Provenance for the Description Logic $\mathcal{ELH}^r$

Camille Bourgaux[1] , **Ana Ozaki**[2,4] , **Rafael Peñaloza**[3] and **Livia Predoiu**[4]
[1]DI ENS, ENS, CNRS, PSL University & Inria, Paris, France
[2]University of Bergen, Norway
[3]University of Milano-Bicocca, Italy
[4]Free University of Bozen-Bolzano, Italy
camille.bourgaux@ens.fr, ana.ozaki@uib.no, rafael.penaloza@unimib.it, livia.predoiu@unibz.it

Abstract

We address the problem of handling provenance information in $\mathcal{ELH}^r$ ontologies. We consider a setting recently introduced for ontology-based data access, based on semirings and extending classical data provenance, in which ontology axioms are annotated with provenance tokens. A consequence inherits the provenance of the axioms involved in deriving it, yielding a provenance polynomial as annotation. We analyse the semantics for the $\mathcal{ELH}^r$ case and show that the presence of conjunctions poses various difficulties for handling provenance, some of which are mitigated by assuming multiplicative idempotency of the semiring. Under this assumption, we study three problems: ontology completion with provenance, computing the set of relevant axioms for a consequence, and query answering.

1 Introduction

Description logics (DLs) are a well-known family of first-order logic fragments in which conceptual knowledge about a particular domain and facts about specific individuals are expressed in an ontology, using unary and binary predicates called *concepts* and *roles* [Baader *et al.*, 2007a]. Important reasoning tasks performed over DL ontologies are axiom entailment, i.e. deciding whether a given DL axiom follows from the ontology; and query answering. Since scalability is crucial when using large ontologies, DLs with favorable computational properties have been investigated. In particular, the $\mathcal{EL}$ language and some of its extensions allow for axiom entailment in polynomial time, and conjunctive query entailment in NP [Baader *et al.*, 2005; Baader *et al.*, 2008a]. Many real-world ontologies, including SNOMED CT, use languages from the $\mathcal{EL}$ family, which underlies the OWL 2 EL profile of the Semantic Web standard ontology language.

In many settings it is crucial to know *how* a consequence—e.g. an axiom or a query—has been derived from the ontology. In the database community, provenance has been studied for nearly 30 years [Buneman, 2013] and gained traction when the connection to semirings, so called *provenance semirings* [Green *et al.*, 2007; Green and Tannen, 2017] was discovered. *Provenance semirings* serve as an abstract algebraic tool to record and track provenance information; that is, to keep track of the specific database tuples used for deriving the query, and of the way they have been processed in the derivation. Besides explaining a query answer, provenance has many applications like: computing the probability or the degree of confidence of an answer, counting the different ways of producing an answer, handling authorship, data clearance, or user preferences [Senellart, 2017; Suciu *et al.*, 2011; Lukasiewicz *et al.*, 2014; Ives *et al.*, 2008]. Semiring provenance has drawn interest beyond relational databases (e.g. [Buneman and Kostylev, 2010; Zimmermann *et al.*, 2012; Deutch *et al.*, 2014; Ramusat *et al.*, 2018; Dannert and Grädel, 2019]), and in particular has recently been considered for ontology-based data access, a setting where a database is enriched with a DL-Lite$_\mathcal{R}$ ontology and mappings between them [Calvanese *et al.*, 2019]. In the latter, the ontology axioms are annotated with *provenance variables*. Queries are then annotated with *provenance polynomials* that express their provenance information.

Example 1. *Consider the facts* $\mathsf{mayor}(\mathsf{Venice}, \mathsf{Brugnaro})$ *and* $\mathsf{mayor}(\mathsf{Venice}, \mathsf{Orsoni})$*, stating that Venice has mayors Brugnaro and Orsoni, annotated respectively with provenance information* v_1 *and* v_2*, and the DL axiom* $\mathsf{ran}(\mathsf{mayor}) \sqsubseteq \mathsf{Mayor}$*, expressing that the range of the role* mayor *is the concept* Mayor*, annotated with* v_3*. The query* $\exists x.\mathsf{Mayor}(x)$ *asks if there is someone who is a mayor. The answer is* yes *and it can be derived using* $\mathsf{ran}(\mathsf{mayor}) \sqsubseteq \mathsf{Mayor}$ *together with any of the two facts, interpreting* x *by* $\mathsf{Brugnaro}$ *or* Orsoni*. This is expressed by the provenance polynomial* $v_1 \times v_3 + v_2 \times v_3$*. Intuitively,* $\times$ *expresses the joint use of axioms in a derivation path of the query, and* $+$ *the alternative derivations.*

We adapt the provenance semantics of Calvanese *et al.* for the $\mathcal{ELH}^r$ variant of $\mathcal{EL}$, extending it to those $\mathcal{ELH}^r$ axioms that do not occur in DL-Lite$_\mathcal{R}$. It turns out that handling the conjunction allowed in $\mathcal{ELH}^r$ axioms is not trivial. To obtain models from which we can derive meaningful provenance-annotated consequences, we adopt $\times$-idempotent semirings and a syntactic restriction on $\mathcal{ELH}^r$ (preserving the expressivity of full $\mathcal{ELH}^r$ when annotations are not considered). After introducing the basic definitions and the semantics for DL ontologies and queries annotated with provenance information, we present a completion algorithm and show that it solves annotated axiom entailment and instance queries in $\mathcal{ELH}^r$ in polynomial time in the size of the ontology and polynomial space in the size of the provenance polynomial. We then show that we can compute the set of relevant provenance variables for an entailment in polynomial time. Finally, we investigate conjunctive query answering. Note that the query answering methods developed by Calvanese *et al.* cannot be extended to $\mathcal{ELH}^r$ since they rely on the FO-rewritability of conjunctive queries in DL-Lite$_\mathcal{R}$, a property that does not hold for $\mathcal{ELH}^r$ [Bienvenu *et al.*, 2013]. Therefore, we adapt the combined approach for query answering in $\mathcal{EL}$ [Lutz *et al.*, 2009] to provenance-annotated $\mathcal{ELH}^r$ ontologies.

Detailed proofs are available in [Bourgaux *et al.*, 2020].

2 Provenance for $\mathcal{ELH}^r$

We first introduce our framework for provenance for $\mathcal{ELH}^r$ ontology and discuss our design choices.

2.1 Basic Notions

In the database setting, commutative semirings have proven to be convenient for representing various kinds of provenance information [Green *et al.*, 2007; Green and Tannen, 2017]. In a commutative semiring $(K, +, \times, 0, 1)$, the product $\times$ and the addition $+$ are commutative and associative binary operators over K, and $\times$ distributes over $+$. Given a countably infinite set $\mathsf{N_V}$ of *variables* that are used to annotate the database tuples, a *provenance semiring* is a semiring over a space of *annotations*, or provenance expressions, with variables from $\mathsf{N_V}$. Green and Tannen present a hierarchy of expressiveness for provenance annotations [2017]. The most expressive form of annotations is provided by the *provenance polynomials* semiring $\mathbb{N}[\mathsf{N_V}] = (\mathbb{N}[\mathsf{N_V}], +, \times, 0, 1)$ of polynomials with coefficients from $\mathbb{N}$ and variables from $\mathsf{N_V}$, and the usual operations. The semiring $\mathbb{N}[\mathsf{N_V}]$ is universal, i.e., for any other commutative semiring $K = (K, +, \times, 0, 1)$, any function $\nu : \mathsf{N_V} \to K$ can be extended to a semiring homomorphism $h : \mathbb{N}[\mathsf{N_V}] \to K$, allowing the computations for K to factor through the computations for $\mathbb{N}[\mathsf{N_V}]$ [Green, 2011]. Hence, provenance polynomials provide the most informative provenance annotations and correspond to so-called *how-provenance* [Cheney *et al.*, 2009]. Less general provenance semirings are obtained by restricting the operations $+$ and $\times$ to idempotence and/or absorption [Green and Tannen, 2017]. In this work, we focus on $\times$-idempotent semirings, i.e., for every $v \in K$, $v \times v = v$. This corresponds to the *Trio semiring* $\mathsf{Trio}(\mathsf{N_V})$, defined in [Green, 2011] as the quotient semiring of $\mathbb{N}[\mathsf{N_V}]$ by the equivalence kernel $\approx_\mathsf{trio}$ of the function $\mathsf{trio} : \mathbb{N}[\mathsf{N_V}] \to \mathbb{N}[\mathsf{N_V}]$ that "drops exponents." An annotation is a polynomial p that is understood to represent its equivalence class $p/{\approx_\mathsf{trio}}$. $\mathsf{Trio}(\mathsf{N_V})$ encompasses in the hierarchy the well-known *why-provenance semiring* $\mathsf{Why}(\mathsf{N_V})$ obtained by restricting $+$ to be idempotent as well, where an annotation corresponds to the set of sets of tuples used to derive the result [Cheney *et al.*, 2009].

We use the following notation. A *monomial* is a finite product of variables in $\mathsf{N_V}$. Let $\mathsf{N_M}$ be the set of monomials, and $\mathsf{N_P}$ the set of all finite sums of monomials, i.e., $\mathsf{N_P}$ contains polynomials of the form $\sum_{1 \leq i \leq n} \prod_{1 \leq j_i \leq m_i} v_{i,j_i}$, with $v_{i,j_i} \in \mathsf{N_V}$; $n, m_i > 0$. By distributivity, every polynomial can be written into this form. The *representative* $[m]$ of a monomial m is the product of the variables occuring in m, in lexicographic order. Two monomials which are equivalent w.r.t. $\approx_\mathsf{trio}$ (i.e. are syntactically equal modulo commutativity, associativity and $\times$-idempotency) have the same representative, e.g., $v \times u$ and $u \times v \times u$ have representative $u \times v$. $\mathsf{N_{[M]}}$ denotes the set $\{[m] \mid m \in \mathsf{N_M}\}$.

As ontology language we use a syntactic restriction of $\mathcal{ELH}^r$. Consider three mutually disjoint countable sets of *concept-* $\mathsf{N_C}$, *role-* $\mathsf{N_R}$, and *individual names* $\mathsf{N_I}$, disjoint from $\mathsf{N_V}$. $\mathcal{ELH}^r$ *general concept inclusions* (GCIs) are expressions of the form $C \sqsubseteq D$, built according to the grammar rules

$$C ::= A \mid \exists R.C \mid C \sqcap C \mid \top \qquad\qquad D ::= A \mid \exists R,$$

where $R \in \mathsf{N_R}$, $A \in \mathsf{N_C}$. *Role inclusions* (RIs) and *range restrictions* (RRs) are expressions of the form $R \sqsubseteq S$ and $\mathsf{ran}(R) \sqsubseteq A$, respectively, with $R, S \in \mathsf{N_R}$ and $A \in \mathsf{N_C}$. An *assertion* is an expression of the form $A(a)$ or $R(a, b)$, with $A \in \mathsf{N_C}$, $R \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. An *axiom* is a GCI, RI, RR, or assertion. An $\mathcal{ELH}^r$ ontology is a finite set of $\mathcal{ELH}^r$ axioms. $\mathcal{ELH}^r$ usually allows GCIs of the form $C \sqsubseteq C$, but these can be translated into our format by exhaustively applying the rules: (i) replace $C \sqsubseteq C_1 \sqcap C_2$ by $C \sqsubseteq C_1$ and $C \sqsubseteq C_2$, (ii) replace $C_1 \sqsubseteq \exists R.C_2$ by $C_1 \sqsubseteq \exists S$, $S \sqsubseteq R$ and $\mathsf{ran}(S) \sqsubseteq C_2$ where S is a fresh role name. The reason for syntactically restricting $\mathcal{ELH}^r$ is that conjunctions or qualified restrictions of a role on the right-hand side of GCIs lead to counter-intuitive behavior when adding provenance annotations. We discuss this later in this section.

2.2 Annotated Ontologies

Provenance information is stored as annotations. An *annotated axiom* has the form (α, m) with α an axiom and $m \in \mathsf{N_M}$. An *annotated* $\mathcal{ELH}^r$ *ontology* $\mathcal{O}$ is a finite set of annotated $\mathcal{ELH}^r$ axioms of the form (α, v) with $v \in \mathsf{N_V} \cup \{1\}$. We denote by $\mathsf{ind}(\mathcal{O})$ the set of individual names occurring in $\mathcal{O}$.

The semantics of annotated ontologies extends the classical notion of interpretations to track provenance. An *annotated interpretation* is a triple $\mathcal{I} = (\Delta^\mathcal{I}, \Delta^\mathcal{I}_\mathsf{m}, \cdot^\mathcal{I})$ where $\Delta^\mathcal{I}, \Delta^\mathcal{I}_\mathsf{m}$ are non-empty disjoint sets (the *domain* and *domain of monomials* of $\mathcal{I}$, respectively), and $\cdot^\mathcal{I}$ maps

- every $a \in \mathsf{N_I}$ to $a^\mathcal{I} \in \Delta^\mathcal{I}$;
- every $A \in \mathsf{N_C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}_\mathsf{m}$;
- every $R \in \mathsf{N_R}$ to $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I} \times \Delta^\mathcal{I}_\mathsf{m}$; and
- every $m, n \in \mathsf{N_M}$ to $m^\mathcal{I}, n^\mathcal{I} \in \Delta^\mathcal{I}_\mathsf{m}$ s.t. $m^\mathcal{I} = n^\mathcal{I}$ iff $m \approx_\mathsf{trio} n$.[1]

We extend $\cdot^\mathcal{I}$ to complex $\mathcal{ELH}^r$ expressions as usual:

$$\begin{aligned}
(\top)^\mathcal{I} &= \Delta^\mathcal{I} \times \{1^\mathcal{I}\};\\
(\exists R)^\mathcal{I} &= \{(d, m^\mathcal{I}) \mid \exists e \in \Delta^\mathcal{I} \text{ s.t. } (d, e, m^\mathcal{I}) \in R^\mathcal{I}\};\\
(C \sqcap D)^\mathcal{I} &= \{(d, (m \times n)^\mathcal{I}) \mid (d, m^\mathcal{I}) \in C^\mathcal{I}, (d, n^\mathcal{I}) \in D^\mathcal{I}\};\\
(\mathsf{ran}(R))^\mathcal{I} &= \{(e, m^\mathcal{I}) \mid \exists d \in \Delta^\mathcal{I} \text{ s.t. } (d, e, m^\mathcal{I}) \in R^\mathcal{I}\};\\
(\exists R.C)^\mathcal{I} &= \{(d, (m \times n)^\mathcal{I}) \mid \exists e \in \Delta^\mathcal{I} \text{ s.t.}\\
&\qquad (d, e, m^\mathcal{I}) \in R^\mathcal{I}, (e, n^\mathcal{I}) \in C^\mathcal{I}\}.
\end{aligned}$$

The annotated interpretation $\mathcal{I}$ *satisfies*:

[1] or iff $m = n$ if we consider $\mathbb{N}[\mathsf{N_V}]$ instead of $\mathsf{Trio}(\mathsf{N_V})$

$(R \sqsubseteq S, m)$, if, for all $n \in \mathsf{N_M}, (d, e, n^{\mathcal{I}}) \in R^{\mathcal{I}}$ implies $(d, e, (m \times n)^{\mathcal{I}}) \in S^{\mathcal{I}}$;
$(C \sqsubseteq D, m)$, if, for all $n \in \mathsf{N_M}, (d, n^{\mathcal{I}}) \in C^{\mathcal{I}}$ implies $(d, (m \times n)^{\mathcal{I}}) \in D^{\mathcal{I}}$;
$(A(a), m)$, if $(a^{\mathcal{I}}, m^{\mathcal{I}}) \in A^{\mathcal{I}}$; and
$(R(a, b), m)$, if $(a^{\mathcal{I}}, b^{\mathcal{I}}, m^{\mathcal{I}}) \in R^{\mathcal{I}}$.

$\mathcal{I}$ is a model of the annotated ontology $\mathcal{O}$, denoted $\mathcal{I} \models \mathcal{O}$, if it satisfies all annotated axioms in $\mathcal{O}$. $\mathcal{O}$ *entails* (α, m), denoted $\mathcal{O} \models (\alpha, m)$, if $\mathcal{I} \models (\alpha, m)$ for every model $\mathcal{I}$ of $\mathcal{O}$.

Remark. *While it may appear counter-intuitive at first sight that $C^{\mathcal{I}}$ differs from $(C \sqcap C)^{\mathcal{I}}$, this is in line with the intuition behind the provenance of a conjunction. In the database setting, the Trio-provenance of tuple (a) being an answer to query $\exists yz.R(x, y) \wedge R(x, z)$ over $\{R(a, b), R(a, c)\}$ is also different from that of (a) being an answer to $\exists y.R(x, y)$.*

Example 2 illustrates the semantics and some differences with the DL-Lite$_{\mathcal{R}}$ case from [Calvanese *et al.*, 2019].

Example 2. *Consider the following annotated ontology.*

$$\begin{aligned}\mathcal{O} = \{&(\mathsf{mayor}(\mathsf{Venice}, \mathsf{Orsoni}), v_1),\\ &(\mathsf{predecessor}(\mathsf{Brugnaro}, \mathsf{Orsoni}), v_2),\\ &(\exists \mathsf{predecessor}.\mathsf{Mayor} \sqsubseteq \mathsf{Mayor}, v_3),\\ &(\mathsf{ran}(\mathsf{mayor}) \sqsubseteq \mathsf{Mayor}, v_4)\}.\end{aligned}$$

Let $\mathcal{I}$ be s.t. $\Delta^{\mathcal{I}} = \{\mathsf{Brugnaro}, \mathsf{Orsoni}, \mathsf{Venice}\}$, $\Delta^{\mathcal{I}}_{\mathsf{m}} = \mathsf{N_{[M]}}$, individual names are interpreted by themselves, monomials by their representatives and

$$\begin{aligned}\mathsf{mayor}^{\mathcal{I}} &= \{(\mathsf{Venice}, \mathsf{Orsoni}, v_1)\},\\ \mathsf{predecessor}^{\mathcal{I}} &= \{(\mathsf{Brugnaro}, \mathsf{Orsoni}, v_2)\},\\ \mathsf{Mayor}^{\mathcal{I}} &= \{(\mathsf{Orsoni}, v_1 \times v_4),\\ &\quad (\mathsf{Brugnaro}, v_1 \times v_2 \times v_3 \times v_4)\}.\end{aligned}$$

$\mathcal{I} \models \mathcal{O}$ by the semantics of annotated $\mathcal{ELH}^r$. Moreover, it can be verified that if $\mathcal{I} \models (\alpha, m)$, then $\mathcal{O} \models (\alpha, m)$. Note that $\mathcal{O}$ entails $(\mathsf{Mayor}(\mathsf{Brugnaro}), v_1 \times v_2 \times v_3 \times v_4)$ *whose provenance monomial contains v_1 and v_2, witnessing that the two assertions of $\mathcal{O}$ have been used to derive* $\mathsf{Mayor}(\mathsf{Brugnaro})$. *Combining two assertions to derive another one is not possible in DL-Lite$_{\mathcal{R}}$. The rewriting-based approach by Calvanese* et al. *cannot be applied here as* $\exists\mathsf{predecessor}.\mathsf{Mayor} \sqsubseteq \mathsf{Mayor}$ *leads to infinitely many rewritings.*

2.3 Discussion on Framework Restrictions

Example 2 shows that conjunction and qualified role restriction lead to a behavior different from DL-Lite$_{\mathcal{R}}$. They are also the reason for some features of our setting. First, the next example illustrates the $\times$-idempotency impact for the $\mathcal{EL}$ family.

Example 3. *Let $\mathcal{O} = \{(A \sqsubseteq B_1, v_1), (A \sqsubseteq B_2, v_2), (B_1 \sqcap B_2 \sqsubseteq C, v_3)\}$. If $\mathcal{I}$ is a model of $\mathcal{O}$ and $(e, n^{\mathcal{I}}) \in A^{\mathcal{I}}$, then $(e, (n \times v_1)^{\mathcal{I}}) \in B_1^{\mathcal{I}}$ and $(e, (n \times v_2)^{\mathcal{I}}) \in B_2^{\mathcal{I}}$ so $(e, (n \times v_1 \times n \times v_2)^{\mathcal{I}}) \in (B_1 \sqcap B_2)^{\mathcal{I}}$, i.e. $(e, (n \times v_1 \times v_2)^{\mathcal{I}}) \in (B_1 \sqcap B_2)^{\mathcal{I}}$ by $\times$-idempotency, which implies $(e, (n \times v_1 \times v_2 \times v_3)^{\mathcal{I}}) \in C^{\mathcal{I}}$. Thus $\mathcal{O} \models (A \sqsubseteq C, v_1 \times v_2 \times v_3)$. This intuitive entailment is lost if $\times$ is not idempotent. Indeed, assume that $\times$ is not idempotent and let $\mathcal{I}$ be the interpretation defined as follows (where $\Delta^{\mathcal{I}} = \{e\}$ and $\Delta^{\mathcal{I}}_{\mathsf{m}}$ contains all monomials with variables in lexicographic order).*

$$\begin{aligned}A^{\mathcal{I}} =&\{(e, u)\} \quad B_1^{\mathcal{I}} = \{(e, u \times v_1)\} \quad B_2^{\mathcal{I}} = \{(e, u \times v_2)\}\\ C^{\mathcal{I}} =&\{(e, u \times u \times v_1 \times v_2 \times v_3)\}.\end{aligned}$$

$\mathcal{I}$ is a model of $\mathcal{O}$ such that $\mathcal{I} \not\models (A \sqsubseteq C, v_1 \times v_2 \times v_3)$.

A downside of $\times$-idempotency is a loss of the expressive power of provenance, neglecting the number of times an axiom is used in a derivation. Let $\mathcal{O} = \{(A \sqsubseteq B, v_1), (B \sqsubseteq A, v_2)\}$. With $\times$-idempotency, $\mathcal{O} \models (A \sqsubseteq B, v_1^k \times v_2^l)$ for $k \geq 1$ and $l \geq 0$ because for $k, l \geq 1$, $v_1^k \times v_2^l$ is interpreted by $(v_1 \times v_2)^{\mathcal{I}}$ in any interpretation $\mathcal{I}$. In contrast, if $\times$ is not idempotent, we only obtain $\mathcal{O} \models (A \sqsubseteq B, v_1^{k+1} \times v_2^k)$ for $k \geq 0$ (in particular $\mathcal{O} \not\models (A \sqsubseteq B, v_1 \times v_2)$), which is a more informative result. Some useful semirings are not $\times$-idempotent; e.g. the Viterbi semiring $([0, 1], \max, \times, 0, 1)$, where $\times$ is the usual product over real numbers, which is applied for representing confidence scores. We limit ourselves to $\times$-idempotent semirings because we are interested in computing provenance not only for assertions or queries, but also for GCIs. In particular, when a non-annotated ontology entails the GCI $C \sqsubseteq D$, we want the annotated version of the ontology to entail $(C \sqsubseteq D, m)$ for some monomial m. The non-idempotent case could be relevant when one is not concerned with provenance for GCI entailment, and is left as future work.

Many useful semirings are $\times$-idempotent. Examples of these are: the Boolean semiring, used for probabilistic query answering in databases; the security semiring, used to determine the minimal level of clearance required to get the consequence; and the fuzzy semiring which allows to determine the truth degree of the consequence (see e.g. [Senellart, 2017] for details on these semirings and more examples).

Second, let us explain the restrictions on the form of the right-hand side of the GCIs. Example 4 illustrates the case of conjunctions. Qualified role restrictions lead to the same kind of behavior (they can be seen as implicit conjunctions).

Example 4. *Let $\mathcal{O} = \{(A \sqsubseteq B \sqcap C, v), (A(a), u)\}$. All the following interpretations which interpret a by itself and monomials by their representatives are models of $\mathcal{O}$:*

$$\begin{aligned}&A^{\mathcal{I}_1} = \{(a, u)\}, B^{\mathcal{I}_1} = \{(a, u \times v)\}, C^{\mathcal{I}_1} = \{(a, u \times v)\}\\ &A^{\mathcal{I}_2} = \{(a, u)\}, B^{\mathcal{I}_2} = \{(a, u)\}, \qquad C^{\mathcal{I}_2} = \{(a, v)\}\\ &A^{\mathcal{I}_3} = \{(a, u)\}, B^{\mathcal{I}_3} = \{(a, 1)\}, \qquad C^{\mathcal{I}_3} = \{(a, u \times v)\}\end{aligned}$$

Since the semantics does not provide a unique way to "split" the monomial $u \times v$ between the two elements of the conjunction, $\mathcal{O} \not\models (B(a), m)$ for any $m \in \mathsf{N_M}$, and in particular, $\mathcal{O} \not\models (B(a), u \times v)$. It is arguably counter-intuitive since we intuitively know that a is in A with provenance u and that A is a subclass of the intersection of B and C with provenance v.

Partially normalizing the ontology before annotating it, or more specifically, replacing e.g. annotated GCIs of the form $(C \sqsubseteq C_1 \sqcap C_2, v)$ by $(C \sqsubseteq C_1, v)$ and $(C \sqsubseteq C_2, v)$, may be acceptable in most cases, even if the rewritten ontology leads to additional—arguably natural—consequences compared to the original one. For instance, even if $\mathcal{O} \not\models (A \sqsubseteq B, v)$ in Example 4, in many cases a user would accept to change the GCI of $\mathcal{O}$ to $(A \sqsubseteq B, v)$ and $(A \sqsubseteq C, v)$ as it may reflect the original intention of the GCI since $\{A \sqsubseteq B, A \sqsubseteq C\}$ and $A \sqsubseteq B \sqcap C$ are semantically equivalent.

One could argue that it would be better to define the semantics so that only $\mathcal{I}_1$ was a model of $\mathcal{O}$ in Example 4, instead of restricting the language as we do. We explain next why this is not so simple.

One possibility is to change the definition of satisfaction of a GCI by an interpretation such that $\mathcal{I} \models (A \sqsubseteq B \sqcap C, m)$ iff for every $(d, n^{\mathcal{I}}) \in A^{\mathcal{I}}$, then $(d, (m \times n)^{\mathcal{I}}) \in B^{\mathcal{I}}$ and $(d, (m \times n)^{\mathcal{I}}) \in C^{\mathcal{I}}$, and similarly for qualified role restrictions. This approach leads to a counter-intuitive behavior. For instance if $\mathcal{O} = \{(A(a), u), (B(a), v)\}$, then $\mathcal{O} \not\models (A \sqcap B \sqsubseteq A \sqcap B, 1)$, since $(a, (u \times v)^{\mathcal{I}}) \in (A \sqcap B)^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{O}$, but there is a model $\mathcal{I}$ of $\mathcal{O}$ such that $(a, (u \times v)^{\mathcal{I}}) \notin A^{\mathcal{I}}$ (and $(a, (u \times v)^{\mathcal{I}}) \notin B^{\mathcal{I}}$). In contrast, our definition of satisfaction ensures that for every interpretation $\mathcal{I}$ and concept C, $\mathcal{I} \models (C \sqsubseteq C, 1)$.

Another possibility is to modify the interpretation of conjunctions and qualified role restrictions such that $(C \sqcap D)^{\mathcal{I}} = \{(d, m^{\mathcal{I}}) \mid (d, m^{\mathcal{I}}) \in C^{\mathcal{I}}, (d, m^{\mathcal{I}}) \in D^{\mathcal{I}}\}$ and $(\exists R.C)^{\mathcal{I}} = \{(d, m^{\mathcal{I}}) \mid \exists e \in \Delta^{\mathcal{I}} \text{ s.t. } (d, e, m^{\mathcal{I}}) \in R^{\mathcal{I}}, (e, m^{\mathcal{I}}) \in C^{\mathcal{I}}\}$. In this case, we lose even basic entailments from annotated ABoxes; e.g., $\{(A(a), u), (B(a), v)\} \not\models ((A \sqcap B)(a), u \times v)$. We also lose the entailment of the GCI from Example 3.

Hence, restricting the syntax to prevent conjunctions on the right and defining the semantics as usual in DLs seems to be the most natural way of handling provenance in DL languages with conjunction. Since $\mathcal{EL}$ ontologies are often already expressed in normal form, the main restriction in our language is the avoidance of qualified existential restrictions on the right-hand side.

2.4 Annotated Queries

Following Calvanese *et al.* [2019], we extend DL conjunctive queries with binary and ternary predicates, where the last term of the tuple is used for provenance information. Recall that by the semantics of annotated ontologies, tuples can only contain monomials. A *Boolean conjunctive query (BCQ)* q is a sentence $\exists \vec{x}.\varphi(\vec{x}, \vec{a})$, where φ is a conjunction of (unique) atoms of the form $A(t_1, t)$, $R(t_1, t_2, t)$; t_i is an individual name from $\vec{a}$, or a variable from $\vec{x}$; and t (the last term of the tuple) is a variable from $\vec{x}$ that does not occur anywhere else in q (Calvanese *et al.* call such a query *standard*). We use $P(\vec{t}, t)$ to refer to an atom which is either $A(t_1, t)$ or $R(t_1, t_2, t)$, and $P(\vec{t}, t) \in q$ if $P(\vec{t}, t)$ occurs in q.

A *match* of the BCQ $q = \exists \vec{x}.\varphi(\vec{x}, \vec{a})$ in the annotated interpretation $\mathcal{I}$ is a function $\pi : \vec{x} \cup \vec{a} \to \Delta^{\mathcal{I}} \cup \Delta^{\mathcal{I}}_{\mathsf{m}}$, such that $\pi(b) = b^{\mathcal{I}}$ for all $b \in \vec{a}$, and $\pi(\vec{t}, t) \in P^{\mathcal{I}}$ for every $P(\vec{t}, t) \in q$, where $\pi(\vec{t}, t)$ is a shorthand for $(\pi(t_1), \pi(t))$ or $(\pi(t_1), \pi(t_2), \pi(t))$ depending on the arity of P. $\mathcal{I}$ satisfies the BCQ q, written $\mathcal{I} \models q$, if there is a match of q in $\mathcal{I}$. A BCQ q is *entailed by* an annotated ontology $\mathcal{O}$, denoted $\mathcal{O} \models q$, if every model of $\mathcal{O}$ satisfies q. For a BCQ q and an interpretation $\mathcal{I}$, $\nu_{\mathcal{I}}(q)$ denotes the set of all matches of q in $\mathcal{I}$. The *provenance* of q on $\mathcal{I}$ is the expression

$$\mathsf{prov}_{\mathcal{I}}(q) := \textstyle\sum_{\pi \in \nu_{\mathcal{I}}(q)} [\prod_{P(\vec{t}, t) \in q} \pi^{-}(t)]$$

where $\pi(t)$ is the last element of the tuple $\pi(\vec{t}, t) \in P^{\mathcal{I}}$; and $\pi^{-}(t)$ is the only $m \in \mathsf{N}_{[\mathsf{M}]}$ s.t. $m^{\mathcal{I}} = \pi(t)$. For $p \in \mathsf{N}_{\mathsf{P}}$, 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 its representative in $\mathsf{prov}_{\mathcal{I}}(q)$. $\mathcal{I}$ *satisfies* q with provenance $p \in \mathsf{N}_{\mathsf{P}}$, denoted $\mathcal{I} \models (q, p)$, if $\mathcal{I} \models q$ and $p \subseteq \mathsf{prov}_{\mathcal{I}}(q)$. $\mathcal{O} \models (q, p)$, if $\mathcal{O} \models q$ and $p \subseteq \mathsf{prov}_{\mathcal{I}}(q)$, for all $\mathcal{I} \models \mathcal{O}$. We call (q, p) an *annotated query*.

Remark. *When $\mathcal{O}$ contains only assertions (no GCIs, RIs, and RRs), we can compare the provenance annotations we obtain to the database case. Similarly to Trio-provenance, the sums of monomials distinguish different ways the query atoms can be mapped to annotated interpretations. For example, given $\mathcal{O} = \{(R(a, b), v_1), (R(b, a), v_2)\}$ and query $q = \exists xytt'.R(x, y, t) \wedge R(y, x, t')$, it holds that $\mathcal{O} \models (q, v_1 \times v_2 + v_1 \times v_2)$. The provenance annotation $v_1 \times v_2 + v_1 \times v_2$ distinguishes among two derivations using the same axioms, contrary to the why-provenance $v_1 \times v_2$. Note that given an axiom α and $\mathcal{O}$ that may contain GCIs, RIs, and RRs, the sum over all monomials m such that $\mathcal{O} \models (\alpha, m)$ represents all possible derivations of α, in the why-provenance spirit.*

The *size* $|X|$ of an annotated ontology, a polynomial or a BCQ X is the length of the string representing X, where elements of N_{C}, N_{R}, N_{I} and N_{V} in X are of length one. We often omit 'annotated' and refer to 'ontologies,' 'queries,' 'assertions,' etc. when it is clear from the context.

3 Reasoning with Annotated $\mathcal{ELH}^r$ Ontologies

We present a completion algorithm for deriving basic entailments from an $\mathcal{ELH}^r$ ontology. As usual with completion algorithms, we restrict to ontologies in normal form. The annotated $\mathcal{ELH}^r$ ontology $\mathcal{O}$ is in *normal form* if for every GCI $(\alpha, v) \in \mathcal{O}$, α is of the form $A \sqsubseteq B$, $A \sqcap A' \sqsubseteq B$, $A \sqsubseteq \exists R$, or $\exists R.A \sqsubseteq B$, with $A, A' \in \mathsf{N}_{\mathsf{C}} \cup \{\top\}$, $B \in \mathsf{N}_{\mathsf{C}}$. Every annotated $\mathcal{ELH}^r$ ontology can be transformed, in polynomial time, into an ontology in normal form which entails the same axioms over the ontology signature, using the following rules where $\widehat{C}, \widehat{D} \notin \mathsf{N}_{\mathsf{C}} \cup \{\top\}$ and A is a fresh concept name:

$$\begin{aligned}
\mathsf{NF}_1 &: (C \sqcap \widehat{D} \sqsubseteq E,\, v) \longrightarrow (\widehat{D} \sqsubseteq A,\, 1), (C \sqcap A \sqsubseteq E,\, v)\\
\mathsf{NF}_2 &: \ (\exists R.\widehat{C} \sqsubseteq D,\, v) \longrightarrow (\widehat{C} \sqsubseteq A,\, 1), (\exists R.A \sqsubseteq D,\, v)\\
\mathsf{NF}_3 &: \qquad (\widehat{C} \sqsubseteq \exists R,\, v) \longrightarrow (\widehat{C} \sqsubseteq A,\, 1), (A \sqsubseteq \exists R,\, v).
\end{aligned}$$

Theorem 5. *Let $\mathcal{O}$ be an annotated $\mathcal{ELH}^r$ ontology, α an axiom, and m a monomial. Let $\mathcal{O}'$ be obtained by applying exhaustively Rules NF_1-NF_3 to $\mathcal{O}$.*

- *If $\mathcal{O} \models (\alpha, m)$, then $\mathcal{O}' \models (\alpha, m)$.*
- *If $\mathcal{O}' \models (\alpha, m)$ and every concept name occurring in α occurs in $\mathcal{O}$, then $\mathcal{O} \models (\alpha, m)$.*

Before describing the reasoning algorithm in detail, we present an important property of entailment; namely, that all entailment problems can be polynomially reduced to each other. This allows us to focus on only one problem. In particular, we focus on entailment of annotated assertions.

Theorem 6. *Let $\mathcal{O}$ be an annotated ontology, and (α, m) an annotated GCI, RR, or RI. One can construct in polynomial time an ontology $\mathcal{O}'$ and an annotated assertion (β, n) such that $\mathcal{O} \models (\alpha, m)$ iff $\mathcal{O}' \models (\beta, n)$. Conversely, if (α, m) is an*

	if	then (if $\Phi \notin \mathcal{O}$)
CR_0	$X \in \mathsf{N_C} \cup \mathsf{N_R} \cup \{\top\}$ occurs in $\mathcal{O}$	add $\Phi = (X \sqsubseteq X, 1)$ to $\mathcal{O}$
CR_1	$(R_1 \sqsubseteq R_2, m_1), (R_2 \sqsubseteq R_3, m_2) \in \mathcal{O}$	add $\Phi = (R_1 \sqsubseteq R_3, [m_1 \times m_2])$ to $\mathcal{O}$
CR_2	$(R \sqsubseteq S, m_1), (\mathsf{ran}(S) \sqsubseteq A, m_2) \in \mathcal{O}$	add $\Phi = (\mathsf{ran}(R) \sqsubseteq A, [m_1 \times m_2])$ to $\mathcal{O}$
CR_3	$(A \sqsubseteq \exists R, m_1), (R \sqsubseteq S, m_2) \in \mathcal{O}$	add $\Phi = (A \sqsubseteq \exists S, [m_1 \times m_2])$ to $\mathcal{O}$
CR_4	$(A \sqsubseteq B, m_1), (B \sqsubseteq C, m_2) \in \mathcal{O}$	add $\Phi = (A \sqsubseteq C, [m_1 \times m_2])$ to $\mathcal{O}$
CR_5	$(A \sqsubseteq B, m_1), (B \sqsubseteq \exists R, m_2) \in \mathcal{O}$	add $\Phi = (A \sqsubseteq \exists R, [m_1 \times m_2])$ to $\mathcal{O}$
CR_6	$(A \sqsubseteq B_1, m_1), (A \sqsubseteq B_2, m_2), (B_1 \sqcap B_2 \sqsubseteq C, m_3) \in \mathcal{O}$	add $\Phi = (A \sqsubseteq C, [m_1 \times m_2 \times m_3])$ to $\mathcal{O}$
CR_7	$(\mathsf{ran}(R) \sqsubseteq B_1, m_1), (\mathsf{ran}(R) \sqsubseteq B_2, m_2), (B_1 \sqsubseteq C_1, m_3), (B_2 \sqsubseteq C_2, m_4), (C_1 \sqcap C_2 \sqsubseteq C, m_5) \in \mathcal{O}$	add $\Phi = (\mathsf{ran}(R) \sqsubseteq C, [m_1 \times m_2 \times m_3 \times m_4 \times m_5])$ to $\mathcal{O}$
CR_8	$(A \sqcap B \sqsubseteq C, m_1), (\top \sqsubseteq B, m_2) \in \mathcal{O}$	add $\Phi = (A \sqsubseteq C, [m_1 \times m_2])$ to $\mathcal{O}$
CR_9	$(A \sqsubseteq \exists S, m_1), (\mathsf{ran}(S) \sqsubseteq B, m_2), (B \sqsubseteq C, m_3), (S \sqsubseteq R, m_4), (\exists R.C \sqsubseteq D, m_5) \in \mathcal{O}$	add $\Phi = (A \sqsubseteq D, [m_1 \times m_2 \times m_3 \times m_4 \times m_5])$ to $\mathcal{O}$
CR_{10}	$(A \sqsubseteq \exists R, m_1), (\top \sqsubseteq B, m_2), (\exists R.B \sqsubseteq C, m_3) \in \mathcal{O}$	add $\Phi = (A \sqsubseteq C, [m_1 \times m_2 \times m_3])$ to $\mathcal{O}$
CR_{11}	$a \in \mathsf{ind}(\mathcal{O})$	add $\Phi = (\top(a), 1)$ to $\mathcal{O}$
CR_{12}	$(R(a,b), m_1), (R \sqsubseteq S, m_2) \in \mathcal{O}$	add $\Phi = (S(a,b), [m_1 \times m_2])$ to $\mathcal{O}$
CR_{13}	$(A(a), m_1), (A \sqsubseteq B, m_2) \in \mathcal{O}$	add $\Phi = (B(a), [m_1 \times m_2])$ to $\mathcal{O}$
CR_{14}	$(A_1(a), m_1), (A_2(a), m_2), (A_1 \sqcap A_2 \sqsubseteq B, m_3) \in \mathcal{O}$	add $\Phi = (B(a), [m_1 \times m_2 \times m_3])$ to $\mathcal{O}$
CR_{15}	$(R(a,b), m_1), (A(b), m_2), (\exists R.A \sqsubseteq B, m_3) \in \mathcal{O}$	add $\Phi = (B(a), [m_1 \times m_2 \times m_3])$ to $\mathcal{O}$
CR_{16}	$(R(a,b), m_1), (\mathsf{ran}(R) \sqsubseteq A, m_2) \in \mathcal{O}$	add $\Phi = (A(b), [m_1 \times m_2])$ to $\mathcal{O}$

Table 1: Completion rules. $A, \ldots, D \in \mathsf{N_C} \cup \{\top\}$, $R, S, R_i \in \mathsf{N_R}$, $m, m_i \in \mathsf{N_M}$.

annotated concept (resp. role) assertion, one can construct in polynomial time an ontology $\mathcal{O}'$ and two annotated concept (resp. role) inclusions (β, n), (γ, n) such that $\mathcal{O} \models (\alpha, m)$ iff $\mathcal{O}' \models (\beta, n)$ or $\mathcal{O}' \models (\gamma, n)$.

We adapt the classical $\mathcal{EL}$ completion rules to handle annotated $\mathcal{ELH}^r$ ontologies in normal form. The algorithm starts with the original ontology $\mathcal{O}$, and extends it through an iterative application of the rules from Table 1 until $\mathcal{O}$ becomes *saturated*; i.e., no more rules are applicable. We cannot use the rules of [Baader *et al.*, 2008a] which eliminate range restrictions by adding GCIs with qualified role restrictions on the right so we designed rules for $\mathcal{ELH}^r$.

A rule application may add axioms annotated with monomials, and other assertions $(\top(a), 1)$, which are not foreseen in the definition of annotated ontologies. Still, $\times$-idempotency ensures that all monomials have at most $|\mathcal{O}|$ factors. To show that the completion algorithm is sound and complete for deciding assertion entailment, we prove a stronger result. The *k-saturation* of $\mathcal{O}$ is the saturated ontology $\mathcal{O}_k$ obtained from $\mathcal{O}$ through the completion algorithm restricted to monomials of length at most k. We show that $\mathcal{O}_k$ suffices for deciding entailment of annotated assertions (α, m) where m is a monomial of length at most k.

Theorem 7. *If $\mathcal{O}_k$ is the k-saturation of $\mathcal{O}$, then*

1. *$\mathcal{O}_k$ is computable in polynomial time w.r.t. the size of $\mathcal{O}$, and in exponential time w.r.t. k,*
2. *for every assertion α and monomial m_k with at most k variables, $\mathcal{O} \models (\alpha, m_k)$ iff $(\alpha, [m_k]) \in \mathcal{O}_k$.*

This theorem states that to decide whether an assertion (α, m) is entailed by $\mathcal{O}$, one just needs to find the k-saturation of $\mathcal{O}$, where k is the number of variables in m, and then check whether $(\alpha, [m]) \in \mathcal{O}_k$. Due to the first point of Theorem 7 and Theorem 6, we obtain the following corollary.

Corollary 8. *For every axiom α, $\mathcal{O} \models (\alpha, m)$ is decidable in polynomial time in $|\mathcal{O}|$ and in exponential time in $|m|$.*

In general there is no need to interrupt the completion algorithm; the ontology saturated without restricting the monomial length can be used to decide all relevant entailments regardless of the length of the monomial. Using $\mathcal{O}_k$ is merely an optimisation when one is only interested in a short monomial.

While the polynomial time upper bound w.r.t. the ontology size is positive, and in line with the complexity of the $\mathcal{EL}$ family, the exponential time bound on the monomial size does not scale well for entailments with larger monomials. Recall that these bounds are based on the number of annotated axioms generated by the completion rules. The following example illustrates the potential exponential blow-up.

Example 9. *Consider $\mathcal{O} = \{(A \sqsubseteq A_i, v_i), (A_i \sqsubseteq B, u_i) \mid 0 \leq i \leq n\} \cup \{(B \sqsubseteq A, u)\}$. If $\mathcal{O}'$ is the result of applying the completion algorithm to $\mathcal{O}$, then for every $S \subseteq \{1, \ldots, n\}$, $(B \sqsubseteq A, [u \times \Pi_{i \in S} u_i \times v_i]) \in \mathcal{O}'$.*

Following Hutschenreiter and Peñaloza [2017], we can see the completion algorithm as an automaton. More precisely, given $\mathcal{O}$ and (α, m), we can construct a tree automaton $\mathcal{A}$, whose states correspond exactly to all the elements in $\mathcal{O}_k$, such that $(\alpha, [m]) \in \mathcal{O}_k$ iff $\mathcal{A}$ accepts at least one tree. Briefly, $\mathcal{A}$ is constructed by reading the rule applications backwards, allowing transitions from the consequence to the premises of the rule; see [Hutschenreiter and Peñaloza, 2017] for details. The number of states in $\mathcal{A}$ is exactly the cardinality of $\mathcal{O}_k$ and hence potentially exponential on k. However, the size of each state is bounded polynomially on k; the arity of the automaton is bounded by the maximum number of premises in a rule, in this case 5; and one can bound polynomially on k the number of different states that may appear in any successful run of $\mathcal{A}$. Thus, $\mathcal{A}$ satisfies the conditions for a PSpace emptiness test [Baader *et al.*, 2008b], which yields the following result.

Proposition 10. *For every axiom α, $\mathcal{O} \models (\alpha, m)$ is decidable in polynomial space in $|m|$.*

Interestingly, these results allow us to bound the full complexity of answering *instance queries* (IQ) of the form $C(a)$ where C is an $\mathcal{ELH}^r$ concept and $a \in \mathsf{N_I}$.

Theorem 11. *Let $\mathcal{O}$ be an ontology, $C(a)$ an IQ and $m \in \mathsf{N_M}$. $\mathcal{O} \models (C(a), m)$ is decidable in polynomial time in $|\mathcal{O}|$ and $|C(a)|$, and polynomial space in $|m|$.*

4 Computing Relevant Provenance Variables

An interesting question is whether a given variable appears in the provenance of a query q; i.e., whether a given axiom occurs in some derivation of q. Formally, $v \in \mathsf{N_V}$ is *relevant* for q (w.r.t. ontology $\mathcal{O}$) iff $\exists m \in \mathsf{N_M}$ s.t. $\mathcal{O} \models (q, v \times m)$. For IQs and $\mathcal{ELH}^r$ this problem can be solved in polynomial time, via an algorithm computing all the relevant variables for all queries of the form $A(a)$, with $a \in \mathsf{N_I}$, $A \in \mathsf{N_C}$. We modify the completion algorithm (Section 3) to combine all monomials from a derivation, instead of storing them separately.

As in Section 3, the algorithm assumes normal form and keeps as data structure a set $\mathcal{S}$ of annotated axioms (α, m), where α uses the vocabulary of $\mathcal{O}$, and $m \in \mathsf{N_M}$. $\mathcal{S}$ is initialised as the original ontology where annotations of the same axiom are merged into a single monomial:

$$\mathcal{S} := \{(\alpha, [\Pi_{v \in V_\alpha} v]) \mid (\alpha, u) \in \mathcal{O}, V_\alpha = \{v \mid (\alpha, v) \in \mathcal{O}\}\},$$

and extended by exhaustively applying the rules in Table 1, where rule applications change $\mathcal{S}$ into

$$\mathcal{S} \uplus (\alpha, m) := \begin{cases} \mathcal{S} \cup \{(\alpha, m)\} \text{ if there is no } (\alpha, n) \in \mathcal{S} \\ \mathcal{S} \setminus \{(\alpha, n)\} \cup \{(\alpha, [m \times n])\} \text{ if } (\alpha, n) \in \mathcal{S}; \end{cases}$$

i.e., add the axiom α with an associated monomial if it does not yet appear in $\mathcal{S}$, and modify the monomial associated to α to include new variables otherwise. To ensure termination, a rule is only applied if it modifies $\mathcal{S}$. The rules are applied until no new rule is applicable; i.e., $\mathcal{S}$ is *saturated*.

Example 12. *The relevance algorithm on the ontology of Example 9, yields the saturated set* $\mathcal{S} = \{(A \sqsubseteq A, m), (B \sqsubseteq B, m), (A \sqsubseteq B, m), (B \sqsubseteq A, m)\} \cup \{(A_i \sqsubseteq B, m), (B \sqsubseteq A_i, m), (A_i \sqsubseteq A, m), (A \sqsubseteq A_i, m) \mid (1 \leq i \leq n\} \cup \{(A_i \sqsubseteq A_j, m) \mid 1 \leq i, j \leq n\}$ *with* $m = u \times \Pi_{i=1}^n u_i \times \Pi_{i=1}^n v_i$.

Each rule application either adds a new axiom, or adds to the label of an existing axiom more variables. As the number of concept and role names, and variables appearing in $\mathcal{S}$ is linear on $\mathcal{O}$, at most polynomially many rules are applied, each requiring polynomial time; i.e, the algorithm is polynomial.

Lemma 13. *If* $\mathcal{S}$ *is the saturated set obtained from* $\mathcal{O}$, $a \in \mathsf{N_I}$, $A \in \mathsf{N_C}$, *and* $v \in \mathsf{N_V}$, *then* v *is relevant for* $A(a)$ *iff* v *occurs in* m *for some* $(A(a), m) \in \mathcal{S}$.

The algorithm decides relevance for assertion entailment in $\mathcal{ELH}^r$, yielding a polynomial-time upper bound for this problem. As in Section 3, axioms and IQs can be handled in polynomial time as well.

Theorem 14. *Relevance for axiom and IQ entailment in* $\mathcal{ELH}^r$ *can be decided in polynomial time.*

This result shows that if we only need to know which axioms are used to derive an axiom or an IQ, the complexity is the same as reasoning in $\mathcal{ELH}^r$ without provenance. This contrasts with *axiom pinpointing* : the task of finding the axioms responsible for a consequence to follow, in the sense of belonging to some *minimal* subontology entailing it (a MinA). Deciding whether an axiom belongs to a MinA is NP-hard for Horn-$\mathcal{EL}$ [Peñaloza and Sertkaya, 2010]. Relevance is easier in our context since provenance does not require minimality: if $\mathcal{O} = \{(A \sqsubseteq B, v_1), (B \sqsubseteq C, v_2), (C \sqsubseteq B, v_3)\}$, v_2 and v_3 are relevant for $A \sqsubseteq B$, but the only MinA is $\{A \sqsubseteq B\}$ so other axioms are not relevant for axiom pinpointing.

Provenance relevance is related to *lean kernels* (LKs) [Peñaloza *et al.*, 2017], which approximate the union of MinAs. The LK of a consequence c is the set of axioms appearing in at least one proof of c in a given inference method, generalizing the notion from propositional logic, where an LK is the set of clauses appearing in a resolution proof for unsatisfiability. The sets of variables computed by our algorithm are the sets of axioms used in the derivations by the completion algorithm, which is a consequence-based method for $\mathcal{ELH}^r$. Thus they correspond to LKs for the associated axioms and our algorithm is an alternative way of computing LKs in $\mathcal{ELH}^r$.

5 Query Answering with Provenance

Even if $\mathcal{ELH}^r$ is expressive enough to reduce entailment of rooted tree-shaped BCQs to assertion entailment, the methods presented in Section 3 do not apply to other kinds of BCQs.

Example 15. *For* $\mathcal{O}{=}\{(R(a,a), u_1), (A(a), u_2), (A{\sqsubseteq}\exists R, v_1), (\mathsf{ran}(R){\sqsubseteq}A, v_2)\}$ *and* $q = \exists xyztt't''.R(x,x,t) \wedge R(x,y,t') \wedge R(z,y,t'')$, $\mathcal{O} \models (q, u_1)$ *but* $\mathcal{O} \not\models (q, u_2 \times v_1 \times v_2)$*:* $\mathcal{O}$ *has a model* $\mathcal{I}$ *with* $R^{\mathcal{I}} = \{(a,a,u_1), (a, b_1, u_2 \times v_1), (a, c_1, u_1 \times v_1 \times v_2)\} \cup \{(b_i, b_{i+1}, u_2 \times v_1 \times v_2) \mid i \geq 1\} \cup \{(c_i, c_{i+1}, u_1 \times v_1 \times v_2) \mid i \geq 1\}$.

We adapt the combined approach by Lutz *et al.* [2009] to trace provenance. Assume that queries contain only individual names occurring in the ontology $\mathcal{O}$. The combined approach builds a canonical model for $\mathcal{O}$ and shows that every query q can be rewritten into a query q^* that holds in this canonical model iff $\mathcal{O} \models q$. We first define the canonical model $\mathcal{I}_\mathcal{O}$ of an ontology $\mathcal{O}$ annotated with provenance information.

Assume that $\mathcal{O}$ is in normal form; $\mathsf{mon}(\mathcal{O})$ denotes the set of monomial representatives built using variables of $\mathsf{N_V}$ occurring in $\mathcal{O}$, and $\mathsf{rol}(\mathcal{O})$ is the set of role names occurring in $\mathcal{O}$. Also assume that $(*)$ if there is $B \in \mathsf{N_C}$, $R \in \mathsf{N_R}$, and $n \in \mathsf{N_M}$ such that $\mathcal{O} \models (\mathsf{ran}(R) \sqsubseteq B, n)$, then $(\mathsf{ran}(R) \sqsubseteq B, [n]) \in \mathcal{O}$. This simplifies the presentation of the construction of the canonical model. Let $\mathsf{aux}(\mathcal{O}) := \{d_R^m \mid R \in \mathsf{rol}(\mathcal{O}), m \in \mathsf{mon}(\mathcal{O})\}$. Assume that $\mathsf{ind}(\mathcal{O}) \cap \mathsf{aux}(\mathcal{O}) = \emptyset$. We define the domain of $\mathcal{I}_\mathcal{O}$ and the domain of monomials of $\mathcal{I}_\mathcal{O}$ as follows:

$$\Delta^{\mathcal{I}_\mathcal{O}} := \mathsf{ind}(\mathcal{O}) \mathbin{\dot\cup} \mathsf{aux}(\mathcal{O}) \qquad\qquad \Delta_{\mathsf{m}}^{\mathcal{I}_\mathcal{O}} := \mathsf{N}_{[\mathsf{M}]}$$

We define the interpretation function of $\mathcal{I}_\mathcal{O}$ as the union of $\cdot^{\mathcal{I}_\mathcal{O}^i}$, $i \geq 0$. The function $\cdot^{\mathcal{I}_\mathcal{O}^0}$ sets $a^{\mathcal{I}_\mathcal{O}^0} = a$ for all $a \in \mathsf{ind}(\mathcal{O})$ (for $a \in \mathsf{N_I} \setminus \mathsf{ind}(\mathcal{O})$ the mapping $a^{\mathcal{I}_\mathcal{O}}$ is irrelevant), $m^{\mathcal{I}_\mathcal{O}^0} = [m]$ for all $m \in \mathsf{N_M}$, and for all $A \in \mathsf{N_C}$ and all $R \in \mathsf{N_R}$,

$$A^{\mathcal{I}_\mathcal{O}^0} := \{(a, [m]) \mid \mathcal{O} \models (A(a), m)\}$$

$$R^{\mathcal{I}_\mathcal{O}^0} := \{(a, b, [m]) \mid \mathcal{O} \models (R(a,b), m)\}.$$

If $\mathcal{I}_\mathcal{O}^i$ is defined, we define $\mathcal{I}_\mathcal{O}^{i+1}$ by choosing an annotated axiom $\alpha \in \mathcal{O}$ and applying one of the following rules in a fair way (i.e., every applicable rule is eventually applied).

R1 $\alpha = (C \sqsubseteq A, m)$: if there is $d \in \Delta^{\mathcal{I}_\mathcal{O}}$ and $n \in \mathsf{mon}(\mathcal{O})$ s.t. $(d, [n]) \in C^{\mathcal{I}_\mathcal{O}^i}$, then add $(d, [m \times n])$ to $A^{\mathcal{I}_\mathcal{O}^i}$.

R2 $\alpha = (C \sqsubseteq \exists R, m)$: if there is $n \in \mathsf{mon}(\mathcal{O})$, $d \in \Delta^{\mathcal{I}_\mathcal{O}}$ s.t. $(d, [n]) \in C^{\mathcal{I}_\mathcal{O}^i}$, then add $(d, d_R^{[m \times n]}, [m \times n])$ to $R^{\mathcal{I}_\mathcal{O}^i}$.

R3 $\alpha = (R \sqsubseteq S, m)$: if there are $d, d' \in \Delta^{\mathcal{I}_\mathcal{O}}$, $n \in \mathsf{mon}(\mathcal{O})$ s.t. $(d, d', [n]) \in R^{\mathcal{I}^i_\mathcal{O}}$, then add $(d, d', [m \times n])$ to $S^{\mathcal{I}^i_\mathcal{O}}$.

Example 16. *For our running example, $\mathcal{I}_\mathcal{O}$ is as follows:*

$$\begin{aligned}
A^{\mathcal{I}_\mathcal{O}} = \{&(a, u_2), (a, u_1 \times v_2), (d_R^{u_2 \times v_1}, u_2 \times v_1 \times v_2),\\
&(d_R^{u_1 \times v_1 \times v_2}, u_1 \times v_1 \times v_2),\\
&(d_R^{u_2 \times v_1 \times v_2}, u_2 \times v_1 \times v_2)\}\\
R^{\mathcal{I}_\mathcal{O}} = \{&(a, a, u_1), (a, d_R^{u_2 \times v_1}, u_2 \times v_1),\\
&(a, d_R^{u_1 \times v_1 \times v_2}, u_1 \times v_1 \times v_2),\\
&(d_R^{u_2 \times v_1}, d_R^{u_2 \times v_1 \times v_2}, u_2 \times v_1 \times v_2),\\
&(d_R^{u_1 \times v_1 \times v_2}, d_R^{u_1 \times v_1 \times v_2}, u_1 \times v_1 \times v_2),\\
&(d_R^{u_2 \times v_1 \times v_2}, d_R^{u_2 \times v_1 \times v_2}, u_2 \times v_1 \times v_2)\}.
\end{aligned}$$

Proposition 17 formalises the fact that $\mathcal{I}_\mathcal{O}$ is a model of $\mathcal{O}$.

Proposition 17. *$\mathcal{I}_\mathcal{O}$ is a model of $\mathcal{O}$.*

We define the rewriting q^* of a query q, closely following Lutz *et al.* [2009]. It contains an additional predicate Aux, always interpreted as $(\Delta^{\mathcal{I}_\mathcal{O}} \setminus \mathsf{ind}(\mathcal{O})) \times \{1^{\mathcal{I}_\mathcal{O}}\}$ in $\mathcal{I}_\mathcal{O}$. Let $\sim_q$ be the smallest transitive relation over terms of q, $\mathsf{term}(q)$, that includes identity relation, and satisfies the closure condition

$$(\dagger)\ \ R_1(t_1, t_2, t), R_2(t'_1, t'_2, t') \in q, t_2 \sim_q t'_2 \implies t_1 \sim_q t'_1.$$

Clearly, the relation $\sim_q$ is computable in polynomial time in the size of q. Define for any equivalence class χ of $\sim_q$, the set

$$\mathsf{pre}(\chi) = \{t_1 \mid \exists R \in \mathsf{N_R} \text{ s.t. } R(t_1, t_2, t) \in q \text{ and } t_2 \in \chi\}.$$

We define the sets Cyc and $\mathsf{Fork}_=$ whose main purpose in the translation is to prevent spurious matches (e.g., with cycles) of a query in the anonymous part of the canonical model.

- $\mathsf{Fork}_=$ is the set of pairs $(\mathsf{pre}(\chi), \chi)$ with $\mathsf{pre}(\chi)$ of cardinality at least two.
- Cyc is the set of variables x in $\mathsf{term}(q)$ such that there are $R_0(t_1^0, t_2^0, t^0), \dots, R_m(t_1^m, t_2^m, t^m), \dots, R_n(t_1^n, t_2^n, t^n)$ in q with $n, m \geq 0$, $x \sim_q t_1^j$ for some $j \leq n$, $t_2^i \sim_q t_1^{i+1}$ for all $i < n$, and $t_2^n \sim_q t_1^m$.

$\mathsf{Fork}_=$, and Cyc can also be computed in polynomial time in the size of q. For each equivalence class χ of $\sim_q$, we choose a representative $t_\chi \in \chi$. For $q = \exists \vec{x}.\psi$, the rewritten query q^* is defined as $\exists \vec{x}.(\psi \wedge \varphi_1 \wedge \varphi_2)$, where

$$\varphi_1 := \bigwedge_{x \in \mathsf{Cyc}} \neg \mathsf{Aux}(x, 1)$$

$$\varphi_2 := \bigwedge_{(\{t_1, \dots, t_k\}, \chi) \in \mathsf{Fork}_=} (\mathsf{Aux}(t_\chi, 1) \rightarrow \bigwedge_{1 \leq i < k} t_i = t_{i+1}).$$

Example 18. *The rewriting q^* of q in Example 16 is $\exists xyztt't''.(R(x, x, t) \wedge R(x, y, t') \wedge R(z, y, t'') \wedge \neg \mathsf{Aux}(x, 1) \wedge (\mathsf{Aux}(y, 1) \rightarrow x = z))$. φ_1 prevents mapping x to some d_R^m, avoiding the R-loops in the anonymous part of $\mathcal{I}_\mathcal{O}$ to satisfy $R(x, x, t)$. φ_2 enforces that if y is mapped in the anonymous part, then x and z are mapped to the same object, which avoids R-loops in the anonymous part of $\mathcal{I}_\mathcal{O}$.*

Our construction differs from the original rewriting of Lutz *et al.* [2009]. In particular, in their rewriting there is a formula φ_3, which is not necessary in our case. Intuitively, this is because we keep the information of the role name used to connect an element of $\mathsf{aux}(\mathcal{O})$ to the rest of the model. Theorem 19 establishes that q^* is as required.

Theorem 19. *Let $\mathcal{O}$ be an ontology in normal form and (q, p) be an annotated query. Then, $\mathcal{O} \models (q, p)$ iff $\mathcal{I}_\mathcal{O} \models (q^*, p)$.*

Although the domain of monomials is infinite, since only elements of $\mathsf{mon}(\mathcal{O})$ are relevant, an exponential size structure representing $\mathcal{I}_\mathcal{O}$ is sufficient to check whether $\mathcal{I}_\mathcal{O} \models (q^*, p)$. The size of the resulting structure is exponential in $|\mathcal{O}|$ and can be constructed in exponential time using the completion algorithm (Theorem 7) to check entailment of assertions and RRs.

Corollary 20. *Let $\mathcal{O}$ be an ontology, q a BCQ and $p \in \mathsf{N_P}$. $\mathcal{O} \models (q, p)$ is decidable in exponential time in $|\mathcal{O}| + |(q, p)|$.*

6 Discussion and Conclusions

We study the problem of computing the provenance of an axiom or a BCQ entailment from $\mathcal{ELH}^r$ ontologies. In particular, entailment of annotated axioms or IQs for a fixed monomial size is tractable, and the set of relevant provenance variables can be computed in polynomial time. For the more challenging problem of CQ answering, we adapt the combined approach.

Related work. Explaining inferences in DLs has been studied mostly focusing on explaining axiom entailment, in particular concept subsumption, through *axiom pinpointing* [Schlobach and Cornet, 2003; Kalyanpur *et al.*, 2007; Baader *et al.*, 2007b]. Few approaches address *query answer explanation* for DL-Lite or existential rules [Borgida *et al.*, 2008; Croce and Lenzerini, 2018; Ceylan *et al.*, 2019; Bienvenu *et al.*, 2019]. However, current explanation services in DLs provide *minimal* explanations, which is crucially different to provenance, since provenance takes into account *all* derivations (cf. discussion in Section 4).

Closest to our work is provenance for OBDA [Calvanese *et al.*, 2019]. However, the challenges in enriching the $\mathcal{EL}$ family with provenance were not investigated. We also study additional problems such as axiom entailment and relevance. Dannert and Grädel [2019] consider provenance in the DL $\mathcal{ALC}$. The setting is not the same as ours since they only consider annotated assertions (not annotated GCIs), do not study BCQs, and the semantics is different as well. There are several proposals for handling provenance in RDF(S), most notably an algebraic deductive system for annotated RDFS [Buneman and Kostylev, 2010]. The approach by Bourgaux and Ozaki [2019] for attributed DL-Lite fundamentally differs by using GCIs and RIs to express constraints on provenance.

Acknowledgments

This work was supported by Camille Bourgaux's CNRS PEPS grant; contract ANR-18-CE23-0003 (CQFD); the University of Bergen; the Free University of Bozen-Bolzano projects PROVDL and FO2S; and the Italian PRIN project HOPE.

References

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *IJCAI*, 2005.

[Baader *et al.*, 2007a] 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.

[Baader *et al.*, 2007b] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *KI*, 2007.

[Baader *et al.*, 2008a] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope further. In *OWLED*, 2008.

[Baader *et al.*, 2008b] Franz Baader, Jan Hladik, and Rafael Peñaloza. Automata can show PSpace results for description logics. *Inf. Comput.*, 206(9-10), 2008.

[Bienvenu *et al.*, 2013] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. First-Order Rewritability of Atomic Queries in Horn Description Logics. In *IJCAI*, 2013.

[Bienvenu *et al.*, 2019] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Computing and explaining query answers over inconsistent DL-Lite knowledge bases. *J. Artif. Intell. Res.*, 64, 2019.

[Borgida *et al.*, 2008] Alexander Borgida, Diego Calvanese, and Mariano Rodriguez-Muro. Explanation in the DL-Lite family of description logics. In *OTM*, 2008.

[Bourgaux and Ozaki, 2019] Camille Bourgaux and Ana Ozaki. Querying attributed DL-Lite ontologies using provenance semirings. In *AAAI*, 2019.

[Bourgaux *et al.*, 2020] Camille Bourgaux, Ana Ozaki, Rafael Peñaloza, and Livia Predoiu. Provenance for the description logic ELHr, 2020. arXiv:2001.07541 [cs.LO].

[Buneman and Kostylev, 2010] Peter Buneman and Egor V. Kostylev. Annotation algebras for RDFS data. In *SWPM@ISWC*, 2010.

[Buneman, 2013] Peter Buneman. The providence of provenance. In *BNCOD*, 2013.

[Calvanese *et al.*, 2019] Diego Calvanese, Davide Lanti, Ana Ozaki, Rafael Peñaloza, and Guohui Xiao. Enriching ontology-based data access with provenance. In *IJCAI*, 2019.

[Ceylan *et al.*, 2019] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, and Andrius Vaicenavicius. Explanations for query answers under existential rules. In *IJCAI*, 2019.

[Cheney *et al.*, 2009] James Cheney, Laura Chiticariu, and Wang Chiew Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4), 2009.

[Croce and Lenzerini, 2018] Federico Croce and Maurizio Lenzerini. A framework for explaining query answers in DL-Lite. In *EKAW*, 2018.

[Dannert and Grädel, 2019] Katrin M. Dannert and Erich Grädel. Provenance analysis: A perspective for description logics? In *Description Logic, Theory Combination, and All That*, 2019.

[Deutch *et al.*, 2014] Daniel Deutch, Tova Milo, Sudeepa Roy, and Val Tannen. Circuits for datalog provenance. In *ICDT*, 2014.

[Green and Tannen, 2017] Todd J. Green and Val Tannen. The semiring framework for database provenance. In *PODS*, 2017.

[Green *et al.*, 2007] Todd J. Green, Gregory Karvounarakis, and Val Tannen. Provenance semirings. In *PODS*, 2007.

[Green, 2011] Todd J. Green. Containment of conjunctive queries on annotated relations. *Theory of Computing Systems*, 49, 2011.

[Hutschenreiter and Peñaloza, 2017] Lisa Hutschenreiter and Rafael Peñaloza. An automata view to goal-directed methods. In *LATA*, 2017.

[Ives *et al.*, 2008] Zachary G. Ives, Todd J. Green, Grigoris Karvounarakis, Nicholas E. Taylor, Val Tannen, Partha Pratim Talukdar, Marie Jacob, and Fernando C. N. Pereira. The ORCHESTRA collaborative data sharing system. *Sigmod Record*, 37, 2008.

[Kalyanpur *et al.*, 2007] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all justifications of OWL DL entailments. In *ISWC*, 2007.

[Lukasiewicz *et al.*, 2014] Thomas Lukasiewicz, Maria Vanina Martínez, Cristian Molinaro, Livia Predoiu, and Gerardo I. Simari. Answering ontological ranking queries based on subjective reports. In *SUM*, 2014.

[Lutz *et al.*, 2009] Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *IJCAI*, 2009.

[Peñaloza and Sertkaya, 2010] Rafael Peñaloza and Baris Sertkaya. On the complexity of axiom pinpointing in the EL family of description logics. In *KR*, 2010.

[Peñaloza *et al.*, 2017] Rafael Peñaloza, Carlos Mencía, Alexey Ignatiev, and João Marques-Silva. Lean kernels in description logics. In *ESWC*, 2017.

[Ramusat *et al.*, 2018] Yann Ramusat, Silviu Maniu, and Pierre Senellart. Semiring provenance over graph databases. In *TaPP*, 2018.

[Schlobach and Cornet, 2003] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI*, 2003.

[Senellart, 2017] Pierre Senellart. Provenance and probabilities in relational databases. *SIGMOD Record*, 46(4), 2017.

[Suciu *et al.*, 2011] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management, Morgan and Claypool Publishers, 2011.

[Zimmermann *et al.*, 2012] Antoine Zimmermann, Nuno Lopes, Axel Polleres, and Umberto Straccia. A general framework for representing, reasoning and querying with annotated semantic web data. *J. Web Sem.*, 11, 2012.