Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

How Hard is it to Decide if a Fact is Relevant to a Query?

Meghyn Bienvenu , Diego Figueira , Pierre Lafourcade
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France
{meghyn.bienvenu, diego.figueira, pierre.lafourcade}@u-bordeaux.fr

Abstract

We consider the following fundamental problem: given a database D, Boolean conjunctive query (CQ) q, and fact $f \in D$, decide whether f is *relevant* to q w.r.t. D, i.e., does f belong to a minimal subset $S \subseteq D$ such that $S \models q$. Despite being of central importance to query answer explanation, the combined complexity of deciding query relevance has not been studied in detail, leaving open what makes this problem hard, and which restrictions can yield lower complexity. Relevance has already been shown to be harder than query evaluation: namely, Σ_2^p-complete for CQs, even over a binary signature. We further observe that NP-hardness applies already to (acyclic) chain CQs. Our work identifies self-joins (multiple atoms with the same relation) as the culprit. Indeed, we prove that if we forbid or bound the occurrence of self-joins, then relevance has the same complexity as query evaluation, namely, NP (without structural restrictions) and LogCFL (for bounded hypertreewidth classes). In the ontology setting, we establish an analogous result for ontology-mediated queries consisting of a CQ and DL-Lite$_\mathcal{R}$ ontology, namely that relevance is no harder than query answering provided that we bound the interaction width (which generalizes both self-join width and a recently introduced 'interaction-free' condition). Our results thus pinpoint what makes relevance harder than query evaluation and identify natural classes of queries which admit efficient relevance computation.

1 Introduction

There has been considerable interest in both the database and knowledge representation communities on devising methods for explaining why a given query answer holds, resulting in a diversity of approaches. In the database area, proposals include qualitative notions based upon causality (Meliou et al. 2010) and provenance (Green and Tannen 2017), as well as quantitative notions of explanation based upon (causal or Shapley-value-based) responsibility measures, which involve assigning scores to facts based upon their contribution to making the query hold (Livshits et al. 2021; Bienvenu, Figueira, and Lafourcade 2024b; Bienvenu, Figueira, and Lafourcade 2025a). In the setting of ontology-mediated query answering (OMQA) (Poggi et al. 2008; Bienvenu and Ortiz 2015; Xiao et al. 2018), where query answers are computed by reasoning over the information contained in the data and the ontology, there have also been a diversity of explanation notions advanced in the literature. Some of these rely upon proofs which show step-by-step how a query answer is obtained (Borgida, Calvanese, and Rodriguez-Muro 2008; Alrabbaa et al. 2022), others compute minimal subsets of facts that make the query hold (Bienvenu, Bourgaux, and Goasdoué 2019; Ceylan et al. 2019; Ceylan et al. 2020; Ceylan et al. 2025), while some recent work has begun to explore quantitative responsibility measures (Bienvenu, Figueira, and Lafourcade 2024a; Bienvenu, Figueira, and Lafourcade 2025b).

In the present paper, our focus will be on explanations given as *minimal supports* (a.k.a. causes, MinEXes), i.e. subset-minimal sets of facts that suffice to obtain the query answer. More precisely, we shall be interested in the fundamental problem of *query relevance*, which is to decide whether a given fact from the dataset is *relevant* to the query, in the sense that it belongs to at least one minimal support. Indeed, identifying the relevant facts can be considered a central task in query answer explanation, which is useful both for summarizing the minimal supports and for aiding users in debugging unexpected query results. Relevance is also closely related to recently explored Shapley-based responsibility measures. Indeed, for classes of monotone queries such as conjunctive queries (with or without an ontology), a fact will be assigned a positive score iff it is relevant (cf. discussions in (Bienvenu, Figueira, and Lafourcade 2025a)). Finally, it is worth noting that notions of relevance have also been considered for other kinds of explanations, including abductive explanations (Eiter and Gottlob 1995) justifications for ontology axiom entailment (Peñaloza and Sertkaya 2017; Chen et al. 2022), explanations for query non-answers (Calvanese et al. 2013), and explanations for query (non)answers under various inconsistency-tolerant semantics (Bienvenu, Bourgaux, and Goasdoué 2019).

We briefly summarize here what is known about the complexity of the query relevance task, which has been explored for ontology-mediated queries (OMQs) formulated in description logic and existential rule ontology languages (Ceylan et al. 2019; Ceylan et al. 2020; Ceylan et al. 2025). For data complexity, in which the size of the query and ontology are treated as fixed, the task is known to be tractable for (unions) of conjunctive queries (CQs), as well as for ontology-mediated queries that can be rewritten as UCQs (i.e. for so-called FO-rewritable ontology languages). Indeed, this is an easy consequence of the fact that the size

	Query evaluation	Relevance	
		sjw<∞	sjw ∞
hw<∞	LogCFL-c	LogCFL-c	NP-c†
hw ∞	NP-c	NP-c	Σ_2^p-c‡

Table 1: Complexity results for classes of conjunctive queries with (un)bounded hypertree width (hw) and self-join width (sjw). Shaded cells indicate new results. † NP-hardness applies to chain CQs over a single binary relation. ‡ General Σ_2^p-hardness is known, we show it holds even for CQs with a single binary relation.

	OMQ evaluation		Relevance	
	iw<∞	iw ∞	iw<∞	iw ∞
tw=1, ℓ<∞	NL-c	LogCFL-c	NL-c	NP-c
tw <∞	LogCFL-c	NP-c	LogCFL-c	NP-h
tw ∞	NP-c		NP-c	Σ_2^p-c

Table 2: Complexity results for classes of ontology-mediated queries in ($\mathsf{DL\text{-}Lite}_{\mathcal{R}}, \mathsf{CQ}$), with (un)bounded treewidth (tw), interaction width (iw), or number of (ℓ) in acyclic CQs (iw= 1). Shaded cells indicate new results.

of minimal supports is bounded by a constant, making it possible to solve the relevance problem by reduction to first-order query evaluation. By contrast, for (ontology-mediated) query languages for which the size of minimal supports is unbounded, the relevance problem becomes NP-hard in data complexity. For combined complexity, the relevance task has been proven Σ_2^p-complete for conjunctive queries, even in the absence of an ontology and with only binary relations (Vaicenavičius 2020). Additional combined complexity results have been established for various kinds of OMQs, with complexities ranging from Σ_2^p to 2EXPTIME, depending on the expressivity of the ontology language.

While the preceding results show that relevance is generally harder than query evaluation, to the best of our knowledge, there has not been any exploration of what precisely makes relevance harder than query evaluation, nor the related question of which restrictions can lower the complexity. The purpose of our paper will thus be to pinpoint the sources of difficulty and identify classes of queries for which relevance is no harder than query evaluation, in particular, leading to the identification of tractable classes.

Contributions Our first contribution is to show that for conjunctive queries (without an ontology), it is the presence of self-joins (i.e. multiple query atoms using the same relation) that make relevance harder than query evaluation. We start by observing that for 'well-behaved' classes of conjunctive queries, like acyclic CQs and CQs of bounded hypertreewidth, for which query evaluation is tractable, the relevance problem is NP-complete. In fact, relevance is NP-hard even for acyclic chain CQs of the form $R(x_1, x_2) \wedge R(x_2, x_3) \wedge \cdots \wedge R(x_{n-1}, x_n)$ on a single relation (Theorem 4.2), which demonstrates that tractability cannot be regained by simplifying the query structure. However, if we restrict to CQs with 'few' self-joins (via a notion of self-join width, recently introduced by Bienvenu, Figueira, and Lafourcade (2025a)), then the complexity of relevance matches that of query evaluation, namely NP for general CQs or LogCFL if queries have bounded hypertree width (Theorem 4.5). These results are summarized in Table 1.

We then move to considering the relevance problem for OMQs consisting of a CQ and a description logic (DL) ontology. After observing that both conjunction ($\sqcap$) and qualified existential restrictions ($\exists R.C$) can make relevance hard even for atomic queries, we focus on ontologies formulated in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$, a prominent lightweight DL often used for OMQA (Calvanese et al. 2007). We introduce a novel notion of interaction width, which can be seen as generalizing both self-join width and a recently introduced 'interaction-free' property (Bienvenu, Figueira, and Lafourcade 2025b). Our second main contribution is to show that for OMQs consisting of a CQ and $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ ontology, bounding the interaction width makes relevance no harder than query evaluation. The obtained results, summarized in Table 2, show that simultaneously bounding the interaction width and treewidth enables efficient (LogCFL) relevance computation, and for acyclic bounded-leaf CQs, relevance drops to NL.

Finally, as a third and last contribution, we investigate a more abstract notion of relevance based on graph homomorphisms, which underlies the CQ relevance problem. Concretely, we study the question of, given two graphs G and G', whether a given edge belongs to a minimal homomorphic image of G in G'. We consider both directed and undirected variants of this problem, finding that they are both Σ_2^p-complete, thereby providing natural and simple Σ_2^p-complete problems of independent interest. As a corollary, we obtain that Σ_2^p-hardness of CQ relevance already holds over signatures consisting of a single binary relation.

Organization After preliminaries in Section 2, we introduce the relevance problem for CQs and OMQs in Section 3 together with some basic observations and reductions. We study the relevance problem for classes of CQs in Section 4 and the relevance problem for OMQs in Section 5. In Section 6, we introduce and study the Minimal Homomorphism Problem for graphs. We conclude in Section 7 with a discussion of future work. Omitted proofs are in the *long version* of this paper, which can be found at https://arxiv.org/abs/2604.22422.

2 Preliminaries

We introduce the key notions, terminology, and notation that will be used in this paper. For detailed introductions to databases and description logics, we refer readers to (Abiteboul, Hull, and Vianu 1995; Baader et al. 2017).

Complexity We shall refer by L and NL the complexity classes of deterministic and nondeterministic logarithmic space, respectively. NP is the class of decision problems solvable by a nondeterministic polynomial-time Turing machine, and Σ_2^p the class of decision problems solvable by a nondeterministic polynomial-time Turing machine with access to an NP oracle. The class LogCFL consists of all decision problems reducible to a context-free language, considered highly parallelizable, where

$$\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{LogCFL} \subseteq \mathsf{AC}^1 \subseteq \mathsf{NC}^2 \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \Sigma_2^p.$$

Databases A (relational) *database* D is a finite set of relational *facts* $R(\mathbf{a})$, where R is a *relation name* of arity $k \geq 1$ and $\mathbf{a}$ is a k-ary vector of *constants*. The *signature* of D is the set of relation names that occur in the facts of D together with their arity, and we speak of a *binary signature* if only unary and binary predicates are used.

Conjunctive Queries *Conjunctive queries* (CQ) are first-order formulas of the form $q(\mathbf{x}) = \exists \mathbf{y}\ \alpha_1 \wedge \cdots \wedge \alpha_n$ where the α_i are relational *atoms* that can contain constants and/or *variables*, where $\mathbf{x}$ is the vector of *free variables* of the formula. By *atoms*(q) we denote $\{\alpha_i\}_i$. We shall only be interested in *Boolean* CQs, that is, CQs with no free variables, and we will henceforth assume that all CQs are Boolean.

For any syntactic object O (e.g., database, query), we will use *vars*(O) and *const*(O) to denote the sets of variables and constants contained in O, and let *terms*$(O) :=$ *vars*$(O) \cup$ *const*(O) denote its set of *terms*.

A *homomorphism* $q \xrightarrow{hom} D$ from a CQ q to a database D is a function $h : terms(q) \to const(D)$ such that $h(c) = c$ for every $c \in const(q)$ and $R(h(t_1), \ldots, h(t_n)) \in D$ for every atom $R(t_1, \ldots, t_n)$ in q. A *homomorphism* $h : q \xrightarrow{hom} q'$ between CQs is defined similarly. A CQ is a *core* if all the homomorphisms $h : q \xrightarrow{hom} q$ are injective. It is known that for any (Boolean) CQ q, $D \models q$ iff there is a homomorphism $h : q \xrightarrow{hom} D$, and that a CQ q entails another CQ q' iff $q' \xrightarrow{hom} q$. We use $h(q)$ for the result of replacing each $x \in vars(q)$ with $h(x)$ and call $h(q)$ a *homomorphic image* of q.

Inequalities A CQ q with inequalities, or *CQ*$^{\neq}$, is a CQ extended with *inequality atoms* of the form $t \neq t'$ where t, t' are terms (i.e., variables or constants). For any database or CQ X, the notion of homomorphism $h : q \xrightarrow{hom} X$ is restricted in the expected way, by further requiring that $h(t) \neq h(t')$ whenever q contains the inequality atom $t \neq t'$.

Evaluation For a class $\mathcal{C}$ of CQs (or of CQ$^{\neq}$s), the *query evaluation problem* for $\mathcal{C}$ is the problem of, given a database D and a CQ $q \in \mathcal{C}$, deciding whether $D \models q$. The problem is well known to be NP-complete for the class of all CQs (Chandra and Merlin 1977), but it becomes tractable under structural restrictions. Except when explicitly stated otherwise, we always mean *combined complexity*, where both the query and database are part of the input (as compared to *data complexity*, where only D is considered as input).

Tree-like Queries An *acyclic CQ* is one in which there are no cycles in its underlying (hyper)graph. In the case of binary signatures this means that the underlying undirected graph, containing the edge $\{t, t'\}$ for each query atom $R(t, t')$, has no cycles. For signatures of bounded arity, the acyclicity is generalized to a measure of tree-likeness via the notion of *treewidth*. We refer the reader to (Gottlob et al. 2016, §3.1) for a definition. For unbounded signatures, several generalizations have been proposed, one of the most prominent being *hypertree width*; in particular, acyclic CQs are precisely the queries of hypertree width 1. We refer readers to (Gottlob et al. 2016, Definition 3.1 & following paragraph) for a definition. The hypertree width of a CQ$^{\neq}$ is defined to be that of the CQ obtained by considering $\neq$ as any ordinary binary relation.

Theorem 2.1 ((Gottlob, Leone, and Scarcello 2002))**.** *For every fixed $k > 0$, the query evaluation problem for CQ$^{\neq}$s of hypertree width at most k is* LogCFL*-complete under log-space reductions. The result holds also for self-join free CQs. Further, checking if a CQ$^{\neq}$ has hypertree width at most k is in* LogCFL.[1]

Description Logics A *description logic* (DL) *knowledge base* (KB) $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ consists of an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$, constructed from mutually disjoint sets $\mathsf{N_C}$ of *concept names* (unary relation names), $\mathsf{N_R}$ of *role names* (binary relation names), and $\mathsf{N_I}$ of *individual names* (here called constants). An *inverse role* has the form R^-, with $R \in \mathsf{N_R}$, and we use $\mathsf{N}_R^{\pm} = \mathsf{N_R} \cup \{R^- \mid R \in \mathsf{N_R}\}$ for the set of *roles*. The *ABox* is a finite set of facts (i.e., a database) over the binary signature given by $\mathsf{N_C}$ and $\mathsf{N_R}$. The *TBox* (ontology) is a finite set of *axioms*. Its form depends on the DL in question.

Our results primarily concern lightweight DLs of the *DL-Lite* family (Calvanese et al. 2007). We shall in particular consider the DL-Lite$_\mathcal{R}$ dialect, whose TBox axioms take the form of *concept inclusions* $B \sqsubseteq C$ and *role inclusions* $P \sqsubseteq S$, built according to the following grammar

$$B := A \mid \exists P \quad C := B \mid \neg B \quad S := P \mid \neg P$$

where $A \in \mathsf{N_C}$ and $P \in \mathsf{N}_R^{\pm}$. The logic DL-Lite$_{\mathsf{core}}$ is obtained from DL-Lite$_\mathcal{R}$ by disallowing role inclusions.

Another prominent lightweight DL that we shall briefly mention is $\mathcal{EL}$, where the TBox consists of *general concept inclusions* (GCIs) $D_1 \sqsubseteq D_2$ between concepts of the form:

$$D := \top \mid A \mid D \sqcap D \mid \exists R.D \qquad A \in \mathsf{N_C}, R \in \mathsf{N_R}$$

The semantics of DL KBs is defined using *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where the *domain* $\Delta^\mathcal{I}$ is a non-empty set and $\cdot^\mathcal{I}$ maps each $a \in \mathsf{N_I}$ to $a^\mathcal{I} \in \Delta^\mathcal{I}$, each $A \in \mathsf{N_C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and each $R \in \mathsf{N_R}$ to $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is extended to complex concepts and roles: $\top^\mathcal{I} = \Delta^\mathcal{I}$, $(\exists P)^\mathcal{I} = \{d \mid \exists e \in \Delta^\mathcal{I}, (d, e) \in P^\mathcal{I}\}$, $(R^-)^\mathcal{I} = \{(e, d) \mid (d, e) \in R^\mathcal{I}\}$, $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$. An interpretation $\mathcal{I}$ *satisfies a fact* $A(a)$ (resp. $P(b, c)$) if $b^\mathcal{I} \in A^\mathcal{I}$ (resp. $(b^\mathcal{I}, c^\mathcal{I}) \in P^\mathcal{I}$). $\mathcal{I}$ *satisfies a (concept or role) inclusion* $G \sqsubseteq H$ if $G^\mathcal{I} \subseteq H^\mathcal{I}$. We call $\mathcal{I}$ a *model of a KB* $\mathcal{K}$, denoted $\mathcal{I} \models \mathcal{K}$, if $\mathcal{I}$ satisfies all axioms in $\mathcal{T}$ and all facts in $\mathcal{A}$. A KB $\mathcal{K}$ is *consistent* if it has a model. A KB $\mathcal{K}$ entails an inclusion or fact γ, written $\mathcal{K} \models \gamma$, if every model $\mathcal{I}$ of $\mathcal{K}$ satisfies γ. Likewise, we write $\mathcal{K} \models \exists P(a)$ to mean $a^\mathcal{I} \in (\exists P)^\mathcal{I}$ in every model $\mathcal{I}$ of $\mathcal{K}$.

Ontology-Mediated Queries We say that a (Boolean) CQ q is *entailed from a KB* $\mathcal{K} = (\mathcal{A}, \mathcal{T})$, written $\mathcal{K} \models q$, if $\mathcal{I} \models q$ for every model $\mathcal{I}$ of $\mathcal{K}$. We may alternatively group $\mathcal{T}$ and q together as a *ontology-mediated query* (*OMQ*) $Q = (\mathcal{T}, q)$, in which case we may write $\mathcal{A} \models Q$ to mean $(\mathcal{A}, \mathcal{T}) \models q$. The notation $(\mathcal{L}, \mathcal{Q})$ is used for the class of all OMQs $(\mathcal{T}, q)$

[1] While the literature does not cover the class of CQ$^{\neq}$ strictly speaking, it can easily be seen that $\neq$-atoms can be handled in LogCFL for the query evaluation problem.

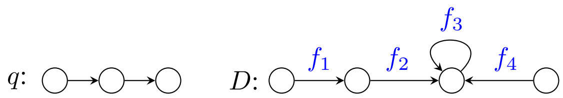

Figure 1: An example query and database to illustrate the notion of relevance. Arrows represent R atoms/facts.

consisting of a TBox formulated in the DL $\mathcal{L}$ and a query $q \in \mathcal{Q}$. Aside from conjunctive queries (CQ), we also consider classes of OMQs whose component CQs are *atomic queries* (AQ), i.e. CQs with a single atom.

The *query evaluation problem for a class of OMQs* is defined analogously as before: given an ABox $\mathcal{A}$ and an OMQ Q from the considered class, decide whether $\mathcal{A} \models Q$. The following theorem recalls some complexity results for query evaluation in subclasses of $(\textsf{DL-Lite}_\mathcal{R}, \textsf{CQ})$ (see (Artale et al. 2009; Bienvenu et al. 2018) and references therein).

Theorem 2.2. *Query evaluation for* $(\textsf{DL-Lite}_\mathcal{R}, \textsf{CQ})$ *is NP-complete. For fixed* $m \geq 1$, $\ell > 1$, *the problem is* LogCFL-*complete for the classes obtained by restricting to* $(\mathcal{T}, q)$ *s.t.*

- *q is acyclic and has at most ℓ leaves, or*
- *$\mathcal{T}$ is a* $\textsf{DL-Lite}_{\textsf{core}}$ *TBox and q has treewidth at most m*

Query evaluation is NL-*complete for* $(\textsf{DL-Lite}_\mathcal{R}, \textsf{AQ})$.

Canonical Model Every consistent $\textsf{DL-Lite}_\mathcal{R}$ KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ admits a so-called *canonical model* $\mathcal{I}_{\mathcal{A},\mathcal{T}}$, which maps homomorphically into every model of $\mathcal{K}$. As we shall use $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ in some of our results, we recall here its definition. For the domain $\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$, we use the set of all words $aP_1 \dots P_n$ $(n \geq 0)$ such that $a \in \mathit{const}(\mathcal{A})$, $P_i \in \mathsf{N}_R^{\pm}$, and:

- if $n \geq 1$, then $(\mathcal{A}, \mathcal{T}) \models \exists P_1(a)$
- for $1 \leq i < n$, $\mathcal{T} \models \exists P_i^- \sqsubseteq \exists P_{i+1}$ and $P_i^- \neq P_{i+1}$.

Elements in $\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \setminus \mathit{const}(\mathcal{A})$ will be called *anonymous elements*. The interpretation function maps each $a \in \mathit{const}(\mathcal{A})$ to itself and interprets concept and role names as follows:

$$\begin{aligned}
A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = &\{a \in \mathit{const}(\mathcal{A}) \mid (\mathcal{A}, \mathcal{T}) \models A(a)\} \cup \\
&\{aP_1 \dots P_n \mid n \geq 1 \text{ and } \mathcal{T} \models \exists P_n^- \sqsubseteq A\}) \\
R^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = &\{(a, b) \mid R(a, b) \in \mathcal{A}\} \cup \\
&\{(w_1, w_2) \mid w_2 = w_1 P \text{ and } \mathcal{T} \models P \sqsubseteq R\} \cup \\
&\{(w_2, w_1) \mid w_2 = w_1 P \text{ and } \mathcal{T} \models P \sqsubseteq R^-\})
\end{aligned}$$

Our results exploit the following well-known property:

Theorem 2.3. *For every* $\textsf{DL-Lite}_\mathcal{R}$ *KB* $(\mathcal{A}, \mathcal{T})$, $(\mathcal{A}, \mathcal{T}) \models q$ *iff* $\mathcal{I}_{\mathcal{A},\mathcal{T}} \models q$ *iff there is a homomorphism* $q \xrightarrow{hom} \mathcal{I}_{\mathcal{A},\mathcal{T}}$.

3 Query Relevance Problem

To improve the usability of information systems, it is important to be able to explain to users why a given answer $\mathbf{a}$ was obtained for the query q. Equivalently, we can rephrase this question in terms of Boolean queries, asking why the Boolean query $q(\mathbf{a})$ obtained by instantiating the free variables of the query with the answer tuple holds. Therefore, in what follows, we will focus w.l.o.g. on Boolean queries.

For queries that are monotone (i.e., such that adding facts can never change the result from true to false), such as CQs, a simple and natural way to explain why a query q holds is to exhibit minimal subset(s) of the data that make q true.

Definition 3.1. A *support* of a CQ q in a database D is any subset $S \subseteq D$ such that $S \models q$. It is a *minimal support* if further it does not strictly contain another support.

However, when a query has many minimal supports, it may be more useful to first present users with the set of facts that occur in at least one support. This is especially the case if a query (answer) unexpectedly holds, in which case one will need to determine which facts are at fault.

Definition 3.2. For a CQ q, database D and fact $f \in D$, we say that f is *relevant* to q in D if it belongs to some minimal support of q in D.

Example 3.3. *Consider the CQ q and database D depicted in Figure 1. There are two minimal supports for q in D, which are $\{f_1, f_2\}$ and $\{f_3\}$. This shows that f_1, f_2, and f_3 are all relevant. However, f_4 is not relevant despite appearing in a homomorphic image of q in D, because the only support containing f_4 is $\{f_3, f_4\}$, which is not minimal.* $\triangle$

To define analogous notions for ontology-mediated queries, one must decide how to treat the case where the knowledge base is inconsistent. Namely, do we want to consider minimal inconsistent subsets as supports for an OMQ? We believe it is more natural to separate debugging of inconsistencies from explaining query results, and so we shall require in our definitions that the KB is consistent (we shall discuss the case of inconsistent KBs in Section 7).

Definition 3.4. Consider an ontology-mediated query $Q = (\mathcal{T}, q)$ with $\mathcal{T}$ a DL TBox and q a CQ q, and an ABox $\mathcal{A}$ such that $(\mathcal{A}, \mathcal{T})$ is consistent. A *minimal support* for Q in $\mathcal{A}$ is an inclusion-minimal subset $S \subseteq \mathcal{A}$ such that $S \models Q$, and a fact $f \in \mathcal{A}$ is relevant to Q in $\mathcal{A}$ if f belongs to some minimal support of Q in $\mathcal{A}$.

Our focus will be on analyzing the complexity of deciding relevance for different classes of (ontology-mediated) CQs. The decision problems are formally defined as follows.

Relevance problem for a class $\mathcal{Q} \subseteq \textsf{CQ}$	
Input:	Query $q \in \mathcal{Q}$, database D, and fact $f \in D$
Question:	Is f relevant for q in D?

Relevance problem for OMQ class $(\mathcal{L}, \mathcal{Q})$ $(\mathcal{Q} \subseteq \textsf{CQ})$	
Input:	OMQ $(\mathcal{T}, q) \in (\mathcal{L}, \mathcal{Q})$, ABox $\mathcal{A}$ such that $(\mathcal{A}, \mathcal{T})$ is consistent, and fact $f \in \mathcal{A}$
Question:	Is f relevant for $(\mathcal{T}, q)$ in $\mathcal{A}$?

Remark 3.5. We treat these as standard decision problems in the complexity-theoretic sense, rather than 'promise problems'. That is, an algorithm for $\mathcal{Q}$ must first verify that the input string is a valid encoding of a query from $\mathcal{Q}$. $\triangle$

We conclude this section by stating some easy upper and lower bounds. First, we can observe that the relevance problem can be decided using a straightforward guess-and-check procedure: guess a subset $S \subseteq D$ that contains the given fact f, and verify that S is a minimal support. The latter can be checked by making repeated calls to a query evaluation oracle: we first check that the query holds in S, then check that the query no longer holds if any fact in S is removed. We thus obtain the following generic upper bounds:

Proposition 3.6. *For any class $\mathcal{Q}$ of (ontology-mediated) CQs, the relevance problem for $\mathcal{Q}$ is in NP^{C}, where C is the complexity class of the query evaluation problem for $\mathcal{Q}$.*

Regarding lower bounds, we cannot reasonably hope to achieve better complexity than for query evaluation. Indeed, a query q holds in a database D just in the case that some fact in D is relevant for q, which allows us to relate the complexities of the two problems as follows:

Proposition 3.7. *For any class $\mathcal{Q}$ of (ontology-mediated) CQs, the query evaluation problem for $\mathcal{Q}$ is in L^{C}, where C is the complexity class of the relevance problem for $\mathcal{Q}$.*

While the preceding result can be used to show that the relevance problem is intractable whenever query evaluation is, its formulation in terms of oracle calls is not the most convenient for establishing precise lower bounds. The following result provides a more direct (many-one) reduction:

Proposition 3.8. *Let $\mathcal{Q}$ be any class of CQs such that $q \in \mathcal{Q}$ implies that there exists $q \wedge \exists x.A(x) \in \mathcal{Q}$, for some relation A that does not occur in q. Then there is a many-one logspace reduction from the query evaluation problem for $\mathcal{Q}$ to the relevance problem for $\mathcal{Q}$. An analogous statement holds for OMQ classes, except that we must further require that the relation A does not occur in the TBox.*

Proof. Given $q \in \mathcal{Q}$ and a database D, let A be any unary relation not appearing in q. Clearly, $q' = q \wedge \exists x.A(x)$ and $D' = D \cup \{A(c)\}$ can be constructed from q, D using a logspace transducer. It then suffices to observe that $D \models q$ iff $A(c)$ is relevant to q' in D. ☐

The preceding proposition applies in particular to the whole class CQ, classes of CQs having hypertree width $< m$ (for any $m > 0$), and classes of acyclic CQs with at most ℓ leaves. It does not however cover classes of CQs with a fixed finite signature, nor the class of connected CQs.

4 Relevance for Conjunctive Queries

The relevance problem for unrestricted CQs is known to be harder than the evaluation problem, in fact one step higher in the polynomial hierarchy.

Theorem 4.1. (Vaicenavičius 2020, Thm. 4.30)[2] *The relevance problem for CQs is Σ_2^p-complete, and Σ_2^p-hardness holds already for constant-free CQs over a binary signature.*

We will see in the next subsections how two quantitative structural notions of queries affect this problem: the hypertree width and the amount of variables from different atoms that can be homomorphically merged into one fact.

[2]While the theorem statement in the dissertation is on *unions* of CQs, the proof makes use only of CQs over a binary signature.

4.1 CQs with Bounded Treewidth

While query evaluation of CQs is NP-hard, structural restrictions have been identified that make query evaluation tractable. This is in particular the case for any class of CQs with bounded hypertree width (Theorem 2.1). By Proposition 3.6, this lowers the complexity of relevance to NP. Interestingly, the NP lower bound applies even to (acyclic) *chain CQs* of the form $R(x_1, x_2) \wedge R(x_2, x_3) \wedge \cdots \wedge R(x_{n-1}, x_n)$ on a single binary relation R.

Theorem 4.2. *For any fixed $k \geq 1$, the relevance problem for the class of all CQs of hypertree width at most k is NP-complete. Hardness holds even for the class of chain CQs.*

Proof sketch. The minimal supports for the n-atom chain CQ q_n are either simple cycles or simple paths of length n: indeed, any set strictly containing a cycle cannot be a minimal support since the cycle satisfies q_n. We can then reduce from the existence of a Hamiltonian path on directed graphs. Given a directed graph G with n vertices we can produce a graph G' by adding two new vertices u, v and the edges (u, v) and (v, v') for every $v' \in V(G)$. Since (u, v) is in no cycle, checking whether (u, v) is relevant for q_{n+1} in G' is then equivalent to testing whether G contains a simple path of length $n - 1$ (i.e., a Hamiltonian path). ☐

4.2 CQs with Bounded Self-Join Width

Observe that the chain CQs of Theorem 4.2 need unboundedly many atoms on the same relation name, or 'self-joins'. A *self-join* is a pair of atoms on the same relation name, and a *self-join free* CQ is a CQ with no self-joins.

We show that if the number of self-join atoms in CQs is bounded, then the complexity of the relevance problem matches that of query evaluation, namely NP or even LogCFL if queries have bounded hypertree width (such as chain CQs). Instead of counting the number of self-joins, we use a more fine-grained notion of 'self-join width', which will allow showing tractability for broad classes of CQs.

For a tuple $\mathbf{t}$ or atom $\alpha = R(\mathbf{t})$ we write $\mathbf{t}[i]$ and $\alpha[i]$ to denote the i-th element of the $\mathbf{t}$-tuple. Two atoms α, β of a CQ q are *mergeable* if there are two homomorphisms $h_\alpha : \alpha \xrightarrow{hom} D$, $h_\beta : \beta \xrightarrow{hom} D$ to an arbitrary database D such that $h_\alpha(\alpha) = h_\beta(\beta)$ (in particular they must have the same relation name). An atom α is (individually) mergeable if it is mergeable with some other atom in q.

Definition 4.3 (Self-join width)**.** The *self-join width* of a CQ q is the cardinality of the following set of variables

$$\mathcal{M}_q := \{\alpha[i] : \alpha, \alpha' \text{ are mergeable atoms of } q \text{ with } \alpha[i] \neq \alpha'[i] \text{ and } \alpha[i] \in \mathit{vars}(q)\}.$$

This definition can be viewed as a generalization and improvement of a prior notion of self-join width introduced in (Bienvenu, Figueira, and Lafourcade 2025a). Note that any self-join free query has self-join width 0, but the converse does not hold for CQs with constants.

Example 4.4. *The CQ $q := \exists xy.R(c, x) \wedge R(c', y)$, where c, c' are distinct constants, has self-join width 0 since it has no two mergeable atoms. The CQ $q' := \exists xyz\ R(x, y) \wedge R(x, c) \wedge S(y, z)$ has self-join width 1 as $\mathcal{M}_{q'} = \{y\}$.* △

For classes of bounded self-join width CQs, the complexity of relevance matches that of query evaluation:

Theorem 4.5. *For any fixed $k \geq 0, \ell > 0$:*

- *The relevance problem for CQs of self-join width at most k is* NP*-complete.*
- *The relevance problem for CQs of self-join width at most k and hypertree width at most ℓ is* LogCFL*-complete.*

The rest of this section is devoted to proving this result. Let us fix the class $\mathcal{SJ}_k$ of CQs of self-join width at most k.

Lemma 4.6. *(a) Detecting if two given atoms are mergeable and more generally testing whether a CQ q is in $\mathcal{SJ}_k$ belongs to* NL. *(b) Computing $\mathcal{M}_q$ from a CQ $q \in \mathcal{SJ}_k$ is in* L.

We shall need to consider all the different possible ways of 'merging' two mergeable atoms by means of making some variables of $\mathcal{M}_q$ equal among them or among the constants contained in the query. For this, consider the set $\mathcal{E}_q$ of all equivalence relations over $\mathcal{M}_q \cup \mathit{const}(q)$ such that no two distinct constants are in the same equivalence class.

Remark 4.7. There is only a polynomial number of such equivalence relations $E \in \mathcal{E}_q$ and further the cardinality $|E|$ of each equivalence relation is of constant size $\leq k$, under the suitable encoding where singleton classes are implicit, and hence E can be stored in logarithmic space. △

For any equivalence relation $E \in \mathcal{E}_q$ let q_E be the result of collapsing all the terms in the same equivalence class in q and removing repeated atoms. Let us further define $q_E^{\neq}$ as the result of adding a inequality atom $t \neq t'$ to q_E for each pair $(t, t') \in (\mathcal{M}_q \times (\mathcal{M}_q \cup \mathit{const}(q))) \setminus E$. Intuitively, $q_E^{\neq}$ is the query stating that variables of $\mathcal{M}_q$ shall be collapsed[3] exactly according to E.

Lemma 4.8. *The problems of testing, given $q \in \mathcal{SJ}_k$ and two equivalence relations E, E' from $\mathcal{E}_q$, whether $q_E \xrightarrow{hom} q_{E'}$ holds, whether $q_E^{\neq} \xrightarrow{hom} q_{E'}^{\neq}$ holds, and whether q_E is a core, are all in* L.

The interest of the $q_E^{\neq}$ queries stems from the fact that, for a well-chosen subset of $\mathcal{E}_q$, they completely characterize the minimal supports of q, as the next lemma shows.

Lemma 4.9. *For every $q \in \mathcal{SJ}_k$ there exists a set $\widetilde{\mathcal{E}}_q \subseteq \mathcal{E}_q$ of equivalence relations, such that the following statements are equivalent for every database D:*

1. *$S \subseteq D$ is a minimal support for q,*
2. *$S = h(q_E)$ for some $E \in \widetilde{\mathcal{E}}_q$ and homomorphism $h : q_E^{\neq} \xrightarrow{hom} D$.*

Further, given an equivalence class $E \in \mathcal{E}_q$, one can test in L *whether $E \in \widetilde{\mathcal{E}}_q$ (under the encoding of Remark 4.7).*

Proof. The existence of $\widetilde{\mathcal{E}}_q$ essentially follows from the developments in (Bienvenu, Figueira, and Lafourcade 2025a, proof of Theorem 6.6), but here we need to adapt definitions to have a better space complexity. Concretely, we define

$$\widetilde{\mathcal{E}}_q := \{E \in \mathcal{E}_q : q_E \text{ is a core, and for all } E' \in \mathcal{E}_q, \text{ if } q_{E'}^{\neq} \xrightarrow{hom} q_E^{\neq} \text{ then } q_E^{\neq} \xrightarrow{hom} q_{E'}^{\neq}\}.$$

Claim 4.10. *$\widetilde{\mathcal{E}}_q$ satisfies the 1 ⇔ 2 equivalence.*

Since the cardinalities of the equivalence classes in $\mathcal{E}_q$ and $\widetilde{\mathcal{E}}_q$ are bounded by a constant (cf. Remark 4.7), they can be stored and manipulated using only logarithmic space. The complexity statement then follows from Lemma 4.8. ☐

With the previous lemma in place, Theorem 4.5 follows:

Proof sketch of Theorem 4.5. We are given a fact $f = R(\mathbf{c})$, database D and a CQ q. We first need to test if q meets the hypothesis. We test $q \in \mathcal{SJ}_k$ in NL due to Lemma 4.6-(a), and for the second statement we test hypertree width in LogCFL due to Theorem 2.1.

We now compute the set $\mathcal{M}_q$ in L via Lemma 4.6-(b), and we then iterate in logspace over all $E \in \mathcal{E}_q$ checking if there is one such E meeting the following criteria. We first check that $E \in \widetilde{\mathcal{E}}_q$ in L by Lemma 4.9. We next check that there is some R-atom $R(\mathbf{t})$ in q_E 'consistent' with our input fact $R(\mathbf{c})$, that is, so that $R(\mathbf{t}) \xrightarrow{hom} R(\mathbf{c})$. Consider now the result $\hat{q}$ of replacing in $q_E^{\neq}$ every $\mathbf{t}[i]$ which is a variable with the constant $\mathbf{c}[i]$. Observe that, due to Lemma 4.9, any homomorphic image of $\hat{q}$ is a minimal support of q containing f, and further if there exists a minimal support of q containing f, there must be some $E \in \widetilde{\mathcal{E}}_q$ and atom consistent with f as described before. All these operations can be performed in logarithmic space since there are logspace transductions from q to $q_E^{\neq}$, and from $q_E^{\neq}$ to $\hat{q}$, which can be composed. Note that the hypertree width of $\hat{q}$ is at most that of q plus k.

We finally check $D \models \hat{q}$ by a call to the query evaluation problem. This is in NP in general (Chandra and Merlin 1977, Theorem 7), or in LogCFL if we started with a class of bounded hypertree width by Theorem 2.1.

The lower bounds hold by Proposition 3.8 due to equivalent bounds for query evaluation of self-join free CQs. ☐

5 Relevance for Ontology-Mediated Queries

Now that we have a clear picture of the combined complexity of relevance for different classes of CQs, we shall push further and consider ontology-mediated queries $(\mathcal{T}, q)$ where $\mathcal{T}$ is a description logic ontology and q is a CQ. We will again be interested in understanding how the complexity of the relevance task compares to that of query evaluation.

5.1 New Sources of Hardness

The addition of an ontology introduces new sources of hardness for the relevance problem, which make the problem difficult even when restricted to atomic queries. Indeed, a first source of hardness stems from the ability to capture reachability in the data, which in DLs can be done using qualified existential restrictions, present in $\mathcal{EL}$ and its extensions. This makes relevance intractable even in data complexity:

[3] If the equivalence class X contains a constant c, then all variables of X are replaced by c; otherwise, we choose a representative variable $x \in X$ and replace all variables of X by x.

Proposition 5.1 ((Ceylan et al. 2020)). *Relevance for $(\mathcal{L}, \mathsf{AQ})$ is* NP*-hard in data complexity (hence also in combined complexity) for any DL $\mathcal{L}$ that can express $\exists R.A \sqsubseteq A$.*

Proof. Any directed graph G can be represented as an ABox $\mathcal{A}_G$ which contains $R(v_1, v_2)$ for each directed edge (v_1, v_2). Then, to decide whether an edge (v_1, v_2) in G lies on a simple path from s to t (an NP-hard problem, cf. (Khalil and Kimelfeld 2023, Lemma 5.3)), it suffices to check whether $R(v_1, v_2)$ is relevant for the OMQ $(\{\exists R.A \sqsubseteq A\}, A(s))$ w.r.t. the ABox $\mathcal{A}_G \cup \{A(t)\}$. □

Observe that the preceding result precludes the possibility of obtaining tractability for classes of OMQs defined by bounding any combination of width notions or any other parameters that assign a finite value to each OMQ.

In fact, $\mathcal{EL}$ contains a further source of hardness: concept conjunction. Indeed, if the ontology language can express propositional definite Horn clauses, then deciding relevance is hard (in combined complexity) even for atomic queries:

Proposition 5.2. *Relevance for the OMQ class $(\mathcal{L}, \mathsf{AQ})$ is* NP*-hard for any DL $\mathcal{L}$ with concept conjunction and GCIs.*

Proof. We reduce from SAT. Consider a CNF formula $\phi := \bigwedge_{j=1}^{m} c_j$ with variables in $V := \{v_1 \dots v_n\}$. We build the AQ $q := A(d)$, the ABox $\mathcal{A} := \{X(d), P_i(d), N_i(d) \mid 1 \leq i \leq n\}$, and the TBox $\mathcal{T} := \mathcal{T}_t \cup \mathcal{T}_p \cup \mathcal{T}_n \cup \mathcal{T}_c$ with:

$$\mathcal{T}_t : \{P_i \sqcap N_i \sqsubseteq A \mid i \in [n]\} \quad \mathcal{T}_p : \{P_i \sqsubseteq C_j \mid v_i \in c_j\}$$
$$\mathcal{T}_c : \{X \sqcap \textstyle\bigsqcap_{j=1}^{m} C_j \sqsubseteq A\} \qquad \mathcal{T}_n : \{N_i \sqsubseteq C_j \mid \overline{v}_i \in c_j\}$$

It can be verified that the fact $X(d)$ is relevant to $(\mathcal{T}, q)$ w.r.t. $\mathcal{A}$ iff ϕ is satisfiable. □

This implies that if the DL allows for concept conjunction, then we cannot hope to identify tractable classes of OMQs by bounding parameters which are dominated by the query size, since NP-hardness holds already for atomic queries.

In view of these results, the only DLs for which we can hope to obtain tractability results by imposing conditions on the query are logics that contain neither conjunction nor admit qualified existential restrictions on the left-hand-side of axioms. This naturally leads us to explore core fragments of the DL-Lite family, which verify these requirements.

5.2 General Results for DL-Lite

We shall henceforth focus on OMQs $(\mathcal{T}, q)$ whose TBox $\mathcal{T}$ is formulated in DL-Lite$_\mathcal{R}$. We consider DL-Lite$_\mathcal{R}$ since it is a well-known core dialect (notably underlying the OWL 2 QL profile) for which the combined complexity of OMQ evaluation has been well explored (Bienvenu et al. 2018).

First, we determine the combined complexity of testing relevance for arbitrary OMQs in (DL-Lite$_\mathcal{R}$, CQ), obtaining the same complexity as for CQs in databases:

Proposition 5.3. *The relevance problem for* (DL-Lite$_\mathcal{R}$, CQ) *is Σ_2^p-complete.*

Proof. The lower bound directly follows from Theorem 4.1, as the hardness proof only uses binary relations. The upper bound follows from Proposition 3.6 and NP membership for query evaluation in (DL-Lite$_\mathcal{R}$, CQ), see Theorem 2.2. □

Next, we pinpoint the complexity of well-behaved OMQ classes whose evaluation problem has been previously shown to be tractable (specifically, LogCFL-complete):

Proposition 5.4. *The relevance problem is* NP*-complete for the following classes of OMQs:*

- *class of OMQs $(\mathcal{T}, q) \in$* (DL-Lite$_\mathcal{R}$, CQ) *such that q is acyclic and has at most ℓ leaves (for any fixed $\ell > 1$)*
- *class of OMQs $(\mathcal{T}, q) \in$* (DL-Lite$_{\mathsf{core}}$, CQ) *such that q has treewidth at most m (for any fixed $m \geq 1$)*

Proof. The NP lower bounds follows from Theorem 4.2. For the upper bounds, we combine Proposition 3.6 with existing LogCFL results for OMQ answering in the considered classes, recalled in Theorem 2.2. □

The preceding results show that, just as in the plain database setting, the worst-case complexity of relevance is one level higher than query evaluation. Inspired by the positive impact of restricting self-joins, we shall next explore how an analogous notion can be employed to obtain lower complexities for deciding relevance of OMQs.

5.3 Bounded Interaction Width OMQs

The most obvious way of translating the notion of self-join-free queries to the OMQA setting would be to consider OMQs $(\mathcal{T}, q)$ where q is a self-join-free CQ. However, the resulting notion does not have the desired properties due to interactions between atoms that arise from the ontology:

Proposition 5.5. *The hardness of Proposition 5.3 holds with the restriction that the component CQs are self-join free.*

Proof. We reduce from the relevance problem of CQs with relations of arity 2. Let q be such a query, and D a database on the same signature, which we can see as an ABox since it does not have any fact of arity > 2. We remove all self-joins in q by replacing each instance of a relation R with a fresh R_i, along with the axiom $R_i \sqsubseteq R$ that makes R_i a particular instance of a R. Since D does not contain any of those R_i, the resulting OMQ will behave exactly as q on D and its subsets, hence the equivalence between the two when it comes to the relevance problem. □

This lead Bienvenu, Figueira, and Lafourcade (2025b) to define a notion of interaction-free OMQ, whose purpose is to ensure that an ABox fact can only be used to satisfy a single atom of the query (in a single way). We recall below the definition of interaction-free OMQs[4].

Definition 5.6 (Interacting query atoms). Given an OMQ $Q = (\mathcal{T}, q) \in$ (DL-Lite$_\mathcal{R}$, CQ), we say that distinct atoms α, β of q *interact* if there exists a fact f such that $(\{f\}, \mathcal{T}) \models \alpha$ and $(\{f\}, \mathcal{T}) \models \beta$ (with α and β treated as Boolean CQs). An atom α *interacts* with itself if there exist a fact f, homomorphisms $h_1 : \alpha \xrightarrow{hom} \mathcal{I}_{\{f\},\mathcal{T}}$ and $h_2 : \alpha \xrightarrow{hom} \mathcal{I}_{\{f\},\mathcal{T}}$, and variable $x \in \mathit{vars}(\alpha)$ such that $h_1(x) \in \mathit{const}(f)$ and $h_2(x) \neq h_1(x)$. We denote by *int-atoms*(Q) the set of atoms in Q that (self-)interact.

[4] We reformulated slightly the original definition to suit our purposes, but it yields the same notion of interaction-free OMQ.

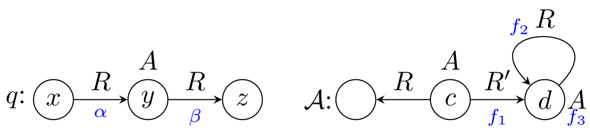

Figure 2: Interacting atoms where $\mathcal{T} := \{R' \sqsubseteq R, R' \sqsubseteq R^-\}$.

Definition 5.7 (Interaction-free OMQ)**.** An OMQ $Q \in (\mathsf{DL\text{-}Lite}_\mathcal{R}, \mathsf{CQ})$ is *interaction-free* if *int-atoms*$(Q) = \emptyset$.

Example 5.8. *Consider the KB* $(\mathcal{A}, \mathcal{T})$ *and CQ* q *depicted in Figure 2.* α *interacts with* β *via* f_1*, but both also interact with themselves by the two distinct homomorphisms that map to* (c, d) *in either direction, since* $\mathcal{I}_{\{f_1\},\mathcal{T}} \supseteq c \overset{R}{\leftrightarrow} d$*. This is especially important because, out of* $(x, y) \mapsto (c, d)$ *and* $(x, y) \mapsto (d, c)$*, only the latter witnesses the relevance of* f_1*: the former can only be extended with* $z \mapsto d$*, whose image strictly contains the minimal support* $\{f_2, f_3\}$*.* △

We propose to generalize interaction-free OMQs by introducing the notion of interaction width:

Definition 5.9 (Interaction width)**.** The *interaction width* of an OMQ $Q \in (\mathsf{DL\text{-}Lite}_\mathcal{R}, \mathsf{CQ})$ is $|$*int-atoms*$(Q)|$.

Observe that, as expected, interaction-free OMQs have interaction width zero. It is also worth noting that, differently from self-join width, we count the number of atoms in *int-atoms*(Q), not the number of variables in such atoms (but this difference is insignificant on binary signatures).

Example 5.10. *The OMQ* Q *from Figure 2 has interaction width 2, since int-atoms*$(Q) = \{\alpha, \beta\}$*. The OMQ* $Q'_n = (\mathcal{T}_n, q_n)$ *where* $q_n := \bigwedge_{i=1}^{n} R_i(x_{i-1}, x_i)$ *and* $\mathcal{T} := \{R_i \sqsubseteq R\}$ *has interaction width* n *since* $\alpha \xrightarrow{hom} \mathcal{I}_{\{R(c,d)\}}$ *for every atom* α *of* q*, hence int-atoms*$(Q'_n) =$ *atoms*(Q'_n)*.* △

The remainder of the section will be dedicated to establishing the following theorem, which shows that by bounding the interaction width, we can obtain the same complexity for query relevance as for query evaluation.

Theorem 5.11. *The relevance problem is in* NP *for every subclass of* $(\mathsf{DL\text{-}Lite}_\mathcal{R}, \mathsf{CQ})$ *having bounded interaction width.* LogCFL *(resp.* NL*) membership holds if we further require component CQs have bounded treewidth (resp. the component CQs are acyclic with bounded number of leaves).*

In what follows, we suppose that we have an input OMQ $Q = (\mathcal{T}, q)$ of interaction width k and an ABox $\mathcal{A}$. We shall call a fact $f \in \mathcal{A}$ *potentially relevant* to atom α in $(\mathcal{T}, q)$ if $(\{f\}, \mathcal{T}) \models \alpha$. The next lemma distinguishes two kinds of potentially relevant facts:

Lemma 5.12. *If a fact* $f \in \mathcal{A}$ *is relevant to* $Q = (\mathcal{T}, q)$ *w.r.t.* $\mathcal{A}$*, then* f *is potentially relevant to either (i) a single atom in atoms*$(q) \setminus$*int-atoms*(Q) *(and no other atom), or (ii) one or more atoms in int-atoms*(Q)*.*

We start by testing whether the given fact $f \in \mathcal{A}$ is potentially relevant to some atom of q, and return no if not. This can be done in NL by guessing a query atom and performing atomic query evaluation. We may thus focus on testing relevance for potentially relevant facts of types (i) or (ii).

Consider first the case where f is potentially relevant to a single atom $\alpha_f \in$ *atoms*$(q) \setminus$ *int-atoms*$(\mathcal{T}, q)$. Then there exists $h : \alpha_f \xrightarrow{hom} \mathcal{I}_{\{f\},\mathcal{T}}$, and moreover, every such homomorphism agrees on which variables are sent to which ABox constants (and which are mapped to the anonymous part). Define $q_{\text{-}\alpha_f}$ as the CQ obtained from q by removing α_f and replacing variable x by $h(x)$ if x occurs both in α_f and another atom of q. The following lemma provides a direct reduction of relevance to query evaluation:

Lemma 5.13. *Let* $f \in \mathcal{A}$ *be potentially relevant to* $\alpha_f \in$ *atoms*$(q) \setminus$ *int-atoms*$(\mathcal{T}, q)$*. Then* f *is relevant to* $(\mathcal{T}, q)$ *w.r.t.* $\mathcal{A}$ *iff* $(\mathcal{A}, \mathcal{T}) \models q_{\text{-}\alpha_f}$*.*

It remains to consider the more challenging case, where f is potentially relevant to some atom(s) in *int-atoms*(Q). The basic idea will be to iterate over subsets $S \subseteq \mathcal{A}$ of at most k facts which contain f and make true the subquery given by *int-atoms*(Q) and can be extended to build a minimal support for q. The following lemma makes precise which properties of S to check in order to conclude that f is relevant. It refers to the set *shared-vars*$(\mathcal{T}, q)$ of variables that occur both in *int-atoms*(Q) and in *atoms*$(q) \setminus$ *int-atoms*$(\mathcal{T}, q)$.

Lemma 5.14. *Let* $f \in \mathcal{A}$ *be potentially relevant to at least one atom in int-atoms*$(\mathcal{T}, q)$*. Then* f *is relevant to* $(\mathcal{T}, q)$ *w.r.t.* $\mathcal{A}$ *iff there exists a subset* $S \subseteq \mathcal{A}$ *with* $f \in S$ *and* $|S| \leq k$ *and* $h_S :$ *int-atoms*$(Q) \xrightarrow{hom} \mathcal{I}_{S,\mathcal{T}}$ *such that:*

1. *if* $x \in$ *shared-vars*$(\mathcal{T}, q)$*, then* $h_S(x) \in$ *const*$(\mathcal{A})$
2. $(\mathcal{A}, \mathcal{T}) \models q_{h_S}$ *where* q_{h_S} *is obtained by removing all atoms in int-atoms*$(\mathcal{T}, q)$ *and replacing* $x \in$ *shared-vars*$(\mathcal{T}, q)$ *by* $h_S(x)$
3. *there is no* $S' \subsetneq S$ *and* $h_{S'} :$ *int-atoms*$(Q) \xrightarrow{hom} \mathcal{I}_{S',\mathcal{T}}$ *such that* $h_{S'}(x) = h_S(x)$ *for every* $x \in$ *shared-vars*$(\mathcal{T}, q)$*.*

This suggests the following procedure for deciding relevance of facts of type (ii): iterate (in logspace) over all candidate sets S and homomorphisms h_S and check whether the required conditions hold. We terminate either when some S has been shown to satisfy the conditions (outputting 'relevant') or when there are no more candidates to test ('not relevant'). Note that the second condition requires a query evaluation check, the cost of which will depend on the form of the query q_{h_S}. For the third condition, we must perform a second (logspace) exploration of possible $S' \subsetneq S$ and $h_{S'}$.

If we consider CQs of bounded interaction width, without further structural restrictions, we can obtain an optimal NP upper bound by implementing the sketched procedure using a non-deterministic polytime Turing machine (using a non-deterministic guess to verify $(\mathcal{A}, \mathcal{T}) \models q_{h_S}$ in condition 2).

For the LogCFL and NL results, we further need to establish the complexity of query evaluation for structurally restricted classes of OMQs of bounded interaction width:

Theorem 5.15. *Query evaluation is in* LogCFL *for the class of OMQs* $(\mathcal{T}, q) \in (\mathsf{DL\text{-}Lite}_\mathcal{R}, \mathsf{CQ})$ *with interaction width at most* k *and treewidth at most* m *(for any fixed* $k \geq 0$*,* $m \geq 1$*). It is in* NL *if we further restrict to* $(\mathcal{T}, q)$ *such that* q *is acyclic and has at most* ℓ *leaves (for fixed* $\ell \geq 2$*).*

Proof sketch. Assuming w.l.o.g. that q is connected, we separately test for the existence of a homomorphism $h : q \xrightarrow{hom} \mathcal{I}_{\mathcal{A},\mathcal{T}}$ that fully maps q into the anonymous part, or one which sends at least one variable to an ABox constant. In the former case, we argue that $|q| \leq k$, so we can guess and check a (compact representation of a) potential homomorphism in NL. For the second kind of homomorphism, we show that all variables are mapped either to an ABox constant or to an anonymous element $aP_1 \dots P_n$ with $n \leq k$, enabling the reuse of techniques developed for bounded-depth ontologies (Bienvenu et al. 2018). □

Although Theorem 5.15 cannot be used directly to evaluate q_{h_S} (as the instantiation of shared variables can make an acyclic query become cyclic, or increase the treewidth), it is possible to simulate constants using fresh unary predicates, thereby retaining q's good structural properties and obtaining the desired LogCFL (resp. NL) upper bounds.

It is worth observing that for interaction-free OMQs,we can only have type (i) potentially relevant facts. Thus, we can decide relevance using AQ checks (to establish potential relevance), followed by evaluating the OMQ $(\mathcal{T}, q_{\text{-}\alpha_f})$. This makes relevance for this class of OMQs easily implementable on top of any OMQA system.

6 Minimal Homomorphisms on Graphs

Theorem 4.1 shows that the relevance problem for CQs is a step higher in the polynomial hierarchy than the corresponding query evaluation problem, and that the Σ_2^p-hardness already holds for binary signatures. We will now study the problem for simplest kind of databases or queries: *graphs*, either directed or undirected, which we call *digraphs* and *graphs*, respectively. We show that the problem remains Σ_2^p-complete even on this simple setting, which can be seen as a result of independent interest.

For digraphs G, G', a *G-homomorphic image* on G' is any subgraph $\hat{G}'$ of G' such that $V(\hat{G}') = Im(h)$ and $E(\hat{G}') = \{(h(v), h(v')) : (v, v') \in E(G)\}$ for some $h : G \xrightarrow{hom} G'$. We shall sometimes write $h(G)$ to denote $\hat{G}'$. The definition for graphs is analogous. Such G-homomorphic image is said to be *minimal* if it does not strictly contain any other G-homomorphic image. We can now define the *Minimal Homomorphism Problem* in its directed ($\mathsf{MinHom^d}$) and undirected ($\mathsf{MinHom^u}$) versions.

$\mathsf{MinHom^u}$ (resp. $\mathsf{MinHom^d}$) problem	
Input:	A pair G, G' of graphs (resp. digraphs), and an edge $e \in E(G')$.
Question:	Is e in some minimal G-homomorphic image on G'?

The problem $\mathsf{MinHom^d}$ is the natural equivalent to the relevance problem on a single binary relation:

Lemma 6.1. *There are logspace many-one reductions between* $\mathsf{MinHom^d}$ *and the relevance problem for the class of constant-free CQs over a single binary relation.*

As we show, it remains Σ_2^p-complete.

Theorem 6.2. $\mathsf{MinHom^d}$ *is Σ_2^p-complete.*

Proof sketch. The upper bound is straightforward. For the lower bound, we reduce from the relevance problem over binary relations $R_1, \dots, R_n$, known to be Σ_2^p-hard (Theorem 4.1). Given an input D, q, f of the relevance problem we consider digraphs G_q, G_D and an edge $e_f \in E(G_D)$ so that this is a positive instance of $\mathsf{MinHom^d}$ iff D, q, f is positive for relevance. The idea is simply to replace each atom or fact $R_i(a, b)$ with a long directed path from a to b containing a short directed cycle of a prime length p_i in the middle, where $p_1 < \dots < p_n$ are the first n prime numbers (which can be computed in polynomial time due to (Rosser 1939, Thm. 2)). In this way, we can build both G_q from q and G_D from D. Since prime p_i-cycles can only be homomorphically mapped to p_i-cycles, and the long paths prevent creating any other short cycle, a homomorphism $q \xrightarrow{hom} D$ can be equivalently seen as a homomorphism $G_q \xrightarrow{hom} G_D$ from prime cycles to prime cycles. By taking any edge e_f of the cycle contained in the replacement of fact f in G_D, we have that e_f belongs to a minimal G_q-homomorphic image on G_D iff f is relevant for q in D. □

As a corollary, we have that the relevance problem is already hard for CQs using a single binary relation.

Corollary 6.3 (of Theorem 6.2 and Lemma 6.1)**.** *The relevance problem for CQs is Σ_2^p-complete even on signatures having a single binary relation.*

Theorem 6.4. $\mathsf{MinHom^u}$ *is Σ_2^p-complete.*

Proof sketch. The lower bound is more involved than for $\mathsf{MinHom^d}$ since for undirected graphs we no longer have that prime p-cycles can only be homomorphically mapped to p-cycles (in fact they can map to any smaller odd cycle). We can however reduce from $\mathsf{MinHom^d}$ by making use of graph-theoretic techniques known as 'replacement methods'. □

7 Conclusion and Discussion

The main takeaway from our study is that the difficulty of the relevance problem for (DL-Lite$_\mathcal{R}$-mediated) CQs hinges fundamentally on the number of atoms (or of variables from different atoms) that may interact with a fact in the following sense: if we restrict the number of self-join variables or interacting atoms, then the complexity of relevance is essentially as good or bad as that of query evaluation. These insights allowed us to identify natural classes of queries for which relevance can be efficiently decided, and we expect that they will also prove useful when designing practical algorithms for the relevance problem.

Complexity Dichotomies & Parameterized Complexity It is a natural question whether bounding the self-join width or interaction width is a necessary condition for obtaining tractability or lower complexity. More precisely, one might try to prove a dichotomy result along the following lines: "For every recursively enumerable class C of conjunctive queries, if C has bounded self-join width and bounded (hyper)treewidth, then relevance is in polynomial time, and otherwise it is NP-hard". Unfortunately, it is not at all clear that such a dichotomy holds, as there is currently no such dichtomy known for the combined complexity of CQ evaluation, and existing research provides strong evidence against its existence (Grohe 2007). By contrast, an FPT / W[1]-hard dichotomy has been proven for the *parameterized* evaluation of CQs, using the treewidth (modulo cores) as the parameter (Grohe 2007). Hence, a promising direction is to investigate the parameterized complexity of the relevance problem.

Other Ontology Languages We conjecture that the notion of interaction width we introduced for $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ can be fruitfully applied also to linear existential rules, as they similarly enjoy the singleton support property (i.e. only one fact is needed to satisfy a query atom). For ontology languages which admit conjunction, and hence do not enjoy the singleton support property (since multiple facts may needed to infer a query atom), such as $\mathsf{DL\text{-}Lite}_{\mathsf{Horn}}$, new restrictions will be needed to define well-behaved OMQ classes admitting tractable relevance computation. Indeed, as noted in Section 5.1, the NP-hardness of relevance for atomic queries precludes any positive results for classes of $\mathsf{DL\text{-}Lite}_{\mathsf{Horn}}$ OMQs obtained via bounding parameters dominated by the query size. For non-first-order-rewritable DLs like $\mathcal{EL}$, Proposition 5.1 effectively rules out any tractability results in combined complexity, but it may still be worthwhile to develop pragmatic algorithms for deciding relevance in such logics.

Relevance w.r.t. Inconsistent KBs When studying relevance of OMQs, we focused on *consistent* knowledge bases. Since explaining the inconsistency of a given KB can be seen as explaining the Boolean query 'is the KB consistent?', we could naturally consider the relevance problem for inconsistency. We should be able to transfer upper bounds from the query setting, but not necessarily lower bounds, as such 'inconsistency queries' may be restricted in ways that lower the complexity. Indeed, in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$, relevance of inconsistency is easily shown to be in NL as minimal inconsistent subsets are of size at most 2. Alternatively, one may study the relevance problem for explanations of query answers under inconsistency-tolerant semantics. Interestingly, existing proposals of explanations for repair-based semantics (Bienvenu, Bourgaux, and Goasdoué 2019) are defined via minimal supports, so we expect that our results will prove useful.

Interaction Width Beyond Relevance We expect that our new notion of interaction width may prove useful for identifying tractable OMQ classes for other explanation-related tasks. In particular, it was recently shown that the class of interaction-free OMQs admits efficient counting of minimal supports (Bienvenu, Figueira, and Lafourcade 2025b), and we are cautiously optimistic that this positive result can be extended classes of OMQs in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ with bounded interaction width (in line with an analogous result shown in the database setting for classes of CQs with bounded self-join width (Bienvenu, Figueira, and Lafourcade 2025a)). As detailed in the cited works, counting minimal supports has applications in query answer explanation as the number of minimal supports can be used to assign numeric scores to facts based upon their contribution to making the query hold.

Acknowledgements

This work has been partially supported by ANR grants INTENDED (ANR-19-CHIA-0014) and EXPAND (ANR-25-CE23-1215).

AI Declaration

The authors have not employed any Generative AI tools.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of databases*, volume 8. Addison-Wesley Reading.

Alrabbaa, C.; Borgwardt, S.; Koopmann, P.; and Kovtunova, A. 2022. Explaining ontology-mediated query answers using proofs over universal models. In *Proceedings of the International Joint Conference on Rules and Reasoning (RuleML+RR)*, 167–182.

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

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

Bienvenu, M., and Ortiz, M. 2015. Ontology-mediated query answering with data-tractable description logics. In *Tutorial Lectures of the Reasoning Web (RW) Summer School*, volume 9203 of *Lecture Notes in Computer Science*. 218–307.

Bienvenu, M.; Kikot, S.; Kontchakov, R.; Podolskii, V. V.; and Zakharyaschev, M. 2018. Ontology-mediated queries: Combined complexity and succinctness of rewritings via circuit complexity. *Journal of the ACM* 65(5):28:1–28:51.

Bienvenu, M.; Bourgaux, C.; and Goasdoué, F. 2019. Computing and explaining query answers over inconsistent DL-Lite knowledge bases. *Journal of Artificial Intelligence Research (JAIR)* 64:563–644.

Bienvenu, M.; Figueira, D.; and Lafourcade, P. 2024a. Shapley value computation in ontology-mediated query answering. In *Principles of Knowledge Representation and Reasoning (KR)*.

Bienvenu, M.; Figueira, D.; and Lafourcade, P. 2024b. When is Shapley value computation a matter of counting? *Proceedings of the ACM on Management of Data (PACMMOD)* 2(2:PODS):1–24.

Bienvenu, M.; Figueira, D.; and Lafourcade, P. 2025a. Shapley revisited: Tractable responsibility measures for query answers. *Proceedings of the ACM on Management of Data (PACMMOD)* 3(2:PODS):1–26.

Bienvenu, M.; Figueira, D.; and Lafourcade, P. 2025b. Tractable responsibility measures for ontology-mediated query answering. In *Principles of Knowledge Representation and Reasoning (KR)*.

Borgida, A.; Calvanese, D.; and Rodriguez-Muro, M. 2008. Explanation in the DL-Lite family of description logics. In *Proceedings of the International Conference: On the Move to Meaningful Internet Systems (OTM)*, 1440–1457.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)*.

Calvanese, D.; Ortiz, M.; Simkus, M.; and Stefanoni, G. 2013. Reasoning about explanations for negative query answers in DL-Lite. *Journal of Artificial Intelligence Research (JAIR)* 48:635–669.

Ceylan, İ. İ.; Lukasiewicz, T.; Malizia, E.; and Vaicenavicius, A. 2019. Explanations for query answers under existential rules. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1639–1646.

Ceylan, İ. İ.; Lukasiewicz, T.; Malizia, E.; and Vaicenavicius, A. 2020. Explanations for ontology-mediated query answering in description logics. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 672–679.

Ceylan, İ. İ.; Lukasiewicz, T.; Malizia, E.; and Vaicenavicius, A. 2025. Explanations for query answers under existential rules. *Artif. Intell.* 341:104294.

Chandra, A. K., and Merlin, P. M. 1977. Optimal implementation of conjunctive queries in relational data bases. In *Symposium on Theory of Computing (STOC)*, 77–90. ACM Press.

Chen, J.; Ma, Y.; Peñaloza, R.; and Yang, H. 2022. Union and intersection of all justifications. In *International Conference on Semantic Web (ESWC)*, volume 13261 of *Lecture Notes in Computer Science*, 56–73. Springer.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42(1):3–42.

Gottlob, G.; Greco, G.; Leone, N.; and Scarcello, F. 2016. Hypertree decompositions: Questions and answers. In *ACM Symposium on Principles of Database Systems (PODS)*, 57–74. ACM Press.

Gottlob, G.; Leone, N.; and Scarcello, F. 2002. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences (JCSS)* 64(3):579–627.

Green, T. J., and Tannen, V. 2017. The semiring framework for database provenance. In *ACM Symposium on Principles of Database Systems (PODS)*, 93–99. ACM Press.

Grohe, M. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM* 54(1):1:1–1:24.

Khalil, M., and Kimelfeld, B. 2023. The complexity of the Shapley value for regular path queries. In *International Conference on Database Theory (ICDT)*, 11:1–11:19.

Livshits, E.; Bertossi, L.; Kimelfeld, B.; and Sebag, M. 2021. The Shapley value of tuples in query answering. *Logical Methods in Computer Science (LMCS)* Volume 17, Issue 3:6942.

Meliou, A.; Gatterbauer, W.; Moore, K. F.; and Suciu, D. 2010. The complexity of causality and responsibility for query answers and non-answers. *Proc. VLDB Endow.* 4(1):34–45.

Peñaloza, R., and Sertkaya, B. 2017. Understanding the complexity of axiom pinpointing in lightweight description logics. *Artif. Intell.* 250:80–104.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. In *Journal on Data Semantics X*, 133–173.

Rosser, B. 1939. The n-th prime is greater than $n\,log(n)$. *Proceedings of the London Mathematical Society* 2(1):21–44.

Vaicenavičius, A. 2020. *Explanations for ontology-mediated query answers*. Ph.D. Dissertation, University of Oxford, UK.

Xiao, G.; Calvanese, D.; Kontchakov, R.; Lembo, D.; Poggi, A.; Rosati, R.; and Zakharyaschev, M. 2018. Ontology-based data access: A survey. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 5511–5519.