Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Tractable Responsibility Measures for Ontology-Mediated Query Answering

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

Recent work on quantitative approaches to explaining query answers employs responsibility measures to assign scores to facts in order to quantify their respective contributions to obtaining a given answer. In this paper, we study the complexity of computing such responsibility scores in the setting of ontology-mediated query answering, focusing on a very recently introduced family of Shapley-value-based responsibility measures defined in terms of weighted sums of minimal supports (WSMS). By exploiting results from the database setting, we can show that such measures enjoy polynomial data complexity for classes of ontology-mediated queries that are first-order-rewritable, whereas the problem becomes #P-hard when the ontology language can encode reachability queries (via axioms like $\exists R.A \sqsubseteq A$). To better understand the tractability frontier, we next explore the combined complexity of WSMS computation. We prove that intractability applies already to atomic queries if the ontology language supports conjunction, as well as to unions of 'well-behaved' conjunctive queries, even in the absence of an ontology. By contrast, our study yields positive results for common DL-Lite dialects: by means of careful analysis, we identify classes of structurally restricted conjunctive queries (which intuitively disallow undesirable interactions between query atoms) that admit tractable WSMS computation.

☞ This pdf contains internal links: clicking on a notion leads to its *definition*. A long version of this paper can be found at https://arxiv.org/abs/2507.23191.

1 Introduction

The question of how to explain query answers has received significant attention in both the database and ontology settings. Qualitative notions of explanation, based e.g. on minimal supports of fact or proofs, have been more extensively explored, particular in the ontology setting, cf. (Borgida, Calvanese, and Rodriguez-Muro 2008; Alrabbaa et al. 2022; Bienvenu, Bourgaux, and Goasdoué 2019; Ceylan et al. 2019; Ceylan et al. 2020). However, there has been recent interest in quantitative notions of explanation based upon *responsibility measures*, which assign scores to the dataset facts to quantify their respective contributions to obtaining a given answer. Prior work on responsibility measures for query answers has predominantly focused on the so-called 'drastic Shapley value' (Livshits et al. 2021; Deutch et al. 2022; Khalil and Kimelfeld ; Kara, Olteanu, and Suciu 2024; Reshef, Kimelfeld, and Livshits 2020; Bienvenu, Figueira, and Lafourcade 2024b; Karmakar et al. 2024; Bienvenu, Figueira, and Lafourcade 2024a). The drastic Shapley value is defined as the Shapley value of the 0/1 modeling of the (Boolean) query. It was motivated by the appealing theoretical characterization of the Shapley value as a concept in cooperative game theory, which is the only distribution of wealth among players respecting certain guarantees, known as 'Shapley axioms' (Shapley 1953). The choice of modelling the query as a 0/1 cooperative game, however, has not been justified.

Unfortunately, the computation of the drastic Shapley value is generally intractable (#P-hard in data complexity), even in the absence of ontologies and for very simple (conjunctive) queries (Livshits et al. 2021; Bienvenu, Figueira, and Lafourcade 2024b). Furthermore, it has recently been argued in (Bienvenu, Figueira, and Lafourcade 2025) that: (i) not all Shapley axioms yield desirable properties when translated into the query answering setting, and (ii) the genuinely desirable properties for responsibility measures of query answers do not pinpoint a single best score function.

In light of this, (Bienvenu, Figueira, and Lafourcade 2025) has very recently proposed a family of responsibility measures, based on *weighted sums of minimal supports* (*WSMS*), where the score of a fact is defined as a weighted sum of the sizes of the query's minimal supports containing it. The cited work shows that WSMS satisfy several desirable properties and that they enjoy more favourable computational properties compared to the drastic Shapley value in the database setting. Further, WSMS can also be defined as the Shapley value of suitable cooperative games.

The positive results for WSMS in the database setting motivate us to investigate the complexity of computing WSMS responsibility scores in the more challenging setting of *ontology-mediated query answering* (*OMQA*) (Poggi et al. 2008; Bienvenu and Ortiz 2015; Xiao et al. 2018). For this first study of WSMS in the ontology setting, we focus on description logic (DL) ontologies (Baader et al. 2017), paying particular attention to DLs of the DL-Lite family (Calvanese et al. 2007), which are the most commonly adopted in OMQA, due to their favourable computational properties. We thus consider *ontology-mediated queries (OMQs)* of the form $(\mathcal{T}, q)$, where $\mathcal{T}$ is formulated in some DL and q is

either a conjunctive query (CQ) or atomic query.

Contributions Our results show that the good computational behaviour of WSMS in the database setting exhibited in (Bienvenu, Figueira, and Lafourcade 2025) extends to some relevant classes of ontology-mediated queries. This is in sharp contrast to the intractability of the drastic Shapley measure considered in the database (Livshits et al. 2021; Bienvenu, Figueira, and Lafourcade 2024b) and ontology (Bienvenu, Figueira, and Lafourcade 2024a) settings.

More precisely, we observe that WSMS computation is tractable in data complexity for any UCQ-rewritable OMQ. In particular, this covers the class of OMQs $(\mathcal{T}, q)$ consisting of a DL-Lite ontology and CQ q (Theorem 4). We show in fact that WSMS computation for such OMQs can be implemented using relational database systems via simple SQL queries (Theorem 5 & Corollary 8). We further define a class of 'well-behaved' OMQs composed of DL-Lite ontologies and bounded treewidth CQs for which we establish tractability also in combined complexity (corollary of Theorem 15).

We also identify DL constructs that render WSMS computation intractable. In particular, we show that the data complexity becomes #P-hard for classes of OMQs exhibiting reachability behaviour, e.g. admitting axioms $\exists R.A \sqsubseteq A$ (Corollary 10). Furthermore, the presence of concept conjunction, present in lightweight DLs like as $\mathcal{EL}$ and Horn dialects of DL-Lite, leads to #P-hardness in combined complexity (Proposition 11). Furthermore, we show that while UCQ-rewritable OMQs enjoy tractable data complexity, this is not the case for combined complexity, even if the rewriting falls within a very restrictive fragment of UCQs (namely, acyclic and self-join free) (Proposition 13).

Organization After reviewing basic terminology and notation, we recall in Section 3 the different responsibility measures, in particular, the recently introduced WSMS, explain how they can be applied in the OMQA setting, and define the WSMS computation task. Section 4 focuses on (in)tractability results on data complexity. The remaining sections consider combined complexity. Section 5 presents (in)tractability results for OMQs based upon atomic queries. Section 6 shows that UCQs are generally intractable for WSMS computation. Finally, Section 7 exhibits a condition ensuring polynomial-time tractability. We finish with some concluding remarks in Section 8.

2 Preliminaries

We recall key definitions and notation concerning description logics and ontology-mediated query answering.

Description Logic Knowledge Bases 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 predicates), $\mathsf{N_R}$ of *role names* (binary predicates), and $\mathsf{N_I}$ of *individual names* (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 *concept assertions* of the form $A(c)$ with $A \in \mathsf{N_C}, c \in \mathsf{N_I}$ and *role assertions* $r(b, c)$ with $r \in \mathsf{N_R}, b, c \in \mathsf{N_I}$. We use $\mathit{ind}(\mathcal{A})$ for the set of individual names in $\mathcal{A}$, and we write $R(b, c)$ to mean $r(b, c)$ if $R = r \in \mathsf{N_R}$ and $r(c, b)$ if $R = r^-$. The *TBox* (ontology) is a finite set of *axioms* whose form depends on the DL in question.

Many of our results concern lightweight DLs of the *DL-Lite* family. 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* $R \sqsubseteq S$, built according to the following grammar

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

where $A \in \mathsf{N_C}$ and $R \in \mathsf{N_R^\pm}$. The logic DL-Lite$_\mathsf{core}$ is obtained from DL-Lite$_\mathcal{R}$ by disallowing role inclusions. Another prominent lightweight DL is $\mathcal{EL}$, whose TBoxes consist of concept inclusions $D_1 \sqsubseteq D_2$ between $\mathcal{EL}$-concepts built as follows:

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

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 the interpretation function $\cdot^\mathcal{I}$ maps each $a \in \mathsf{N_I}$ to $a^\mathcal{I} \in \Delta^\mathcal{I}$, each $A \in \mathsf{N_C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, 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: $\top^\mathcal{I} = \Delta^\mathcal{I}$, $(\exists R)^\mathcal{I} = \{d \mid \exists e \in \Delta^\mathcal{I}, (d, e) \in R^\mathcal{I}\}$, $(r^-)^\mathcal{I} = \{(e, d) \mid (d, e) \in r^\mathcal{I}\}$, $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$. An interpretation $\mathcal{I}$ *satisfies an assertion* $A(a)$ (resp. $r(b, c)$) if $b^\mathcal{I} \in A^\mathcal{I}$ (resp. $(b^\mathcal{I}, c^\mathcal{I}) \in r^\mathcal{I}$). $\mathcal{I}$ *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 $\mathcal{T}$ (written $\mathcal{I} \models \mathcal{T}$) and all assertions in $\mathcal{A}$ (written $\mathcal{I} \models \mathcal{A}$). It will also be convenient to introduce notation $\mathcal{K} \models \exists R(a)$ (with $R \in \mathsf{N_R^\pm}$) to indicate that $a^\mathcal{I} \in (\exists R)^\mathcal{I}$ in every model $\mathcal{I}$ of $\mathcal{K}$. We use $\mathsf{Mod}(\mathcal{K})$ for the set of models of $\mathcal{K}$. A KB $\mathcal{K}$ is *consistent* if it has a model.

Databases While our main interest is in DL KBs, we shall import definitions and techniques from the database literature, so we briefly introduce some notation and terminology for databases. Formally, a (relational) *database* $\mathcal{D}$ is a finite set of relational *facts* $P(\vec{a})$, where P is a relational predicate of arity $k \geq 1$ and $\vec{a}$ is a k-ary vector of *constants* (which we may assume drawn from $\mathsf{N_I}$). The *signature* of $\mathcal{D}$ is the set of predicates that occur in the facts of $\mathcal{D}$, and we speak of a binary signature if only unary and binary predicates are used. We will use $\alpha \in \mathcal{D}$ to indicate that fact α occurs in $\mathcal{D}$.

Every database $\mathcal{D}$ can be equivalently viewed as a finite first-order interpretation $\mathcal{I}_\mathcal{D}$ whose domain is the set of constants in $\mathcal{D}$, which interprets every constant from $\mathcal{D}$ as itself, and which interprets each predicate P as follows: $P^\mathcal{I} = \{\vec{a} \mid P(\vec{a}) \in \mathcal{D}\}$. Moreover, it will sometimes prove convenient to treat an ABox $\mathcal{A}$ as a database and to consider the associated finite interpretation $\mathcal{I}_\mathcal{A}$.

Queries An (abstract, non-numeric) *query* of arity $k \geq 0$ is a function that maps every first-order interpretation $\mathcal{I}$ to a set $q(\mathcal{I})$ of k-tuples of domain elements. We will mostly work with *Boolean queries*, of arity 0, for which $q(\mathcal{I})$ can only take two values: $\emptyset$ which we interpret as 'false', and $\{()\}$ which we interpret as 'true'. In the latter case, we write

$\mathcal{I} \models q$. Queries may also be (and in fact are usually) evaluated over databases, in which case $q(\mathcal{D})$ returns a set of tuples of constants from $\mathcal{D}$ (called *answers*) and $\mathcal{D} \models q$ means that q evaluates to 'true' in the associated interpretation $\mathcal{I}_\mathcal{D}$.

We consider various concrete classes of *first-order queries* (FO-queries for short), which are defined as first-order logic formulas. Note that when querying DL KBs, the relational atoms in queries will use predicates from $\mathsf{N_C} \cup \mathsf{N_R}$, whereas in the database context, the atoms will use the available database predicates. The most prominent class of FO queries is CQ, the class of *conjunctive queries*, which are defined by formulas of the form $q(\vec{x}) = \exists \vec{y}. \alpha_1 \wedge \cdots \wedge \alpha_n$ where the α_i are relational *atoms* that can contain constants and/or *variables* in $\mathit{vars}(q) := \vec{x} \cup \vec{y}$, where $\vec{x}$ is the vector of free variables of the formula. A CQ q can be partitioned into is *connected components* which are the inclusion-maximal connected subqueries of q. When a CQ has no free variables it is naturally Boolean. We shall also consider the subclass AQ of CQs with a single atom called *atomic queries* (AQs), the class UCQ of *unions of conjunctive queries* defined as finite disjunctions of CQs with the same set of free variables, and the class (U)CQ$^{\neq}$ of (U)CQ with inequality atoms.

We will often treat CQs as sets of atoms and use notation like $\alpha \in q$ to indicate that atom α is a conjunct of the CQ q. It is well known that when q is a constant-free Boolean CQ, $\mathcal{I} \models q$ iff there is a homomorphism $h : q \xrightarrow{\mathit{hom}} \mathcal{I}$, i.e. a function h from $\mathit{vars}(q)$ to $\Delta^\mathcal{I}$ such that $P(\vec{x}) \in q$ implies that $h(\vec{x}) \in P^\mathcal{I}$ (or $P(h(\vec{x})) \in \mathcal{D}$ if we evaluate q over a database $\mathcal{D}$). This characterization extends to Boolean CQs with constants by adding the requirement that h maps every constant c to $c^\mathcal{I}$ (or to c itself, if we work with databases). Note that one can define in the same manner homomorphisms between two queries or between two interpretations.

Ontology-Mediated Query Answering We say that a Boolean query q is *entailed from a KB* $\mathcal{K}$, written $\mathcal{K} \models q$, if $\mathcal{I} \models q$ for every $\mathcal{I} \in \mathsf{Mod}(\mathcal{K})$. The *certain answers* to a non-Boolean k-ary query $q(\vec{x})$ *w.r.t.* a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ are the tuples $\vec{a} \in \mathit{ind}(\mathcal{A})^k$ such that $\mathcal{K} \models q(\vec{a})$, with $q(\vec{a})$ the Boolean query obtained by substituting $\vec{a}$ for $\vec{x}$.

Alternatively, we can treat $\mathcal{T}$ and q together as constituting a composite *ontology-mediated query* (*OMQ*) $Q = (\mathcal{T}, q)$, in which case we will write $\mathcal{A} \models (\mathcal{T}, q)$ to mean $(\mathcal{A}, \mathcal{T}) \models q$. The notation $(\mathcal{L}, \mathcal{Q})$ will be used to designate the class of all OMQs $(\mathcal{T}, q)$ consisting of a TBox formulated in the DL $\mathcal{L}$ and a query $q \in \mathcal{Q}$.

A common approach to computing certain answers (or checking query entailment) is to rewrite an OMQ into another query that can be directly evaluated using a database system. Formally, we call a query $q^*(\vec{x})$ a *rewriting of an OMQ* $(\mathcal{T}, q)$ if for every ABox $\mathcal{A}$ and candidate answer $\vec{a}$:

$$\mathcal{A} \models (\mathcal{T}, q(\vec{a})) \quad \text{iff} \quad \mathcal{A} \models q^*(\vec{a})$$

If we restrict the above definition by requiring that q^* belong to a class of queries $\mathcal{C}$ (*e.g.* FO or UCQ$^{\neq}$), we call it a *$\mathcal{C}$-rewriting*. Several dialects of DL-Lite, including DL-Lite$_\mathcal{R}$, are known to guarantee the existence of FO-rewritings, meaning that every OMQ from (DL-Lite$_\mathcal{R}$, CQ) possesses an FO-rewriting. Moreover, such rewritings in fact can be expressed in UCQ (or in UCQ$^{\neq}$, if the DL admits functional roles or number restrictions).

In Horn DLs, like $\mathcal{EL}$ and DL-Lite$_\mathcal{R}$, every consistent KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ admits a *canonical model* $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ with the property that for every model $\mathcal{J}$ of $\mathcal{K}$, there is a homomorphism $h : \mathcal{I}_{\mathcal{A},\mathcal{T}} \xrightarrow{\mathit{hom}} \mathcal{J}$ that is the identity on $\mathit{ind}(\mathcal{A})$. Importantly, if $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ admits a canonical model $\mathcal{I}_{\mathcal{A},\mathcal{T}}$, then for every CQ q and candidate answer tuple $\vec{a}$:

$$\mathcal{K} \models q(\vec{a}) \quad \text{iff} \quad \mathcal{I}_{\mathcal{A},\mathcal{T}} \models q(\vec{a})$$

The precise definition of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ depends on the particular Horn DL. In the case of DL-Lite$_\mathcal{R}$, the domain $\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$ of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ consists of all words $aR_1 \dots R_n$ ($n \geq 0$) such that $a \in \mathit{ind}(\mathcal{A})$, $R_i \in \mathsf{N}_R^{\pm}$, and:

- if $n \geq 1$, then $(\mathcal{A}, \mathcal{T}) \models \exists R_1(a)$ and there is no $b \in \mathit{ind}(\mathcal{A})$ such that $(\mathcal{A}, \mathcal{T}) \models R_1(a, b)$
- for $1 \leq i < n$, $\mathcal{T} \models \exists R_i^- \sqsubseteq \exists R_{i+1}$ and $R_i^- \neq R_{i+1}$.

Elements in $\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \setminus \mathit{ind}(\mathcal{A})$ will be called *anonymous elements*. The interpretation function is defined as follows:

$$\begin{aligned}
a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= a \text{ for all } a \in \mathit{ind}(\mathcal{A}) \\
A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \cap (\{a \in \mathit{ind}(\mathcal{A}) \mid (\mathcal{A}, \mathcal{T}) \models A(a)\} \cup \\
&\qquad \{aR_1 \dots R_n \mid n \geq 1 \text{ and } \mathcal{T} \models \exists R_n^- \sqsubseteq A\}) \\
r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} &= \left(\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}\right)^2 \cap (\{(a, b) \mid r(a, b) \in \mathcal{A}\} \cup \\
&\qquad \{(w_1, w_2) \mid w_2 = w_1 R' \text{ and } \mathcal{T} \models R' \sqsubseteq r\} \cup \\
&\qquad \{(w_2, w_1) \mid w_2 = w_1 R' \text{ and } \mathcal{T} \models R' \sqsubseteq r^-\})
\end{aligned}$$

Complexity We assume familiarity with the class FP of functions that can be computed in deterministic polynomial time, as well as the class #P of functions defined as counting the accepting runs of a nondeterministic Turing machine. All hardness results are defined from *polynomial-time Turing reductions*: we say that P_1 reduces to P_2 and write $P_1 \leqslant_\mathsf{P} P_2$ if the problem P_1 can be solved in polynomial time given access to a unit-cost oracle that solves P_2.

3 Responsibility Measures & Shapley Value

In this section, we recall the notion of responsibility measure, which provides quantitative explanations of query answers, and some concrete Shapley-value-based responsibility measures. We also explain and illustrate how these notions, defined for databases, transfer to the OMQA setting.

3.1 Responsibility Measures for Query Answers

Although we shall be interested in employing responsibility measures to quantify the contribution of facts to obtaining an answer $\vec{a}$ to a query $q(\vec{x})$, it will actually be more convenient to consider the equivalent task of quantifying contributions to satisfying the associated Boolean query $q(\vec{a})$ (obtained by instantiating the free variables $\vec{x}$ of q with $\vec{a}$). For this reason, in the remainder of the paper, we shall w.l.o.g. restrict ourselves to Boolean queries.

We shall further focus on monotone Boolean queries, defined in the database setting as queries q such that $\mathcal{D}_1 \models$

$q \Rightarrow \mathcal{D}_2 \models q$ whenever $\mathcal{D}_1 \subseteq \mathcal{D}_2$. Such queries notably include the class of homomorphism-closed queries, which covers most well-known classes of OMQs, as well as $\mathsf{UCQ}^{\neq}$. Note that a natural qualitative approach to explaining why a monotone Boolean query q holds in a database $\mathcal{D}$ is to consider the set $\mathsf{MS}_q(\mathcal{D})$ of *minimal supports* of q in $\mathcal{D}$, defined as inclusion-minimal subsets $\mathcal{D}' \subseteq \mathcal{D}$ s.t. $\mathcal{D}' \models q$.

Our focus will be on providing *quantitative* explanations in the form of *responsibility measures*, which are functions that assign a score to every fact in the data, reflecting their contributions to making the query hold. Such measures have been formally defined, in the database setting, as ternary functions ϕ that take as input a database $\mathcal{D}$, a (Boolean) query q and a fact $\alpha \in \mathcal{D}$, and output a numerical value. As this definition is extremely permissive, (Bienvenu, Figueira, and Lafourcade 2025, §4.1)[1] identifies a set of desirable properties that ϕ ought to satisfy, focusing on the case of (Boolean) monotone queries. While the formal definitions are rather technical and outside the scope of this paper, these properties intuitively state: *(Sym-db)* if two facts are interchangeable w.r.t. the query, they should have equal responsibility; *(Null-db)* if a fact $\alpha \in \mathcal{D}$ is *irrelevant* in the sense that $S \cup \{\alpha\} \models q$ iff $S \models q$ for all $S \subseteq \mathcal{D}$, then $\phi(\mathcal{D}, q, \alpha) = 0$, otherwise $\phi(\mathcal{D}, q, \alpha) > 0$; and *(MS1)* (*resp. (MS2)*) all other things being equal, a fact that appears in smaller (resp. more) minimal supports should have higher responsibility.

The notions of responsibility measures and minimal supports can be straightforwardly translated into the OMQA setting: it suffices to take the ABox as the database and use an OMQ $(\mathcal{T}, q)$ for the database query.

Example 1. *Consider the $\mathcal{EL}$ KB $(\mathcal{A}, \mathcal{T})$ defined in Figure 1 and the CQ* $\mathsf{FishBased}(x)$*, which we instantiate with the answer $\{x \mapsto cancalaiseSole\}$ to obtain the Boolean CQ $q := \mathsf{FishBased}(cancalaiseSole)$. There are 3 minimal supports for the OMQ $Q := (\mathcal{T}, q)$ in $\mathcal{A}$: $\{f_1, f_2\}$, $\{f_3, f_4, f_5\}$ and $\{f_3, f_6, f_7\}$. We illustrate how the properties defined above translate in this context: by (Sym-db), we have $\phi(\mathcal{D}, Q, f_1) = \phi(\mathcal{D}, Q, f_2)$ (as f_1 and f_2 appear in the same minimal supports); by (Null-db), we have $\phi(\mathcal{D}, Q, f_0) = 0$ (as it is irrelevant); by (MS1), we have $\phi(\mathcal{D}, Q, f_1) > \phi(\mathcal{D}, Q, f_4)$ (as f_1 appears in smaller supports); and by (MS2), $\phi(\mathcal{D}, Q, f_3) > \phi(\mathcal{D}, Q, f_4)$ (as f_3 appears in more supports).* △

3.2 Shapley-Based Responsibility Measures

The responsibility measures we consider in this paper are based on the 'Shapley value'. Originally defined in (Shapley 1953), it takes as input a cooperative game consisting of a finite set P of players and a *wealth function* $\xi : 2^P \to \mathbb{Q}$ that assigns a value to each coalition (*i.e.*, set) of players, with $\xi(\emptyset) = 0$. The *Shapley value* then assigns to each player $p \in P$ a value $\mathrm{Sh}(P, \xi, p)$ that should be seen as a fair share of the total wealth $\xi(P)$ of the game that should be awarded to p based on the respective contributions of all players. By "fair share" we mean that the Shapley value is, provably, the only wealth distribution scheme that satisfies a very natural set of axioms (Shapley 1953, Axioms 1 to 3).

To obtain a responsibility measure from the Shapley value, one needs to model the input instance $(\mathcal{D}, q)$ as a cooperative game (P, ξ). The set P contains the elements that will receive a score, hence it should naturally be the set $\mathcal{D}$ itself. As for the wealth function, it must assign a numerical score to every database, reflecting in some way the satisfaction of the query. Formally, one needs to provide a *wealth function family* $\Xi^\star := (\xi^\star_q)_q$ which associates a wealth function $\xi^\star_q$ with each query q. A responsibility measure can be straightforwardly obtained by applying the Shapley value to the game $(\mathcal{D}, \xi^\star_q)$:

$$\phi(\mathcal{D}, q, \alpha) := \mathrm{Sh}(\mathcal{D}, \xi^\star_q, \alpha)$$

The first wealth function family that was considered in the literature is Ξ^{dr}, defined by: $\xi^{\mathsf{dr}}_q(\mathcal{D}) := 1$ if $\mathcal{D} \models q$ and 0 otherwise (Livshits et al. 2021), which gives rise to the *drastic Shapley value* $\mathrm{Sh}(\mathcal{D}, \xi^{\mathsf{dr}}_q, \alpha)$. In fact, Ξ^{dr} was until recently the only wealth function used to define Shapley-based responsibility measures for Boolean queries.[2]

Very recently, however, a new family of responsibility measures called *weighted sums of minimal supports* (WSMSs) has been defined as follows:

$$\phi^w_{\mathsf{wsms}}(\mathcal{D}, q, \alpha) := \sum_{\substack{S \in \mathsf{MS}_q(\mathcal{D}) \\ \alpha \in S}} w(|S|, |\mathcal{D}|) \qquad (1)$$

based upon some *weight function* $w : \mathbb{N} \times \mathbb{N} \to \mathbb{Q}$ (Bienvenu, Figueira, and Lafourcade 2025). It has been shown that all WSMSs can be equivalently defined via the Shapley value: for every weight function w, there exists a wealth function family $\Xi^w = (\xi^w_q)_q$ such that

$$\phi^w_{\mathsf{wsms}}(\mathcal{D}, q, \alpha) = \mathrm{Sh}(\mathcal{D}, \xi^w_q, \alpha)$$

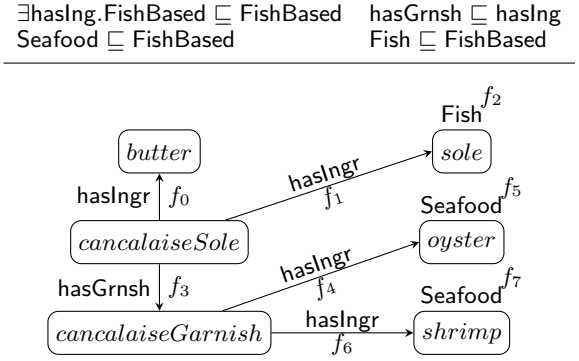

Figure 1: An example KB, with data and knowledge about a recipe from (Escoffier 1903). The arrows represent role assertions and labels around boxes (*e.g.* Fish) represent concept assertions.

[1] The definitions in the cited paper slightly differ from the ones we present here since we choose to assign scores to all facts, whereas in prior work, the database is partitioned into sets of *exogenous* and *endogenous* facts, with only endogenous facts assigned scores. Removing this distinction simplifies the technical presentation, while still covering what is arguably the most relevant practical setting (in which all facts are treated as endogenous).

[2] As a consequence, the drastic Shapley value is simply called 'Shapley value' in many papers.

$\star$	f_0	f_1, f_2	f_3	f_4, f_5, f_6, f_7
dr	0	$\frac{1224}{5040}$	$\frac{1056}{5040}$	$\frac{384}{5040}$
ms	0	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{3}$

Table 1: Values of $\mathrm{Sh}(\mathcal{D}, \xi_q^\star, \_)$ for the instance in Figure 1 . The facts that are equivalent by (Sym-db) are grouped together.

(Bienvenu, Figueira, and Lafourcade 2025, Proposition 4.4). The wealth function family $\Xi^{\mathsf{ms}} := \Xi^w$ induced by the inverse weight function $w : (n, k) \mapsto 1/n$ is of particular interest as its wealth function $\xi_q^{\mathsf{ms}}(\mathcal{D})$ is simply the number of minimal supports for q in $\mathcal{D}$, which constitutes a very natural measure of how 'robust' the entailment $\mathcal{D} \models q$ is.

It should be noted that the Shapley values obtained from Ξ^{dr} and from Ξ^w (for any positive and non-decreasing weight function w) yield responsibility measures that satisfy the properties (Sym-db)–(MS2) (Bienvenu, Figueira, and Lafourcade 2025, Propositions B.1 and B.2). As the following example illustrates, however, these measures do not always coincide, as the properties do not identify a unique 'reasonable' responsibility measure.

Example 2. *Reconsider the DL-Lite$_{\mathsf{core}}$ KB $(\mathcal{A}, \mathcal{T})$ from Figure 1 and the CQ $q := \mathsf{FishBased}(cancalaiseSole)$. Although the properties (Sym-db)–(MS2) enforce many conditions, they do not restrict the relative values of f_1 and f_3. Indeed, we can observe that $\mathrm{Sh}(\mathcal{D}, \xi_q^{\mathsf{dr}}, f_1) > \mathrm{Sh}(\mathcal{D}, \xi_q^{\mathsf{dr}}, f_3)$, but $\mathrm{Sh}(\mathcal{D}, \xi_q^{\mathsf{ms}}, f_1) < \mathrm{Sh}(\mathcal{D}, \xi_q^{\mathsf{ms}}, f_3)$, see Table 1. E.g., $\mathrm{Sh}(\mathcal{D}, \xi_q^{\mathsf{ms}}, f_3) = 1/3 + 1/3$ since f_3 is in two minimal supports, both of size 3, and hence each contributing $1/3$.* △

Note that while we focus on Ξ^{ms} as the archetypical WSMS, it should be observed that the weight function w can be adjusted to suit the needs of particular settings by giving more or less weight to the size of the minimal supports relative to their numbers (intuitively tuning the relative importance of (MS1) and (MS2)). At the extremes, (Bienvenu, Figueira, and Lafourcade 2025, §4.4) introduced two representative WSMS: Ξ^{s} that always favours appearing in the smallest minimal supports, and $\Xi^{\#}$ that always favours the highest total number of minimal supports.

Following (Bienvenu, Figueira, and Lafourcade 2025), for any wealth function family $\Xi^\star$ and class $\mathcal{C}$ of queries, we denote by $\mathrm{SVC}_{\mathcal{C}}^\star$ the problem of computing $\mathrm{Sh}(\mathcal{D}, \xi_q^\star, \alpha)$ given any database $\mathcal{D}$, fact $\alpha \in \mathcal{D}$, and query $q \in \mathcal{C}$. We also consider the problem $\mathrm{SVC}_q^\star$ associated with a single fixed query q. Our focus in this paper will be on the case $\Xi^\star = \Xi^w$ for some weight function w, in particular Ξ^{ms}, in which case we will speak of *WSMS computation*. Moreover, we shall study these tasks in the OMQA setting, so $\mathcal{C}$ will be a class $(\mathcal{L}, \mathcal{Q})$ of OMQs, and q will be a particular OMQ Q.

4 Data Complexity

We initiate our study of the complexity of WSMS computation by considering the *data complexity* of the task $\mathrm{SVC}^w_{(\mathcal{L},\mathcal{Q})}$, for different classes $(\mathcal{L}, \mathcal{Q})$ of OMQs. As usual, data complexity means that complexity is measured only w.r.t. the size of the ABox, while the size of the OMQ is treated as a constant. Observe that $\mathrm{SVC}^w_{(\mathcal{L},\mathcal{Q})}$ enjoys FP data complexity just in the case that SVC^w_Q is in FP for every OMQ $Q \in (\mathcal{L}, \mathcal{Q})$. Likewise, $\mathrm{SVC}^w_{(\mathcal{L},\mathcal{Q})}$ is #P-hard iff there is some OMQ $Q \in (\mathcal{L}, \mathcal{Q})$ for which SVC^w_Q is #P-hard.

Our data complexity results for OMQs naturally build upon existing results from the database setting (Bienvenu, Figueira, and Lafourcade 2025). That work establishes a key lemma that essentially tells us that, assuming the weight function w is tractable and reversible, the SVC^w_Q task boils down to counting minimal supports, no more nor less. It would be superfluous to give the precise definition of *tractable* and *reversible* weight functions here: simply note that they are minor assumptions that are trivially satisfied by the three explicit instances we consider (Ξ^{ms}, Ξ^{s} and $\Xi^{\#}$).To present this lemma in our setting, we fix a Boolean OMQ Q (*i.e.* $Q = (\mathcal{T}, q)$ with q a Boolean query) and consider two *numeric queries* (*i.e.* queries that output a number instead of a set of answers) derived from Q:

- $\#^{\mathsf{fms}}_Q$ outputs the number $\#^{\mathsf{fms}}_Q(k, \mathcal{A})$ of minimal supports for Q of size[3] k in the ABox $\mathcal{A}$, for every size k
- $\#^{\mathsf{ms}}_Q$ outputs the total number of minimal supports of Q.

We denote by EVAL-$\#^{\mathsf{fms}}_Q$ and EVAL-$\#^{\mathsf{ms}}_Q$ the problems of computing these numeric queries over any input ABox. We now state the key lemma, phrased for OMQs:

Lemma 3. *(Bienvenu, Figueira, and Lafourcade 2025, Lemmas 5.1 and 5.5) For every reversible tractable weight function w, and Boolean monotone OMQ Q,*

$$\text{EVAL-}\#^{\mathsf{ms}}_Q \leqslant_{\mathsf{P}} \mathrm{SVC}^w_Q \leqslant_{\mathsf{P}} \text{EVAL-}\#^{\mathsf{fms}}_Q.$$

4.1 Tractability Result for Rewritable OMQs

With this lemma at hand, we can forget about the particular weight function w and concentrate on the conceptually simpler task EVAL-$\#^{\mathsf{fms}}_Q$ of counting the number of minimal supports, per size. This can be achieved in polynomial time (in data complexity) whenever there is a data-independent bound on the size of minimal supports, since we can then simply iterate over all bounded-size subsets of the ABox. In particular, this holds for OMQs that can be rewritten into a $\mathsf{UCQ}^{\neq}$, yielding the following tractability result:

Theorem 4. *(Corollary of (Bienvenu, Figueira, and Lafourcade 2025, Theorem 5.2))* $\mathrm{SVC}^w_Q \in \mathsf{FP}$ *for every tractable weight function w and every Boolean OMQ Q that is $\mathsf{UCQ}^{\neq}$-rewritable. This implies, in particular, that $\mathrm{SVC}^w_{(\text{DL-Lite}_{\mathcal{R}},\mathsf{UCQ})}$ enjoys FP data complexity.*

Theorem 4 mentions $(\text{DL-Lite}_{\mathcal{R}}, \mathsf{UCQ})$ to give a concrete example of a tractable OMQ class, but naturally the same holds for other prominent DL-Lite dialects (like DL-Lite$_{\mathcal{F}}$ and DL-Lite$_{\mathsf{Horn}}$) known to be $\mathsf{UCQ}^{\neq}$-rewritable. Description logics outside of the DL-Lite family typically do not guarantee the existence of rewritings. However, techniques for identifying FO-rewritable (in fact, $\mathsf{UCQ}^{\neq}$-rewritable)

[3]FMS stands for *Fixed-size Minimal Supports*.

OMQs for various Horn DLs are known (Bienvenu et al. 2016) and have been successfully implemented for OMQs in $(\mathcal{EL}, \mathsf{CQ})$ (Hansen and Lutz 2017). The preceding tractability result is therefore relevant for a wider range of DLs.

Beyond providing theoretical results, rewriting is a powerful tool from a practical standpoint and has proven to be a key technique for obtaining efficient implementations of OMQA by leveraging existing and highly optimized relational database systems. Computing SVC^w_Q for $\mathsf{UCQ}^{\neq}$-rewritable OMQs Q is no exception, as it can be done by evaluating simple SQL queries in parallel. We phrase the next result for databases, as it is relevant to both the pure database and OMQA settings.

Theorem 5. *For every $\mathsf{UCQ}^{\neq}$ $\widetilde{q}$ and $k \in \mathbb{N}$, there exists a set $\mathcal{Q}^{\neq}$ of $\mathsf{CQ}^{\neq}$ queries such that, for every database $\mathcal{D}$,*

$$\#^{\mathsf{fms}}_{\widetilde{q}}(k, \mathcal{D}) = \sum_{q \in \mathcal{Q}^{\neq}} \#^{\mathsf{hom}}_q(\mathcal{D}) \cdot \gamma_q,$$

where $\#^{\mathsf{hom}}_q(\mathcal{D})$ is the number of homomorphisms from q to $\mathcal{D}$, and γ_q is a rational number computable from q. Further, every $q \in \mathcal{Q}^{\neq}$ is of at most quadratic size.

Proof. Given a query $\widetilde{q} \in \mathsf{UCQ}^{\neq}$ and a number k, we show how to build the set $\mathcal{Q}^{\neq}$ with the desired properties of the statement. Let us say that q' is a *reduct* of $q \in \mathsf{CQ}^{\neq}$ if q' is the result of collapsing variables of q and removing repeated atoms; in particular, q' is a homomorphic image[4] of q. Similarly, q' is a reduct of $\widetilde{q}$ if it is a reduct of a $\mathsf{CQ}^{\neq}$ therein. Consider the set $\mathcal{R}_k$ of all reducts q of $\widetilde{q}$ such that

(a) q has exactly k relational atoms;
(b) there is no reduct q' of $\widetilde{q}$ having strictly less than k relational atoms such that $q' \xrightarrow{hom} q$.

Let $\mathcal{Q}$ be the result of removing from $\mathcal{R}_k$ all redundant queries. Concretely, we initialize $\mathcal{Q} := \mathcal{R}_k$ and we apply the following *redundancy-removal-rule* until no more queries can be deleted: find a query $q \in \mathcal{Q}$ such that $q' \xrightarrow{hom} q$ for some distinct $q' \in \mathcal{Q}$ and update $\mathcal{Q} := \mathcal{Q} \setminus \{q\}$. Finally, let $\mathcal{Q}^{\neq}$ be the result of adding, for each $q \in \mathcal{Q}$ and distinct $x, y \in \mathit{vars}(q)$, the atom $x \neq y$ to q. In this way, homomorphisms from $\mathcal{Q}^{\neq}$ queries must necessarily be injective.

Claim 6. $\{\mathsf{MS}_q(\mathcal{D})\}_{q \in \mathcal{Q}^{\neq}}$ *is a partition of* $\{M \in \mathsf{MS}_{\widetilde{q}}(\mathcal{D}) : |M| = k\}$.

An *automorphism* of q is a homomorphism $h : q \xrightarrow{hom} q$; we denote by $\mathrm{Aut}(q)$ the set of all automorphisms of q.

Claim 7. $\#^{\mathsf{hom}}_q(\mathcal{D}) = |\mathrm{Aut}(q)| \cdot |\mathsf{MS}_q(\mathcal{D})|$ *for all* $q \in \mathcal{Q}^{\neq}$.

It is easy to see that all queries $q \in \mathcal{Q}^{\neq}$ are of quadratic size. Together with the preceding claims, this shows that $\mathcal{Q}^{\neq}$ and numbers $\gamma_q := 1/|\mathrm{Aut}(q)|$ have the required properties. □

Corollary 8. *Computing $\mathrm{SVC}^w_{\widetilde{q}}$ for a $\mathsf{UCQ}^{\neq}$ query $\widetilde{q}$ can be achieved by evaluating simple and short (quadratic) 'SELECT COUNT(*)' SQL queries in parallel.*

[4] A homomorphism $h : q \xrightarrow{hom} q'$ between $q, q' \in \mathsf{CQ}^{\neq}$ is defined in the expected way, *i.e.*, if $x \neq y \in q$, then $h(x) \neq h(y)$.

4.2 Intractability Results

Theorem 4 applies to OMQs which are $\mathsf{UCQ}^{\neq}$-rewritable (which is equivalent to being FO-rewritable for common DLs), and we conjecture, in line with (Bienvenu, Figueira, and Lafourcade 2025, Conjecture 5.7), that this is precisely the tractability frontier, i.e. if an OMQ Q does not admit a $\mathsf{UCQ}^{\neq}$-rewriting, then $\mathrm{SVC}^{\mathsf{ms}}_Q$ is $\#\mathsf{P}$-hard. We recall a related result for regular path queries (RPQs):

Theorem 9. *(Bienvenu, Figueira, and Lafourcade 2025, Theorem 5.6) Let w be a reversible tractable weight function, and a* regular path query *(RPQ) $q := \mathcal{L}(c, d)$ (i.e., q tests if there is a path from c to d conforming to a regular language $\mathcal{L}$). Then $\mathrm{SVC}^w_q \in \mathsf{FP}$ if $\mathcal{L}$ is finite or $\epsilon \in \mathcal{L}$ and $c = d$, and $\#\mathsf{P}$-hard otherwise.*

Exploiting the fact that reachability can be expressed using atomic queries in DLs such as $\mathcal{EL}$ that admit qualified existential restrictions, we get the following corollary:

Corollary 10. *Let w be a reversible tractable weight function, and $\mathcal{L}$ be any DL that can express the axiom $\exists r.A \sqsubseteq A$, where A is a concept name and r a role name. Then, there exists an OMQ $Q \in (\mathcal{L}, \mathsf{AQ})$ such that SVC^w_Q is $\#\mathsf{P}$-hard. Thus, $\mathrm{SVC}^w_{(\mathcal{L},\mathcal{Q})}$ is $\#\mathsf{P}$-hard in data complexity.*

5 Combined Complexity: Atomic OMQs

In light of the positive data complexity results for $\mathsf{UCQ}^{\neq}$-rewritable OMQs, such as those based upon DL-Lite ontologies, a natural question is whether we can achieve tractability even in combined complexity, i.e. when also taking into account the size of the OMQ. Naturally, this will only be possible if the considered class of OMQs admits PTime query evaluation. A natural candidate are atomic OMQs, i.e. OMQs $(\mathcal{T}, q)$ where $q \in \mathsf{AQ}$ (we shall consider restricted classes of (U)CQs in Sections 6 and 7). Note that due to Lemma 3, it suffices to consider $\textsc{eval}\text{-}\#^{\mathsf{ms}}_{(\mathcal{L},\mathsf{AQ})}$ and $\textsc{eval}\text{-}\#^{\mathsf{fms}}_{(\mathcal{L},\mathsf{AQ})}$ to obtain, respectively, lower and upper bounds on $\mathrm{SVC}^w_{(\mathcal{L},\mathsf{AQ})}$, for any reversible and tractable w.

5.1 DLs with Conjunction

We first show that WSMS computation is intractable in combined complexity for atomic OMQs whenever the considered DL allows for concept conjunction. This is the case for the lightweight DL $\mathcal{EL}$ and all of its extensions, but also for so-called Horn dialects of DL-Lite (which enjoy tractable data complexity due to Theorem 4).

Proposition 11. *Let $\mathcal{L}_\sqcap$ be the DL that only allows for axioms of the form $A \sqcap B \sqsubseteq C$, for $A, B, C \in \mathsf{N_C}$. Then $\textsc{eval}\text{-}\#^{\mathsf{ms}}_{(\mathcal{L}_\sqcap,\mathsf{AQ})}$ is $\#\mathsf{P}$-hard.*

Proof. We reduce from the problem #MINVERTEXCOVER which is to count, given an input graph $G = (V, E)$, the number of inclusion-minimal subsets $S \subseteq V$ such that every edge $e \in E$ has at least one endpoint in S. This problem has been shown to be $\#\mathsf{P}$-hard in (Valiant 1979, Problem 4). We prove that $\#\textsc{MinVertexCover} \leqslant_{\mathsf{P}} \textsc{eval}\text{-}\#^{\mathsf{ms}}_{(\mathcal{L}_\sqcap,\mathsf{AQ})}$.

Given a graph $G = (V, E)$, we define the instance:

$$\mathcal{T}_G := \{A_u \sqsubseteq B_{(u,v)}, A_v \sqsubseteq B_{(u,v)} \mid (u,v) \in E\} \cup \{\sqcap_{e \in E} B_e \sqsubseteq C\}$$
$$\mathcal{A}_G := \{A_u(c) \mid u \in V\} \qquad q_G := C(c)$$

For simplicity, $\mathcal{T}_G$ uses $\sqcap$ of arbitrary arity, but this can be simulated in a standard way (see Appendix B) of the full version). By construction, the minimal supports for $(\mathcal{T}_G, q_G)$ in $\mathcal{A}_G$ correspond to the minimal vertex covers of G. □

5.2 DL-Lite Dialects with Singleton Supports

Many of the more common dialects of DL-Lite, however, do not allow for conjunction and instead enjoy the following singleton-support property. A DL $\mathcal{L}$ *has singleton supports* if for every atomic OMQ $(\mathcal{T}, q) \in (\mathcal{L}, \mathsf{AQ})$ and every ABox $\mathcal{A}$ consistent with $\mathcal{T}$, all sets in $\mathsf{MS}_Q(\mathcal{A})$ are singletons.

Proposition 12. *Let $\mathcal{L}$ be any DL that has singleton supports and for which atomic OMQs admit* PTime *evaluation in combined complexity. Then* EVAL-$\#^{\mathsf{fms}}_{(\mathcal{L},\mathsf{AQ})}$ *is in* FP *for combined complexity. This holds in particular when $\mathcal{L} = $ DL-Lite$_\mathcal{R}$.*

Proof. Consider an OMQ $Q = (\mathcal{T}, q) \in (\mathcal{L}, \mathsf{AQ})$ and an ABox $\mathcal{A}$. Since $\mathcal{L}$ has singleton supports, we know that every minimal support for Q in $\mathcal{A}$ consists of a single assertion from $\mathcal{A}$. We can thus consider the linearly many such singleton sets and use the PTime evaluation procedure to check which ones entail the given atomic query. The fact that DL-Lite$_\mathcal{R}$ has singleton supports is folklore and implicit in existing OMQA algorithms. For example, standard rewriting algorithms will rewrite OMQs from (DL-Lite$_\mathcal{R}$, AQ) into unions of AQs, cf. (Calvanese et al. 2007). □

6 Digression: Unions of Conjunctive Queries

Our next goal will be to extend the tractability result for (DL-Lite$_\mathcal{R}$, AQ) to cover some suitable subclass of (DL-Lite$_\mathcal{R}$, CQ), where the user query is a non-atomic CQ. Since every $Q \in$ (DL-Lite$_\mathcal{R}$, CQ) can be rewritten into a UCQ q', one idea would be to identify conditions on OMQs that guarantee that the rewritten query belongs to some class $\mathcal{C}$ of UCQs for which EVAL-$\#^{\mathsf{fms}}_\mathcal{C}$ is known to be tractable.

Currently, however, it has only been shown that EVAL-$\#^{\mathsf{fms}}_\mathcal{C} \in$ FP for any class $\mathcal{C}$ of CQs having bounded *generalized hypertreewidth* and bounded '*self-join width*' (Bienvenu, Figueira, and Lafourcade 2025, Theorem 6.6).[5] This covers in particular CQs that are both *acyclic* (i.e. the undirected graph underlying the query is acyclic) and *self-join free* (i.e. no two atoms share the same predicate). No combined complexity results exist for UCQs, and it is not a priori clear if the preceding tractability result can be suitably extended to identify a 'nice' class of UCQs.

Our next result shows that positive results for well-behaved CQs do not in fact transfer to their unions. Indeed, EVAL-$\#^{\mathsf{ms}}$ is #P-hard already for the restricted class sjf-AUCQ of acyclic self-join free UCQs.

[5] Since we do not need the precise definitions, we direct the interested reader to (Bienvenu, Figueira, and Lafourcade 2025, §6.3).

Proposition 13. EVAL-$\#^{\mathsf{ms}}_{\text{sjf-AUCQ}}$ *is* #P*-hard under polynomial-time* 1-Turing reductions[6]*, even on binary signatures and in the absence of constants.*

Proof sketch. This is an adaptation of the #P-hardness proof of (Pichler and Skritek 2013, Theorem 6) for counting the number of answers of unions of acyclic full conjunctive queries (where 'full' means that queries have no existentially quantified variables). The adaptation must address some extra requirements, namely: (i) accounting for counting minimal supports instead of answers, (ii) ensuring that the queries are self-join free, and (iii) working on binary signatures instead of ternary. While this makes the reduction considerably more technical, the underlying idea remains that of Pichler and Skritek.

The reduction, from the perfect matching counting problem, builds a database $\mathcal{D}$ and constant-free queries $q_1, q_2 \in$ sjf-AUCQ such that the number of perfect matchings on G is equal to $\#^{\mathsf{ms}}_{q_1}(\mathcal{D}) - \#^{\mathsf{ms}}_{q_2}(\mathcal{D})$. In fact, q_1 is an (acyclic, self-join free) CQ rather than a UCQ, and its evaluation is in polynomial time by (Bienvenu, Figueira, and Lafourcade 2025, Remark 6.5 and paragraph after). Hence, this is a 1-Turing reduction. □

7 From Atomic to Conjunctive Queries

We return to the question of how to extend the tractability result for (DL-Lite$_\mathcal{R}$, AQ) to multiple atoms, covering suitable subclasses of (DL-Lite$_\mathcal{R}$, CQ). As seen in Section 6, we cannot simply rewrite the OMQ and appeal to tractability results for database queries. Instead, we shall introduce a class of well-behaved OMQs, inspired by the class of self-join free CQs. We establish tractability for such OMQs by characterizing their minimal supports in terms of the minimal supports of the atomic OMQs associated with their atoms.

To simplify the presentation, we assume throughout this section that the TBox is formulated in DL-Lite$_\mathcal{R}$, but our results also apply to other DL-Lite dialects satisfying the conditions of Proposition 12.

7.1 Interaction-Free OMQs

Recall that, by Lemma 3, it suffices to study EVAL-$\#^{\mathsf{fms}}$ to obtain the tractability of SVC$^{\mathsf{ms}}$ (and more generally, SVC^w, for well-behaved w). The method developed in (Bienvenu, Figueira, and Lafourcade 2025) for counting minimal supports of CQs essentially boils down to a reduction to the well-studied problem of counting the homomorphisms of the CQ into the database. The main issue is that, for arbitrary CQs, it is possible that several homomorphisms map to the same minimal support. Consider for instance the CQ $\exists xy.r(x,y) \wedge r(y,x)$ and its minimal support $\{r(c,d), r(d,c)\}$, which is the image of two homomorphisms: $(x,y) \mapsto (c,d)$ and $(x,y) \mapsto (d,c)$. Observe that such situations cannot arise for self-join free CQs, as each fact can only be used to satisfy a single atom of the query. As a consequence, counting minimal supports reduces to counting homomorphisms of the CQ into the database, which

[6] I.e., Turing reductions that only allow a single call to an oracle.

is tractable if there is a bound on the generalized hypertreewidth of the considered CQs (Pichler and Skritek 2013). Our aim will be to exhibit a class of OMQs which retains the desirable property that a single fact may not be used to satisfy multiple query atoms, thereby allowing us to characterize the minimal supports of such OMQs in terms of the minimal supports of the atomic OMQs of their atoms.

We now define our tractability criterion. Let anon be a special constant not in $\mathsf{N_I}$, and for every subset $C \subseteq \mathsf{N_I}$ denote by C° the set $C \cup \{\mathsf{anon}\}$. Given an ABox $\mathcal{A}$, TBox $\mathcal{T}$, CQ q and assignment $\mu : \mathit{vars}(q) \to \mathit{ind}(\mathcal{A})^\circ$, we write $(\mathcal{A}, \mathcal{T}) \models_\mu q$ if there exists a homomorphism $h : q \to \mathcal{I}_{\mathcal{A},\mathcal{T}}$ such that, for every $x \in \mathit{vars}(q)$, $h(x) = \mu(x)$ if $\mu(x) \in \mathit{ind}(\mathcal{A})$ and $h(x) \notin \mathit{ind}(\mathcal{A})$ if $\mu(x) = \mathsf{anon}$. Essentially, the variables assigned to anon map to anonymous (non-ABox) elements of the canonical model $\mathcal{I}_{\mathcal{A},\mathcal{T}}$.

An OMQ $(\mathcal{T}, q) \in (\text{DL-Lite}_\mathcal{R}, \mathsf{CQ})$ is *interaction-free* if, for every assertion f, atoms α, β of q and assignments $\mu_\alpha : \mathit{vars}(\alpha) \to \mathit{ind}(f)^\circ$, $\mu_\beta : \mathit{vars}(\beta) \to \mathit{ind}(f)^\circ$, we cannot have both $(\{f\}, \mathcal{T}) \models_{\mu_\alpha} \alpha$ and $(\{f\}, \mathcal{T}) \models_{\mu_\beta} \beta$, unless $(\alpha, \mu_\alpha) = (\beta, \mu_\beta)$. We denote by $\mathsf{itf}\text{-}(\text{DL-Lite}_\mathcal{R}, \mathsf{CQ})$ the class of all interaction-free OMQs in $(\text{DL-Lite}_\mathcal{R}, \mathsf{CQ})$.

We believe that $\mathsf{itf}\text{-}(\text{DL-Lite}_\mathcal{R}, \mathsf{CQ})$ is a practically relevant class of OMQs: 8 of the 14 queries from the well-known LUBM benchmark (Guo, Pan, and Heflin 2005) correspond to interaction-free OMQs and the remaining 6 are interaction-free after removing obviously redundant atoms. In the case where $\mathcal{T} = \emptyset$, the above condition can only be violated by two distinct atoms α, β and two assignments $\mu_\alpha : \mathit{vars}(\alpha) \to \mathsf{N_I}, \mu_\beta : \mathit{vars}(\beta) \to \mathsf{N_I}$ (with no anon in their image) such that $\mu_\alpha(\alpha) = \mu_\beta(\beta)$. As it turns out, this corresponds to saying that α and β are '*mergeable*' in the jargon of (Bienvenu, Figueira, and Lafourcade 2025, §6.3). The notion of interaction-free thus generalizes the notion of queries with no mergeable atoms (corresponding to self-join width 0) , which includes all self-join free CQs.

Example 14. *(a) The OMQ $(\emptyset, \exists x.r(c,x) \wedge r(d,x))$, with distinct $c, d \in \mathsf{N_I}$, is interaction-free despite its self-join, because no fact can satisfy both $r(c,x)$ and $r(d,x)$. (b) The OMQ $(\{\exists r \sqsubseteq A; \exists r^- \sqsubseteq A\}, \exists x.A(x))$ isn't interaction-free despite the query having a single atom, because the fact $r(c,d)$ satisfies it in two different ways ($x \mapsto c$ and $x \mapsto d$). (c) The OMQ $(\exists x, y.A(x) \wedge r(x,y), \{A \sqsubseteq \exists r\})$ isn't interaction-free either, because the fact $A(c)$ would satisfy both atoms thanks to the ontology.* △

7.2 Theorem Statement and Proof Idea

As previously mentioned, we aim to reduce the problem of counting the minimal supports of the input OMQ to counting the minimal supports of its component atomic OMQs. Formally, we prove:

Theorem 15. *Let $\mathcal{C}$ be a subclass of $\mathsf{itf}\text{-}(\text{DL-Lite}_\mathcal{R}, \mathsf{CQ})$ such that the set of queries $\{q \mid (\mathcal{T}, q) \in \mathcal{C}\}$ has bounded treewidth. Then $\text{EVAL-}\#^{\mathsf{fms}}_{\mathcal{C}} \leqslant_{\mathsf{P}} \text{EVAL-}\#^{\mathsf{fms}}_{(\text{DL-Lite}_\mathcal{R}, \mathsf{AQ})}$. Further, $\text{EVAL-}\#^{\mathsf{fms}}_{\mathcal{C}} \in \mathsf{FP}$.*

For the proof of Theorem 15, we focus on the reduction, since the tractability will readily follow from it and Proposition 12. Before diving into the details, we first give the following incomplete but informative formula:

$$\#^{\mathsf{ms}}_{(\mathcal{T},q)}(\mathcal{A}) \approx \sum_{\mu : \mathit{vars}(q) \to \mathit{ind}(\mathcal{A})} \prod_{\alpha \in q} \#^{\mathsf{ms}}_{(\mathcal{T},\mu(\alpha))}(\mathcal{A}) \quad (2)$$

Intuitively, this formula enumerates every possible assignment μ, computes the number of minimal supports associated with μ by multiplying the number of possibilities for each atom, then sums everything up. This formula does not give the correct result for all OMQs in $\mathsf{itf}\text{-}(\text{DL-Lite}_\mathcal{R}, \mathsf{CQ})$ (hence the $\approx$), for reasons that will be explained and addressed in Section 7.4, but it does work in many cases thanks to the following consequence of the absence of interactions.

Lemma 16. *Let $(\mathcal{T}, q) \in \mathsf{itf}\text{-}(\textit{DL-Lite}_\mathcal{R}, \mathsf{CQ})$, and $\mathcal{A}$ be an ABox. Then for every assignment $\mu : \mathit{vars}(q) \to \mathit{ind}(\mathcal{A})$:*

$$\#^{\mathsf{ms}}_{(\mathcal{T},\mu(q))}(\mathcal{A}) = \prod_{\alpha \in q} \#^{\mathsf{ms}}_{(\mathcal{T},\mu(\alpha))}(\mathcal{A})$$

Even when Equation (2) holds, it has two issues: (a) it does not directly yield a polynomial-time procedure as it sums over an exponential number of mappings μ (this will be addressed in Section 7.3); and (b) it computes the value of $\#^{\mathsf{ms}}$ while we actually need $\#^{\mathsf{fms}}$. However, (b) is not actually a problem: as we observed in the proof of Lemma 16, $\mathcal{T}$ has singleton supports and q is interaction-free, so the minimal supports for $(\mathcal{T}, q)$ all have the same size as q.

7.3 Efficient Summation Over Assignments

Observe that in Equation (2) we can ignore all assignments μ such that $(\mathcal{A}, \mathcal{T}) \not\models_\mu q$ since the summand is 0. In other words, Equation (2) is a sum over all homomorphisms $h : q \to \mathcal{I}_{\mathcal{A},\mathcal{T}}$ whose image is contained in $\mathit{ind}(\mathcal{A})$. As it turns out, efficient summation over homomorphisms has been studied in the context of databases annotated with semirings, or *weighted databases*. These are defined as $\mathcal{D} = (\mathcal{D}^\dagger, \omega)$, where $\mathcal{D}^\dagger$ is a database and $\omega : \mathcal{D}^\dagger \to \mathbb{N}$ assigns, to each fact, a 'weight'. The weight $q(\mathcal{D}) \in \mathbb{N}$ associated to the evaluation of a Boolean CQ $q = \exists \vec{x}. \bigwedge_{i=1}^k \alpha_i$ to a weighted database $\mathcal{D}$ is $q(\mathcal{D}) := \sum_{h : q \xrightarrow{hom} \mathcal{D}^\dagger} \prod_{i=1}^k \omega(h(\alpha_i))$. Equation (2) can thus be seen as q applied to the weighted database $\mathcal{D}^q_\mathcal{A} := (\mathcal{D}^\dagger, \{\beta \mapsto \#^{\mathsf{ms}}_{(\mathcal{T},\beta)}(\mathcal{A})\}_{\beta \in \mathcal{D}^\dagger})$ for $\mathcal{D}^\dagger = \{\mu(\alpha) \mid \alpha \in q, \mu : \mathit{vars}(\alpha) \to \mathit{ind}(\mathcal{A})\}$. Since each atom of q contains at most 2 variables, $\mathcal{D}^q_\mathcal{A}$ has at most $|q| \cdot |\mathit{ind}(\mathcal{A})|^2$ facts. For each fact $\beta \in \mathcal{D}^\dagger$, the weight $\#^{\mathsf{ms}}_{(\mathcal{T},\beta)}(\mathcal{A})$ can be computed by a call to the $\text{EVAL-}\#^{\mathsf{ms}}_{(\text{DL-Lite}_\mathcal{R}, \mathsf{AQ})}$ oracle. Overall, $\mathcal{D}^q_\mathcal{A}$ can be built in polynomial time with the oracle. The last step is to compute $q(\mathcal{D}^q_\mathcal{A})$, which can be done in polynomial time for any class of CQs with bounded treewidth.[7]

7.4 Variables Mapped Outside the ABox

As mentioned earlier, Equation (2) is inaccurate with respect to DL-Lite$_\mathcal{R}$, as evidenced by the following example.

[7] This is a trivial adaptation of the algorithm of (Flum and Grohe 2004, Proposition 3.5) for counting homomorphisms. It is also a basic case of the more general tractability results of (Khamis, Ngo, and Rudra 2016; Joglekar, Puttagunta, and Ré 2016).

Example 17. *Take* $\mathcal{T} = \{A \sqsubseteq \exists r; \exists r^- \sqsubseteq B\}$, $q = \exists x.B(x)$ *and* $\mathcal{A} = \{A(c)\}$. *Then the left-hand side of Equation* (2) *equals* 1 *because* $\mathcal{A}$ *is a minimal support for* $(\mathcal{T}, q)$, *but the right-hand side equals 0 because the only possible* μ *is* $x \mapsto c$, *but* $(\mathcal{A}, \mathcal{T}) \not\models B(c)$. △

The issue highlighted by Example 17 is that some minimal supports may only be witnessed by homomorphisms that map some variable to an anonymous element of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ rather than an ABox individual. Such minimal supports will thus be missed by Equation (2). However, by exploiting the structure of canonical models in DL-Lite$_\mathcal{R}$, we can show that the interaction-free condition ensures that all *shared variables* (i.e. variables $x \in \mathit{vars}(q)$ that appear in multiple atoms of q) are necessarily mapped to ABox individuals:

Lemma 18. *Let* $Q = (\mathcal{T}, q) \in \mathsf{itf}$-(*DL-Lite*$_\mathcal{R}$, CQ) *and* x *be a shared variable of* q. *Then, for every homomorphism* $h : q \xrightarrow{hom} \mathcal{I}_{\mathcal{A},\mathcal{T}}$, *we have* $h(x) \in \mathit{ind}(\mathcal{A})$.

Proof. Assume for a contradiction that $h : q \xrightarrow{hom} \mathcal{I}_{\mathcal{A},\mathcal{T}}$ and $h(x) \notin \mathit{ind}(\mathcal{A})$ for some shared variable x. From the definition of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$, we know that $h(x) = aR_1 \dots R_n$ for some $a \in \mathit{ind}(\mathcal{A})$ and roles R_i. For convenience we suppose $R_1 \in \mathsf{N_R}$, but the proof is analogous if R_1 is an inverse role. As x is shared, it must appear in at least two distinct atoms α_1 and α_2 of q. If $\alpha_i = A(x)$, then we must have $h(x) \in A^{\mathcal{I}_{\mathcal{A},\mathcal{T}}}$. If $\alpha_i = s(x, y)$ or $\alpha_i = s(y, x)$, then $h(y)$ must either be equal to the unique 'predecessor' of $h(x)$, which is $aR_1 \dots R_{n-1}$, or to some immediate 'successor' of $h(x)$, which must have the form $aR_1 \dots R_n R_{n+1}$ for some role R_{n+1}. This follows from the way roles are interpreted in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. Thus, all terms in α_1 and α_2 must be mapped by h either to a or to some anonymous element with prefix aR_1.

The existence of $aR_1 \dots R_n$ in the domain means there is some assertion $f \in \mathcal{A}$ such that $(\{f\}, \mathcal{T}) \models \exists R_1(a)$ and $(\{f\}, \mathcal{T}) \not\models R_1(a, b)$ for any individual b. It follows that $\mathcal{I}_{\{f\},\mathcal{T}}$ will also contain aR_1, and in fact all of the anonymous elements of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ having prefix aR_1, and it will interpret concept and role names on these elements in precisely the same way as in $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. Thus, h witnesses the satisfaction of α_1 and α_2 in $\mathcal{I}_{\{f\},\mathcal{T}}$. But this means that we can use h to define assignments $\mu_1 : \mathit{vars}(\alpha_1) \to \mathit{ind}(f)^\circ$, $\mu_2 : \mathit{vars}(\alpha_2) \to \mathit{ind}(f)^\circ$ such that both $(\{f\}, \mathcal{T}) \models {}_{\mu_1}\alpha_1$ and $(\{f\}, \mathcal{T}) \models {}_{\mu_2}\alpha_2$. This is impossible as $\alpha_1 \neq \alpha_2$ and $(\mathcal{T}, q)$ is interaction-free. □

We now have three kinds of query atoms to consider. First, those that contain no shared variables, which can be treated by a separate call to the EVAL-$\#^{\mathsf{ms}}_{(\text{DL-Lite}_\mathcal{R},\text{AQ})}$ oracle, due to the following easy lemma:

Lemma 19. *Let* $(\mathcal{T}, q) \in \mathsf{itf}$-(*DL-Lite*$_\mathcal{R}$, CQ) *with* $q = q_1 \wedge q_2$ *such that* $\mathit{vars}(q_1) \cap \mathit{vars}(q_2) = \emptyset$. *Then for any ABox* $\mathcal{A}$, $\#^{\mathsf{ms}}_{(\mathcal{T},q)}(\mathcal{A}) = \#^{\mathsf{ms}}_{(\mathcal{T},q_1)}(\mathcal{A}) \times \#^{\mathsf{ms}}_{(\mathcal{T},q_2)}(\mathcal{A})$.

Next, we have the atoms that contain only shared variables. By Lemma 18, their variables must be mapped to $\mathit{ind}(\mathcal{A})$, hence such atoms are already accounted for by the weighted database $\mathcal{D}^{(\mathcal{T},q)}_\mathcal{A}$ built in Section 7.3.

The third class of atoms are the role atoms which contain one shared variable and one unshared variable. These will be addressed by constructing a weighted database $\overline{\mathcal{D}}^{(\mathcal{T},q)}_\mathcal{A}$ that extends $\mathcal{D}^{(\mathcal{T},q)}_\mathcal{A}$ with some extra facts. Consider one such atom $\alpha = R(x, y) \in q$, with x and y being shared and unshared respectively, and $R \in \mathsf{N}^\pm_R$ (since the unshared variable could come first). Again by Lemma 18, we know that x must necessarily be instantiated by an individual $c \in \mathit{ind}(\mathcal{A})$, but y might be mapped to an anonymous element. We thus add to $\overline{\mathcal{D}}^{(\mathcal{T},q)}_\mathcal{A}$ a fact $R(c, c_\alpha)$, with c_α a fresh individual, for every individual $c \in \mathit{ind}(\mathcal{A})$ such that, for some $f \in \mathcal{A}$, $(\{f\}, \mathcal{T}) \models \exists R(c)$ but $(\{f\}, \mathcal{T}) \not\models R(c, d)$ for all $d \in \mathcal{A}$, and set the weight of this $R(c, c_\alpha)$ to be the number of such $\{f\}$.

Lemma 20. *Let* $(\mathcal{T}, q) \in \mathsf{itf}$-(*DL-Lite*$_\mathcal{R}$, CQ) *such that* q *is connected and* $|q| \geqslant 2$. *Then* $q\left(\overline{\mathcal{D}}^{(\mathcal{T},q)}_\mathcal{A}\right) = \#^{\mathsf{ms}}_{(\mathcal{T},q)}(\mathcal{A})$.

7.5 Putting Everything Together

The construction has been presented in a progressive manner for ease of understanding. We now recapitulate the argument in a more direct fashion.

Proof of Theorem 15. The algorithm goes as follows. For every connected component q_c of q with at least 2 atoms we: (1) build the weighted database $\mathcal{D}^{q_c}_\mathcal{A}$ described in Section 7.3 using the EVAL-$\#^{\mathsf{ms}}_{(\text{DL-Lite}_\mathcal{R},\text{AQ})}$ oracle; (2) extend $\mathcal{D}^{q_c}_\mathcal{A}$ into $\overline{\mathcal{D}}^{q_c}_\mathcal{A}$ as described in Section 7.4; (3) compute $q_c(\overline{\mathcal{D}}^{q_c}_\mathcal{A})$ using standard weighted database algorithms.

By Lemma 20, this yields the value of $\#^{\mathsf{ms}}_{(\mathcal{T},q_c)}(\mathcal{A})$. The remaining connected components consist in a single atom, so the corresponding value can then be obtained by a direct call to the EVAL-$\#^{\mathsf{ms}}_{(\text{DL-Lite}_\mathcal{R},\text{AQ})}$ oracle. Once all the values are obtained, we finally multiply them all together, which yields the desired $\#^{\mathsf{ms}}_{(\mathcal{T},q)}(\mathcal{A})$ by Lemma 19.

Regarding the consequence that EVAL-$\#^{\mathsf{fms}}_\mathcal{C} \in \mathsf{FP}$, this is a direct application of Proposition 12 to the above. □

8 Conclusion and Future Work

Our work explores the recently introduced class of Shapley-based responsibility measures, known as WSMS, in the context of ontology-mediated query answering. Our complexity analysis pinpoints sources of intractability but also identifies relevant classes of OMQs for which WSMS computation is tractable in data (and sometimes also combined) complexity and can moreover be computed using standard database systems. It would be interesting in future work to test out the approach in practice and try to generalize the 'interaction-free' condition to identify further tractable cases.

While we focused on DLs, many results extend to other ontology formalisms such as existential rules. In particular, the data tractability result extends to UCQ-rewritable rule-sets, and the tractability result for atomic queries extends to bounded-arity linear existential rules because they satisfy the conditions of Proposition 12. An interesting future step would be to see if a useful notion of 'interaction-free' could be defined in order to obtain tractability in combined complexity for linear existential rules with CQs.

Acknowledgements

This work was partially supported by ANR AI Chair INTENDED (ANR-19-CHIA-0014).

References

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.

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.; Hansen, P.; Lutz, C.; and Wolter, F. 2016. First order-rewritability and containment of conjunctive queries in Horn description logics. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 965–971.

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)):105.

Bienvenu, M.; Figueira, D.; and Lafourcade, P. 2025. Shapley revisited: Tractable responsibility measures for query answers. *Proc. ACM Manag. Data* 3(2):112:1–112:26.

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)*.

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.

Deutch, D.; Frost, N.; Kimelfeld, B.; and Monet, M. 2022. Computing the Shapley value of facts in query answering. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 1570–1583. ACM.

Escoffier, A. 1903. *Le guide culinaire : Aide-mémoire de cuisine pratique*. Bibliothèque Professionnelle.

Flum, J., and Grohe, M. 2004. The parameterized complexity of counting problems. *SIAM Journal on Computing* 33(4):892–922.

Guo, Y.; Pan, Z.; and Heflin, J. 2005. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics* 3(2-3):158–182.

Hansen, P., and Lutz, C. 2017. Computing FO-rewritings in *EL* in practice: From atomic to conjunctive queries. In *Proceedings of the 16th International Semantic Web Conference (ISWC)*, 347–363.

Joglekar, M. R.; Puttagunta, R.; and Ré, C. 2016. AJAR: aggregations and joins over annotated relations. In *ACM Symposium on Principles of Database Systems (PODS)*, 91–106. ACM Press.

Kara, A.; Olteanu, D.; and Suciu, D. 2024. From Shapley value to model counting and back. *Proceedings of the ACM on Management of Data (PACMMOD)* 2(2 (PODS)):79.

Karmakar, P.; Monet, M.; Senellart, P.; and Bressan, S. 2024. Expected Shapley-like scores of Boolean functions: Complexity and applications to probabilistic databases. *Proc. ACM Manag. Data* 2(2):92:1–92:26.

Khalil, M., and Kimelfeld, B. The complexity of the Shapley value for regular path queries. In *International Conference on Database Theory (ICDT)*, volume 255 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 11:1–11:19.

Khamis, M. A.; Ngo, H. Q.; and Rudra, A. 2016. FAQ: Questions asked frequently. In *ACM Symposium on Principles of Database Systems (PODS)*, 13–28. ACM Press.

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

Pichler, R., and Skritek, S. 2013. Tractable counting of the answers to conjunctive queries. *Journal of Computer and System Sciences (JCSS)* 79(6):984–1001.

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.

Reshef, A.; Kimelfeld, B.; and Livshits, E. 2020. The impact of negation on the complexity of the Shapley value in conjunctive queries. In *ACM Symposium on Principles of Database Systems (PODS)*, 285–297. ACM.

Shapley, L. S. 1953. A value for n-person games. In *Contributions to the Theory of Games II*. Princeton University Press. 307–317.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *SIAM Journal on Computing* 8(3):410–421.

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.