Provenance in Ontology-based Data Access

Ana Ozaki and Rafael Peñaloza

KRDB Research Centre, Free University of Bozen-Bolzano, Italy

Abstract. We present an approach for dealing with provenance information in an ontology-based data access (OBDA) setting. Our approach is based on provenance semirings, which were studied in database theory as an abstract tool to relate the result of a query with the possible matches of the query in the data. We investigate the problems of (i) deciding whether an ontology annotated with provenance information entails a conjunctive query associated with a provenance polynomial, and (ii) computing a polynomial representing the provenance of a query entailed by a provenance annotated ontology. We show that these polynomials may be infinite in general. We then study a special case where the semiring is idempotent and hence the polynomial is guaranteed to be finite. We instantiate these problems to DL-Lite$_\mathcal{R}$ and provide some complexity results.

1 Introduction

Description logics (DLs) [3] have been sucessfully applied to integrate data coming from multiple and heterogeneous sources. In this setting, commonly known as ontology-based data access (OBDA), an ontology enriches the data with background knowledge, providing the user with a high-level conceptual view of the data, among other beneficial properties [9, 13]. Data integration can be useful to support users with a convenient vocabulary for queries, and help them to obtain accurate results.

Querying over multiple data sources also increases the challenge of establishing the authorship and of determining the reliability of the data. More importantly, this challenge transfers to all the (implicit) consequences that can be derived from the data and the ontology. In this scenario, one may not just be interested in knowing the result of a query, but also *how* it was produced, how *likely* or *reliable* the result may be, *how many* different ways are there to derive it, or how *dependent* it is to certain parts of the data, among many other questions [18].

To address all these issues, *provenance semirings* were introduced in database theory [8] as an abstract tool to record and track provenance information; that is, to keep track of the specific database tuples that are responsible for a derivation, and additional information associated to them. In this work, we present an approach based on provenance semirings for dealing with provenance information in an OBDA setting. Following the approach from database theory, we assume that facts are annotated with a label and we want to know which combinations of these labels lead to the entailment of a query. Such information is expressed in the form of a *provenance polynomial*, as we illustrate in the following example.

Example 1. Consider the DL assertions

$$\mathsf{headGov}(\mathsf{Renier}, \mathsf{Venice}),\ \mathsf{headGov}(\mathsf{Brugnaro}, \mathsf{Venice}),\ \mathsf{City}(\mathsf{Venice})$$

being annotated with the sources p, q, and r respectively. Based on these assertions, the answer to the query $\exists xy.(\mathsf{headGov}(x, y) \wedge \mathsf{City}(y))$ can be derived by using any of the first two assertions (the role assertions) together with the last assertion. Based on the source annotations, this information can be expressed through the provenance polynomial $(p \oplus q) \otimes r$.

In our approach for an OBDA setting, we do not only label the facts in the ABox, but also the concept inclusions from the TBox. These labels are interpreted and propagated according to a semiring semantics, as we illustrate in the next example.

Example 2. Consider the inclusion $\exists\mathsf{headGov}.\top \sqsubseteq \mathsf{Mayor}$ annotated with the source s. Based on this inclusion and the annotated assertions from Example 1, the answer to $\exists x.(\mathsf{Mayor}(x))$ can be derived by using any of the first two assertions together with the inclusion. Based on the source annotations, this information can be expressed through the provenance polynomial $(p \oplus q) \otimes s$.

We investigate (i) the complexity of deciding whether an ontology annotated with provenance information entails a conjunctive query associated with a provenance polynomial, and (ii) the problem of computing polynomials representing the provenance of a query entailed by a provenance annotated ontology. We instantiate these problems to DL-Lite$_\mathcal{R}$, the logic underpinning the OWL 2 QL profile [1], and show that, for the first problem, the complexity remains NP-complete in combined complexity, and in LogSpace in data complexity (as in the classical query answering problem in DL-Lite$_\mathcal{R}$). Regarding problem (ii), we show that the set of polynomials representing provenance may be infinite in general. So we study a special case where the semiring is multiplicatively-idempotent and the set of polynomials is guaranteed to be finite. We show that computing the provenance polynomials can be hard, i.e., there is a DL-Lite$_\mathcal{R}$ ontology and a query such that the set of provenance polynomials cannot be represented with polynomial space.

2 Basic Definitions

Following the original ideas from provenance semirings in databases [8], we represent the provenance information through a so-called *positive algebra provenance semiring* (or *provenance semiring* for short). Formally, given a set X of *variables*, the provenance semiring is the algebra $\mathcal{S} = (\mathbb{N}[X], \oplus, \otimes, 0, 1)$, where the product $\otimes$ and the addition $\oplus$ are two commutative, and associative binary operators, and the product distributes over the addition. These operators are defined as usual over the space $\mathbb{N}[X]$ of finite polynomials with integer coefficients on the variables X. We denote by $\mathsf{N_P}$ the set of all (finite) polynomials over the variables X that are of *expanded* form; that is $\mathsf{N_P}$ contains only polynomials of the form

$\sum_{1\leq i\leq n}\prod_{1\leq j\leq m} a_{i,j}$, with $a_{i,j} \in X$, and $n, m > 0$. In words, a polynomial in expanded form is a finite set of *monomials*, each formed by a finite product of variables in X. Notice that distributivity of product over additions entails that every polynomial can be equivalently rewritten in expanded form; however, the expanded form of a polynomial may become exponentially larger. For instance, the polynomial $(p \oplus q) \otimes r$ from Example 1 is equivalent to $(p \otimes r) \oplus (q \otimes r)$.

2.1 Provenance Annotated Ontologies

The provenance information for each axiom in an ontology is stored in the form of an annotation. For the scope of this paper, we focus on ontologies written in the description logic DL-Lite$_\mathcal{R}$ [2]. Consider three mutually disjoint countable sets of *concept names* $\mathsf{N_C}$, *role names* $\mathsf{N_R}$, and *individual names* $\mathsf{N_I}$. A DL-Lite$_\mathcal{R}$ *concept* (resp. *role*) *assertion* is an expression of the form $A(a)$ (resp. $R(a, b)$), with $A \in \mathsf{N_C}$ (resp. $R \in \mathsf{N_R}$) and $a, b \in \mathsf{N_I}$. DL-Lite$_\mathcal{R}$ *role* and *concept inclusions* are expressions of the form $S \sqsubseteq T$ and $B \sqsubseteq C$, respectively, where S, T are role expressions and B, C are concepts built according to the grammar rules

$$S ::= R \mid R^- \qquad T ::= S \mid \neg S \qquad B ::= A \mid \exists S \qquad C ::= B \mid \neg B,$$

with $R \in \mathsf{N_R}$ and $A \in \mathsf{N_C}$. A DL-Lite$_\mathcal{R}$ *axiom* is either a DL-Lite$_\mathcal{R}$ assertion, role inclusion, or concept inclusion. A *provenance annotated* DL-Lite$_\mathcal{R}$ *ontology* is a set of *annotated axioms* of the form (α, p) where α is a DL-Lite$_\mathcal{R}$ axiom and $p \in X \subseteq \mathsf{N_P}$ is a variable of a provenance semiring, such that each variable appears in at most one axiom. We call DL-Lite$_\mathcal{R}^\mathsf{P}$ this annotated extension of DL-Lite$_\mathcal{R}$.

The semantics of DL-Lite$_\mathcal{R}^\mathsf{P}$ is given by interpretations, which extend the classical interpretations of DL-Lite$_\mathcal{R}$ to track provenance, when relevant. Formally, a DL-Lite$_\mathcal{R}^\mathsf{P}$ interpretation is a triple $\mathcal{I} = (\Delta^\mathcal{I}, \Delta_\mathsf{p}^\mathcal{I}, \cdot^\mathcal{I})$ where $\Delta^\mathcal{I}$ is a non-empty set (called the *domain* of $\mathcal{I}$), $\Delta_\mathsf{p}^\mathcal{I}$ is a non-empty set (called the *domain of polynomials* of $\mathcal{I}$), and $\cdot^\mathcal{I}$ is the *interpretation function* mapping

- every $p, q \in \mathsf{N_P}$ to some $p^\mathcal{I}, q^\mathcal{I}$ in the set $\Delta_\mathsf{p}^\mathcal{I}$ with the condition that $p^\mathcal{I} = q^\mathcal{I}$ iff the polynomials p and q are mathematically equal (e.g., $(p \otimes q)^\mathcal{I} = (q \otimes p)^\mathcal{I}$);
- every $A \in \mathsf{N_C}$ to some $A^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta_\mathsf{p}^\mathcal{I}$; and
- every $R \in \mathsf{N_R}$ to some $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I} \times \Delta_\mathsf{p}^\mathcal{I}$.

We extend the mapping $\cdot^\mathcal{I}$ to further DL-Lite$_\mathcal{R}$ expressions in the natural way:

$$(R^-)^\mathcal{I} = \{(e, d, p^\mathcal{I}) \mid (d, e, p^\mathcal{I}) \in R^\mathcal{I}\}\ ,\ (\neg S)^\mathcal{I} = (\Delta^\mathcal{I} \times \Delta^\mathcal{I} \times \Delta_\mathsf{p}^\mathcal{I}) \setminus S^\mathcal{I},$$
$$(\exists S)^\mathcal{I} = \{(d, p^\mathcal{I}) \mid \exists e \in \Delta^\mathcal{I} : (d, e, p^\mathcal{I}) \in S^\mathcal{I}\} \text{ and } (\neg B)^\mathcal{I} = (\Delta^\mathcal{I} \times \Delta_\mathsf{p}^\mathcal{I}) \setminus B^\mathcal{I}.$$

The DL-Lite$_\mathcal{R}^\mathsf{P}$ interpretation $\mathcal{I}$ *satisfies*: $(A(a), p)$ (respectively $(R(a, b), p)$) if $(a, p^\mathcal{I}) \in A^\mathcal{I}$ (resp. $(a, b, p^\mathcal{I}) \in R^\mathcal{I}$); $(C \sqsubseteq D, p)$ if, for all $q \in \mathsf{N_P}$, $(d, q^\mathcal{I}) \in C^\mathcal{I}$ implies that $(d, (q \otimes p)^\mathcal{I}) \in D^\mathcal{I}$; and $(S \sqsubseteq T, p)$ if $(d, e, q^\mathcal{I}) \in S^\mathcal{I}$ implies that $(d, e, (q \otimes p)^\mathcal{I}) \in T^\mathcal{I}$. $\mathcal{I}$ satisfies an annotated ontology $\mathcal{O}$, in symbols $\mathcal{I} \models \mathcal{O}$, if it satisfies all annotated axioms in $\mathcal{O}$.

Example 3. Consider the ontology $\mathcal{O}$ with the assertions of Example 1 and the inclusion of Example 2. Let $\mathcal{I}$ be a DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ interpretation with domain $\Delta^{\mathcal{I}} = \{\mathsf{Renier}, \mathsf{Venice}, \mathsf{Brugnaro}\}$ and $\Delta^{\mathcal{I}}_{\mathsf{p}}$ containing $\{p, q, r, s, (p \otimes s), (q \otimes s)\}$, which interprets individual names and polynomials in $\mathcal{O}$ by themselves, and

- $\mathsf{headGov}^{\mathcal{I}} = \{(\mathsf{Renier}, \mathsf{Venice}, p), (\mathsf{Brugnaro}, \mathsf{Venice}, q)\}$,
- $\mathsf{Mayor}^{\mathcal{I}} = \{(\mathsf{Renier}, p \otimes s), (\mathsf{Brugnaro}, q \otimes s)\}$,
- $\mathsf{City}^{\mathcal{I}} = \{(\mathsf{Venice}, r)\}$.

$\mathcal{I}$ is a model of $\mathcal{O}$.

2.2 Provenance Annotated Queries

We extend the notion of conjunctive queries in DLs by allowing binary and ternary predicates, where the last term of the tuple can either be a variable or a monomial in $\mathsf{N_P}$ (by definition of the semantics of DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$, tuples can only contain monomials, not sums). More specifically, a *Boolean conjunctive query (BCQ)* q is a sentence $\exists \vec{x}.\varphi(\vec{x}, \vec{a}, \vec{p})$, where φ is a conjunction of atoms of the form $A(t_1, t)$, $R(t_1, t_2, t)$, and t_i is either an individual name from $\vec{a}$, or a variable from $\vec{x}$, and t (the last term of each tuple) is either an element of $\mathsf{N_P}$ in the list $\vec{p}$ or a variable from $\vec{x}$. We often write $P(\vec{t})$ to refer to an atom which can be either $A(t_1, t)$ or $R(t_1, t_2, t)$ and write $P(\vec{t}) \in q$ if $P(\vec{t})$ is an atom occurring in q.

A *match* of the BCQ $q = \exists \vec{x}.\varphi(\vec{x}, \vec{a}, \vec{p})$ in the DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ interpretation $\mathcal{I}$ is a function π mapping $\vec{x} \cup \vec{a} \cup \vec{p}$ to $\Delta^{\mathcal{I}} \cup \Delta^{\mathcal{I}}_{\mathsf{p}}$, such that, for all $b \in \vec{a} \cup \vec{p}$, $\pi(b) = b^{\mathcal{I}}$ and, for every atom $P(\vec{t}) \in q$, $\pi(\vec{t}) \in P^{\mathcal{I}}$. The interpretation $\mathcal{I}$ satisfies the BCQ q, written $\mathcal{I} \models q$, if there is a match of q in $\mathcal{I}$. A BCQ is *entailed by* $\mathcal{O}$ if it is satisfied by every model of $\mathcal{O}$. For a BCQ q and an interpretation $\mathcal{I}$, we denote by $\nu_{\mathcal{I}}(q)$ the set of all matches of q in $\mathcal{I}$. The *provenance* of q on $\mathcal{I}$, denoted $\mathsf{prov}_{\mathcal{I}}(q)$, is the (potentially infinite) expression:

$$\sum_{\pi \in \nu_{\mathcal{I}}(q)} \prod_{P(\vec{t}) \in q} \mathsf{prov}_{\mathcal{I}}(P(\pi(\vec{t})))$$

where $\mathsf{prov}_{\mathcal{I}}(P(\pi(\vec{t})))$ is the last element of the tuple $\pi(\vec{t}) \in P^{\mathcal{I}}$. For $p \in \mathsf{N_P}$, we write $p \subseteq \mathsf{prov}_{\mathcal{I}}(q)$ if p is a sum of monomials and for each occurrence of a monomial in p we find an occurrence of it in $\mathsf{prov}_{\mathcal{I}}(q)$. We say that $\mathcal{I}$ satisfies q with provenance $p \in \mathsf{N_P}$, in symbols $\mathcal{I} \models (q, p)$, if $\mathcal{I} \models q$ and $p \subseteq \mathsf{prov}_{\mathcal{I}}(q)$. We say that a DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontology $\mathcal{O}$ *entails* q, in symbols $\mathcal{O} \models q$ if for all DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ interpretations $\mathcal{I}$, if $\mathcal{I} \models \mathcal{O}$ then $\mathcal{I} \models q$; and $\mathcal{O} \models (q, p)$ if $\mathcal{O} \models q$ and for all DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ interpretations $\mathcal{I}$ satisfying $\mathcal{O}$ we have that $p \subseteq \mathsf{prov}_{\mathcal{I}}(q)$.

We often write $|Y|$ to denote the size of a DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontology, a polynomial or a BCQ Y, defined as the length of the string that represents Y, where role and concept names are considered to be of length one.

2.3 Reasoning Problems

We consider the problem of *query entailment* w.r.t. a provenance polynomial, defined as follows: given an ontology $\mathcal{O}$ in an ontology language $\mathcal{L}$, a query q and

a polynomial $p \in \mathsf{N_P}$ we want to decide whether $\mathcal{O} \models (q, p)$. Another important and related question is how to compute the provenance of a query. We define the problem of computing the provenance of a query as follows: given an ontology $\mathcal{O}$ in an ontology language $\mathcal{L}$ and a query q, we want to compute the set of all $p \in \mathsf{N_P}$ such that $\mathcal{O} \models (q, p)$. In our formalism, this second problem depends on whether there is a finite set of polynomials which we can compute. The following example shows that in DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ the set of provenance polynomials can be infinite.

Example 4. Consider a DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ ontology $\mathcal{O}$ with the annotated assertions of Example 1, the annotated inclusion of Example 2 and $(\mathsf{Mayor} \sqsubseteq \exists \mathsf{headGov}.\top, t)$. Then, for all $n \in \mathbb{N}$, it holds that $\mathcal{O} \models (\mathsf{Mayor}(\mathsf{Renier}), p \otimes s^{n+1} \otimes t^n)$. Indeed, any model $\mathcal{I}$ of $\mathcal{O}$ satisfies the mentioned axioms, so $(\mathsf{Renier}, (p \otimes s)^{\mathcal{I}}) \in \mathsf{Mayor}^{\mathcal{I}}$ implies $(a, (p \otimes s \otimes t)^{\mathcal{I}}) \in (\exists \mathsf{headGov}.\top)^{\mathcal{I}}$, which implies $(\mathsf{Renier}, (p \otimes s^2 \otimes t)^{\mathcal{I}}) \in \mathsf{Mayor}^{\mathcal{I}}$, and so on.

In Sections 3 and 4 we consider the problem of query entailment w.r.t. a provenance polynomial. Notice that in Example 4, if the semiring is multiplicatively idempotent (i.e., $s \otimes s = s$), then the set of provenance polynomials is finite; indeed, the only polynomial will be $p \otimes s \otimes t$. This is not a coincidence; in fact, under multiplicative-idempotency, the set of provenance polynomials is always finite. The following proposition states that multiplicative-idempotency is not only necessary (as shown in Example 4) but also sufficient to guarantee that the set of polynomials is finite.

Proposition 5. *Under multiplicative-idempotency, for any satisfiable DL-Lite*$_{\mathcal{R}}^{\mathsf{P}}$ *ontology* $\mathcal{O}$ *and BCQ* q*, the set of* $p \in \mathsf{N_P}$ *such that* $\mathcal{O} \models (q, p)$ *is finite.*

Sketch. Under multiplicative-idempotency, for any DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ ontology $\mathcal{O}$ and BCQ q, the number of possible monomials occurring $p \in \mathsf{N_P}$ such that $\mathcal{O} \models (q, p)$ is finite. Thus, the only possibility for the set to be infinite is if these monomials can repeat an unlimited number of times. To entail such polynomials, the arbitrarily large number of repetitions should happen in all models of $\mathcal{O}$. However, under multiplicative-idempotency, DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ enjoys the finite domain property. That is, if a DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ ontology has a model then it has a model $\mathcal{I}$ with $\Delta^{\mathcal{I}}$ finite. □

In Section 5 we study the properties gained under idempotent semirings. In particular, we consider the problem of computing the provenance of a query.

3 Characterization

In this section, we show how to reduce the problem of query entailment w.r.t. a provenance polynomial to the query entailment problem in standard DLs for a particular class of provenance annotated DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ ontologies which we call *marked*. We use our results for marked ontologies, in Section 4, to solve the query answering problem for provenance annotated DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ ontologies in general. We say that a DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ ontology $\mathcal{O}_m$ is marked if there is a DL-Lite$_{\mathcal{R}}^{\mathsf{P}}$ ontology $\mathcal{O}$ such that $\mathcal{O}_m$ is the result of:

– replacing each $(R(a,b),p)$ by $(R_{a,b}(a,b),p)$, where $R_{a,b}$ is a fresh role name;
– for all $a,b \in \mathsf{N_I}$ and all $R \in \mathsf{N_R}$ occurring in $\mathcal{O}$, adding a concept inclusion $(\exists R^{(-)}_{a,b} \sqsubseteq C, v)$ for each $(\exists R^{(-)} \sqsubseteq C, u) \in \mathcal{O}$, where v is fresh;
– for all $a,b \in \mathsf{N_I}$ and all $R,S \in \mathsf{N_R}$ occurring in $\mathcal{O}$, adding a role inclusion $(R^{(-)}_{a,b} \sqsubseteq S^{(-)}_{a,b}, v)$ for each $(R^{(-)} \sqsubseteq S^{(-)}, u) \in \mathcal{O}$, where v is fresh.

Intuitively, we want to ensure that there is a model of $\mathcal{O}_m$ in which elements in the anonymous part (i.e., not in the image of $\mathsf{N_I}$) connected (via roles) to the image of an individual are associated with monomials containing at least one variable of the semiring which is not shared by anonymous elements connected to the image of another individual. In other words, we want to 'mark' monomials associated to elements derived from assertions of named individuals.

We now show that given a marked ontology $\mathcal{O}$, a BCQ q and a polynomial $p \in \mathsf{N_P}$, we can translate the query q and the polynomial p into a set of queries such that $\mathcal{O}$ entails (q,p) iff it entails at least one of the queries from this set. We first show how to translate a BCQ where all terms are variables (no individual names and no polynomials). For a BCQ $q = \exists \vec{x}\ \varphi(\vec{x})$ with m atoms and $p \in \mathsf{N_P}$ with n monomials, we define $\mathsf{Tr}(q,p)$ as the set of all BCQs:

$$\exists \vec{y} \bigwedge_{1 \leq i \leq n} \varphi_i(\vec{x_i}), \tag{1}$$

where $\vec{y} = \vec{x_1}, \ldots, \vec{x_n}$ and each $q_i = \exists \vec{x_i}\ \varphi_i(\vec{x_i})$ is a 'copy' of q in which we replace each variable $x \in \vec{x}$ by a fresh variable $x_i \in \vec{x_i}$. It remains to check whether we can find the monomials of the polynomial in these matches. We do this by replacing the last variable in each j-th atom of q_i by a monomial $p_{i,j}$ built from symbols in $X \subseteq \mathsf{N_P}$ occurring in p such that $\prod_{1 \leq j \leq m} p_{i,j} = p_i$ for some $p_i \in \mathsf{N_P}$, with $1 \leq i \leq n$; and $\sum_{1 \leq i \leq n} p_i = p$.

The translation of a BCQ with individual names is done in a similar manner, except that we also have to add such individual names in each copy of the query, that is, we would replace the corresponding variable in the translation with the individual name occurring in the query. Theorem 8 formalises the correctness of our translation. In the following, we write $\mathcal{O} \models \mathsf{Tr}(q,p)$ (resp. $\mathcal{I} \models \mathsf{Tr}(q,p)$) to express that there is $q' \in \mathsf{Tr}(q,p)$ such that $\mathcal{O} \models q'$ (resp. $\mathcal{I} \models q'$).

Example 6. Consider the query $q = \exists xyzw.(\mathsf{headGov}(x,y,z) \wedge \mathsf{City}(y,w))$ and the polynomial $p = (s \otimes t) \oplus (s \otimes r)$. Then,

$$\exists x_1 y_1 x_2 y_2.(\mathsf{headGov}(x_1,y_1,s) \wedge \mathsf{City}(y_1,t) \wedge \mathsf{headGov}(x_2,y_2,s) \wedge \mathsf{City}(y_2,r))$$

is in $\mathsf{Tr}(q,p)$.

To show Theorem 8 we use the following technical lemma. whose proof uses the construction of the canonical model and the assumption that the DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontology is in marked form.

Lemma 7. *Let $\mathcal{O}$ be a satisfiable DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ marked ontology, q a BCQ and p a polynomial in $\mathsf{N_P}$. If $\mathcal{O} \models (q,p)$ then for any two monomials p_1, p_2 appearing in p, it holds that p_1 and p_2 are mathematically distinct.*

With the help of this lemma, we can state the main result of this section; namely, that the translation for marked ontologies is correct.

Theorem 8. *Let $\mathcal{O}$ be a DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontology, q a BCQ and $p \in \mathsf{N_P}$ a polynomial formed of mathematically distinct monomials. Then,*

$$\mathcal{O} \models (q, p) \text{ if, and only if, } \mathcal{O} \models \mathsf{Tr}(q, p).$$

In the next section, we use our characterization for marked ontologies to present a query rewriting method for answering provenance queries.

4 Query Rewriting

We now show how to adapt the classical query rewriting algorithm PerfectRef [7] to deal with provenance annotated ontologies and queries in DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$. We use our characterization in Section 3 for marked ontologies and restrict to the case where BCQs q are such that, for all $P(\vec{t}) \in q$, the last element of $\vec{t}$ is a monomial in $\mathsf{N_P}$ (i.e., variables do not occur in the last parameter of an atom). In Theorem 11 we explain how we solve the problem of query entailment w.r.t. a provenance polynomial for arbitrary DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontologies and annotated queries. Whenever possible, we use the same definitions and terminology in [7, Section 5.1], adapting some of them to our setting, when necessary.

In the following, the symbol "$\_$" represents non-distinguished non-shared variables. A positive inclusion I is a provenance annotated role or concept inclusion without negations. We say that I is applicable to an atom $A(x, p)$ if I is annotated with v occurring in p and it has A in its right-hand side. A positive inclusion I is applicable to an atom $R(x, y, p)$ if (i) $x = \_$, I is annotated with v occurring in p, and the right-hand side of I is $\exists R$, or (ii) I is a role inclusion annotated with v occurring in p and its right-hand side is either R or R^-.

To simplify the presentation, for each role R^- occurring in a marked DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontology $\mathcal{O}$, we add to $\mathcal{O}$ the annotated role inclusions $(R^- \sqsubseteq \overline{R}, p_R)$ and $(\overline{R} \sqsubseteq R^-, p'_R)$, where $\overline{R}$ is a fresh role name and p_R, p'_R are fresh variables of a provenance semiring. We can then assume w.l.o.g. that inverse roles only occur in such role inclusions by replacing any other occurrence of R^- with $\overline{R}$. Given a monomial $p \in \mathsf{N_P}$ and a variable $v \in X \subseteq \mathsf{N_P}$ of the semiring occurring in p, we denote by $p_{|v}$ the result of removing one occurrence of v from p.

Definition 9. *Let g be an atom and I be a positive inclusion that is applicable to g. Then, the atom obtained from g by applying I, denoted by $gr(g, I)$, is:*

- $gr(g, I) = A_1(x, p_{|v})$, *if* $g = A(x, p)$ *and* $I = (A_1 \sqsubseteq A, v)$;
- $gr(g, I) = R(x, \_, p_{|v})$, *if* $g = A(x, p)$ *and* $I = (\exists R \sqsubseteq A, v)$;
- $gr(g, I) = A(x, p_{|v})$, *if* $g = R(x, \_, p)$ *and* $I = (A \sqsubseteq \exists R, v)$;
- $gr(g, I) = R_1(x, \_, p_{|v})$, *if* $g = R(x, \_, p)$ *and* $I = (\exists R_1 \sqsubseteq \exists R, v)$;
- $gr(g, I) = R_1(x, y, p_{|v})$, *if* $g = R(x, y, p)$ *and* $I = (R_1 \sqsubseteq R, v)$;
- $gr(g, I) = R_1(y, x, p_{|v})$, *if* $g = R(x, y, p)$ *and either* $I = (R_1 \sqsubseteq R^-, v)$ *or* $I = (R_1^- \sqsubseteq R, v)$.

Algorithm 1 Algorithm PerfectRef

Input: a BCQ q, a set of positive inclusions $\mathcal{O}_\mathcal{T}$
Output: a union of BCQs PR

1: $PR := \{q\}$
2: **repeat**
3: $\quad PR' := PR$
4: $\quad$ **for all** $q \in PR'$ **do**
5: $\quad\quad$ **for all** $g \in q$ **do**
6: $\quad\quad\quad$ **for all** $I \in \mathcal{O}_\mathcal{T}$ **do**
7: $\quad\quad\quad\quad$ **if** I is applicable to g **then**
8: $\quad\quad\quad\quad\quad PR := PR \cup \{q[g/gr(g, I)]\}$
9: $\quad\quad$ **for all** $g_1, g_2 \in q$ **do**
10: $\quad\quad\quad$ **if** g_1 and g_2 unify **then**
11: $\quad\quad\quad\quad PR := PR \cup \{\tau(\mathsf{reduce}(q, g_1, g_2))\}$
12: **until** $PR' = PR$
13: **return** PR

We use the same algorithm PerfectRef (Algorithm 1) originally presented in [7], except that the notion of applicability of a positive inclusion I to an atom g is as previously described and $gr(g, I)$ is as in Definition 9. Let $q[g/g']$ denote the BCQ obtained from q by replacing the atom g with a new atom g'. Let τ be a function that takes as input a BCQ q and returns a new BCQ obtained by replacing each occurrence of an unbound variable in q with the symbol '_'; and let reduce be a function that takes as input a BCQ q and two atoms g_1, g_2 and returns a BCQ obtained by applying to q the most general unifier between g_1 and g_2. We denote by $\mathsf{PerfectRef}(q, \mathcal{O}_\mathcal{T})$ the output of the algorithm PerfectRef with a BCQ q (with a monomial in $\mathsf{N_P}$ in the last parameter of each atom) and a set $\mathcal{O}_\mathcal{T}$ of positive inclusions of a marked DL-Lite$^\mathsf{P}_\mathcal{R}$ ontology $\mathcal{O}$ as input.

Example 10. Consider a DL-Lite$^\mathsf{P}_\mathcal{R}$ ontology $\mathcal{O}$ containing the annotated assertions of Example 1 and the annotated inclusion I of Example 2. Assume that Algorithm 1 receives $\mathcal{O}_\mathcal{T}$ and $q = \exists x.\mathsf{Mayor}(x, p \otimes s)$ as input. Since I is applicable to $g = \mathsf{Mayor}(x, p \otimes s)$, in Line 8, Algorithm 1 adds to PR the result of replacing g by $gr(g, I) = \mathsf{headGov}(x, \_, p)$ in q. Hence we get that

$$q^\dagger = \exists x, y\ \mathsf{headGov}(x, y, p) \in \mathsf{PerfectRef}(q, \mathcal{O}_\mathcal{T}).$$

Indeed $q^\dagger$ is a rewriting of q.

Termination of our modified version of PerfectRef follows the same lines as [7, Lemma 34], except that now the number of terms is exponential in the size of monomials occurring in the query, and thus, in the size of the query. This is due to Definition 9, where we 'break' the monomial into a smaller one. Our modification does not change the upper bounds obtained with the algorithm, since in data complexity only the data is considered as part of the input (so the query is fixed) and the upper bound for combined complexity, which we establish in Theorem 11, is obtained by a non-deterministic version of the algorithm.

Theorem 11. *In DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$, answering provenance annotated queries is* NP*-complete (combined complexity).*

The following theorem states the complexity of answering provenance annotated queries in DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ when (i) only provenance annotated role and concept inclusions are considered as the input, and (ii) only the assertions are the input.

Theorem 12. *In DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$, answering provenance annotated queries is (i) in* PTime *in the size of the role and concept inclusions, and (ii) in* LogSpace *in the size of the assertions (data complexity).*

Proof. These complexity results follow from the fact that: (1) satisfiability of DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ is in LogSpace, as in DL-Lite$_{\mathcal{R}}$ [2]; (2) the query rewriting can be computed in time polynomial in the number of positive inclusions in a DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontology $\mathcal{O}$ and in constant time in the number of assertions in $\mathcal{O}$; (3) (our modified version of) PerfectRef is correct [7] (see Theorem 11); and (4) evaluation of a query over a database can be computed in LogSpace w.r.t. the size of the database (follows from the fact that queries are a fragment of first-order logic and their size is fixed for the complexity results in this theorem). □

5 Idempotency

We now turn our attention to a special case, where the semiring describing the provenance of knowledge is multiplicatively-idempotent; that is, the operation $\otimes$ is such that for every polynomial $p \in \mathsf{N_P}$, $p \otimes p = p$. Such would be the case, for example, if the provenance refers to the name of the source of the knowledge; then, having several times the same name does not affect the result. Alternatively, one can consider access rights to observe certain pieces of knowledge from the ontology. For the scope of this section, we remove the assumption that the labels of each of the axioms are unique. That is, while every axiom is still labelled with one variable from the provenance space, several axioms may share the same label. This is consistent, again, with the idea of the labels representing the source or the accessibility level of a piece of knowledge.

Consider first the problem of computing the provenance of a query, as described in Section 2. Notice, however, that whenever the semiring is fully idempotent (that is, $p \oplus p = p$ also holds), the task can be simplified to the computation of relevant monomials. More precisely, in this special case we are interested in computing the set of all monomials p such that $\mathcal{O} \models (q, p)$. Clearly, the polynomial of the query is the result of adding all these monomials. This definition is equivalent to the general one since the semiring is idempotent: repetitions of the monomials do not affect the result, and repetitions of a variable in a given monomial can be removed. If the semiring is only multiplicatively idempotent, then computing monomials does not suffice, since some of them may need to appear several times. However, the problem is still simplified to find the (finite) number of repeated monomials to be observed.

The first thing that we notice is that, in general, we cannot avoid an exponential runtime of any method computing this polynomial, since it may require the addition of exponentially many monomials, as shown in the following example.

Example 13. Consider the DL-Lite$^{\mathsf{P}}_{\mathcal{R}}$ ontology $\mathcal{O}$ containing the axioms:

$$\begin{array}{ccccc} (A \sqsubseteq B_1, x), & (A \sqsubseteq C_1, x), & (B_n \sqsubseteq D, x), & (C_n, \sqsubseteq D, x), & (A(a), p), \\ (B_i \sqsubseteq B_{i+1}, x_i), & (B_i \sqsubseteq C_{i+1}, y_i), & (C_i \sqsubseteq B_{i+1}, x_i), & (C_i \sqsubseteq C_{i+1}, y_i), & 1 \leq i < n, \end{array}$$

and the simple query $q = D(a)$. Notice that every monomial $p = x \otimes \prod_{i=1}^{n} z_i$, where each $z_i \in \{x_i, y_i\}$ is such that $\mathcal{O} \models (q, p)$ (and none other). Hence, the polynomial of the query q is formed by the sum of 2^n different monomials; that is, it is exponential on the size of the ontology.

Perhaps more interesting, though, is that the provenance of some queries cannot be expressed through a provenance polynomial whose length is polynomial on the size of the ontology, even if we allow the expression to not be in expanded form. This is, in fact, a direct consequence of the results by Karchmer and Wigderson on monotone complexity [11, 12]. In a nutshell, Karchmer and Wigderson show that there is no monotone Boolean formula of polynomial length that can express all the paths between two nodes in a graph.[1] In fact, the result holds already for the special case of a *complete* graph. As hinted in Example 13, graphs can be described in DL-Lite$_{\mathcal{R}}$ (and in even simpler logics) using basic inclusion axioms. Moreover, monotone Boolean formulas can be seen as a provenance polynomial over an idempotent semiring; indeed, the conjunction $\wedge$ serves as product and $\vee$ as the addition in this algebra, and both operators are idempotent. Hence we have the following result (see also [14]).

Proposition 14. *There exists a DL-Lite$_{\mathcal{R}}$ ontology $\mathcal{O}$ and a query q such that the provenance polynomial of q w.r.t. $\mathcal{O}$ cannot be represented in polynomial space. The result holds even if the semiring is idempotent, and every axiom in $\mathcal{O}$ has a unique label.*

Proof. Let N be a set with n concept names such that $A, B \in N$. Define the ontology

$$\mathcal{O} = \{(C \sqsubseteq D, x_{CD}) \mid C, D \in N\} \cup \{A(a), x_A\},$$

and the query $q = B(a)$. Notice that $\mathcal{O}$ simulates a complete graph over the nodes in N. Every derivation of $B(a)$ represents a path from A to B in this graph. Hence, if the provenance of q could be expressed with polynomial space, there would be a monotone Boolean formula representing all the paths from A to B in the complete graph with n nodes, contradicting the results from [11, 12]. □

If in addition we assume as in the previous sections that every axiom has a unique variable as a provenance label, then it is possible to show that the provenance polynomial for *instance queries* can be computed efficiently, whenever

[1] A *monotone Boolean formula* is a propositional formula built using only the connectives $\wedge$ and $\vee$ (without negations).

its length does not increase greatly; that is, it can be computed in polynomial time on the size of the input *and the output*. The proof of the following lemma follows the same ideas presented in [15,16], based on the fact that all the relevant monomials that form the provenance can be enumerated with polynomial delay.

Lemma 15. *The provenance p for an instance query w.r.t. the ontology $\mathcal{O}$ can be computed in polynomial time on the size of $\mathcal{O}$* and the polynomial p.

All the results presented in this section refer to the complexity obtained from considering the whole ontology as an input. In fact, more precise complexity analyses based on fixing some of the parameters are left for future work. Moreover, most of the queries used were simple instance or Boolean queries. In particular, this means that all the hardness results can be transferred directly to more complex (conjunctive) queries. On the other hand, it is worth noting that the upper bounds developed in the previous section for the general case, obviously apply also in the restricted setting of idempotency.

6 Conclusions

In this paper, we have studied the problem of provenance in an OBDA scenario where the data and the ontological axioms are annotated with provenance information. In particular, we have shown that query rewriting techniques developed for the classical OBDA setting can be adapted to handle provenance as well. Using this argument, we were able to study the complexity of finding the provenance of a query. We also studied special cases in which the operators from the provenance semiring are idempotent.

It is important to emphasise that, despite some apparent similarities, the problem of provenance is very different from that of axiom pinpointing [6,10,17]. Although both in problems we are interested in tracing the causes of a consequence, axiom pinpointing focuses on those that use *minimal* sets of axioms. In contrast, all possible derivations are relevant for provenance, independently of whether a subset of axioms might already suffice. Hence, although the set of monotone Boolean formulas is a semiring, the provenance polynomial over this semiring does not necessarily coincide with the pinpointing formula [4,5]. The main reason for this difference is that the provenance label represents a key which can be used to encode other information or values, such as trust or probability; hence each derivation contributes to the final value represented by the polynomial.

As future work, we plan to study the properties of provenance-based OBDA in more detail. In particular, we are interested in understanding how the choice of the semiring affects the computational complexity of answering different problems, and obtain a more fine-grained analysis of the complexity results. We also plan to investigate our approach for dealing with provenance information in more expressive logics.

References

1. OWL 2 Web Ontology Language Profiles (Second Edition). Technical report, W3C, 2012.
2. Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *Journal of artificial intelligence research*, 36(1):1–69, 2009.
3. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, second edition, 2007.
4. Franz Baader and Rafael Peñaloza. Automata-based axiom pinpointing. *J. Autom. Reasoning*, 45(2):91–129, 2010.
5. Franz Baader and Rafael Peñaloza. Axiom pinpointing in general tableaux. *J. Log. Comput.*, 20(1):5–34, 2010.
6. Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *KI*, pages 52–67, 2007.
7. Diego Calvanese, Guiseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
8. Todd J. Green, Gregory Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, pages 31–40, 2007.
9. Stijn Heymans, Li Ma, Darko Anicic, Zhilei Ma, Nathalie Steinmetz, Yue Pan, Jing Mei, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, Edith Schonberg, Kavitha Srinivas, Cristina Feier, Graham Hench, Branimir Wetzstein, and Uwe Keller. Ontology reasoning with large data repositories. In *Ontology Management*, 2008.
10. Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all justifications of OWL DL entailments. In *ISWC*, pages 267–280, 2007.
11. M. Karchmer and A. Wigderson. Monotone circuits for connectivity require superlogarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.
12. Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 539–550, New York, NY, USA, 1988. ACM.
13. Roman Kontchakov, Mariano Rodríguez-Muro, and Michael Zakharyaschev. Ontology-based data access with databases: A short course. In *Proceedings of the 9th International Conference on Reasoning Web: Semantic Technologies for Intelligent Data Access*, RW'13, pages 194–229, 2013.
14. Rafael Peñaloza. *Axiom pinpointing in description logics and beyond*. PhD thesis, Dresden University of Technology, 2009.
15. Rafael Peñaloza. Inconsistency-tolerant instance checking in tractable description logics. In Stefania Costantini, Enrico Franconi, William Van Woensel, Roman Kontchakov, Fariba Sadri, and Dumitru Roman, editors, *Proceedings of the International Joint Conference on Rules and Reasoning (RuleML+RR 2017)*, volume 10364 of *Lecture Notes in Computer Science*, pages 215–229. Springer, 2017. doi: `https://doi.org/10.1007/978-3-319-61252-2_15`.
16. Rafael Peñaloza and Barış Sertkaya. Understanding the complexity of axiom pinpointing in lightweight description logics. *Artificial Intelligence*, 250:80–104, September 2017. doi: https://doi.org/10.1016/j.artint.2017.06.002.

17. Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 355–360. Morgan Kaufmann Publishers Inc., 2003.
18. Pierre Senellart. Provenance and probabilities in relational databases. *SIGMOD Record*, 46(4):5–15, 2017.