{CQ})$, an OMQ language that admits non-approximate OMQ evaluation in PTIME in data complexity. Ontology strengthening approximation looks more promising. We consider the fragment \mathcal{ELIU}_{\perp} of \mathcal{ALC} and show that \mathcal{ELIU}_{\perp} -ontology relaxing OMQ evaluation in $(\mathcal{ELIU}_{\perp}, \mathcal{Q})$ is FPT with double exponential overall running time (thus in PTIME in data complexity) and 2EXPTIME-complete in combined complexity for $\mathcal{Q} \in \{\text{AQ}, \text{CQ}, \text{UCQ}\}$. Note that non-approximate OMQ evaluation in $(\mathcal{ELIU}_{\perp}, \text{AQ})$ is only EXPTIME-complete (Lutz 2008), and thus this is another case where approximate OMQ evaluation is harder in combined complexity than non-approximate OMQ evaluation. An appendix with full proofs is available at (Haga et al. 2021).

Related work. Several approaches achieve practically efficient OMQ evaluation by a pragmatic translation of ontologies into languages that enjoy PTIME data complexity such as Datalog, OWL 2 QL, or OWL 2 EL. This includes Screech (Hitzler and Vrandečić 2005; Tserendorj et al. 2008), TrOWL (Pan and Thomas 2007; Thomas, Pan, and Ren 2010), and PAGoDA (Zhou et al. 2015), see also (Carral et al. 2014). Approximations are computed both from below and above, in the spirit of knowledge compilation (Selman and Kautz 1996). Approximations of ontologies in tractable languages with stronger guarantees were recently studied in (Bötcher, Lutz, and Wolter 2019; Haga et al. 2020), but querying and data were (mostly) not considered. In database theory, approximate querying (without ontologies) was studied in (Fink and Olteanu 2011; Barceló, Libkin, and Romero 2014; Barceló, Romero, and Zeume 2018). The approximation and rewriting of OMQs in(to) datalog is studied in (Bienvenu et al. 2014; Kaminski, Nenov, and Grau 2016; Feier, Kuusisto, and Lutz 2019). In the context of Horn DLs, OMQ approximation that achieves FPT was considered in (Barceló et al. 2019).

2 Preliminaries

Description Logics. Let \mathbb{N}_C and \mathbb{N}_R be countably infinite sets of *concept names* and *role names*. Further fix a countably infinite supply of *constants*. A *role* is a role name r or an *inverse role* r^- , with r a role name and $(r^-)^- = r$. An *ALCI-concept* is defined according to the syntax rule

$$C, D ::= \top \mid \perp \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C$$

where A ranges over concept names and r over roles. We use $C \sqcup D$ as abbreviation for $\neg(\neg C \sqcap \neg D)$, $C \rightarrow D$ for $\neg C \sqcup D$, and $\forall r.C$ for $\neg \exists r.\neg C$. An \mathcal{ELI}_\perp -concept is an \mathcal{ALCI} -concept that does not use negation “ \neg ” and an \mathcal{ELI}_\perp^u -concept is an \mathcal{ELI}_\perp -concept that may use the *universal role* u in place of a role name. \mathcal{ELI} -concepts and \mathcal{ELI}^u -concepts do not admit \perp . Let $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ELI}, \mathcal{ELI}^u, \mathcal{ELI}_\perp, \mathcal{ELI}_\perp^u\}$. An \mathcal{L} -ontology is a finite set of *concept inclusion (CIs)* $C \sqsubseteq D$ with C and D \mathcal{L} -concepts. We sometimes also consider infinite ontologies that, however, only serve the purpose of defining a semantics and never have to be represented explicitly. For \mathcal{ELI}_\perp^u -ontologies, we assume w.l.o.g. that \perp occurs only in CIs of the form $C \sqsubseteq \perp$ where \perp does not occur in C . A *database* is a finite set of *facts* of the form $A(a)$ or $r(a, b)$ where $A \in \mathbb{N}_C \cup \{\top\}$, $r \in \mathbb{N}_R$, and a, b are constants. We use $\text{adom}(\mathcal{D})$ to denote the set of constants used in database \mathcal{D} , also called its *active domain*.

A *signature* Σ is a set of concept and role names, uniformly referred to as *symbols*. We use $\text{sig}(X)$ to denote the set of symbols used in any syntactic object X such as a concept or an ontology. A Σ -database is a database \mathcal{D} with $\text{sig}(\mathcal{D}) \subseteq \Sigma$. The *size* of a (finite) syntactic object X , denoted $\|X\|$, is the number of symbols needed to write it as a word using a suitable encoding.

The semantics is given in terms of *interpretations* \mathcal{I} , which we define to be a (possibly infinite and) non-empty set of facts. We use $\Delta^\mathcal{I}$ to denote the set of individual names in \mathcal{I} , define $A^\mathcal{I} = \{a \mid A(a) \in \mathcal{I}\}$ for all $A \in \mathbb{N}_C$, and $r^\mathcal{I} = \{(a, b) \mid r(a, b) \in \mathcal{I}\}$ for all $r \in \mathbb{N}_R$. The extension $C^\mathcal{I}$ of \mathcal{ALCI} -concepts C is then defined as usual (Baader et al. 2017). The universal role u is always interpreted as $u^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. This definition of interpretation is slightly different from the usual one, but equivalent; its virtue is uniformity as every database is a (finite) interpretation. Interpretation \mathcal{I} *satisfies* CI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, fact $A(a)$ if $a \in A^\mathcal{I}$, and fact $r(a, b)$ if $(a, b) \in r^\mathcal{I}$. We thus make the standard names assumption, that is, we interpret constants as themselves. For $S \subseteq \Delta^\mathcal{I}$, we use $\mathcal{I}|_S$ to denote the restriction of \mathcal{I} to facts that only contain constants from S .

Interpretation \mathcal{I} is a *model* of an ontology or database if it satisfies all inclusions or facts in it. A database \mathcal{D} is *satisfiable* w.r.t. an ontology \mathcal{O} if there is a model \mathcal{I} of \mathcal{O} and \mathcal{D} . We write $\mathcal{O} \models \mathcal{O}'$ if every model of ontology \mathcal{O} is also a model of ontology \mathcal{O}' . We associate every interpretation \mathcal{I} with an undirected graph $G_\mathcal{I} = (V, E)$ where $V = \Delta^\mathcal{I}$ and $E = \{\{d, e\} \mid (d, e) \in r^\mathcal{I} \text{ for some } r \in \mathbb{N}_R\}$. We say that \mathcal{I} is a *tree* if $G_\mathcal{I}$ is acyclic without self loops and multi-edges, that is, $(d, e) \in r_1^\mathcal{I}$ implies $(d, e) \notin r_2^\mathcal{I}$ for all distinct roles r_1, r_2 . Note that, somewhat unusually, our trees need thus not be connected.

Queries. A *conjunctive query (CQ)* is of the form $q(\bar{x}) = \exists \bar{y} \varphi(\bar{x}, \bar{y})$, where \bar{x} and \bar{y} are tuples of variables and $\varphi(\bar{x}, \bar{y})$ is a conjunction of *atoms* of the form $A(x)$ and $r(x, y)$, $A \in \mathbb{N}_C$, $r \in \mathbb{N}_R$, and x, y variables from $\bar{x} \cup \bar{y}$. We require that all variables in \bar{x} are used in φ , call the variables in \bar{x} the *answer variables* of q , and use $\text{var}(q)$ to denote $\bar{x} \cup \bar{y}$. We take the liberty to write $\alpha \in q$ to indicate that α is an atom in q and sometimes write $r^-(x, y) \in q$ in place of $r(y, x) \in q$. The CQ q gives rise to a database \mathcal{D}_q , often called the canonical database for q , obtained by viewing the variables in q as constants and the atoms as facts. For $V \subseteq \text{var}(q)$, we use $q|_V$ to denote the restriction of q to the atoms that use only variables in V .

A *homomorphism* from interpretation \mathcal{I}_1 to interpretation \mathcal{I}_2 is a function $h : \Delta^{\mathcal{I}_1} \rightarrow \Delta^{\mathcal{I}_2}$ such that $d \in A^{\mathcal{I}_1}$ implies $h(d) \in A^{\mathcal{I}_2}$ and $(d, e) \in r^{\mathcal{I}_1}$ implies $(h(d), h(e)) \in r^{\mathcal{I}_2}$ for all $d, e \in \Delta^{\mathcal{I}_1}$, $A \in \mathbb{N}_C$, and $r \in \mathbb{N}_R$. A homomorphism from CQ q to interpretation \mathcal{I} is a homomorphism from \mathcal{D}_q to \mathcal{I} . A tuple $\bar{d} \in (\Delta^\mathcal{I})^{|\bar{x}|}$ is an *answer* to q on \mathcal{I} if there is a homomorphism h from q to \mathcal{I} with $h(\bar{x}) = \bar{d}$. A *contraction* of a CQ p is a CQ that can be obtained from q by identifying variables. The identification of two answer variables is not admitted and the identification of an answer variable x with a quantified variable y results in x .

A *union of conjunctive queries (UCQ)* $q(\bar{x})$ is a disjunction of CQs that all have the same answer variables \bar{x} . A tuple $\bar{d} \in (\Delta^\mathcal{I})^{|\bar{x}|}$ is an *answer* to q on interpretation \mathcal{I} , written $\mathcal{I} \models q(\bar{d})$, if \bar{d} is an answer to some CQ in q on \mathcal{I} . We use $q(\mathcal{I})$ to denote set of all answers to q on \mathcal{I} . The *arity* of q is the length of \bar{x} and q is *Boolean* if it is of arity zero.

An \mathcal{ELI} -query (ELIQ) is a unary CQ $q(x)$ such that \mathcal{D}_q is a connected tree and a *Boolean \mathcal{ELI} -query (BELIQ)* is a Boolean CQ $q()$ such that \mathcal{D}_q is a connected tree. One can alternatively define ELIQs as being of the form $C(x)$ with C an \mathcal{ELI} -concept, and BELIQs as being of the form $\exists u.C$ with C an \mathcal{ELI} -concept and we may thus use ELIQs as \mathcal{ELI} -concepts and BELIQs as \mathcal{ELI}^u -concepts, and vice versa. For uniformity, we use *BELIQ* to refer to a CQ that is either an ELIQ or a BELIQ. An *atomic query (AQ)* is an ELIQ of the form $A(x)$, A a concept name.

Ontology-Mediated Queries. An *ontology-mediated query (OMQ)* is a triple $Q = (\mathcal{O}, \Sigma, q)$ with \mathcal{O} an ontology, $\Sigma \subseteq \text{sig}(\mathcal{O}) \cup \text{sig}(q)$ a signature called the *data signature*, and q a query such as a UCQ. We write $Q(\bar{x})$ to indicate that the answer variables of q are \bar{x} . The signature Σ expresses the promise that Q is only evaluated on Σ -databases. Let \mathcal{D} be such a database. Then $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$ is an *answer* to Q on \mathcal{D} , written $\mathcal{D} \models Q(\bar{a})$, if $\mathcal{I} \models q(\bar{a})$ for all models \mathcal{I} of \mathcal{O} and \mathcal{D} . When more convenient, we might alternatively write $\mathcal{D}, \mathcal{O} \models q(\bar{a})$. We further write $Q(\mathcal{D})$ to denote the set of all answers to Q on \mathcal{D} . For OMQs $Q_1(\bar{x})$ and $Q_2(\bar{x})$, $Q_i = (\mathcal{O}_i, \Sigma, q_i)$, we say that Q_1 is *contained* in Q_2 and write $Q_1 \subseteq Q_2$, if for every Σ -database \mathcal{D} , $Q_1(\mathcal{D}) \subseteq Q_2(\mathcal{D})$. We say that Q_1 is *equivalent* to Q_2 and write $Q_1 \equiv Q_2$, if $Q_1 \subseteq Q_2$ and $Q_2 \subseteq Q_1$. We use $(\mathcal{L}, \mathcal{Q})$ to denote the *OMQ language* that contains all OMQs Q in which \mathcal{O} is formulated in DL \mathcal{L} and q in query language \mathcal{Q} , such as in $(\mathcal{ALCI}, \text{UCQ})$ and $(\mathcal{ELI}, \text{AQ})$.

Treewidth. Treewidth is a widely used notion that measures the degree of tree-likeness of a graph. As common for example in the area of constraint satisfaction problems, we are interested in two parameters of tree decompositions instead of only one. A *tree decomposition* of an interpretation \mathcal{I} is a triple $(V, E, (B_v)_{v \in V})$ where (V, E) is an undirected tree and $(B_v)_{v \in V}$ is a family of subsets of $\Delta^{\mathcal{I}}$, often referred to as bags, such that:

1. for all $d \in \Delta^{\mathcal{I}}$, $\{v \in V \mid d \in B_v\}$ is nonempty and connected in (V, E) ;
2. if $(d_1, d_2) \in r^{\mathcal{I}}$ for any role name r , then there is a $v \in V$ with $d_1, d_2 \in B_v$.

We call $(V, E, (B_v)_{v \in V})$ an (ℓ, k) -*tree decomposition* if for all distinct $v, v' \in V$, $|B_v \cap B_{v'}| \leq \ell$ and $|B_v| \leq k$. An interpretation \mathcal{I} has *treewidth* (ℓ, k) if it admits an (ℓ, k) -tree decomposition. It has *treewidth* k if it has treewidth $(k, k + 1)$. As usual, the ‘+1’ is used to achieve that trees have treewidth 1.

We also speak of the treewidth of a CQ $q = \exists \bar{y} \varphi(\bar{x}, \bar{y})$, which is that of $\mathcal{D}_q|_{\bar{y}}$, and of the treewidth of a UCQ q , which is the maximum of the treewidths of the CQs in q . Note that answer variables do not contribute to treewidth. For $\ell, k \geq 1$ with $\ell < k$, we use $\text{CQ}_{\ell, k}^{\text{tw}}$ (resp. CQ_k^{tw}) to denote the class of CQs of treewidth (ℓ, k) (resp. of treewidth k), and likewise for $\text{UCQ}_{\ell, k}^{\text{tw}}$ (resp. UCQ_k^{tw}) and UCQs.

Tuple-Generating Dependencies. A *tuple-generating dependency* (TGD) is a first-order sentence ϑ of the form $\forall \bar{x} \forall \bar{y} (\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \psi(\bar{x}, \bar{z}))$ such that $\exists \bar{y} \phi(\bar{x}, \bar{y})$ and $\exists \bar{z} \psi(\bar{x}, \bar{z})$ are CQs. For simplicity, we write ϑ as $\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \psi(\bar{x}, \bar{z})$. We call ϕ and ψ the *body* and *head* of ϑ . The body may be the empty conjunction, i.e. logical truth, then denoted by \top , and the head may be logical falsity denoted by \perp . TGDs with head \perp are often called *denial constraints* (Fan and Geerts 2012). The *frontier variables* of TGD ϑ are the variables that occur unquantified in both body and head. We say that ϑ is *frontier one* if it has at most one frontier variable. An interpretation \mathcal{I} *satisfies* ϑ , denoted $\mathcal{I} \models \vartheta$, if $q_\phi(\mathcal{I}) \subseteq q_\psi(\mathcal{I})$.

We also consider ontologies that are sets of TGDs and, more generally, sentences formulated in first-order logic (FO). What we mean here is the version of FO in which only unary and binary relation symbols are used, which are from N_C and N_R , respectively. Function symbols, constants, and equality are not permitted. An interpretation is a *model* of an FO ontology if it satisfies all sentences in \mathcal{O} . With an *FO-fragment* \mathcal{L} , we mean a class of FO-sentences and an *\mathcal{L} -ontology* is a finite set of sentences from \mathcal{L} . As in the DL case, we sometimes also consider infinite ontologies. We use TGD to denote the FO-fragment that consists of all TGDs, and thus speak of *TGD-ontologies*. It is easy to see that every \mathcal{ELI}_\perp^u -ontology is also a frontier one TGD-ontology.

A standard tool for dealing with TGD ontologies \mathcal{O} is the *chase* that constructs from \mathcal{O} and a database \mathcal{D} a universal model of \mathcal{O} and \mathcal{D} , that is, a model $\text{ch}_{\mathcal{O}}(\mathcal{D})$ of \mathcal{D} and \mathcal{O} that homomorphically embeds into every model of \mathcal{D} and \mathcal{O} and thus satisfies $\mathcal{D}, \mathcal{O} \models q(\bar{a})$ iff $\text{ch}_{\mathcal{O}}(\mathcal{D}) \models q(\bar{a})$ for all CQs q and tuples \bar{a} . Details are given in the appendix, see also (Johnson and Klug 1982; Cali, Gottlob, and Kifer 2013).

3 OMQ Approximation

We introduce two notions of OMQ approximation from below: one where we relax the ontology and one where we relax the database. We start with the former.

For an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q)$, a Σ -database \mathcal{D} , and an ontology language \mathcal{L}' , we use $\text{app}_{\mathcal{L}'}(Q, \mathcal{D})$ to denote the set of tuples $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$ such that $\bar{a} \in Q'(\mathcal{D})$ for some OMQ $Q' = (\mathcal{O}', \Sigma, q)$ where \mathcal{O}' is a (finite) \mathcal{L}' -ontology with $\mathcal{O} \models \mathcal{O}'$. Note that the ontology \mathcal{O}' might contain symbols that do not occur in \mathcal{O} , we will see later that this in fact results in additional answers. Every choice of an OMQ language $(\mathcal{L}, \mathcal{Q})$ and an ontology language \mathcal{L}' gives rise to an approximate OMQ evaluation problem, as follows.

\mathcal{L}' -ontology relaxing OMQ evaluation in $(\mathcal{L}, \mathcal{Q})$
 INPUT: OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{L}, \mathcal{Q})$,
 Σ -database \mathcal{D} , tuple $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$
 OUTPUT: ‘yes’ if $\bar{a} \in \text{app}_{\mathcal{L}'}(Q, \mathcal{D})$ and ‘no’ otherwise

It follows from the definition that ontology relaxing approximation is sound, in the sense that $\text{app}_{\mathcal{L}'}(Q, \mathcal{D}) \subseteq Q(\mathcal{D})$ for all OMQs $Q = (\mathcal{O}, \Sigma, q)$ and Σ -databases \mathcal{D} . We concentrate on the case where \mathcal{L}' is an ontology language such that $(\mathcal{L}', \mathcal{Q})$ enjoys PTIME data complexity. For $(\mathcal{L}, \mathcal{Q}) = (\mathcal{ALCL}, \text{CQ})$, for instance, we might choose $\mathcal{L}' = \mathcal{ELI}_\perp^u$.

In the definition of $\text{app}_{\mathcal{L}'}(Q, \mathcal{D})$, one can equivalently replace the infinitely many \mathcal{L}' -ontologies \mathcal{O}' with the single but infinite \mathcal{L}' -ontology $\mathcal{O}_{\mathcal{L}'}$, that consists of all \mathcal{L}' -sentences φ with $\mathcal{O} \models \varphi$. In fact, the following lemma is a consequence of compactness. For an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q)$ and an ontology language \mathcal{L}' , we use $Q_{\mathcal{L}'}^{\approx}$ to denote the OMQ $(\mathcal{O}_{\mathcal{L}'}, \Sigma, q)$.

Lemma 1. *Let $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\text{FO}, \text{UCQ})$ be an OMQ and \mathcal{L}' an ontology language. Then for every Σ -database \mathcal{D} , $\text{app}_{\mathcal{L}'}(Q, \mathcal{D}) = Q_{\mathcal{L}'}^{\approx}(\mathcal{D})$.*

Note that we do not insist that the infinitely many \mathcal{O}' or the infinite $\mathcal{O}_{\mathcal{L}'}$ is ever explicitly generated when computing approximate answers.

We next define a mode of approximation that is based on relaxing the database by replacing it with a homomorphic pre-image. A *pointed database* is a pair (\mathcal{D}, \bar{b}) with \mathcal{D} a database and \bar{b} a tuple over $\text{adom}(\mathcal{D})$. For an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q)$, a Σ -database \mathcal{D} , and a class of pointed databases \mathfrak{D} , we use $\text{app}_{\mathfrak{D}}(Q, \mathcal{D})$ to denote the set of tuples $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$ such that for some $(\mathcal{D}', \bar{b}) \in \mathfrak{D}$ with $\bar{b} \in Q(\mathcal{D}')$ there is a homomorphism h from \mathcal{D}' to \mathcal{D} with $h(\bar{b}) = \bar{a}$. Every choice of $(\mathcal{L}, \mathcal{Q})$ and \mathfrak{D} gives rise to an approximate OMQ evaluation problem, as follows.

\mathfrak{D} -database relaxing OMQ evaluation in $(\mathcal{L}, \mathcal{Q})$
 INPUT: OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{L}, \mathcal{Q})$,
 Σ -database \mathcal{D} , tuple $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$
 OUTPUT: ‘yes’ if $\bar{a} \in \text{app}_{\mathfrak{D}}(Q, \mathcal{D})$ and ‘no’ otherwise

Answers to any OMQ $Q = (\mathcal{O}, \Sigma, q) \in (\text{FO}, \text{UCQ})$ are preserved under homomorphisms if \mathcal{O} does not use equality, that is, if $\mathcal{D}_1, \mathcal{D}_2$ are databases, h is a homomorphism from \mathcal{D}_1 to \mathcal{D}_2 , and $\bar{a} \in Q(\mathcal{D}_1)$ for an OMQ Q , then $h(\bar{a}) \in$

$Q(\mathcal{D}_2)$ (Bienvenu et al. 2014). As a consequence, database relaxing approximation is sound.

We are interested in choosing \mathfrak{D} such that evaluating OMQs from $(\mathcal{L}, \mathcal{Q})$ on \mathfrak{D} enjoys PTIME data complexity. An important example are classes of databases that are of bounded treewidth, the simplest case being the class of databases that are trees. More precisely, we use \mathfrak{D}_λ (with ‘ λ ’ symbolizing a tree) to denote the class of all pointed databases (\mathcal{D}, \bar{a}) such that the restriction of \mathcal{D} to domain $\text{adom}(\mathcal{D}) \setminus \bar{a}$ is a tree. Recall that a tree does not need to be connected. The resulting notion of approximation is closely related to \mathcal{ELT}_\perp^u -ontology relaxing approximation.

In the same way in which we have rephrased ontology relaxing approximation in terms of a single infinite ontology, we can sometimes (depending on the choice of \mathfrak{D}) rephrase database relaxing approximation in terms of evaluation on a single infinite database. We illustrate this for the case $\mathfrak{D} = \mathfrak{D}_\lambda$. Let \mathcal{D} be a database and $S \subseteq \text{adom}(\mathcal{D})$. A path in \mathcal{D} is a sequence $p = a_0 r_1 a_1 r_2 \dots r_n a_n$, $n \geq 0$, where $a_0, \dots, a_n \in \text{adom}(\mathcal{D})$, r_1, \dots, r_n are (potentially inverse) roles, and $r_{i+1}(a_i, a_{i+1}) \in \mathcal{D}$ for $0 \leq i < n$. We use $\text{tail}(p)$ to denote a_n . The tree unraveling \mathcal{D}_S^\approx of \mathcal{D} at S is the (potentially infinite) database that contains the following facts: all facts from $\mathcal{D}|_S$, $r(p, prb)$ for every path prb , $A(p)$ for every path p with $A(\text{tail}(p)) \in \mathcal{D}$, and $r(a, p)$ for every $r(a, b) \in \mathcal{D}$ and every path p with $a \in S$ and $\text{tail}(p) = b$ (r a potentially inverse role). Note that \mathcal{D}_S^\approx is a tree if and only if $S = \emptyset$. Thus, the tree unravelings $\mathcal{D}_{\bar{a}}^\approx$ in the following lemma are in general not tree databases.

Lemma 2. *Let $Q = (\mathcal{O}, \Sigma, q) \in (FO, UCQ)$. Then for all Σ -databases \mathcal{D} and $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{a}|}$, $\bar{a} \in \text{app}_{\mathfrak{D}_\lambda}(Q, \mathcal{D})$ iff $\bar{a} \in Q(\mathcal{D}_{\bar{a}}^\approx)$.*

4 \mathcal{ELT}_\perp^u -Ontology Relaxing Approximation

We consider \mathcal{ELT}_\perp^u -ontology relaxing evaluation of OMQs from (\mathcal{ALCI}, UCQ) , starting with an example.

Example 1. *Let $Q(x) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALC}, CQ)$ where*

$$\begin{aligned} \mathcal{O} &= \{\top \sqsubseteq \forall r.(B_1 \rightarrow A) \sqcup \forall r.(B_2 \rightarrow A)\} \\ \Sigma &= \{r, A, B_1, B_2\} \\ q &= \exists y r(x, y) \wedge A(y). \end{aligned}$$

Further let $\mathcal{D} = \{r(a, b_1), r(a, b_2), B_1(b_1), B_2(b_2)\}$. Clearly, $a \in Q(\mathcal{D})$. The ontology $\mathcal{O}_{\mathcal{ELT}_\perp^u}^\approx$ contains CI $\exists r.B_1 \sqcap \exists r.B_2 \sqsubseteq \exists r.A$, thus also $a \in Q_{\mathcal{ELT}_\perp^u}^\approx(\mathcal{D})$.

We next illustrate incompleteness, which cannot be avoided by any notion of approximation from below that attains PTIME data complexity. This follows from the existence of OMQs that express non-3-colorability.

Example 2. *Let $Q() = (\mathcal{O}, \Sigma, \exists x D(x)) \in (\mathcal{ALC}, BELIQ)$ with*

$$\mathcal{O} = \{\top \sqsubseteq R \sqcup G \sqcup B, X \sqcap \exists e.X \sqsubseteq D \mid X \in \{R, G, B\}\}$$

and $\Sigma = \{e\}$. Every Σ -database \mathcal{D} can be viewed as an undirected graph by ‘forgetting’ the direction of e -edges. Then $\mathcal{D} \models Q$ iff the graph is not 3-colorable. In contrast, the careful chase algorithm given below can be used to verify

that $\mathcal{D} \not\models Q_{\mathcal{ELT}_\perp^u}^\approx$ for all Σ -databases \mathcal{D} . It is easy to modify the example so as to use an AQ in place of a BELIQ.

For readers who are disappointed by the extreme incompleteness in the previous example, we remark replacing \mathcal{ELT}_\perp^u with classes of TGDs improves the situation.

We next present three observations regarding our definition of ontology relaxing approximation. The first observation is that it increases completeness to admit in $\mathcal{O}_{\mathcal{ELT}_\perp^u}^\approx$ symbols that do not occur in \mathcal{O} .

Example 3. *Let $Q(x) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALC}, CQ)$ with*

$$\begin{aligned} \mathcal{O} &= \{\top \sqsubseteq \forall r.(B_1 \rightarrow B) \sqcup \forall r.(B_2 \rightarrow B)\} \\ \Sigma &= \{r, A, B, B_1, B_2\} \quad q = \exists y r(x, y) \wedge A(y) \wedge B(y). \end{aligned}$$

Then $\mathcal{O}_{\mathcal{ELT}_\perp^u}^\approx$ contains the CI

$$\exists r.(B_1 \sqcap A) \sqcap \exists r.(B_2 \sqcap A) \sqsubseteq \exists r.(A \sqcap B)$$

despite the fact that A does not occur in \mathcal{O} . Let

$$\mathcal{D} = \{r(a, b_1), r(a, b_2), B_1(b_1), B_2(b_2), A(b_1), A(b_2)\}.$$

Then $a \in Q_{\mathcal{ELT}_\perp^u}^\approx(\mathcal{D})$, but we show in the appendix that this is no longer true when we remove from $\mathcal{O}_{\mathcal{ELT}_\perp^u}^\approx$ all CIs that use a symbol that does not occur in \mathcal{O} .

It is, however, easy to see that it suffices to admit in $\mathcal{O}_{\mathcal{ELT}_\perp^u}^\approx$ the symbols that occur in \mathcal{O} or in Σ while additional symbols do not further increase completeness.

The second observation is that \mathcal{ELT}_\perp^u -ontology relaxing approximation is more complete than \mathcal{ELT}_\perp -ontology relaxing approximation. In fact, it seems to be much more challenging to compute answers for the latter while offering no obvious benefit compared to the former, and thus we do not consider it in this paper.

Example 4. *Let $Q() = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALC}, CQ)$ be the Boolean OMQ with*

$$\mathcal{O} = \{A \sqsubseteq B \sqcup \forall r.B\} \quad \Sigma = \{r, A, B\} \quad q = \exists x B(x).$$

Let $\mathcal{D} = \{A(a), r(a, b)\}$. Then $\mathcal{O}_{\mathcal{ELT}_\perp^u}^\approx$ contains $A \sqcap \exists r.\top \sqsubseteq \exists u.B$ and thus $\mathcal{D} \models Q_{\mathcal{ELT}_\perp^u}^\approx$, but it is shown in the appendix that $\mathcal{D} \not\models Q_{\mathcal{ELT}_\perp}^\approx$.

We remark that \mathcal{ELT}_\perp^u -ontology relaxing approximation is also more complete than \mathcal{ELT}_\perp -ontology relaxing approximation, examples are easy to find.

Third, we observe that defining ontology relaxing approximation in terms of ontologies that are implied by the original ontology does not necessarily result in maximum completeness. In fact, the following example shows that it may pay off to use an \mathcal{ELT}_\perp^u -ontology that is *not* a consequence of the original ontology. This is a very interesting effect, but we do not investigate it further.

Example 5. *We use the ontology \mathcal{O} and signature Σ from Example 4. Let $Q_0() = (\mathcal{O}, \Sigma, q_0)$ for*

$$q_0 = \exists x \exists y A(x) \wedge A(y) \wedge B(y) \wedge r(x, y) \wedge r(y, x),$$

and let $\mathcal{D}_0 = \{A(a), r(a, b), r(b, a), A(b)\}$. While $\mathcal{D}_0 \models Q_0$, the careful chase algorithm below yields $\mathcal{D}_0 \not\models Q_{\mathcal{ELT}_\perp^u}^\approx$. Now take $\mathcal{O}' = \{A \sqsubseteq B\}$ and note that $\mathcal{O} \not\models \mathcal{O}'$ but Q_0 is equivalent to $(\mathcal{O}', \Sigma, q_0)$ as in fact both OMQs are equivalent to the CQ q_0 with atom $B(y)$ dropped.

The following is the main result of this section. When we speak about fixed-parameter tractability (FPT), we generally mean that the parameter is the size of the OMQ, that is, we refer to running time $f(\|Q\|) \cdot O(\|D\|^c)$ where f is a computable function and c a constant.

Theorem 1. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCT}\}$. Then \mathcal{ELT}_\perp^u -ontology relaxing OMQ evaluation is*

1. EXPTIME-complete in combined complexity and PTIME-complete in data complexity in $(\mathcal{L}, \mathcal{Q})$, $\mathcal{Q} \in \{AQ, CQ, UCQ\}$;
2. FPT with single exponential running time in $(\mathcal{L}, \mathcal{Q})$, $\mathcal{Q} \in \{CQ_k^{pw}, UCQ_k^{pw} \mid k \geq 1\}$;
3. in linear time in data complexity in $(\mathcal{L}, \text{bELIQ})$ with running time $2^{O(\|Q\|)} \cdot O(\|D\|)$.

To prove Theorem 1, we first establish the following crucial lemma that relates \mathcal{ELT}_\perp^u -ontology relaxing OMQ evaluation to tree unravelings.

Lemma 3. *Let $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCT}, \text{bELIQ})$, \mathcal{D} be a Σ -database, and $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$. Then*

1. $\bar{a} \in Q_{\mathcal{ELT}_\perp^u}(\mathcal{D})$ iff $\bar{a} \in Q(\mathcal{D}_\emptyset^\approx)$;
2. \mathcal{D} is satisfiable w.r.t. $\mathcal{O}_{\mathcal{ELT}_\perp^u}$ iff $\mathcal{D}_\emptyset^\approx$ is satisfiable w.r.t. \mathcal{O} .

We first prove Point 3 of Theorem 1. We are thus given an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCT}, \text{bELIQ})$, a Σ -database \mathcal{D} , and a tuple $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$ and have to decide whether $\bar{a} \in Q_{\mathcal{ELT}_\perp^u}(\mathcal{D})$. By Point 1 of Lemma 3, it suffices to decide whether $\bar{a} \in Q(\mathcal{D}_\emptyset^\approx)$. This is much more convenient as we are back to the original ontology instead of having to deal directly with $\mathcal{O}_{\mathcal{ELT}_\perp^u}$. In the appendix, we show that deciding $\bar{a} \in Q(\mathcal{D}_\emptyset^\approx)$ can be reduced in linear time (data complexity) to the unsatisfiability of propositional Horn formulas, which is well-known to be in linear time (Dowling and Gallier 1984).

Regarding the upper bounds in Points 1 and 2, we first observe that we can concentrate on CQs rather than UCQs. This follows from the fact that \mathcal{ELT}_\perp^u -ontologies have universal models, even if infinite.

Lemma 4. *Let $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCT}, UCQ)$ with $q = p_1 \vee \dots \vee p_n$, and let \mathcal{D} be a Σ -database. Then $Q_{\mathcal{ELT}_\perp^u}(\mathcal{D}) = Q_1(\mathcal{D}) \cup \dots \cup Q_n(\mathcal{D})$, $Q_i = (\mathcal{O}_{\mathcal{ELT}_\perp^u}, \Sigma, p_i)$ for $1 \leq i \leq n$.*

We now describe an algorithm that establishes the upper bounds in Points 1 and 2 of Theorem 1 and that we refer to as a *careful chase*, see also (Bienvenu et al. 2013).

Assume that we are given an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCT}, CQ)$, a Σ -database \mathcal{D} , and a tuple $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$. We use $\text{trees}(q)$ to denote the set of bELIQs that can be obtained from CQ q by first quantifying all variables, then taking a contraction, then an induced subquery, and then choosing at most one variable as the answer variable. In addition, $\text{trees}(q)$ contains all AQs $A(x)$ with A a concept name used in \mathcal{O} .

The algorithm first extends \mathcal{D} to a database \mathcal{D}' as follows:

- whenever $\mathcal{D}, \mathcal{O}_{\mathcal{ELT}_\perp^u} \models p()$ with $p() \in \text{trees}(q)$ a bELIQ, then take a disjoint copy of \mathcal{D}_p and add it to \mathcal{D} ;

- whenever $\mathcal{D}, \mathcal{O}_{\mathcal{ELT}_\perp^u} \models p(a)$ with $p(x) \in \text{trees}(q)$ an ELIQ, then take a disjoint copy of \mathcal{D}_p and add it to \mathcal{D} , glueing the root x of \mathcal{D}_p to a .

$\mathcal{D}, \mathcal{O}_{\mathcal{ELT}_\perp^u} \models p()$ and $\mathcal{D}, \mathcal{O}_{\mathcal{ELT}_\perp^u} \models p(a)$ can be decided in time $2^{O(\|Q\|)} \cdot O(\|D\|)$ by Point 3 of Theorem 1. Note that \mathcal{D}' is a subdatabase of $\text{ch}_{\mathcal{O}_{\mathcal{ELT}_\perp^u}}(\mathcal{D})$, which is why we speak of a careful chase. The algorithm then carries out the following steps:

- (i) If \mathcal{D} is unsatisfiable w.r.t. $\mathcal{O}_{\mathcal{ELT}_\perp^u}$, then returns ‘yes’;
- (ii) check whether $\bar{a} \in q(\mathcal{D}')$ and return the result.

By Point 2 of Lemma 3, the unsatisfiability check in (i) is equivalent to checking whether $\mathcal{D}_\emptyset^\approx$ is unsatisfiable w.r.t. \mathcal{O} . This is the case if and only if $\mathcal{D}_\emptyset^\approx, \mathcal{O} \models \exists x A(x)$ with A a fresh concept name, which can be decided in time $2^{O(\|Q\|)} \cdot O(\|D\|)$ by Point 3 of Theorem 1.

Checking $\bar{a} \in q(\mathcal{D}')$ in (ii) can be implemented using brute force to attain EXPTIME combined complexity and PTIME data complexity or using as a blackbox an algorithm that runs within the time requirements of fixed-parameter tractability to attain FPT when q is of bounded treewidth. We prove in the appendix that the algorithm is correct and achieves the upper bounds stated in Theorem 1.

The EXPTIME lower bounds in Theorem 1 are proved by a straightforward reduction from the subsumption of concept names in \mathcal{ALC} (Baader et al. 2017): A is subsumed by B w.r.t. \mathcal{O} iff $\{A(a)\}, \mathcal{O} \models B(a)$ iff $\{A(a)\}, \mathcal{O}_{\mathcal{ELT}_\perp^u} \models B(a)$. This trivial reduction also shows that using fragments of \mathcal{ELT}_\perp^u such as \mathcal{EL}_\perp or \mathcal{EL}_\perp^u as a target for ontology approximation cannot improve combined complexity. The PTIME lower bound in Point 1 is inherited from OMQ evaluation in (\mathcal{EL}, AQ) (Calvanese et al. 2015).

5 Tree-Database Relaxing Approximation

We study \mathcal{D}_\perp -database relaxing approximation that for the sake of readability we from now on refer to as tree-database relaxing approximation. We start with observing that tree-database relaxing approximation is incomparable to \mathcal{ELT}_\perp^u -ontology relaxing approximation.

Example 6. *Let $Q(x) = (\mathcal{O}, \Sigma, A(x)) \in (\mathcal{ALCT}, AQ)$ where*

$$\mathcal{O} = \{P \sqcap \exists r. P \sqsubseteq A, \neg P \sqcap \exists r. \neg P \sqsubseteq A\} \quad \Sigma = \{P, r, A\}.$$

Consider $\mathcal{D} = \{r(a, a)\}$. Then $a \notin Q_{\mathcal{ELT}_\perp^u}(\mathcal{D})$, but $a \in Q(\mathcal{D}_{\{a\}}^\approx)$ since $\mathcal{D}_{\{a\}}^\approx = \mathcal{D}$.

Conversely, let $Q(x) = (\emptyset, \{r\}, q) \in (\mathcal{ALCT}, CQ)$ where

$$q(x) = \exists y_1 \exists y_2 \exists y_3 r(x, y_1) \wedge r(y_1, y_2) \wedge r(y_2, y_3) \wedge r(y_3, y_1),$$

and $\mathcal{D} = \{r(a, b_1), r(b_1, b_2), r(b_2, b_3), r(b_3, b_1)\}$. Then $Q_{\mathcal{ELT}_\perp^u}(\mathcal{D}) = \{a\}$, but $Q(\mathcal{D}_{\{a\}}^\approx) = \emptyset$.

Note that the OMQs in Example 6 are based on CQs that are not bELIQs. This is no coincidence, as the following is a consequence of Lemma 3 and the fact that $\mathcal{D}_\emptyset^\approx \subseteq \mathcal{D}_S^\approx$ for all databases \mathcal{D} and $S \subseteq \text{adom}(\mathcal{D})$.

Proposition 1. *In $(\mathcal{ALCI}, bELIQ)$, tree-database relaxing OMQ evaluation is at least as complete as \mathcal{ELT}_\perp^u -ontology relaxing OMQ evaluation.*

The converse of Proposition 1 fails, as per the first part of Example 6. The first main result of this section follows.

Theorem 2. *Tree-database relaxing OMQ evaluation is*

1. *2EXPTIME-complete in combined complexity and in linear time in data complexity (thus FPT) with running time $2^{2^{O(\|Q\|)}} \cdot O(\|D\|)$ in $(\mathcal{ALCI}, \mathcal{Q})$, $\mathcal{Q} \in \{CQ, UCQ, CQ_k^{nw}, UCQ_k^{nw} \mid k \geq 1\}$;*
2. *EXPTIME-complete in combined complexity and in linear time in data complexity (thus FPT) with running time $2^{O(\|Q\|)} \cdot O(\|D\|)$ in $(\mathcal{ALC}, \mathcal{Q})$ and $(\mathcal{ALCI}, \mathcal{Q})$, $\mathcal{Q} \in \{AQ, bELIQ\}$.*

The upper bound in Point 2 of Theorem 2 is proved similarly to Point 3 of Theorem 1, by reduction to the unsatisfiability of propositional Horn formulas. To prove the upper bound in Point 1, we first show that one can construct from an OMQ $Q = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCI}, UCQ)$ and Σ -database \mathcal{D} an OMQ $Q' = (\mathcal{O}, \Sigma', q')$ and a Σ' -database \mathcal{D}' such that $Q(\mathcal{D}) = Q'(\mathcal{D}')$ and q' is a conjunction of disjunctions of BELIQs. This again enables a reduction to the unsatisfiability of propositional Horn formulas. Note that a counterpart of Lemma 4 does not hold for tree-database relaxing approximation and thus we treat UCQs directly. The lower bounds are trivial as (non-approximate) evaluation in all mentioned OMQ languages is hard already on databases of the form $\mathcal{D} = \{A(a)\}$, which satisfy $\mathcal{D}_{\{a\}}^{\approx} = \mathcal{D}$, and for CQs of bounded treewidth (Lutz 2008).

In contrast to \mathcal{ELT}_\perp^u -ontology relaxing approximation, we achieve no improvement in combined complexity in the \mathcal{ALCI} case, but we achieve linear time data complexity also for (U)CQs that are not of bounded treewidth. Informally, this is because database relaxing OMQ evaluation also approximates answers to the actual query (if it is a (U)CQ) while ontology relaxing evaluation only approximates the impact of the ontology. For comparison we recall that without ontologies, evaluating CQs of unbounded treewidth is $W[1]$ -hard, thus most likely not linear time in data complexity and not even FPT (Grohe 2007).

We next prove the second main result of this section. Recall that non-approximate OMQ evaluation in (\mathcal{ALC}, CQ) and (\mathcal{ALC}, UCQ) is EXPTIME-complete in combined complexity and CONP-complete in data complexity. Surprisingly, tree-database relaxing evaluation in these OMQ languages is *harder* in combined complexity than non-approximate evaluation while it is simpler in data complexity (the latter by Theorem 2).

Theorem 3. *Tree-database relaxing OMQ evaluation is EXPSPACE-hard in (\mathcal{ALC}, CQ) and 2EXPTIME-hard in (\mathcal{ALC}, UCQ) .*

The proof of Theorem 3 adapts hardness proofs from (Lutz 2008) for non-approximate OMQ evaluation in (\mathcal{ALCI}, CQ) , simulating inverse roles by making use of the unraveled database.

6 TGD-Ontology Relaxing Approximation

We generalize ontology relaxing approximation from \mathcal{ELT}_\perp^u to TGDs that are frontier-one and have bodies and heads of bounded treewidth. This yields an infinite hierarchy of increasingly more complete approximations.

A TGD $\vartheta = \phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \psi(\bar{x}, \bar{z})$ is an ℓ, k, ℓ', k' -TGD, $\ell < k$ and $\ell' < k'$, if its body has treewidth (ℓ, k) and its head has treewidth (ℓ', k') when viewed as CQs in which all variables are quantified. Every \mathcal{ELT}_\perp^u -CI can be written as a frontier-one 1, 2, 1, 2-TGD, but there are frontier-one 1, 2, 1, 2-TGDs that are not \mathcal{ELT}_\perp^u -CIs, such as $r(x, x) \rightarrow s(x, x)$. From now on, when speaking about ℓ, k, ℓ', k' -TGDs, we generally mean frontier-one TGDs. An ℓ, k, ℓ', k' -TGD-ontology is an FO-ontology that consists only of (frontier-one) ℓ, k, ℓ', k' -TGDs. If any of ℓ, k, ℓ', k' is not bounded, we assign to it value ω .

We study ℓ, k, ℓ', k' -TGD-ontology relaxing OMQ evaluation. Recall that, by Lemma 1, computing such approximations for an OMQ $Q \in (\text{FO}, UCQ)$ corresponds to evaluating the OMQ $Q_{\ell, k, \ell', k'}^{\approx}$ which is based on ontology $\mathcal{O}_{\ell, k, \ell', k'}^{\approx}$. For brevity, we drop the suffix ‘-TGD’ and for instance speak about ℓ, k, ℓ', k' -ontology relaxing OMQ evaluation and write $Q_{\ell, k, \ell', k'}^{\approx}$.

We first observe that restricting the treewidth of the body is necessary to attain PTIME data complexity and that we cannot hope to attain the favourable combined complexity enjoyed by \mathcal{ELT}_\perp^u -ontology relaxing approximation for every value of ℓ' and k' . The following is a consequence of Example 7 below and of hardness proofs in (Lutz 2008).

Proposition 2. *$\omega, \omega, 1, 2$ -ontology relaxing OMQ evaluation in (\mathcal{ALC}, AQ) is CONP-hard in data complexity and there are ℓ', k' such that 1, 2, ℓ', k' -ontology relaxing OMQ evaluation in (\mathcal{ALCI}, CQ) is 2EXPTIME-hard in combined complexity.*

To tame the combined complexity of TGD-ontology-relaxing approximation, we concentrate on the case where $\ell' = 1$. We next consider the choice of values for k and k' , the treewidth of rule bodies and heads. We first show that k gives rise to an infinite hierarchy of increasingly more complete approximations.

Example 7. *Consider the OMQ $Q = (\mathcal{O}, \{e\}, \exists x D(x))$ from Example 2 that expresses non-3-colorability. For every $\{e\}$ -database \mathcal{D} , there is a $k \geq 1$ such that $\mathcal{D} \models Q$ iff $\mathcal{D} \models Q_{1, k, 1, 2}^{\approx}$: for $k = |\text{adom}(\mathcal{D})|$, $\mathcal{O}_{1, k, 1, 2}^{\approx}$ contains $q_{\mathcal{D}} \rightarrow \exists x D(x)$ if \mathcal{D} is not 3-colorable, where $q_{\mathcal{D}}$ is \mathcal{D} viewed as a CQ. Unless PTIME = NP, there are thus no ℓ, k, ℓ', k' such that $Q_{\ell, k, \ell', k'}^{\approx} \supseteq Q_{1, k+p, 1, 2}^{\approx}$ for all $p > 0$. In fact, the existence of such ℓ, k, ℓ', k' implies that 3-colorability is in PTIME by Theorem 4 below.*

Example 7 also shows that TGD-ontology relaxing approximations are more complete than \mathcal{ELT}_\perp^u -ontology relaxing approximations, c.f. Example 2.

For k' , we make the weaker observation that there is no maximum value to be used uniformly for all OMQs.

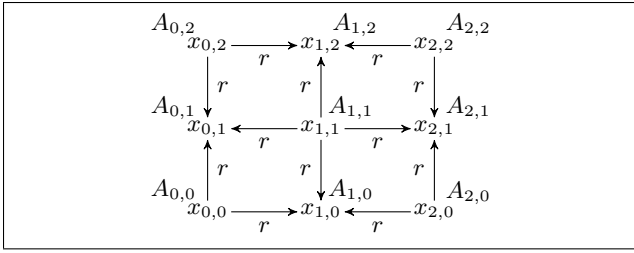


Figure 1: CQ from Example 8 for the case $m = 2$.

Example 8. For $m \geq 2$, consider the Boolean CQ

$$q_m = \bigwedge_{\substack{i,j,i',j' \in \{1,\dots,m\} \\ i+j \text{ is even} \\ |i-i'|+|j-j'|=1}} r(x_{i,j}, x_{i',j'}) \wedge \bigwedge_{i,j \in \{1,\dots,m\}} A_{i,j}(x_{i,j})$$

which takes the form of an $m \times m$ grid in which the direction of the roles alternate and every point in the grid is labeled with a unique concept name, see Figure 1. Consider the following two ways of contracting q_m into a path: identify $x_{i,j}$ with $x_{i',j'}$ if $i+j = i'+j'$ or identify $x_{i,j}$ with $x_{i',j'}$ if $i-j = i'-j'$. Let C_1 and C_2 be the two paths obtained, viewed as \mathcal{ELI} -concepts.

Set $\mathcal{O}_m = \{A \sqsubseteq C_1 \sqcup C_2\}$ and $Q_m() = (\mathcal{O}_m, \{A\}, q_m)$, and let $\mathcal{D} = \{A(a)\}$. Then $(\mathcal{O}_m)_{1,2,1,m^2}^{\approx}$ contains the TGD $A \rightarrow q_m$, so $\mathcal{D} \models (Q_m)_{1,2,1,m^2}^{\approx}$. In contrast, we argue in the appendix that $\mathcal{D} \not\models (Q_m)_{\omega,\omega,1,m^2-1}^{\approx}$.

For every fixed OMQ, however, there is a maximum useful value for k' . The next proposition is established analyzing the algorithm in the proof of Theorem 4 below.

Proposition 3. Let $\ell, k \geq 1$ with $\ell < k$. For every $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCI}, UCQ)$ and $k' \geq |\text{var}(q)|$, $\mathcal{Q}_{\ell,k,1,|\text{var}(q)|}^{\approx} \equiv \mathcal{Q}_{\ell,k,1,k'}^{\approx}$.

The main result of this section is as follows.

Theorem 4. Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ and $\ell, k, k' \geq 1$ with $\ell < k$. Then $\ell, k, 1, k'$ -ontology relaxing OMQ evaluation is

1. EXPTIME-complete in combined complexity and PTIME-complete in data complexity in $(\mathcal{L}, \mathcal{Q})$, $\mathcal{Q} \in \{\mathcal{AQ}, \mathcal{CQ}, \mathcal{UCQ}\}$;
2. FPT in $(\mathcal{L}, \mathcal{Q})$, $\mathcal{Q} \in \{\mathcal{CQ}_p^{hw}, \mathcal{UCQ}_p^{hw} \mid p \geq 1\}$.

So TGD-ontology relaxing approximation inherits the good computational properties of \mathcal{ELI}_\perp^u -ontology relaxing approximation except for linear time for BELIQs, while being significantly more complete.

The lower bounds are proved exactly as for \mathcal{ELI}_\perp^u -ontology relaxing approximation, see Section 4. For the upper bounds, we treat the CQs in a UCQ independently and use a careful chase algorithm that essentially follows the lines of the careful chase presented in Section 4. An important difference is that a counterpart of Lemma 3 in which $Q_{\mathcal{ELI}_\perp^u}^{\approx}$ is replaced with $Q_{\ell,k,1,m^2}^{\approx}$ and $\mathcal{D}_\emptyset^{\approx}$ with an unraveling of \mathcal{D} into a database of treewidth ℓ, k fails to hold if $k' > k$ (the ‘only if’ direction of Point 1 fails). We resort to Theorem 5 below, which is a central ingredient to the proof. For

a database \mathcal{D} and $\ell, k \geq 1$, with $\ell < k$, $\mathcal{D}_{\ell,k}^{\approx}$ denotes the unraveling of \mathcal{D} into a database of treewidth (ℓ, k) , defined in the appendix. While $\text{adom}(\mathcal{D}) \cap \text{adom}(\mathcal{D}_{\ell,k}^{\approx}) = \emptyset$, $\mathcal{D}_{\ell,k}^{\approx}$ contains constants that are ‘copies’ of each $a \in \text{adom}(\mathcal{D})$. We use $\langle \mathcal{D}_{\ell,k}^{\approx}, a \rangle$ to denote a database obtained from $\mathcal{D}_{\ell,k}^{\approx}$ by choosing a copy of a in $\mathcal{D}_{\ell,k}^{\approx}$ and renaming it back to a . With $\langle \mathcal{D}_{\ell,k}^{\approx}, () \rangle$, we mean $\mathcal{D}_{\ell,k}^{\approx}$.

Theorem 5. Let $\ell, k, k' \geq 1$ with $\ell < k$. Given an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCI}, CQ)$ of arity at most one and with \mathcal{D}_q of treewidth $(1, k')$, a Σ -database \mathcal{D} , and $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$, deciding whether $\bar{a} \in Q(\langle \mathcal{D}_{\ell,k}^{\approx}, \bar{a} \rangle)$ is in EXPTIME in combined complexity and FPT.¹

The proof of Theorem 5 uses alternating tree automata.

Both the correctness proof for the careful chase algorithm and the proof of Theorem 5 exploit the regular shape of $\text{ch}_{\mathcal{O}}(\mathcal{D})$ for frontier one TGD-ontologies \mathcal{O} and fail for unrestricted TGDs. In fact, we do not know whether unrestricted TGD-ontology relaxing evaluation is decidable. However, we conjecture that the results in this section generalize to frontier-guarded TGDs (Baget, Leclère, and Mugnier 2010). The gain in completeness appears to be modest.

7 BTW-Database Relaxing Approximation

We study database relaxing approximation based on databases of bounded treewidth. For $\ell, k \geq 1$ with $\ell < k$, let $\mathcal{D}_{\ell,k}$ denote the class of pointed databases (\mathcal{D}, \bar{a}) such that the restriction of \mathcal{D} to domain $\text{adom}(\mathcal{D}) \setminus \bar{a}$ has treewidth ℓ, k . For readability, we speak of ℓ, k -database relaxing approximation in place of $\mathcal{D}_{\ell,k}$ -database relaxing approximation. As for TGD-ontology relaxing approximations, the parameter k gives rise to an infinite hierarchy of increasingly more complete approximations.

We first observe a counterpart of Lemma 2. Let \mathcal{D} be a database, \bar{a} a tuple over $\text{adom}(\mathcal{D})$, and $\ell, k \geq 1$ with $\ell < k$. With $\mathcal{D}_{\bar{a},\ell,k}^{\approx}$, we denote the database obtained by unraveling \mathcal{D} such that $(\mathcal{D}_{\bar{a},\ell,k}^{\approx}, \bar{a}) \in \mathcal{D}_{\ell,k}$. Details are in the appendix.

Lemma 5. Let $Q = (\mathcal{O}, \Sigma, q) \in (FO, UCQ)$ and $\ell, k \geq 1$ with $\ell < k$. Then for all Σ -databases \mathcal{D} and $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$, $\bar{a} \in \text{app}_{\mathcal{D}_{\ell,k}}(Q, \mathcal{D})$ iff $\bar{a} \in Q(\mathcal{D}_{\bar{a},\ell,k}^{\approx})$.

We next relate ℓ, k, ℓ', k' -ontology relaxing approximation to ℓ, k -database relaxing approximation.

Example 9. Let $Q_n() = (\emptyset, \{r\}, q_n) \in (\mathcal{ALCI}, CQ)$ where

$$q_n = \exists x_1 \cdots \exists x_n \bigwedge_{1 \leq i,j \leq n} r(x_i, x_j),$$

and let $\mathcal{D}_n = \{r(a_i, a_j) \mid 1 \leq i, j \leq n\}$. Then $\mathcal{D}_n \models (Q_n)_{1,2,1,2}^{\approx}$, but $(\mathcal{D}_n)_{\emptyset, n-2, n-1}^{\approx} \not\models Q_n$.

Conversely, take $Q() = (\mathcal{O}, \Sigma, q) \in (\mathcal{ALCI}, CQ)$ where

$$\begin{aligned} \mathcal{O} &= \{A \sqsubseteq \forall s. B \sqcup \forall s^2. B \sqcup \forall s^3. B\} \\ \Sigma &= \{A, A_1, A_2, A_3, B, r, s\} \\ q &= \exists x_1 \exists x_2 \exists x_3 \exists y \bigwedge_{1 \leq i \leq 3} r^2(x_i, y) \wedge A_i(x_i) \wedge B(y) \end{aligned}$$

¹Note that \mathcal{D}_q being of treewidth $(1, k')$ is a stricter condition than q being of treewidth $(1, k')$.

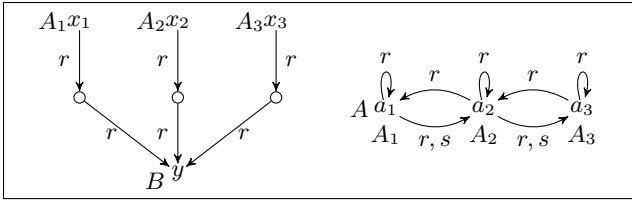


Figure 2: Query $q()$ and database \mathcal{D} from (2nd part of) Example 9.

and let

$$\mathcal{D} = \{A(a_1), A_i(a_i), s(a_i, a_{i+1}), r(a_i, a_{i+1}), r(a_{i+1}, a_i), r(a_i, a_i), r(a_3, a_3), A_3(a_3) \mid 1 \leq i < 3\}.$$

A picture is provided in Figure 2. The algorithms underlying the theorems in this and the previous section can be used to show that $\mathcal{D}_{\emptyset,1,2}^{\approx} \models Q$, but $\mathcal{D} \not\models Q_{\omega,\omega,\omega}^{\approx}$.

A straightforward variation of Example 7 shows that the parameter k indeed gives rise to an infinite hierarchy of increasingly more complete approximations.

Proposition 4. *Let $\ell, k, \ell', k' \geq 1$ with $\ell < k$ and $\ell' < k'$.*

1. In $(\mathcal{ALCC}, UCQ_{\ell,k}^{vw})$, ℓ, k -database relaxing OMQ evaluation is at least as complete as ℓ, k, ℓ', k' -ontology relaxing OMQ evaluation.
2. For OMQ from $(\mathcal{ALCC}, CQ_{\ell',k'}^{vw})$ of arity at most $r \leq 1$, $\ell + r, k + r, \ell', k'$ -ontology relaxing OMQ evaluation is at least as complete as ℓ, k -database relaxing OMQ evaluation.

We remark that Point 2 of Proposition 4 no longer holds if CQs are replaced by UCQs. We now formulate the main result of this section.

Theorem 6. *Let $\ell, k \geq 1$, $\ell < k$. Then ℓ, k -database relaxing OMQ evaluation is*

1. 2EXPTIME-complete in combined complexity and fixed-parameter tractable with double exponential running time in $(\mathcal{ALCC}, \mathcal{Q})$, $\mathcal{Q} \in \{CQ, UCQ, CQ_p^{vw}, UCQ_p^{vw} \mid p \geq 1\}$;
2. EXPTIME-complete in combined complexity and fixed-parameter tractable with single exponential running time in $(\mathcal{ALCC}, \mathcal{Q})$ and $(\mathcal{ALCC}, \mathcal{Q})$, $\mathcal{Q} \in \{AQ, bELIQ\}$.

We thus achieve FPT even for (U)CQs of unbounded treewidth, as for tree-database relaxing approximations, but not linear time. The lower bounds are proved exactly as for Theorem 2. The upper bounds are shown by an elimination approach. We remark that ℓ, k -database relaxing approximation coincides with the answers given by canonical (ℓ, k) -Datalog programs, see (Feier, Kuusisto, and Lutz 2019).

Recall that for tree-database unraveling approximation, the combined complexity *increases* for OMQ languages based on \mathcal{ALC} and (U)CQs. It seems clear that the same is true for BTW-database relaxing approximation and that it can be proved by adapting the proof of Theorem 3. A central idea is to replace single constants in databases by cliques.

Conjecture 1. *Let $\ell, k \geq 1$, $\ell < k$. Then ℓ, k -database relaxing OMQ evaluation in (\mathcal{ALC}, UCQ) is 2EXPTIME-hard.*

8 Approximation from Above

The approximations studied so far are from below, thus sound but incomplete. We define dual approximations from above that are complete but unsound: ontology strengthening approximation and database strengthening approximation. It turns out that these are computationally less well-behaved. While the former may increase combined complexity, the latter does not even enjoy PTIME data complexity.

We start with ontology strengthening approximation. For an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q)$, a Σ -database \mathcal{D} , and an ontology language \mathcal{L}' , we use $\text{app}_{\mathcal{L}'}^{\uparrow}(Q, \mathcal{D})$ to denote the set of tuples $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$ such that $\bar{a} \in Q'(\mathcal{D})$ for all OMQs $Q' = (\mathcal{O}', \Sigma, q)$ where \mathcal{O}' is an \mathcal{L}' -ontology with $\mathcal{O}' \models \mathcal{O}$. Every choice of $(\mathcal{L}, \mathcal{Q})$ and \mathcal{L}' gives rise to an approximate OMQ evaluation problem.

\mathcal{L}' -ontology strengthening evaluation in $(\mathcal{L}, \mathcal{Q})$
 INPUT : OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{L}, \mathcal{Q})$,
 Σ -database \mathcal{D} , tuple $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$
 OUTPUT : ‘yes’ if $\bar{a} \in \text{app}_{\mathcal{L}'}^{\uparrow}(Q, \mathcal{D})$ and ‘no’ otherwise

We consider \mathcal{ELI}_{\perp} -ontology strengthening OMQ evaluation in $(\mathcal{ELIU}_{\perp}, UCQ)$ where \mathcal{ELIU}_{\perp} is the extension of \mathcal{ELI}_{\perp} with disjunction. Note that we can find an implying \mathcal{ELI}_{\perp} -ontology for every \mathcal{ELIU}_{\perp} -ontology, namely $\{\top \sqsubseteq \perp\}$. In contrast to ontology relaxing approximation, it does not seem beneficial to use the universal role. The following example illustrates unsoundness.

Example 10. *Take $Q(x) = (\mathcal{O}, \Sigma, q)$ where*

$$\begin{aligned} \mathcal{O} &= \{A \sqsubseteq A_1 \sqcup A_2, \\ &\quad \exists r.(A_i \sqcap B_1) \sqcap \exists r.(A_i \sqcap B_2) \sqsubseteq B \mid i \in \{1, 2\}\} \\ \Sigma &= \{A, A_1, A_2, B, B_1, B_2, r\} \\ q &= B(x). \end{aligned}$$

Let $\mathcal{D} = \{r(a, b_1), r(a, b_2), A(b_1), B_1(b_1), A(b_2), B_2(b_2)\}$. Then $a \in \text{app}_{\mathcal{ELI}_{\perp}}^{\uparrow}(\mathcal{D})$ as every \mathcal{ELI}_{\perp} -ontology \mathcal{O}' with $\mathcal{O}' \models \mathcal{O}$ implies $A \sqsubseteq A_1$ or $A \sqsubseteq A_2$. But $a \notin Q(\mathcal{D})$.

Let \mathcal{O} be an \mathcal{ELIU}_{\perp} -ontology. A set \mathcal{M} of \mathcal{ELI}_{\perp} -ontologies is an *exhaustive \mathcal{ELI}_{\perp} -approximation set* for \mathcal{O} if $\hat{\mathcal{O}} \models \mathcal{O}$ for every $\hat{\mathcal{O}} \in \mathcal{M}$, and for every \mathcal{ELI}_{\perp} -ontology \mathcal{O}' with $\mathcal{O}' \models \mathcal{O}$, there is an $\hat{\mathcal{O}} \in \mathcal{M}$ such that $\mathcal{O}' \models \hat{\mathcal{O}}$. Such sets \mathcal{M} are interesting because for all OMQs $Q = (\mathcal{O}, \Sigma, q)$ with q a UCQ and all Σ -databases \mathcal{D} , $\text{app}_{\mathcal{ELI}_{\perp}}^{\uparrow}(Q, \mathcal{D}) = \bigcap_{\hat{\mathcal{O}} \in \mathcal{M}} Q_{\hat{\mathcal{O}}}(\mathcal{D})$ where $Q_{\hat{\mathcal{O}}} = (\hat{\mathcal{O}}, \Sigma, q)$. Consider for instance $\mathcal{O} = \{\top \sqsubseteq A_1 \sqcup A_2\}$, for which $\{\mathcal{O}_1, \mathcal{O}_2\}$ is an exhaustive \mathcal{ELI}_{\perp} -approximation set where $\mathcal{O}_i = \{\top \sqsubseteq A_i\}$. In the appendix, we show how to construct a finite exhaustive \mathcal{ELI}_{\perp} -approximation set for any given \mathcal{ELIU}_{\perp} -ontology and use this to prove the upper bounds in the following result.

Theorem 7. *Let $\mathcal{L} \in \{AQ, CQ, UCQ\}$. \mathcal{ELI}_{\perp} -ontology strengthening OMQ evaluation in $(\mathcal{ELIU}_{\perp}, \mathcal{L})$ is 2EXPTIME-complete in combined complexity and FPT with double exponential running time.*

	Ontology relaxing		Database relaxing	
	\mathcal{ELI}_{\perp}^u	TGD	trees	bounded treewidth
$(\mathcal{ALC}(\mathcal{I}), \text{bELIQ})$	LIN / EXP TIME	FPT / EXP TIME	LIN / EXP TIME	FPT / EXP TIME
$(\mathcal{ALC}, \text{CQ})$	P TIME / EXP TIME		FPT / EXP SPACE-2 EXP TIME	FPT / in 2 EXP TIME
$(\mathcal{ALC}, \text{UCQ})$	P TIME / EXP TIME		FPT / 2 EXP TIME	FPT / in 2 EXP TIME
$(\mathcal{ALC}(\mathcal{I}), (\text{UCQ})_k^{\text{hw}})$	FPT / EXP TIME		FPT / 2 EXP TIME	
$(\mathcal{ALC}(\mathcal{I}), (\text{UCQ}))$	P TIME / EXP TIME		FPT / 2 EXP TIME	

Figure 3: Results for approximate OMQ evaluation, data and parametric complexity / combined complexity.

The lower bound is proved by a reduction from the word problem for a suitable kind of alternating Turing machine. We consider the lower bound for $(\mathcal{ELIU}_{\perp}, \text{AQ})$ surprising as non-approximate OMQ evaluation is only EXP TIME-complete (Lutz 2008). Thus, approximate OMQ evaluation is significantly harder, the only result of this kind in the current paper that applies to AQs. It is amusing to note that the lower bound depends only on disjunction on the left hand side of concept inclusions, which are syntactic sugar, but not on the seemingly much more ‘dangerous’ disjunctions on the right hand side. It is in fact a byproduct of our proofs that, without disjunctions on the left, \mathcal{ELI}_{\perp} -ontology strengthening OMQ evaluation in $(\mathcal{ELIU}_{\perp}, \text{UCQ})$ is only EXP TIME-complete. $\mathcal{ALC}(\mathcal{I})$ -ontologies can be rewritten in polynomial time into a ‘nesting-free’ normal form that is often used by reasoners and that has sometimes been presupposed for approximation (Zhou et al. 2015). The rewriting is not equivalence preserving, but only yields a conservative extension. $\mathcal{ALC}(\mathcal{I})$ -ontologies in this form can in turn be rewritten into an equivalent \mathcal{ELIU}_{\perp} -ontology without disjunction on the left. The following example shows that \mathcal{ALC} -ontologies that are not in normal form behave differently in that they may have only infinite exhaustive \mathcal{ELI}_{\perp} -approximation sets.

Example 11. Let $\mathcal{O} = \{\exists r. \top \sqcap \forall r. A \sqsubseteq B_1 \sqcup B_2\}$. Then for each $n \geq 1$, the \mathcal{ELI}_{\perp} -ontology

$$\mathcal{O}_n = \{\exists r. A \sqsubseteq \exists r^n. X, \exists r. (A \sqcap \exists r^{n-1}. X) \sqsubseteq B_1\}$$

is such that $\mathcal{O}_n \models \mathcal{O}$. It is easy to see that $\mathcal{O}_n \not\models \mathcal{O}_m$ when $n \neq m$ and any \mathcal{ELI}_{\perp} -ontology \mathcal{O}'_n with $\mathcal{O}_n \models \mathcal{O}'_n \models \mathcal{O}$ is equivalent to \mathcal{O}_n .

In the appendix, we give another example which shows that the effect pointed out in Example 11 also affects answers to OMQs. We leave the decidability and complexity of \mathcal{ELI}_{\perp} -ontology relaxing approximation in $(\mathcal{ALC}(\mathcal{I}), \text{UCQ})$ (without assuming normal form) as an open problem.

We next turn to database strengthening approximation. For an OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q)$, a Σ -database \mathcal{D} , and a class \mathfrak{D} of pointed databases, we use $\text{app}_{\mathfrak{D}}^{\uparrow}(Q, \mathcal{D})$ to denote the set of tuples $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$ such that for all $\langle \mathcal{D}', \bar{b} \rangle \in \mathfrak{D}$ and all homomorphisms h from \mathcal{D} to \mathcal{D}' with $h(\bar{a}) = \bar{b}$, $\bar{b} \in Q(\mathcal{D}')$. Every choice of $(\mathcal{L}, \mathcal{Q})$ and \mathfrak{D} gives rise to an approximate OMQ evaluation problem.

\mathfrak{D} -database strengthening evaluation in $(\mathcal{L}, \mathcal{Q})$
INPUT : OMQ $Q(\bar{x}) = (\mathcal{O}, \Sigma, q) \in (\mathcal{L}, \mathcal{Q})$,
 Σ -database \mathcal{D} , tuple $\bar{a} \in \text{adom}(\mathcal{D})^{|\bar{x}|}$
OUTPUT : ‘yes’ if $\bar{a} \in \text{app}_{\mathfrak{D}}^{\uparrow}(Q, \mathcal{D})$ and ‘no’ otherwise

A natural choice for \mathfrak{D} are classes of databases of bounded treewidth. We only consider here the class \mathfrak{D}_1 of pointed databases $\langle \mathcal{D}, \bar{a} \rangle$ where $\text{adom}(\mathcal{D}) \setminus \bar{a}$ has treewidth $(1, 2)$, i.e. it is a tree with multi-edge and self-loops admitted.

Example 12. Take $Q(x) = (\mathcal{O}, \Sigma, q)$ where

$$\mathcal{O} = \{A_i \sqcap A_j \sqsubseteq B \mid 1 \leq i < j \leq 3\}$$

$$\Sigma = \{A_1, A_2, A_3, B, r\}$$

$$q = \exists x B(x).$$

Let

$$\mathcal{D} = \{r(a_1, a_2), r(a_2, a_3), r(a_3, a_1), A_1(a_1), A_2(a_2), A_3(a_3)\}.$$

Then $() \in \text{app}_{\mathfrak{D}_1}^{\uparrow}(\mathcal{D})$, but $() \notin Q(\mathcal{D})$.

Examples 10 and 12 also show that \mathfrak{D}_1 -database strengthening approximation and \mathcal{ELI}_{\perp} -ontology strengthening approximation are incomparable as $a \notin \text{app}_{\mathfrak{D}_1}^{\uparrow}(\mathcal{D})$ in Example 10 and $() \notin \text{app}_{\mathcal{ELI}_{\perp}}^{\uparrow}(\mathcal{D})$ in Example 12. It turns out that \mathfrak{D}_1 -database strengthening approximation does not even enjoy P TIME data complexity, and this holds already when the original ontology is formulated in a tractable language.

Theorem 8. \mathfrak{D}_1 -database strengthening approximation is CONP-complete in data complexity in $(\mathcal{ALC}(\mathcal{I}), \text{UCQ})$. The lower bound already holds when the ontology is empty. It also holds in $(\mathcal{EL}, \text{CQ})$.

The interesting part is the lower bound, proved by a non-trivial reduction from the validity of propositional formulas.

9 Conclusion

We have introduced and studied various kinds of OMQ approximations. Our results on approximation from below are summarized in Figure 3 where all entries for combined complexity mean completeness results, unless stated otherwise. We believe that the results in this paper show that ontology relaxing approximation is preferable to database-relaxing approximation in most aspects. First, they are conservative in the sense that they are complete when the ontology is empty. Second, they have lower combined complexity. And third, also for the dual notion of ontology-strengthening approximation one attains P TIME data complexity. It would be interesting to generalize our approach to extensions of $\mathcal{ALC}(\mathcal{I})$ with, for instance, role hierarchies, transitive roles, and functional roles. Moreover, ontology strengthening approximation deserves further study.

Acknowledgements

Haga and Lutz were supported by DFG CRC 1320 Ease. Wolter was supported by EPSRC grant EP/S032207/1.

References

- Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.
- Baget, J.; Leclère, M.; Mugnier, M.; and Salvat, E. 2009. Extending decidable cases for rules with existential variables. In *Proc. of IJCAI*, 677–682.
- Baget, J.; Leclère, M.; and Mugnier, M. 2010. Walking the decidability line for rules with existential variables. In *Proc. of KR*. AAAI Press.
- Barceló, P.; Feier, C.; Lutz, C.; and Pieris, A. 2019. When is ontology-mediated querying efficient? In *Proc. of LICS*, 1–13. IEEE.
- Barceló, P.; Libkin, L.; and Romero, M. 2014. Efficient approximations of conjunctive queries. *SIAM J. Comput.* 43(3):1085–1130.
- Barceló, P.; Romero, M.; and Zeume, T. 2018. A more general theory of static approximations for conjunctive queries. In *Proc. of ICDT*, volume 98 of *LIPICs*, 7:1–7:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Bienvenu, M.; Ortiz, M.; Simkus, M.; and Xiao, G. 2013. Tractable queries for lightweight description logics. In *Proc. of IJCAI*, 768–774. IJCAI/AAAI.
- Bienvenu, M.; ten Cate, B.; Lutz, C.; and Wolter, F. 2014. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Trans. Database Syst.* 39(4):33:1–33:44.
- Bötcher, A.; Lutz, C.; and Wolter, F. 2019. Ontology approximation in Horn description logics. In *Proc. of IJCAI*, 1574–1580. ijcai.org.
- Calì, A.; Gottlob, G.; and Kifer, M. 2013. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.* 48:115–174.
- Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2015. Data complexity of query answering in description logics (extended abstract). In *Proc. of IJCAI*, 4163–4167. AAAI Press.
- Carral, D.; Feier, C.; Grau, B. C.; Hitzler, P.; and Horrocks, I. 2014. \mathcal{EL} -ifying ontologies. In *Proc. of IJCAR*, volume 8562 of *LNCS*, 464–479. Springer.
- Dowling, W. F., and Gallier, J. H. 1984. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Log. Program.* 1(3):267–284.
- Fan, W., and Geerts, F. 2012. *Foundations of Data Quality Management*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.
- Feier, C.; Kuusisto, A.; and Lutz, C. 2019. Rewritability in monadic disjunctive Datalog, MMSNP, and expressive description logics. *Log. Methods Comput. Sci.* 15(2).
- Fink, R., and Olteanu, D. 2011. On the optimal approximation of queries using tractable propositional languages. In *Proc. of ICDT*, 174–185. ACM.
- Gottlob, G.; Morak, M.; and Pieris, A. 2015. Recent advances in datalog[±]. In *Reasoning Web*, volume 9203 of *LNCS*, 193–217. Springer.
- Grohe, M. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM* 54(1):1:1–1:24.
- Haga, A.; Lutz, C.; Marti, J.; and Wolter, F. 2020. A journey into ontology approximation: From non-Horn to Horn. In *Proc. of IJCAI*, 1827–1833. ijcai.org.
- Haga, A.; Lutz, C.; Sabellek, L.; and Wolter, F. 2021. How to approximate ontology-mediated queries. *CoRR* abs/2107.05369.
- Hitzler, P., and Vrandečić, D. 2005. Resolution-based approximate reasoning for OWL DL. In *Proc. of ISWC*, 383–397. Springer.
- Johnson, D. S., and Klug, A. C. 1982. Testing containment of conjunctive queries under functional and inclusion dependencies. In *Proc. of PODS*, 164–169. ACM.
- Kaminski, M.; Nenov, Y.; and Grau, B. C. 2016. Datalog rewritability of disjunctive datalog programs and non-Horn ontologies. *Artif. Intell.* 236:90–118.
- Lutz, C. 2008. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR*, volume 5195 of *LNCS*, 179–193. Springer.
- Pan, J. Z., and Thomas, E. 2007. Approximating OWL-DL ontologies. In *Proc. of AAAI*, 1434–1439. AAAI Press.
- Schaerf, A. 1993. On the complexity of the instance checking problem in concept languages with existential quantification. *J. of Intel. Inf. Systems* 2:265–278.
- Selman, B., and Kautz, H. A. 1996. Knowledge compilation and theory approximation. *J. ACM* 43(2):193–224.
- Thomas, E.; Pan, J. Z.; and Ren, Y. 2010. TrOWL: Tractable OWL 2 reasoning infrastructure. In *Proc. of ESWC*, volume 6089 of *LNCS*, 431–435. Springer.
- Tserendorj, T.; Rudolph, S.; Krötzsch, M.; and Hitzler, P. 2008. Approximate OWL-reasoning with Screech. In *Proc. of RR*, volume 5341 of *LNCS*, 165–180. Springer.
- Zhou, Y.; Grau, B. C.; Nenov, Y.; Kaminski, M.; and Horrocks, I. 2015. PAGOdA: Pay-as-you-go ontology query answering using a datalog reasoner. *J. Artif. Intell. Res.* 54:309–367.