Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Shapley Value Computation in Ontology-Mediated Query Answering

Meghyn Bienvenu[1,2] , **Diego Figueira**[1] , **Pierre Lafourcade**[1]
[1] Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France
[2] Japanese-French Laboratory for Informatics, CNRS, NII, IRL 2537, Tokyo, Japan
{meghyn.bienvenu, diego.figueira, pierre.lafourcade}@u-bordeaux.fr

Abstract

The Shapley value, originally introduced in cooperative game theory for wealth distribution, has found use in KR and databases for the purpose of assigning scores to formulas and database tuples based upon their contribution to obtaining a query result or inconsistency. In the present paper, we explore the use of Shapley values in ontology-mediated query answering (OMQA) and present a detailed complexity analysis of Shapley value computation (SVC) in the OMQA setting. In particular, we establish a FP/#P-hard dichotomy for SVC for ontology-mediated queries $(\mathcal{T}, q)$ composed of an ontology $\mathcal{T}$ formulated in the description logic $\mathcal{ELHI}_\bot$ and a connected constant-free homomorphism-closed query q. We further show that the #P-hardness side of the dichotomy can be strengthened to cover possibly disconnected queries with constants. Our results exploit recently discovered connections between SVC and probabilistic query evaluation and allow us to generalize existing results on probabilistic OMQA.

☞ This pdf contains internal links: clicking on a notion leads to its *definition*.[1]

1 Introduction

The Shapley value was originally proposed in the context of cooperative game theory as a method for fairly distributing the wealth of a coalition of players based upon their respective contributions. It has appealing theoretical properties, having been shown to be the unique wealth distribution measure that satisfies a set of desirable axioms. Since its proposal in (Shapley 1953), it has found application in numerous domains, including various areas of computer science. In artificial intelligence, the Shapley value has been utilized for defining inconsistency measures of propositional (Grant and Hunter 2006; Hunter and Konieczny 2010) and description logic knowledge bases (Deng, Haarslev, and Shiri 2007), and more recently for defining explanations of machine learning models (Lundberg and Lee 2017). The Shapley value has also gained attention in the database area (Bertossi et al. 2023), where it has been employed both for defining inconsistency values of databases (Livshits and Kimelfeld 2022) and also for providing quantitative explanations of query answers (Livshits et al. 2021). While other quantitative measures, such as *causal responsibility* and the *Banzhaf power index* (aka *causal effect*), have also been considered for databases, the Shapley value has thus far garnered the most attention. We direct readers to (Livshits et al. 2021; Abramovich et al. 2024) for more details on alternative measures and how they relate to the Shapley value.

In general, Shapley value computation is known to be computationally challenging, being #P-hard in data complexity for common classes of queries, such as conjunctive queries. This has motivated non-uniform complexity studies aimed at pinpointing which queries admit tractable Shapley value computation (Reshef, Kimelfeld, and Livshits 2020; Khalil and Kimelfeld 2023), in particular, by establishing fruitful connections with probabilistic query evaluation and variants of model counting (Deutch et al. 2022; Kara, Olteanu, and Suciu 2024; Bienvenu, Figueira, and Lafourcade 2024b).

In the present paper, we revisit the use of the Shapley value in the ontology setting, building upon these recent advances in the database area. We shall mostly focus on how the Shapley value can be employed for explaining answers in the context of ontology-mediated query answering (OMQA). We recall that the OMQA is used to improve access to incomplete and possibly heterogeneous data through the addition of ontology layer, which provides a user-friendly vocabulary for query formulation as well as domain knowledge that is taken into account when computing the query answers. Over the past fifteen years, OMQA has grown into a vibrant research topic within both the KR and database communities (Poggi et al. 2008; Calì et al. 2011; Mugnier and Thomazo 2014; Bienvenu and Ortiz 2015; Xiao et al. 2018). With the increasing maturity and deployment of OMQA techniques, there is an acknowledged need to help users understand the query results. Various notions of explanations with different levels of detail can be considered for OMQA, ranging from providing proofs of how an answer can be derived (Borgida, Calvanese, and Rodriguez-Muro 2008; Alrabbaa et al. 2022) to generating minimal subsets of the KB that suffice to obtain the answer or identifying the assertions and/or axioms that are relevant in the sense that they belong to such a minimal subset (Bienvenu, Bourgaux, and Goasdoué 2019; Ceylan et al. 2019; Ceylan et al. 2020). The Shapley value offers a more nuanced, quantitative version of the latter approach, by assigning the relevant assertions and axioms scores based upon

[1] https://ctan.org/pkg/knowledge

their level of responsibility or importance in obtaining the considered query answer (or entailment).

For our study of Shapley value computation, we will work with description logic (DL) knowledge bases (KB), consisting of an ABox (dataset) and TBox (ontology). We introduce some natural ways of defining the Shapley value computation (SVC) problem in the DL setting, by varying what is to be explained (entailment of a TBox axiom, ABox assertion, or query answer), which parts of the KB are assigned values, and how the complexity is measured. To begin our study, we establish the #P-hardness of the Shapley value computation of a simple graph reachability query, which we then employ to show #P-hardness of several variants of the SVC problem, even for inexpressive DLs and atomic queries. In light of these initial negative results, we embark on a non-uniform complexity analysis, whose aim is to classify the data complexity of the Shapley value computation problems SVC_Q associated with each ontology-mediated query (OMQ) $Q = (\mathcal{T}, q)$. By transferring recent results from the database setting, we establish a FP/#P-hard dichotomy result of Shapley value computation problem SVC_Q for OMQs $Q = (\mathcal{T}, q)$ where the TBox $\mathcal{T}$ is formulated in the Horn DL $\mathcal{ELHI}_\bot$ and q is a constant-free connected homomorphism-closed query. Moreover, if restricted to the case where q is a conjunctive query, then the dichotomy is *effective*, i.e. we can decide whether SVC_Q is FP or #P-hard. Our final and most technically challenging result is to show that the #P-hardness part of the dichotomy can be strengthened to cover OMQs based upon a wider range of queries q. Specifically, we show that for any OMQ $Q = (\mathcal{T}, q)$ based upon a $\mathcal{ELHI}_\bot$ TBox and a UCQ q (which may be disconnected and/or contain constants), non-FO-rewritability of Q implies #P-hardness of SVC_Q. Due to the tight connections holding between Shapley value computation and probabilistic query evaluation, the proof of this result can be further used to obtain a FP/#P-hard dichotomy for probabilistic ontology-mediated queries from $(\mathcal{ELHI}_\bot, \mathrm{UCQ})$, substantially generalizing existing results.

The paper is structured as follows. Section 2 introduces key notions from databases and description logics, and Section 3 defines Shapley values and recalls useful results about Shapley values in databases. We also prove a new hardness result for graph reachability queries, which we apply in Section 4 to show hardness of Shapley value computation in various ontology settings. In Section 5, we present our FP/#P-hard dichotomy result for OMQs in the Horn DL $\mathcal{ELHI}_\bot$, and in Section 6, we strengthen the #P-hardness result to cover a wider range of queries. We conclude the paper in Section 7 with a summary of our contributions and a discussion of future work. Missing proof details can be found in the *full version* (Bienvenu, Figueira, and Lafourcade 2024a).

2 Preliminaries

We recall some important notions related to description logics (DLs), databases, queries, and complexity, directing readers to (Baader et al. 2017) for a detailed introduction to DLs. Our presentation of DLs and databases slightly differs from the 'usual' ones so as that we may employ some definitions and notations in both settings.

Databases A *database* $\mathcal{D}$ is a finite set of relational facts $P(\vec{a})$, where P is a k-ary symbol drawn from a countably infinite set of relation symbols $\mathsf{N_D}$ and $\vec{a}$ is a k-ary tuple of (individual) *constants* drawn from a countably infinite set $\mathsf{N_I}$. We shall also consider *extended databases* which may contain infinitely many facts $P(\vec{a})$, and where the elements of $\vec{a}$ are drawn from $\mathsf{N_I}$ and from a countably infinite set $\mathsf{N_U}$ of unnamed elements. The *domain* $dom(\mathcal{D})$ of $\mathcal{D}$ contains all constants and unnamed elements occurring in $\mathcal{D}$, and we use $\mathit{const}(\mathcal{D})$ for the constants in $\mathcal{D}$. When $\mathcal{D}$ is a database, $dom(D) = \mathit{const}(\mathcal{D})$.

A *homomorphism* from an extended database $\mathcal{D}$ to an extended database D' is a function $h : dom(\mathcal{D}) \to dom(\mathcal{D}')$ such that $P(h(\vec{a})) \in D'$ for every $P(\vec{a}) \in \mathcal{D}$. We write $\mathcal{D} \xrightarrow{hom} \mathcal{D}'$ to indicate the existence of such an h. If additionally $h(c) = c$ for every $c \in C \cap \mathit{const}(\mathcal{D})$, with $C \subseteq \mathsf{N_I}$, then we call h a C-homomorphism and write $\mathcal{D} \xrightarrow{C\text{-}hom} \mathcal{D}'$.

We say that a (possibly extended) database $\mathcal{D}$ is *connected* if so is the underlying undirected graph with vertices $dom(\mathcal{D}) \cup \mathcal{D}$ and edges $\{(a_i, P(\vec{a})) \mid P(\vec{a}) \in \mathcal{D}\}$. The *connected components* of $\mathcal{D}$ are the maximal subsets of $\mathcal{D}$ that are connected in the underlying graph.

Queries In the most general sense, a k*-ary query* ($k \geq 0$) can be defined as a function q that maps every extended database $\mathcal{D}$ to a set of k-tuples of constants from $\mathit{const}(\mathcal{D})$ (the *answers* to q). Queries of arity 0 are called *Boolean*. When q is a Boolean query, each $\mathcal{D}$ is mapped either to $\{()\}$ or $\{\}$. In the former case, we say that $\mathcal{D}$ *satisfies* q and write $\mathcal{D} \models q$. If additionally $\mathcal{D}' \not\models q$ for every $\mathcal{D}' \subsetneq \mathcal{D}$, then we shall call $\mathcal{D}$ a *minimal support for* q.

A Boolean query q is said to be *closed under homomorphisms*, or *hom-closed*, if $\mathcal{D} \models q$ and $\mathcal{D} \xrightarrow{hom} \mathcal{D}'$ implies that $\mathcal{D}' \models q$. The notion of C*-hom-closed* is defined analogously using $\xrightarrow{C\text{-}hom}$ instead of $\xrightarrow{hom}$. When q is (C-)hom-closed, $\mathcal{D} \models q$ iff $\mathcal{D}$ contains some minimal support for q; we say that q is *connected* if all its minimal supports are connected.

So far we have considered an abstract notion of query, but in practice, queries are often specified in concrete query languages. First-order (FO) queries are given by formulas in first-order predicate logic with equality, whose relational atoms are built from predicates from $\mathsf{N_D}$ and terms drawn from $\mathsf{N_I} \cup \mathsf{N_V}$, with $\mathsf{N_V}$ a countably infinite set of variables, equipped with standard FO logic semantics (i.e. $\mathcal{D} \models q$ if $\mathcal{D}$, viewed as a first-order structure, satisfies the FO sentence q). Two prominent classes of FO queries are *conjunctive queries* (*CQs*), and *unions of conjunctive queries* (*UCQs*) which are finite disjunctions of CQs having the same free variables. We remark that Boolean (U)CQs without constants are hom-closed, and Boolean (U)CQs with constants in C are C-hom-closed. Other examples of (C)-hom-closed queries include Datalog queries and regular path queries.

DL Knowledge Bases A DL *knowledge base* (*KB*) $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ comprises an ABox (dataset) $\mathcal{A}$ and a TBox (ontology) $\mathcal{T}$, which are built from countably infinite sets $\mathsf{N_C}$ of *concept names* (unary predicates) and $\mathsf{N_R}$ of *role names* (binary predicates) with $\mathsf{N_C} \cup \mathsf{N_R} \subseteq \mathsf{N_D}$, and the individual constants from $\mathsf{N_I}$. An *ABox* is a database with relations drawn from $\mathsf{N_C} \cup \mathsf{N_R}$ and thus contains two kinds of facts: *concept assertions* $A(c)$ ($A \in \mathsf{N_C}, c \in \mathsf{N_I}$) and *role assertions* $r(c, d)$ ($r \in \mathsf{N_R}, c, d \in \mathsf{N_I}$). A *TBox* is a finite set of *axioms*, whose form is dictated by the *DL* in question. We use $\mathcal{L}$ *TBox* to refer to a TBox formulated in the DL $\mathcal{L}$. For example, in the DL $\mathcal{ELHI}_\bot$ considered later in this paper, *complex concepts* are constructed as follows:

$$C := \top \mid A \mid C \sqcap C' \mid \exists R.C \qquad A \in \mathsf{N_C}, R \in \mathsf{N_R^\pm}$$

where $\mathsf{N_R^\pm} = \mathsf{N_R} \cup \{r^- \mid r \in \mathsf{N_R}\}$, and $\mathcal{ELHI}_\bot$ TBoxes consist of *concept inclusions* $C \sqsubseteq D$ (with C, D complex concepts) and *role inclusions* $R \sqsubseteq S$ with $R, S \in \mathsf{N_R^\pm}$. We shall also consider DL-Lite$_{\mathsf{core}}$ TBoxes, which are composed of concept inclusions of the form

$$B_1 \sqsubseteq (\neg)B_2 \qquad B_i := A \mid \exists R.\top \quad A \in \mathsf{N_C}, R \in \mathsf{N_R^\pm}$$

The semantics of DL KBs is defined using *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I} \subseteq \mathsf{N_I} \cup \mathsf{N_U}$ is a non-empty set and $\cdot^\mathcal{I}$ a function[2] that maps every $A \in \mathsf{N_C}$ to a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and every $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is straightforwardly extended to interpret complex concepts and roles: $\top^\mathcal{I} = \Delta^\mathcal{I}$, $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$, $(\exists R.C)^\mathcal{I} = \{d \mid \exists e \in C^\mathcal{I} \text{ s.t. } (d, e) \in r^\mathcal{I}\}$, $(\neg G)^\mathcal{I} = \Delta^\mathcal{I} \setminus G^\mathcal{I}$, and $(r^-)^\mathcal{I} = \{(e, d) \mid (d, e) \in r^\mathcal{I}\}$. Note that by requiring that $\Delta^\mathcal{I} \subseteq \mathsf{N_I} \cup \mathsf{N_U}$, we ensure that every interpretation $\mathcal{I}$ can be viewed as an extended database $\mathcal{D}_\mathcal{I} = \{A(e) \mid e \in A^\mathcal{I}\} \cup \{r(d, e) \mid (d, e) \in r^\mathcal{I}\}$, and we shall view $\mathcal{I}$ as an extended database when convenient.

An interpretation $\mathcal{I}$ *satisfies a (concept or role) inclusion* $G \sqsubseteq H$ if $G^\mathcal{I} \subseteq H^\mathcal{I}$, and it *satisfies an assertion* $A(c)$ (resp. $r(c, d)$) if $c \in A^\mathcal{I}$ (resp. $(c, d) \in r^\mathcal{I}$). We call $\mathcal{I}$ a *model of a TBox* $\mathcal{T}$ if it satisfies every axiom in $\mathcal{T}$, a *model of an ABox* $\mathcal{A}$ if it satisfies every assertion in $\mathcal{A}$, and a *model of a KB* $(\mathcal{A}, \mathcal{T})$ if it is a model of both $\mathcal{T}$ and $\mathcal{A}$. We use $\mathsf{Mod}(\mathcal{K})$ for the set of models of a KB $\mathcal{K}$. A KB $\mathcal{K}$ is *consistent* if $\mathsf{Mod}(\mathcal{K}) \neq \emptyset$ (else it is *inconsistent*). An ABox $\mathcal{A}$ is $\mathcal{T}$-consistent when the KB $(\mathcal{A}, \mathcal{T})$ is consistent. An axiom α is *entailed from a TBox* $\mathcal{T}$, written $\mathcal{T} \models \alpha$, if every model of $\mathcal{T}$ satisfies α, and an axiom or assertion α is *entailed from a KB* $\mathcal{K}$, written $\mathcal{K} \models \alpha$, if every model of $\mathcal{K}$ satisfies α.

Querying DL KBs We say that a Boolean query q is entailed from a DL KB $\mathcal{K}$, written $\mathcal{K} \models q$, if $\mathcal{D}_\mathcal{I} \models q$ 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 k-tuples $\vec{a}$ of constants from *const*$(\mathcal{A})$ such that $\mathcal{K} \models q(\vec{a})$, with $q(\vec{a})$ the Boolean query obtained by substituting $\vec{a}$ for the free variables $\vec{x}$. Note that when the KB is inconsistent, every Boolean query is trivially entailed, so every possible tuple $\vec{a}$ of ABox constants counts as a certain answer.

While it is traditional to view queries as being posed to the KB, it is sometimes more convenient to adopt a database perspective and treat $\mathcal{T}$ and q together as constituting a composite *ontology-mediated query* (*OMQ*) $Q = (\mathcal{T}, q)$, which is posed to the ABox $\mathcal{A}$. When we adopt this perspective, we will write $\mathcal{A} \models (\mathcal{T}, q)$ or $\mathcal{A} \models Q$ to mean $(\mathcal{A}, \mathcal{T}) \models q$. When convenient, we will use the notation $(\mathcal{L}, \mathcal{Q})$ to designate the class of all OMQs $(\mathcal{T}, q)$ such that $\mathcal{T}$ is formulated in the DL $\mathcal{L}$ and q is a query from the class of queries $\mathcal{Q}$.

A prominent technique for 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 modify the above definition to only quantify over $\mathcal{T}$-consistent ABoxes, then we speak instead of a *rewriting w.r.t. consistent ABoxes*. When q^* is a first-order query, we call it a *first-order (FO) rewriting*. If an OMQ Q possesses an FO-rewriting, we say that Q is *FO-rewritable*, else it is called *non-FO-rewritable*.

In Horn DLs, like $\mathcal{ELHI}_\bot$ and DL-Lite$_{\mathsf{core}}$, every consistent KB $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ admits a *canonical model* $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ with $\Delta^{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \subseteq$ *const*$(\mathcal{A}) \cup \mathsf{N_U}$ with the special property that it embeds homomorphically into every model of $\mathcal{K}$. More precisely, $\mathcal{D}_{\mathcal{I}_{\mathcal{A},\mathcal{T}}} \xrightarrow{C\text{-hom}} \mathcal{D}_\mathcal{I}$ for every $\mathcal{I} \in \mathsf{Mod}(\mathcal{K})$ and every $C \subseteq \mathsf{N_I}$. While the definition of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ depends on the particular Horn DL, the construction typically involves completing the ABox by adding tree-shaped structures using unnamed elements to satisfy the TBox axioms in the least constrained way possible. Importantly, if $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ admits a canonical model $\mathcal{I}_{\mathcal{A},\mathcal{T}}$, then for every C-hom-closed query q (with $C \subseteq \mathsf{N_I}$):

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

In particular, this holds when q is a (U)CQ.

Complexity We assume familiarity with FP, the set of functions solved in polynomial-time; and #P, the functions which output the number of accepting runs in polynomial-time nondeterministic Turing machines. We will work with *polynomial-time Turing reductions* between computational tasks, and we write $P_1 \equiv_{\mathsf{P}} P_2$ to denote that there are polynomial-time algorithms to compute P_i using unit-cost calls to P_{3-i}, for both $i \in \{1, 2\}$.

3 Shapley Value: Definition & Basic Results

In this section, we formally define the Shapley value, recall relevant existing results, and prove a new intractability result for computing Shapley values in reachability games.

3.1 Definition of Shapley Value

The Shapley value (Shapley 1953) was introduced as a means to fairly distribute wealth amongst players in a a cooperative game, based upon their respective contributions.

A *cooperative game* consists of a finite set of players P and a *wealth function* $\mathbf{v} : \wp(P) \to \mathbb{Q}$ that assigns a value to each coalition (*i.e.*, set) of players, with $\mathbf{v}(\emptyset) = 0$. Picture a scenario where the players arrive one by one in a random order, and each one earns what she added to the current

[2] To simplify the comparison with the database setting, we make the *standard names assumption*, interpreting constants as themselves, but our results also hold under the weaker *unique names assumption*. Moreover, to allow for finite interpretation domains, we do not require all constants to be interpreted.

coalition's wealth on arrival. The *Shapley value* of a player $p \in P$ is defined as her expected earnings in this scenario, which can be expressed as:

$$\mathrm{Sh}(P, \mathbf{v}, p) := \frac{1}{|P|!} \sum_{\sigma \in \mathfrak{S}(P)} (\mathbf{v}(\sigma_{\leqslant p}) - \mathbf{v}(\sigma_{<p})) \quad (1)$$

where $\mathfrak{S}(P)$ denotes the set of permutations of P and $\sigma_{<p}$ (resp. $\sigma_{\leqslant p}$) the set of players that appear before p (resp. before or at p) in the permutation σ. Intuitively, we take the average marginal contribution $\mathbf{v}(\sigma_{\leqslant p}) - \mathbf{v}(\sigma_{<p})$ of p, across all possible orderings σ of the players. Eq. (1) can be equivalently reformulated as:

$$\sum_{B \subseteq P \setminus \{p\}} \frac{|B|!(|P| - |B| - 1)!}{|P|!} (\mathbf{v}(B \cup \{p\}) - \mathbf{v}(B)) \quad (2)$$

which will be more convenient in our proofs.

3.2 Existing Results from the Database Setting

There has been significant interest lately in the problem of computing the Shapley value of database facts as a means of quantifying their contributions to a query answer. The formal setting is as follows: the database $\mathcal{D}$ is *partitioned* into *endogenous* and *exogenous facts*, $\mathcal{D} = \mathcal{D}_\mathsf{n} \uplus \mathcal{D}_\mathsf{x}$,[3] the players of the cooperative game are the endogenous facts $\mathcal{D}_\mathsf{n}$, and the wealth function of a given Boolean query q is defined, for every subset $B \subseteq \mathcal{D}_\mathsf{n}$ of endogenous facts, as $\mathbf{v}_q(B) = v_B - v_\mathsf{x}$ where $v_B = 1$ (resp. $v_\mathsf{x} = 1$) if $\mathcal{D}_\mathsf{x} \cup B \models q$ (resp. if $\mathcal{D}_\mathsf{x} \models q$), and 0 otherwise. Shapley value computation on q, denoted SVC_q, is the problem of computing the Shapley value $\mathrm{Sh}(\mathcal{D}_\mathsf{n}, \mathbf{v}_q, \alpha)$ for the input partitioned database $\mathcal{D}_\mathsf{n} \uplus \mathcal{D}_\mathsf{x}$ and fact $\alpha \in \mathcal{D}_\mathsf{n}$. We will write $\mathrm{SVC}^\mathsf{n}_q$ to refer to the task when restricted to *purely endogenous* databases, *i.e.*, partitioned databases with only endogenous assertions (of the form $\mathcal{D} = (\mathcal{D}_\mathsf{n}, \mathcal{D}_\mathsf{x})$ with $\mathcal{D}_\mathsf{x} = \emptyset$).

Probabilistic Query Evaluation We will exploit known connections between SVC and probabilistic query evaluation. A *tuple-independent probabilistic database* is a pair $\mathcal{D} = (S, \pi)$ where S is a database and $\pi : S \to (0, 1]$ is a probability assignment. For a Boolean query q, $\Pr(\mathcal{D} \models q)$ is the probability of q being true, where each assertion α has independent probability $\pi(\alpha)$ of being in the database. The problem of computing, given a tuple-independent probabilistic database $\mathcal{D}$, the probability $\Pr(\mathcal{D} \models q)$ is known as the *probabilistic query evaluation* problem, or PQE_q.

We consider three restrictions of PQE_q, by limiting the probabilities that appear in the image $Im(\pi)$ of the probability assignment of the input probabilistic database:

- $\mathrm{PQE}_q(1/2)$: input (S, π) is such that $Im(\pi) = \{1/2\}$;
- $\mathrm{PQE}_q(1/2; 1)$: input (S, π) is such that $Im(\pi) = \{1/2, 1\}$;
- *single proper probability query evaluation* (SPPQE_q): input (S, π) is s.t. $Im(\pi) = \{p, 1\}$ for some $p \in (0, 1]$.

[3] By $A \uplus B$ we denote the union $A \cup B$ of two disjoint sets A, B.

These restricted versions of PQE can be also found in the literature under the names of their counting problem counterparts: $\mathrm{PQE}(1/2)$ is also known as the "model counting" or "uniform reliability" problems, and $\mathrm{PQE}(1/2; 1)$ as the "generalized model counting" problem.

Known results In (Livshits et al. 2021), a FP/#P-hard dichotomy was established for *self-join-free* CQs (i.e. not having two atoms with the same relation name). The dichotomy coincides with the FP/#P-hard dichotomy for PQE (Dalvi and Suciu 2004), and the tractable queries admit a syntactical characterization, known as *hierarchical* queries. In fact, the PQE dichotomy extends to the more general class of UCQs (Dalvi and Suciu 2012), where the queries for which PQE is tractable are known as *safe* UCQs (hence, in particular hierarchical CQs are safe). However, it is an open problem whether UCQs (or even CQs with self-joins) also enjoy a dichotomy for SVC. Concretely, it is unknown if SVC is #P-hard for all unsafe UCQs (or even CQs).

Recent work has clarified the relation between the two dichotomies by reducing SVC to PQE (Deutch et al. 2022) and reproving the hardness of SVC (Kara, Olteanu, and Suciu 2024) by reduction from the same model counting problem for *Boolean functions* that had been used to show hardness of PQE for non-hierarchical self-join-free CQs (Dalvi and Suciu 2004). Further, SPPQE and SVC have been shown to be polynomial-time inter-reducible for many fragments of hom-closed queries (Bienvenu, Figueira, and Lafourcade 2024b), in particular for connected queries without constants.

Theorem 1. *(Bienvenu, Figueira, and Lafourcade 2024b, Corollaries 4.1 and 4.2) For every connected hom-closed Boolean query q, $\mathrm{SPPQE}_q \equiv_\mathsf{P} \mathrm{SVC}_q$; further, on graph databases, SVC_q is in FP if q is equivalent to a safe UCQ and #P-hard otherwise.*

In the context of the previous statement, a *graph database* is a database restricted to relations of arity 1 or 2 (hence an ABox can be seen as a graph database). The result above relies crucially on the #P-hardness of $\mathrm{PQE}(1/2)$ (and hence of SPPQE) for non-FO-rewritable (*a.k.a. unbounded*) hom-closed queries on graph databases (Amarilli 2023), and the FP/#P-hard dichotomy of $\mathrm{PQE}(1/2; 1)$ for UCQs (Dalvi and Suciu 2012; Kenig and Suciu 2021).

3.3 Hardness of SVC for Reachability Games

This subsection shows the #P-hardness of the Shapley value computation of graph reachability, which, as we shall see in Section 4, implies the #P-hardness of several problems in the setting of ontologies.

Consider the Boolean query *st-reach* which asks whether there is a directed path from a vertex s to a vertex t in a (directed) graph. A graph can be seen as a graph database using a single binary relation, and s, t are individuals. As we show next, computing the Shapley value for this simple query is already #P-hard, even in the restricted case of purely endogenous databases.

Proposition 2. $\mathrm{SVC}^\mathsf{n}_{\textit{st-reach}}$ *is #P-hard.*

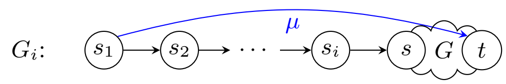

Figure 1: Illustration of the graph G_i.

The result follows via a reduction akin to the one in (Livshits et al. 2021), but from a different #P-hard task, namely S-T CONNECTEDNESS, which is the task of, given a graph G and vertices s, t thereof, counting the number of subgraphs of G which contain a path from s to t (Valiant 1979, problem 11). This reduction follows a technique that will be used again in later proofs, which consists in producing several related variants of an instance, so that when applying Equation (2) we obtain a system of linear combinations of the desired values. This system turns out to be invertible and thus we can obtain the values.

Proof sketch. By reduction from the S-T CONNECTEDNESS task, known to be #P-hard (Valiant 1979, Theorem 1 & Problem 11). Let $G = (V, E)$ be the input to the S-T CONNECTEDNESS. For each $1 \leq i \leq |E|$, let $G_i = (V_i, E_i)$ be the graph having $V_i = V \cup \{s_j : j \in [i]\}$, $E_i = E \cup \{(s_i, s), (s_1, t)\} \cup \{(s_j, s_{j+1}) : j \in [i-1]\}$ and let $\mu = (s_1, t)$, as shown in Figure 1. The argument then follows the same lines as (Bienvenu, Figueira, and Lafourcade 2024b, Lemma 4.2), where one can show that each $\mathrm{Sh}(E_i, \mathbf{v}, \mu)$ for the query s_1t-*reach* is a linear combination (plus constants) of the number of subgraphs of G of a given size on which there is a path from s to t, and that these form a solvable system. It suffices to solve the system and add up all solutions to obtain the total number of subgraphs that connect s to t. $\square$

4 Shapley Values in the Ontology Setting and First Intractability Results

Now that we that have seen how the Shapley value has been applied in the database setting, we can adapt the definitions and techniques to the context of OMQA. There are different ways to formalize this, in particular, the ontology may be considered to be part of the input or not. Remember that in Proposition 2 we identified the inherent difficulty of reachability-like queries for SVC. Since reasoning on TBox axioms inherently involves some form of reachability analysis, we will show that SVC is #P-hard as soon as the TBox is taken as being part of the input. This will motivate us to consider a different way of formulating and analyzing the SVC problem in later sections.

We first present a running example, which showcases the use of the Shapley value in the ontology setting.

Example 3. *Consider the* $\mathcal{ELHI}_\bot$ *KB defined in Figure 2, where the ABox (bottom half) contains information on some ingredients and recipes, and the TBox (top half) defines more complex notions such as a 'land-sea recipe'. For instance,* $\exists\mathsf{HasIngr}.\mathsf{FishBased} \sqsubseteq \mathsf{FishBased}$ *intuitively translates as 'anything that has a fish-based ingredient is fish-based'.*

A user of this KB might obtain poulardeNantua as an answer to the query $\mathsf{LandSea}(x)$*, and wonder which ingredients are the most responsible for this fact. She can thus set everything but the role assertions (which specify ingredients) as exogenous and compute the Shapley values for the Boolean query* $\mathsf{LandSea}(poulardeNantua)$*. The modeling choice to set only role assertions as endogenous corresponds to considering the background knowledge provided by the TBox and the concept assertions as being external to responsibility attribution, since they are not part of recipes.*[4]

We can compute the values via Eq. (1)*. There are* $4! = 24$ *possible permutations over the* 4 *endogenous role assertions* $\{e_1, e_2, e_3, e_4\}$*. 14 out of 24 permutations are s.t.* $\mathbf{v}(\sigma_{\leqslant e_1}) - \mathbf{v}(\sigma_{<e_1}) = 1$*, and similarly 6 for* e_2*, 2 for* e_3*, and 2 for* e_4*, making the respective Shapley values: 14/24, 6/24, 2/24 and 2/24.*

As expected, e_1 *has the highest responsibility because it is necessary to satisfy the query, then comes* e_2 *that only needs to be combined with* e_1 *and finally* e_3 *and* e_4 *that must be used together in addition to* e_1*.*

$\mathsf{FishBased} \sqcap \mathsf{MeatBased} \sqsubseteq \mathsf{LandSea}$	$\mathsf{Fish} \sqsubseteq \mathsf{FishBased}$
$\exists\mathsf{HasIngr}.\mathsf{FishBased} \sqsubseteq \mathsf{FishBased}$	$\mathsf{Seafood} \sqsubseteq \mathsf{FishBased}$
$\exists\mathsf{HasIngr}.\mathsf{MeatBased} \sqsubseteq \mathsf{MeatBased}$	$\mathsf{Crustacean} \sqsubseteq \mathsf{SeaFood}$
$\mathsf{HasSauce} \sqsubseteq \mathsf{HasIngr}$	$\mathsf{Meat} \sqsubseteq \mathsf{MeatBased}$

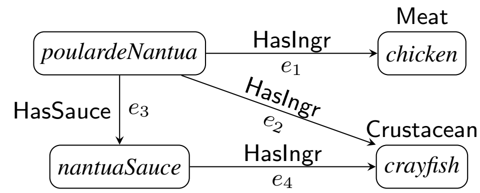

Figure 2: An example KB, with data and knowledge about a recipe from (Escoffier 1903). The arrows represent role assertions and labels on top of boxes (*e.g.* Meat) represent concept assertions.

Of course a naïve application of Eqs. (1) or (2) is not efficient, since they involve an exponential number of permutations or subsets. This raises the natural question of when a tractable approach can be found.

In the remainder of this section we illustrate how the hardness for $\mathrm{SVC}^{\mathsf{n}}_{st\text{-}reach}$ can be used to prove the hardness of many natural applications of the Shapley value using the same example KB depicted in Figure 2.

4.1 Shapley Values for Axiom Entailment

A first application of the Shapley value to ontologies is to focus solely on the TBox and determine which axioms are most responsible for a given TBox entailment, *e.g.* to find out why crustaceans count as fish-based. Unfortunately, the transitive nature of concept inclusions make this a reachability question and hence #P-hard in light of Proposition 2. In fact, hardness holds already for the simplest possible DL $\mathcal{L}_{\min}$ containing only concept name inclusions.

[4]It is especially important in this scenario to exclude the TBox axioms because while the axiom $\mathsf{Meat} \sqsubseteq \mathsf{MeatBased}$ is explicitly part of the KB, the inclusion $\mathsf{Crustacean} \sqsubseteq \mathsf{FishBased}$ is only indirectly inferred. This difference would lower the scores of the fish-based ingredients relative to the meat-based ones if the TBox axioms were also set as endogeneous.

Proposition 4. *For every TBox $\mathcal{T}$ and $S \subseteq \mathcal{T}$ and pair of concept names A, B, define $\mathbf{v}_{A \sqsubseteq B}(S) := 1$ if $S \models A \sqsubseteq B$, and 0 otherwise. The problem of computing, given a $\mathcal{L}_{\min}$ TBox $\mathcal{T}$, concept names A, B, and an axioms $\mu \in \mathcal{T}$, the Shapley value of μ in the game $(\mathcal{T}, \mathbf{v}_{A \sqsubseteq B})$ is $\#\mathsf{P}$-hard.*

Proof. To reduce from $\mathrm{SVC}^{\mathsf{n}}_{\textit{st-reach}}$, let $G = (V, E)$ a directed graph and $s, t \in V$. Consider the TBox $\mathcal{T}_G := \{A_x \sqsubseteq A_y \mid (A_x, A_y) \in E\}$ (it only contains concept name inclusions) and the concept inclusion $A_s \sqsubseteq A_t$. By construction the reachability from s to t is equivalent to the entailment of $A_s \sqsubseteq A_t$, which means the two games are isomorphic. □

4.2 Shapley Values for Query Entailment

We next consider the application of the Shapley value to explaining query entailment *w.r.t.* a DL KB. In our running example, a user may ask why there is a fish-based recipe in the KB of Figure 2 if there is no Fish assertion. She could therefore compute the Shapley values for the CQ $\exists x.\mathsf{FishBased}(x)$ and discover that the *crayfish* is considered fish-based because of the assertion $\mathsf{Crustacean}(\textit{crayfish})$ and the axioms $\mathsf{Crustacean} \sqsubseteq \mathsf{SeaFood} \sqsubseteq \mathsf{FishBased}$. However, the chain of inclusions needed to satisfy the query expresses a form of reachability, and thus we can transfer the $\#\mathsf{P}$-hardness of $\mathrm{SVC}^{\mathsf{n}}_{\textit{st-reach}}$.

In this setting we consider as input a *partitioned KB* $\mathcal{K} = (\mathcal{A}_{\mathsf{n}} \uplus \mathcal{A}_{\mathsf{x}}, \mathcal{T}_{\mathsf{n}} \uplus \mathcal{T}_{\mathsf{x}})$ and the task is to compute the Shapley value of the statements in $\mathcal{A}_{\mathsf{n}} \cup \mathcal{T}_{\mathsf{n}}$, in order to quantify their responsibility in $\mathcal{K}$ entailing q. Concretely, for any Boolean query q we consider the associated cooperative game having $\mathcal{A}_{\mathsf{n}} \cup \mathcal{T}_{\mathsf{n}}$ as players and $\mathbf{v}_q(S) := v_S - v_{\mathsf{x}}$ where $v_S = 1$ (resp. $v_{\mathsf{x}} = 1$) if $(\mathcal{A}_{\mathsf{x}} \cup (S \cap \mathcal{A}_{\mathsf{n}}), \mathcal{T}_{\mathsf{x}} \cup (S \cap \mathcal{T}_{\mathsf{n}})) \models q$ (resp. if $(\mathcal{A}_{\mathsf{x}}, \mathcal{T}_{\mathsf{x}}) \models q$), and 0 otherwise.

Proposition 5. *The problem of computing Shapley values for Boolean CQs over partitioned KBs on $\mathcal{L}_{\min}$ is $\#\mathsf{P}$-hard. Hardness holds even for queries given as ABox assertions.*

Proof. We build a TBox to reduce from $\mathrm{SVC}^{\mathsf{n}}_{\textit{st-reach}}$: let $G = (V, E)$ be a directed graph and $s, t \in V$, from which we define $\mathcal{T}_G := \{A_x \sqsubseteq A_y \mid (A_x, A_y) \in E\}$, with $A_t = A$, and $\mathcal{A} := \{A_s(c)\}$. We set the ABox as exogenous and the TBox as endogenous. Then a subset $\mathcal{X} \subseteq \mathcal{T}_G$ is *s.t.* $(\mathcal{A}, \mathcal{X}) \models q$ iff $\mathcal{X} \models A_s \sqsubseteq A_t$ iff $\mathcal{X}$ defines a subset of E which admits a path from s to t. The game for $(\mathcal{T}_G, \mathcal{A})$ is therefore isomorphic to the one for $\mathrm{SVC}^{\mathsf{n}}_{\textit{st-reach}}$ on G. □

4.3 Shapley Values on Exogenous Ontologies

Propositions 4 and 5 show that computing Shapley values of axioms is inevitably intractable, as reasoning on concept and role inclusions naturally involves reachability. One idea to sidestep this issue would be to treat the whole TBox as exogenous. Conceptually, this corresponds to treating TBox axioms as given or obvious, thereby focusing on explanations in terms of the ABox assertions. In our running example of Figure 2, the user may wonder what ingredients make the Poularde Nantua recipe fish-based. She could compute Shapley values for the CQ $\mathsf{FishBased}(\textit{poulardeNantua})$, setting everything but the $\mathsf{HasIngr}$-roles as exogenous, and discover that the recipe is fish-based because of the two chains of ingredients: $\textit{poulardeNantua} \xrightarrow{\mathsf{HasIngr}} \textit{crayfish}$ and $\textit{poulardeNantua} \xrightarrow{\mathsf{HasSauce}} \textit{nantuaSauce} \xrightarrow{\mathsf{HasIngr}} \textit{crayfish}$.

In terms of complexity the problem will once again be $\#\mathsf{P}$-hard due to reachability, which this time is expressed within the data itself. The culprit is the axioms $\exists\mathsf{HasIngr}.\mathsf{FishBased} \sqsubseteq \mathsf{FishBased}$, which can be found in any DL at least as expressive as $\mathcal{EL}$.

Proposition 6. *Let q be a CQ of the form $A(c)$, for $A \in \mathsf{N_C}$ and $c \in \mathsf{N_I}$. Then the problem of computing Shapley values for q over partitioned KBs on $\mathcal{EL}$ is $\#\mathsf{P}$-hard, even if we assume that the TBox only contains exogenous axioms.*

Proof. Let $G = (V, E)$ be a directed graph and $s, t \in V$, from which we define the partitioned ABox given by $\mathcal{A}_{\mathsf{x}} := \{B(c_s), D(c_t)\}$ and $\mathcal{A}_{\mathsf{n}} := \{r(c_x, c_y) \mid (x, y) \in E\}$ with B, D, r and the c_x being all fresh, except $c_s := c$. Then define the purely exogenous TBox $\mathcal{T} := \{B \sqcap D \sqsubseteq A, \exists r.D \sqsubseteq D\}$. Then a coalition $\mathcal{X} \subseteq \mathcal{A}_{\mathsf{n}}$ is *s.t.* $(\mathcal{X} \cup \mathcal{A}_{\mathsf{x}}, \mathcal{T}) \models q$ iff it defines a subset of G where there is a path from s to t. The corresponding cooperative game is then the same as the one defining $\mathrm{SVC}^{\mathsf{n}}_{\textit{st-reach}}$, hence the desired reduction. □

Interestingly, we can show that the problem stated in the preceding result is tractable for DL-Lite ontologies:

Proposition 7. *The problem of computing Shapley values of CQs of the form $A(c)$ (with $A \in \mathsf{N_C}$ and $c \in \mathsf{N_I}$) over partitioned KBs on DL-Lite$_{\mathsf{core}}$ is in FP when restricted to KBs with only exogenous TBox axioms.*

Proof. Note that one can compute, in polynomial time, a set of facts $\{\alpha_1, \ldots, \alpha_m\} \subseteq \mathcal{A}$ such that for every $\mathcal{A}' \subseteq \mathcal{A}$, we have $\mathcal{A} \models (\mathcal{T}, A(c))$ iff $\mathcal{A}' \models \alpha_1 \vee \cdots \vee \alpha_m$. This is only possible because $\mathcal{T}$ is a DL-Lite$_{\mathsf{core}}$ TBox.

We can then compute the Shapley values for $q^* = \alpha_1 \vee \cdots \vee \alpha_m$ and the database $\mathcal{A}_{\mathsf{n}} \uplus \mathcal{A}_{\mathsf{x}}$, disregarding the TBox $\mathcal{T}$ since it is exogenous. We observe that if $\mathcal{A}_{\mathsf{x}}$ contains some α_i, all facts in $\mathcal{A}_{\mathsf{n}}$ have value zero. Otherwise, only the facts α_i will have a non-zero Shapley value, and their values can be easily computed using Eq. (2) and the observation that there are precisely $\binom{|\mathcal{A}_{\mathsf{n}}| - m}{k}$ subsets $B \subseteq \mathcal{A}_{\mathsf{n}} \setminus \{\alpha_i\}$ of size k such that $\mathbf{v}_{q^*}(B \cup \{\alpha_i\}) - \mathbf{v}_{q^*}(B) = 1$. □

Determining for which classes of *non-atomic* CQs the previous proposition holds w.r.t. DL-Lite ontologies is challenging, as it would require us to first establish a full complexity characterization for plain CQs (without an ontology, on a binary signature), which remains an open question.

4.4 Approximation and Relevance

In view of the hardness results of Propositions 4 to 6, an alternative would be to give up on the precise Shapley value and instead find an approximation, or at the very least distinguish between elements having a zero or non-zero Shapley value. We call the elements with non-zero values *relevant*[5], because they are exactly those that appear in some minimal

[5]The identification of relevant axioms / assertions has been considered in previous work on explaining DL entailments and query answers, see e.g. (Peñaloza and Sertkaya 2010; Ceylan et al. 2020).

support. However, observe that the Shapley value is more informative than relevance: in Example 3 all considered assertions are relevant, but have different Shapley values.

There have been different works on approximating Shapley values in the context of databases (Livshits et al. 2021; Khalil and Kimelfeld 2023), but in the case of $\mathrm{SVC}^{\mathsf{n}}_{st\text{-}reach}$ it has been shown in (Khalil and Kimelfeld 2023, Theorem 5.1) that no multiplicative FPRAS approximation can be found unless $\mathsf{BPP} \subseteq \mathsf{NP}$,[6] because merely deciding relevance is NP-hard —indeed, it can be used to decide if a given edge lies on a simple path from s to t, which is a known NP-complete problem. Hence, in the previous cases we have considered, the intractability stems from the notion of relevance rather than from the precise Shapley value.

5 A Dichotomy for OMQs in $\mathcal{ELHI}_\perp$

The results of Section 4 show that allowing TBoxes to be part of the input, make the Shapley value computation problems $\#\mathsf{P}$-hard. This suggests the interest of analyzing the complexity of Shapley value computation at the level of individual ontology-mediated queries, only taking data as input. Such a non-uniform approach to complexity analysis has previously been undertaken for several OMQA settings, and in particular in the context of probabilistic OMQA (Jung and Lutz 2012). Moreover, this perspective aligns nicely with the formulation of Shapley value computation for database queries and shall allow us to transfer results from the database setting.

In the present section, we will be interested in applying Theorem 1 in a 'black-box' fashion, so we will need to identify a class of OMQs that is hom-closed and connected. To this end, we prove the following lemma, which shows that the addition of a $\mathcal{ELHI}_\perp$ ontology preserves the connectedness of hom-closed queries.

Lemma 8. *Let q be a connected C-hom-closed query and $\mathcal{T}$ an $\mathcal{ELHI}_\perp$ ontology. Then the OMQ $Q := (\mathcal{T}, q)$ is a connected C-hom-closed query.*

Proof. Consider q and $\mathcal{T}$ as in the lemma statement, and define $q_\perp$ as the Boolean query that is satisfied whenever the input ABox $\mathcal{A}$ is inconsistent with $\mathcal{T}$. By inspecting Datalog rewritings of ABox inconsistency for (extensions of) $\mathcal{ELHI}_\perp$ (Eiter et al. 2012; Bienvenu and Ortiz 2015), it can be readily verified that the query $q_\perp$ is both (C-)hom-closed and connected.

Now consider a minimal support $\mathcal{A}$ of Q that is $\mathcal{T}$-consistent. The knowledge base $(\mathcal{A}, \mathcal{T})$ thus admits a canonical model $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ relative to $\mathcal{T}$, and we may suppose that $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ is constructed in a standard way, as in e.g. (Bienvenu and Ortiz 2015). By definition, $\mathcal{I}_{\mathcal{A},\mathcal{T}} \models q$ since $\mathcal{A} \models Q$, so $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ must contain a minimal support $\mathcal{S}$ of q. Moreover, since q is connected, $\mathcal{S}$ must be contained in a single connected component of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$. However, it can be easily seen from the construction of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ that the connected components of $\mathcal{I}_{\mathcal{A},\mathcal{T}}$ are the canonical models of the connected components of $\mathcal{A}$. It follows that $\mathcal{S} \subseteq \mathcal{I}_{\mathcal{A}^*,\mathcal{T}}$ for some connected component $\mathcal{A}^*$ of $\mathcal{A}$. Since $\mathcal{I}_{\mathcal{A}^*,\mathcal{T}}$ contains $\mathcal{S}$, we have $\mathcal{I}_{\mathcal{A}^*,\mathcal{T}} \models q$ and thus $\mathcal{A}^* \models Q$. Since $\mathcal{A}$ is assumed to be minimal, then necessarily $\mathcal{A}^* = \mathcal{A}$ and thus $\mathcal{A}$ is connected. This completes the proof that Q is connected.

To show Q is C-hom-closed, consider ABoxes $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A} \models Q$ and $\mathcal{A} \xrightarrow{C\text{-}hom} \mathcal{B}$. If $\mathcal{A} \models q_\perp$, then $\mathcal{B} \models q_\perp$ since $q_\perp$ is hom-closed, and from $\mathcal{B} \models q_\perp$ we trivially have $\mathcal{B} \models \mathcal{Q}$. Otherwise, the KB $(\mathcal{A}, \mathcal{T})$ is consistent, and if $\mathcal{B} \not\models q_\perp$, then so too is $(\mathcal{B}, \mathcal{T})$. An examination of the canonical model construction in (Bienvenu and Ortiz 2015) reveals that $\mathcal{A} \xrightarrow{C\text{-}hom} \mathcal{B}$ implies $\mathcal{I}_{\mathcal{A},\mathcal{T}} \xrightarrow{C\text{-}hom} \mathcal{I}_{\mathcal{B},\mathcal{T}}$. Since $\mathcal{A} \models Q$, we have $\mathcal{I}_{\mathcal{A},\mathcal{T}} \models q$. As q is C-hom-closed, this yields $\mathcal{I}_{\mathcal{B},\mathcal{T}} \models q$, hence $\mathcal{B} \models Q$ as required. ☐

The first and most direct application of Lemma 8 is to establish a dichotomy for OMQs whose base query is constant-free and connected, as a consequence of the equivalence with probabilistic databases established in (Bienvenu, Figueira, and Lafourcade 2024b).

Theorem 9. *For every connected (constant-free) hom-closed query q and $\mathcal{ELHI}_\perp$ ontology $\mathcal{T}$ we have $\mathrm{SVC}_{(\mathcal{T},q)} \equiv_{\mathsf{P}} \mathrm{SPPQE}_{(\mathcal{T},q)}$. Further, the problem is in FP if the OMQ $(\mathcal{T}, q)$ can be rewritten into a safe UCQ and $\#\mathsf{P}$-hard otherwise.*

Proof. For OMQs $(\mathcal{T}, q)$ from the considered class, Lemma 8 states that $(\mathcal{T}, q)$ is a connected ($\emptyset$-)hom-closed query, hence Theorem 1 gives the desired results. ☐

It is decidable whether or not a given UCQ is safe (Dalvi and Suciu 2012, implicit). It therefore follows that the dichotomy given by Theorem 9 is effective whenever the first-order rewritability is decidable for the considered class of OMQs, and that a first-order rewriting can always be effectively computed when there exists one. This is in particular true for ($\mathcal{ELHI}_\perp$,CQ) (Bienvenu et al. 2016, Theorem 5).

Furthermore, (Jung and Lutz 2012, Theorem 5) gives a syntactic characterisation of which constant-free connected OMQs in $(\text{DL-Lite}_{\mathsf{core}}, CQ)$ are equivalent to safe UCQs. By Theorem 9, it also characterizes constant-free connected OMQs $Q \in (\text{DL-Lite}_{\mathsf{core}}, CQ)$ that are *s.t.* $\mathrm{SVC}_Q \in \mathsf{FP}$.

6 Strengthening the $\#\mathsf{P}$-Hardness Result

The dichotomy of Theorem 9 is limited in two respects:

- The result only covers connected constant-free queries.
- When an OMQ is seen as an abstract query, the distinction between consistent and inconsistent ABoxes is lost. However, one might be interested in explaining answers to a query only over consistent ABoxes.

When considering OMQs in $(\mathcal{ELHI}_\perp, \mathrm{UCQ})$, both of these points can be improved upon by studying the properties of such OMQs that are non-FO-rewritable.

Theorem 10. *Let q be a UCQ and $\mathcal{T}$ a $\mathcal{ELHI}_\perp$ ontology. If the OMQ $Q := (\mathcal{T}, q)$ is non-FO-rewritable w.r.t. consistent ABoxes, then SVC_Q and $\mathrm{PQE}_Q(1/2; 1)$ on consistent ABoxes are both $\#\mathsf{P}$-hard.*

[6] The same article gives a so-called additive approximation, but its use is very limited by the fact that it cannot decide relevance.

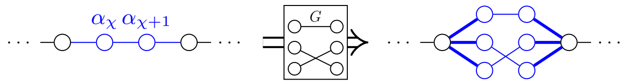

Figure 3: Encoding $\mathcal{P}^G$ of G in $\mathcal{P}$. The endogenous assertions are indicated by thick lines.

As a consequence of Theorem 10, we obtain the following dichotomy for the probabilistic evaluation of OMQs, which extends (Jung and Lutz 2012, Theorem 7) by generalizing from $(\mathcal{ELI}, CQ)$ to $(\mathcal{ELHI}_\bot, UCQ)$.

Theorem 11. *Let Q be a $(\mathcal{ELHI}_\bot, \text{UCQ})$ OMQ. Then* $\text{PQE}_Q(1/2; 1)$, SPPQE_Q *and* PQE_Q *are all in* FP *if Q is FO-rewritable into a safe UCQ, and* $\#$P*-hard otherwise. Further, this dichotomy is effective if $Q \in (\mathcal{ELHI}_\bot, CQ)$.*

The remainder of this section will be devoted to proving Theorem 10. We start in Section 6.1 by explaining the core idea for the reduction with a restricted case, and showing the existence of the necessary structures to build the reduction. Then, in Section 6.2, we show how to apply the idea from the restricted case in general.

6.1 Proof Idea via a Restricted Setting

The idea behind the proof is the fact that any non-FO-rewritable OMQ $Q = (\mathcal{T}, q) \in (\mathcal{ELHI}_\bot, \text{UCQ})$ must have minimal supports that contain arbitrarily deep tree-like structures (see Claim 13). These must contain some arbitrarily long path, where we define a *path* between two constants a_0 and a_k to be any non-empty set of assertions $\mathcal{P} = \{R_1(a_0, a_1), \ldots, R_k(a_{k-1}, a_k)\}$, where $a_0, \ldots, a_k$ are pairwise distinct constants and each $R_i(a_{i-1}, a_i)$ stands for a role assertion of the form $r(a_{i-1}, a_i)$ or $r(a_i, a_{i-1})$. To give the intuition, we shall first consider the restricted case where $\mathcal{P}$ is a minimal support for Q on its own, and where Q is connected and hom-closed.

Similarly to what has been done to prove (Livshits et al. 2021, Proposition 4.6), we wish to reduce from the known $\#$P-hard problem of counting the number of independent sets in a bipartite graph $G = (X, Y, E)$. For a sufficiently large $\mathcal{P}$, we can take two consecutive internal individuals $(a_\chi, a_{\chi+1})$, and duplicate them to encode an arbitrary bipartite graph as an ABox $\mathcal{P}^G$, as shown in Figure 3. We set as endogenous every role assertion linking $a_{\chi-1}$ to a copy of a_χ, or linking a copy of $a_{\chi+1}$ to $a_{\chi+2}$, the remaining assertions are exogenous.

Any coalition $\mathcal{X} \subseteq \mathcal{P}^G_{\mathsf{n}}$ naturally maps to a set $V_\mathcal{X} \subseteq X \cup Y$ of vertices of G. If $V_\mathcal{X}$ is not independent, then there is a path homomorphic to $\mathcal{P}$ in $\mathcal{X} \cup \mathcal{P}^G_{\mathsf{x}}$, hence $\mathcal{X} \cup \mathcal{P}^G_{\mathsf{x}} \models Q$. Otherwise, if $V_\mathcal{X}$ is independent, then any connected subset of $\mathcal{X} \cup \mathcal{P}^G_{\mathsf{x}}$ will homomorphically map to a *proper* subset of $\mathcal{P}$. Since Q is connected and $\mathcal{P}$ is a minimal support for it, this implies that $\mathcal{X} \cup \mathcal{P}^G_{\mathsf{x}} \not\models Q$. From this we can use the same technique as (Livshits et al. 2021, Proposition 4.6) to count the independent sets of G, as we shall see later.

6.2 The General Case

Now that we have seen the underlying idea for the restricted case, we formally show the existence of the path $\mathcal{P}$ upon which we shall build the reduction.

To fix the notations, let $Q = (\mathcal{T}, q) \in (\mathcal{ELHI}_\bot, \text{UCQ})$ be the non-FO-rewritable (w.r.t. consistent ABoxes) OMQ we consider. We write $q = \bigvee_{i \in I} \bigwedge_{j \in J_i} q_{i,j}$ where the $q_{i,j}$ are *connected components*, that is, maximal connected subqueries.[7] For every $q_{i,j}$ we write $C_{i,j} := \mathit{const}(q_{i,j})$, and $C := \bigcup_{i \in I} \biguplus_{j \in J_i} C_{i,j}$. Note that the last union is indeed disjoint because the connected components of a CQ cannot share any constant or variable.

The following is a simple consequence of Q being non-FO-rewritable w.r.t. consistent ABoxes.

Claim 12. *There exist i, j such that for all $N \in \mathbb{N}$, there are $\mathcal{T}$-consistent ABoxes $\mathcal{S}^N_{i,j}, \mathcal{S}^N_{i,\neg j}$ with $|\mathcal{S}^N_{i,j}| \geqslant N$ such that:*

(1) $\mathit{const}(\mathcal{S}^N_{i,j}) \cap \mathit{const}(\mathcal{S}^N_{i,\neg j}) = \emptyset$ and $\mathcal{S}^N_{i,j} \uplus \mathcal{S}^N_{i,\neg j}$ is a minimal support for Q;

(2) $\mathcal{S}^N_{i,j}$ and $\mathcal{S}^N_{i,\neg j}$ are minimal supports for $(\mathcal{T}, q_{i,j})$ and $(\mathcal{T}, q_{i,\neg j})$ respectively, where $q_{i,\neg j} := \bigwedge_{j' \in J_i \setminus \{j\}} q_{i,j}$.

Let us henceforth fix $q_{\bar{\imath},\bar{\jmath}}$ as having the indices $\bar{\imath}, \bar{\jmath}$ satisfying the statement of Claim 12. The next claim is a straightforward application of (Bienvenu et al. 2020, Prop. 23)[8].

Claim 13. *For all $k \in \mathbb{N}$, there exists a minimal support $\mathcal{A}^k$ for $(\mathcal{T}, q_{\bar{\imath},\bar{\jmath}})$ that is $\mathcal{T}$-consistent, individuals $a_0, \ldots, a_k$, and $N \in \mathbb{N}$ s.t.:*

(1) $\mathcal{A}^k \xrightarrow{\text{C-hom}} \mathcal{S}^N_{\bar{\imath},\bar{\jmath}}$;

(2) $\mathcal{A}^k$ has a path $\mathcal{P} = \{R_1(a_0, a_1), \ldots, R_k(a_{k-1}, a_k)\}$, and further $\mathcal{P}$ is the only path between a_0 and a_k in $\mathcal{A}^k$;

(3) $\{a_0, \ldots, a_k\} \cap C = \emptyset$.

We now have the minimal support $\mathcal{A}^\circ := \mathcal{A}^k \uplus \mathcal{S}^N_{\bar{\imath},\neg\bar{\jmath}}$ we need for the reduction. However, it differs from the one in the restricted case of Section 6.1 in two respects:

1. the path $\mathcal{P}$ is merely a subset of the minimal support;
2. the OMQ Q is not necessarily connected.

To deal with the first issue we introduce some notions. Let $\mathcal{A}^k_\chi := \mathcal{A}^k \setminus \{R_\chi(a_{\chi-1}, a_\chi), R_{\chi+1}(a_\chi, a_{\chi+1})\}$. Given an assertion $\alpha \in \mathcal{A}^k$ and an individual a_χ on the path $\mathcal{P}$, we say that α is *below* (resp. *left* of, resp. *right* of) a_χ if α is in the connected component of a_χ in $\mathcal{A}^k_\chi$ (resp. $a_{\chi-1}$, resp. $a_{\chi+1}$). Further, we say that the assertions $R_\chi(a_{\chi-1}, a_\chi)$ and $R_{\chi+1}(a_\chi, a_{\chi+1})$ are respectively left and right of a_χ.

Claim 14. *For every assertion a_χ on the path $\mathcal{P}$, the sets $\mathbf{B}(a_\chi)$, $\mathbf{L}(a_\chi)$ and $\mathbf{R}(a_\chi)$ of assertions that are below, left of and right of a_χ resp. are disjoint. Furthermore, if $\chi < \lambda$, then $\mathbf{R}(a_\chi)$ contains $\mathbf{B}(a_\lambda)$ and $\mathbf{R}(a_\lambda)$, while $\mathbf{L}(a_\lambda)$ contains $\mathbf{B}(a_\chi)$ and $\mathbf{L}(a_\chi)$ (see Figure 4).*

Proof sketch. If any two of $\mathbf{B}(a_\chi)$, $\mathbf{L}(a_\chi)$ and $\mathbf{R}(a_\chi)$ were to intersect, then we could build a path between a_0 and a_k other than $\mathcal{P}$. As for the inclusions, they are fairly intuitive when looking at Figure 4. □

[7]Observe that, for example, the CQ $R(x, c) \wedge S(c, y)$, where c is a constant, has only one connected component.

[8]This paper is the extended version of (Bienvenu et al. 2016).

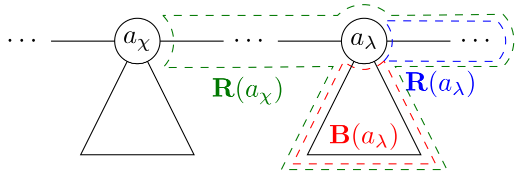

Figure 4: Illustration of why $\mathbf{R}(a_\chi)$ contains $\mathbf{B}(a_\lambda)$ and $\mathbf{R}(a_\lambda)$.

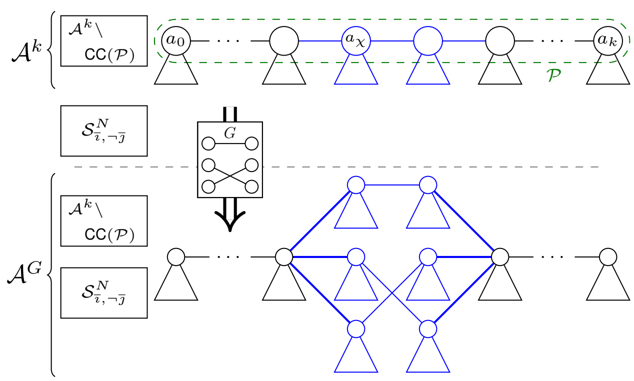

Figure 5: Construction of $\mathcal{A}^G$ from $\mathcal{A}^\circ$ and G. Thick lines represent endogenous assertions. The boxes are sets with no structure, and the triangle below some a is the connected $\mathbf{B}(a)$. $\mathsf{CC}(\mathcal{P})$ denotes the set of assertions in the connected component of $\mathcal{P}$.

Claim 14 allows us to generalize the construction we made for the restricted case. The idea is simple: every time we copied an assertion in the restricted case, we also copy its $\mathbf{B}$ set, always with fresh constants. Note that the construction still depends on a pair $(a_\chi, a_{\chi+1})$ of consecutive individuals of $\mathcal{P}$ that are *internal* (*i.e.*, not in $\{a_0, a_k\}$). We call such pair an *interface*. We overlook the choice of this interface for now, but it will become relevant later.

Starting with a bipartite graph $G = (X, Y, E)$ whose independent sets we wish to count, we build the partitioned ABox $\mathcal{A}^G$ illustrated in Figure 5 (*cf.* with the restricted case of Figure 3), which we formally define as follows. Starting from $\mathcal{A}^\circ$, we focus on the 3 assertions $R_\chi(a_{\chi-1}, a_\chi)$, $R_{\chi+1}(a_\chi, a_{\chi+1})$, $R_{\chi+2}(a_{\chi+1}, a_{\chi+2})$ of $\mathcal{P}$ that intersect the chosen $(a_\chi, a_{\chi+1})$. We replace a_χ (resp. $a_{\chi+1}$) with a family $(b_x)_{x\in X}$ (resp. $(c_y)_{y\in Y}$) of copies with fresh constants. We also copy every assertion below a_χ (resp. below $a_{\chi+1}$), again with fresh constants. Finally, we add the assertions $\{R_\chi(a_{\chi-1}, b_x)\}_{x\in X}$, $\{R_{\chi+1}(b_x, c_y)\}_{(x,y)\in E}$, $\{R_{\chi+2}(c_y, a_{\chi+2})\}_{y\in Y}$ to obtain an ABox, which we denote by $\mathcal{A}^G$. As for the associated partition, we set the endogenous assertions $\mathcal{A}^G_\mathsf{n} := \{R_\chi(a_{\chi-1}, b_x) \mid x \in X\} \cup \{R_{\chi+2}(c_y, a_{\chi+2}) \mid y \in Y\}$ and the rest as exogenous.

We can now try to apply to $\mathcal{A}^G$ the same arguments that we used on $\mathcal{P}^G$ for the restricted case. If we consider a coalition $\mathcal{X} \subseteq \mathcal{P}^G_\mathsf{n}$, it once again naturally maps to a set of vertices of G, via a bijection we denote by $\eta : \mathcal{A}^G_\mathsf{n} \to X \cup Y$.

If $\eta(\mathcal{X})$ is not independent, then there is a path C-homomorphic to $\mathcal{P}$ in $\mathcal{X} \cup \mathcal{A}^G_\mathsf{x}$, and since every assertion in $\mathcal{A}^\circ \setminus \mathcal{P}$ is present in $\mathcal{A}^G_\mathsf{x}$ (or at least has the necessary C-isomorphic copies in the case of the assertions below a_χ and $a_{\chi+1}$), this implies $\mathcal{X} \cup \mathcal{A}^G_\mathsf{x} \models Q$.

If $\eta(\mathcal{X})$ is independent, however, we cannot directly use the argument above, because Q is no longer assumed to be connected. In fact $\mathcal{X} \cup \mathcal{A}^G_\mathsf{x}$ could conceivably contain a disconnected minimal support $\mathcal{S}$ for Q such as depicted in Figure 6.(a). Nevertheless, if we denote by $\rho : \mathcal{A}^G \to \mathcal{A}^\circ$ the C-homomorphism that maps every fresh constant back to its original counterpart (essentially, ρ reverses the construction of $\mathcal{A}^G$), then we must have $\rho(\mathcal{S}) = \mathcal{A}^\circ$. Otherwise, this would contradict the minimality of $\mathcal{A}^\circ$. In fact, by leveraging the fact that the OMQ $(\mathcal{T}, q_{i,j})$ built from any connected component $q_{i,j}$ is connected by Lemma 8; we can further show that such an $\mathcal{S}$ always admits a partition $\mathcal{S} = \mathcal{S}_\leftarrow \uplus \mathcal{S}_\rightarrow$ with specific properties. This motivates the following definition of splittable interfaces.

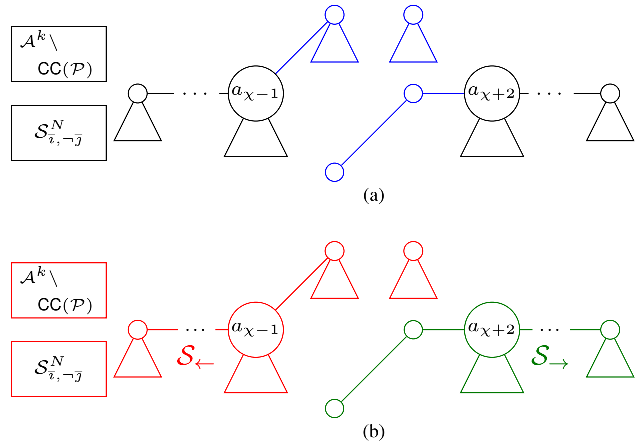

Figure 6: (a) Example of what $\mathcal{S}$ could look like. There is no path between $a_{\chi-1}$ and $a_{\chi+2}$, but the ABox can still collapse back to the whole $\mathcal{A}^\circ$ by ρ. This implies in particular that the whole black regions are present. (b) Partition of $\mathcal{S}$ into $\mathcal{S}_\leftarrow \uplus \mathcal{S}_\rightarrow$ that makes the interface splittable.

An interface $(a_\chi, a_{\chi+1})$ is said to be *splittable* if there exists some $i \in I$, two sets $J_\leftarrow, J_\rightarrow \subseteq J_i$ of connected components of the i^{th} disjunct of q, and a minimal support $\mathcal{S}_\leftarrow$ for $\bigwedge_{j\in J_\leftarrow} q_{i,j}$ which contains every assertion left of a_χ and no assertion right of $a_{\chi+1}$, and the symmetrical property for $J_\rightarrow$, with the added condition that $\mathcal{S}_\rightarrow$ is connected. These two minimal supports correspond to the red and green regions in Figure 6.(b), and they are the only way the construction can fail:

Claim 15. *$\mathcal{A}^G$ is $\mathcal{T}$-consistent and, if it is built from an unsplittable interface $(a_\chi, a_{\chi+1})$, then for every coalition $\mathcal{X} \subseteq \mathcal{A}^G_\mathsf{n}$, $\mathcal{X} \uplus \mathcal{A}^G_\mathsf{x} \models Q$ iff $\eta(\mathcal{X})$ isn't independent in G.*

The final ingredient is the fact that we can always find an unsplittable interface in $\mathcal{P}$ if it is long enough.

Claim 16. *The number of splittable interfaces in $\mathcal{P}$ is bounded by a function of q.*

Proof sketch. Two disjoint interfaces cannot have the same $J_\rightarrow$ otherwise it would contradict the minimality of the supports in the definition of splittability. The total number of splittable interfaces is therefore bounded by 2 times the number of sets of connected components of q. □

Once we have this, Claim 15 gives a bijection between the $\mathcal{X}$ *s.t.* $\mathcal{X} \uplus \mathcal{A}_{\mathsf{x}}^{G} \not\models Q$ and the independent sets of G, and we can conclude by exactly reproducing the end of the proof for (Livshits et al. 2021, Proposition 4.6), which consists in building several variants of the graph G to obtain a solvable linear system for the number of independent sets of G of each size. Proof details can be found in the full version.

7 Discussion

While the Shapley value had previously been suggested for ontology debugging (Deng, Haarslev, and Shiri 2007), its application to ontology-mediated query answering, and a precise analysis of the complexity of SVC, had not yet been considered. The present paper exploits very recent results on Shapley value computation in the database setting to obtain complexity dichotomies for OMQs whose TBoxes are formulated in the well-known Horn DL $\mathcal{ELHI}_\perp$. In particular, Theorem 9 identifies classes of OMQs for which the Shapley value can be computed in FP, while Theorem 10 provides a general #P-hardness result for non-FO-rewritable OMQs. Using the same techniques, and leveraging known connections between Shapley value computation and probabilistic query evaluation, we were further able to obtain a general dichotomy result for probabilistic OMQA (Theorem 11).

The dichotomy result we obtain for probabilistic OMQA is stronger than the one we obtain for SVC. This is to be expected as existing results on SVC in the pure database setting do not cover UCQs that are disconnected or with constants, and it is an important open question whether there is a dichotomy for SVC_q for all Boolean UCQs q. We remark however that any progress on this question can be immediately transferred to the OMQA setting.

In the context of probabilistic databases, existing works prove that $\mathrm{PQE}_q(1/2;1) \equiv_{\mathsf{P}} \mathrm{PQE}_q$ for any q that is either a UCQ or a (constant-free) hom-closed query. Our Theorem 11 shows that this equivalence also holds for any $q \in (\mathcal{ELHI}_\perp, UCQ)$ (with constants), suggesting that it might hold for the full class of C-hom-closed queries.

An interesting but challenging direction for future work is to study SVC for ontologies formulated using tuple-generating dependencies (*a.k.a.* existential rules). We expect that the extension of our results to such ontologies will be non-trivial, due both to the presence of higher-arity predicates and the lack of forest-shaped models. Indeed, in Section 5, the dichotomy for SPPQE that we exploited to obtain Theorem 9 is currently only known for arity 2 relations (*i.e.*, graph databases). The techniques in Section 6, on the other hand, rely on the existence of a unique path between anonymous elements, which was ensured by the existence of forest-shaped canonical models. By contrast, the bounded treewidth canonical models obtained for many classes of existential rules do not satisfy this unique-path condition.

Acknowledgements

This work is partially supported by ANR AI Chair INTENDED (ANR-19-CHIA-0014) and ANR project CQFD (ANR-18-CE23-0003).

References

Abramovich, O.; Deutch, D.; Frost, N.; Kara, A.; and Olteanu, D. 2024. Banzhaf values for facts in query answering. *Proceedings of the ACM on Management of Data* 2(3):123.

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.

Amarilli, A. 2023. Uniform reliability for unbounded homomorphism-closed graph queries. In *Proceedings of the International Conference on Database Theory (ICDT)*, 14:1–14:17.

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

Bertossi, L. E.; Kimelfeld, B.; Livshits, E.; and Monet, M. 2023. The Shapley value in database management. *SIGMOD Record* 52(2):6–17.

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 *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 965–971.

Bienvenu, M.; Hansen, P.; Lutz, C.; and Wolter, F. 2020. First order-rewritability and containment of conjunctive queries in Horn description logics (Long version with appendix). *CoRR* abs/2011.09836.

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 (Long version with appendix). *CoRR* abs/2407.20058.

Bienvenu, M.; Figueira, D.; and Lafourcade, P. 2024b. When is Shapley value computation a matter of counting? In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*.

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.

Calì, A.; Gottlob, G.; Lukasiewicz, T.; and Pieris, A. 2011. A logical toolbox for ontological reasoning. *SIGMOD Record* 40(3):5–14.

Ceylan, İ. İ.; Lukasiewicz, T.; Malizia, E.; and Vaicenavicius, A. 2019. Explanations for query answers under existential rules. In *Proceedings of the 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.

Dalvi, N. N., and Suciu, D. 2004. Efficient query evaluation on probabilistic databases. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, 864–875.

Dalvi, N., and Suciu, D. 2012. The dichotomy of probabilistic inference for unions of conjunctive queries. *Journal of the ACM* 59(6):1–87.

Deng, X.; Haarslev, V.; and Shiri, N. 2007. Measuring Inconsistencies in Ontologies. In *Proceedings of the European Semantic Web Conference*, 326–340.

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.

Eiter, T.; Ortiz, M.; Simkus, M.; Tran, T.; and Xiao, G. 2012. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 726–733.

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

Grant, J., and Hunter, A. 2006. Measuring inconsistency in knowledge bases. *Journal of Intelligent Information Systems* 27(2):159–184.

Hunter, A., and Konieczny, S. 2010. On the measure of conflicts: Shapley inconsistency values. *Artificial Intelligence (AIJ)* 174(14):1007–1026.

Jung, J. C., and Lutz, C. 2012. Ontology-based access to probabilistic data with OWL QL. In *Proceedings of the International Semantic Web Conference (ISWC)*, 182–197.

Kara, A.; Olteanu, D.; and Suciu, D. 2024. From Shapley value to model counting and back. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*.

Kenig, B., and Suciu, D. 2021. A dichotomy for the generalized model counting problem for unions of conjunctive queries. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, 312–324.

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

Livshits, E., and Kimelfeld, B. 2022. The Shapley value of inconsistency measures for functional dependencies. *Logical Methods in Computer Science (LMCS)* 18(2).

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.

Lundberg, S. M., and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

Mugnier, M.-L., and Thomazo, M. 2014. An introduction to ontology-based query answering with existential rules. In *Tutorial Lectures of the International Reasoning Web (RW) Summer School*, volume 8714 of *Lecture Notes in Computer Science (LNCS)*. 245–278.

Peñaloza, R., and Sertkaya, B. 2010. On the complexity of axiom pinpointing in the $\mathcal{EL}$ family of description logics. In Lin, F.; Sattler, U.; and Truszczynski, M., eds., *Principles of Knowledge Representation and Reasoning (KR)*.

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 *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, 285–297.

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 *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 5511–5519.