Introducing Nominals to the Combined Query Answering Approaches for $\mathcal{EL}$

Giorgio Stefanoni, Boris Motik, and Ian Horrocks

Department of Computer Science, University of Oxford

1 Introduction

Description logics (DLs) [1] are a family of knowledge representation formalisms that underpin OWL 2 [2]—an ontology language used in advanced information systems with many practical applications. Answering conjunctive queries (CQs) over ontology-enriched data sets is a core reasoning service in such systems, so the computational aspects of this problem have received a lot of interest lately. For expressive DLs, the problem is at least doubly exponential in query size [3]. The problem, however, becomes easier for the $\mathcal{EL}$ [4] and the DL-Lite [5] families of DLs, which provide the foundation for the OWL 2 EL and the OWL 2 QL profiles of OWL 2. An important goal of this research was to devise not only *worst-case optimal*, but also *practical* algorithms. The known approaches can be broadly classified as follows.

The first group consists of automata-based approaches for DLs such as OWL 2 EL [6] and Horn-$\mathcal{SHOIQ}$ and Horn-$\mathcal{SROIQ}$ [7]. While worst-case optimal, these approaches are typically not suitable for practice since their best-case and worst-case performance often coincide.

The second group consists of rewriting-based approaches. Roughly speaking, these approaches rewrite the ontology and/or the query into another formalism, typically a union of conjunctive queries or a datalog program; the relevant answers can then be obtained by evaluating the rewriting over the data. Rewriting-based approaches were developed for members of the DL-Lite family [5, 8], and the DLs $\mathcal{ELHIO}_\bot$ [9] and Horn-$\mathcal{SHIQ}$ [10], to name just a few. A common problem, however, is that rewritings can be exponential in the ontology and/or query size. Although this is often not a problem in practice, such approaches are not worst-case optimal. An exception is the algorithm by Rosati [11] that rewrites an $\mathcal{ELH}_\bot$ ontology into a datalog program of polynomial size; however, the algorithm also uses a nondeterministic step to transform the CQ into a tree-shaped one, and it is not clear how to implement this step in a goal-directed manner.

The third group consists of *combined approaches*, which use a three-step process: first, they augment the data with certain consequences of the ontology; second, they evaluate the CQ over the augmented data; and third, they filter the result of the second phase to eliminate unsound answers. The third step is necessary because, to ensure termination, the first step is unsound and may introduce facts that do not follow from the ontology; however, this is done in a way that makes the third step feasible. Such approaches have been developed for logics in the DL-Lite [12] and the $\mathcal{EL}$ [13] families, and they are appealing because they are worst-case optimal and practical: only the second step is intractable (in query size), but it can be solved using database techniques.

None of the combined approaches proposed thus far, however, handles *nominals*—concepts containing precisely one individual. Nominals are included in OWL 2 EL, and they are often used to state that all instances of a class have a certain property value, such as 'the sex of all men is male', or 'each German city is located in Germany'. In this paper we present a combined approach for $\mathcal{ELHO}_{\bot}^{r}$—the DL that covers all features of OWL 2 EL apart from transitive roles and complex role inclusions. To the best of our knowledge, this is the first combined approach that handles nominals. Our extension is nontrivial because nominals require equality reasoning, which increases the complexity of the first and the third step of the algorithm. In particular, nominals may introduce recursive dependencies in the filtering conditions used in the third phase; this is in contrast to the known combined approach for $\mathcal{EL}$ [13] in which filtering conditions are not recursive and can be incorporated into the input query. To solve this problem, our algorithm evaluates the original CQ and then uses a polynomial function to check the relevant conditions for each answer.

Following Krötzsch, Rudolph, and Hitzler [14], instead of directly materialising the relevant consequences of the ontology and the data, we transform the ontology into a datalog program that captures the relevant consequences. Although seemingly just a stylistic issue, a datalog-based specification may be beneficial in practice: one can either materialise all consequences of the program bottom-up in advance, or one can use a top-down technique to compute only the consequences relevant for the query at hand. The latter can be particularly useful in informations systems that have read-only access to the data, or where data changes frequently.

We have implemented a prototypical system using our algorithm, and we carried out a preliminary empirical evaluation of (*i*) the blowup in the number of facts introduced by the datalog program, and (*ii*) the number of unsound answers obtained in the second phase. Our experiments show both of these numbers to be manageable in typical cases, suggesting that our algorithm provides a practical basis for answering CQs in an expressive fragment of OWL 2 EL.

The proofs of our technical results are provided in the technical report [15].

2 Preliminaries

Logic Programming. We use the standard notions of variables, constants, function symbols, terms, atoms, formulas, and sentences [16]. We often identify a conjunction with the set of its conjuncts. A substitution σ is a partial mapping of variables to terms; $\mathsf{dom}(\sigma)$ and $\mathsf{rng}(\sigma)$ are the domain and the range of σ, respectively; $\sigma|_S$ is the restriction of σ to a set of variables S; and, for α a term or a formula, $\sigma(\alpha)$ is the result of simultaneously replacing each free variable x occurring in α with $\sigma(x)$. A *Horn clause* C is an expression of the form $B_1 \wedge \ldots \wedge B_m \rightarrow H$, where H and each B_i are atoms. Such C is a *fact* if $m = 0$, and it is commonly written as H. Furthermore, C is *safe* if each variable occurring in H also occurs in some B_i. A *logic program* Σ is a finite set of safe Horn clauses; furthermore, Σ is a *datalog program* if each clause in Σ is function-free.

In this paper, we interpret a logic program Σ in a model that can be constructed bottom-up. The *Herbrand universe* of Σ is the set of all terms built from the constants and the function symbols occurring in Σ. Given an arbitrary set of facts B, let $\Sigma(B)$ be the smallest superset of B such that, for each clause $\varphi \to \psi \in \Sigma$ and each substitution σ mapping the variables occurring in the clause to the Herbrand universe of Σ, if $\sigma(\varphi) \subseteq B$, then $\sigma(\psi) \subseteq \Sigma(B)$. Let I_0 be the set of all facts occurring in Σ; for each $i \in \mathbb{N}$, let $I_{i+1} = \Sigma(I_i)$; and let $I = \bigcup_{i \in \mathbb{N}} I_i$. Then I is the *minimal Herbrand model* of Σ, and it is well known that I satisfies $\forall \vec{x}.C$ for each Horn clause $C \in \Sigma$ and $\vec{x}$ the vector of all variables occurring in C.

In this paper we allow a logic program Σ to contain the equality predicate $\approx$. In first-order logic, $\approx$ is usually interpreted as the identity over the interpretation domain; however, $\approx$ can also be explicitly axiomatised [16]. Let $\Sigma_\approx$ be the set containing clauses (1)–(3), an instance of clause (4) for each n-ary predicate R occurring in Σ and each $1 \leq i \leq n$, and an instance of clause (5) for each n-ary function symbol f occurring in Σ and each $1 \leq i \leq n$.

$$\to x \approx x \tag{1}$$

$$x_1 \approx x_2 \to x_2 \approx x_1 \tag{2}$$

$$x_1 \approx x_2 \wedge x_2 \approx x_3 \to x_1 \approx x_3 \tag{3}$$

$$R(\vec{x}) \wedge x_i \approx x'_i \to R(x_1, \ldots, x'_i, \ldots, x_n) \tag{4}$$

$$x_i \approx x'_i \to f(\ldots, x_i, \ldots) \approx f(\ldots, x'_i, \ldots) \tag{5}$$

The minimal Herbrand model of a logic program Σ that contains $\approx$ is the minimal Herbrand model of $\Sigma \cup \Sigma_\approx$.

Conjunctive Queries. A *conjunctive query* (CQ) is a formula $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ with ψ a conjunction of function-free atoms over variables $\vec{x} \cup \vec{y}$. Variables $\vec{x}$ are the *answer variables* of q. Let $N_T(q)$ be the set of terms occurring in q.

For τ a substitution such that $\mathsf{rng}(\tau)$ contains only constants, let $\tau(q) = \exists \vec{z}.\tau(\psi)$, where $\vec{z}$ is obtained from $\vec{y}$ by removing each variable $y \in \vec{y}$ for which $\tau(y)$ is defined. Note that, according to this definition, non-free variables can also be replaced; for example, given $q = \exists y_1, y_2.R(y_1, y_2)$ and $\tau = \{y_2 \mapsto a\}$, we have $\tau(q) = \exists y_1.R(y_1, a)$.

Let Σ be a logic program, let I be the minimal Herbrand model of Σ, and let $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be a CQ that uses only the predicates occurring in Σ. A substitution π is a *candidate answer* for q in Σ if $\mathsf{dom}(\pi) = \vec{x}$ and $\mathsf{rng}(\pi)$ contains only constants; furthermore, such a π is a *certain answer* to q over Σ, written $\Sigma \models \pi(q)$, if a substitution τ exists such that $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$, $\pi = \tau|_{\vec{x}}$, and $\tau(q) \subseteq I$.

Description Logic. DL $\mathcal{ELHO}^r_\bot$ is defined w.r.t. a signature consisting of mutually disjoint and countably infinite sets N_C, N_R, and N_I of *atomic concepts* (i.e., unary predicates), *roles* (i.e., binary predicates), and *individuals* (i.e., constants), respectively. Furthermore, for each individual $a \in N_I$, expression $\{a\}$ denotes a *nominal*—that is, a concept containing precisely the individual a. Also, we assume that $\top$ and $\bot$ are unary predicates (without any predefined meaning) not occurring in N_C. We consider only *normalised* knowledge bases, as it is well known [4] that each $\mathcal{ELHO}^r_\bot$ KB can be normalised in polynomial time without affecting the answers to CQs. An $\mathcal{ELHO}^r_\bot$ *TBox* is a finite set of axioms of the form shown in the left-hand side of Table 1, where $A_{(i)} \in N_C \cup \{\top\}$, $B \in N_C \cup \{\top, \bot\}$, $R, S \in N_R$, and $a \in N_I$. An *ABox* $\mathcal{A}$ is a finite set of facts constructed using the symbols from $N_C \cup \{\top, \bot\}$, N_R, and N_I. Finally, an $\mathcal{ELHO}^r_\bot$ *knowledge base* (KB) is a tuple $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is an $\mathcal{ELHO}^r_\bot$ TBox $\mathcal{T}$ and an $\mathcal{A}$ is an ABox such that each predicate occurring in $\mathcal{A}$ also occurs in $\mathcal{T}$.

Table 1. Transforming $\mathcal{ELHO}^r_\bot$ Axioms into Horn Clauses

Type	Axiom		Clause
1	$\{a\} \sqsubseteq A$	$\leadsto$	$A(a)$
2	$A \sqsubseteq B$	$\leadsto$	$A(x) \to B(x)$
3	$A \sqsubseteq \{a\}$	$\leadsto$	$A(x) \to x \approx a$
4	$A_1 \sqcap A_2 \sqsubseteq A$	$\leadsto$	$A_1(x) \wedge A_2(x) \to A(x)$
5	$\exists R.A_1 \sqsubseteq A$	$\leadsto$	$R(x,y) \wedge A_1(y) \to A(x)$
6	$A_1 \sqsubseteq \exists R.A$	$\leadsto$	$A_1(x) \to R(x, f_{R,A}(x))$ $A_1(x) \to A(f_{R,A}(x))$
7	$R \sqsubseteq S$	$\leadsto$	$R(x,y) \to S(x,y)$
8	$\mathsf{range}(R, A)$	$\leadsto$	$R(x,y) \to A(y)$

We interpret $\mathcal{K}$ as a logic program. Table 1 shows how to translate a TBox $\mathcal{T}$ into a logic program $\Xi(\mathcal{T})$. Moreover, let $\top(\mathcal{T})$ be the set of the following clauses instantiated for each atomic concept A and each role R occurring in $\mathcal{T}$.

$$A(x) \to \top(x) \quad R(x,y) \to \top(x) \quad R(x,y) \to \top(y)$$

A KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is translated into the logic program $\Xi(\mathcal{K}) = \Xi(\mathcal{T}) \cup \top(\mathcal{T}) \cup \mathcal{A}$. Then, $\mathcal{K}$ is *unsatisfiable* if $\Xi(\mathcal{K}) \models \exists y.\bot(y)$. Furthermore, given a conjunctive query q and a candidate answer π for q, we write $\mathcal{K} \models \pi(q)$ iff $\mathcal{K}$ is unsatisfiable or $\Xi(\mathcal{K}) \models \pi(q)$. Given a candidate answer π for q, deciding whether $\Xi(\mathcal{K}) \models \pi(q)$ holds is NP-complete in *combined complexity*, and PTIME-complete in *data complexity* [6].

3 Datalog Rewriting of $\mathcal{ELHO}^r_\bot$ TBoxes

For the rest of this section, we fix an arbitrary $\mathcal{ELHO}^r_\bot$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. We next show how to transform $\mathcal{K}$ into a datalog program $\mathsf{D}(\mathcal{K})$ that can be used to check the satisfiability of $\mathcal{K}$. In the following section, we then show how to use $\mathsf{D}(\mathcal{K})$ to answer conjunctive queries.

Due to axioms of type 6 (cf. Table 1), $\Xi(\mathcal{K})$ may contain function symbols and is generally not a datalog program; thus, the evaluation of $\Xi(\mathcal{K})$ may not terminate. To ensure termination, we eliminate function symbols from $\Xi(\mathcal{K})$ using the technique by Krötzsch, Rudolph, and Hitzler [14]: for each $A \in N_C \cup \{\top\}$ and each $R \in N_R$ occurring in $\mathcal{T}$, we introduce a globally fresh and unique *auxiliary individual* $o_{R,A}$. Intuitively, $o_{R,A}$ represents all terms in the Herbrand universe of $\Xi(\mathcal{K})$ needed to satisfy the existential concept $\exists R.A$. Krötzsch, Rudolph, and Hitzler [14] used this technique to facilitate taxonomic reasoning, while we use it to obtain a practical CQ answering algorithm. Please note that $o_{R,A}$ depends on both R and A, whereas in the known approaches such individuals depend only on A [13] or R [12].

Definition 1. *Datalog program $\mathsf{D}(\mathcal{T})$ is obtained by translating each axiom of type other than 6 in the TBox $\mathcal{T}$ of $\mathcal{K}$ into a clause as shown in Table 1, and by translating each axiom $A_1 \sqsubseteq \exists R.A$ in $\mathcal{T}$ into clauses $A_1(x) \to R(x, o_{R,A})$ and $A_1(x) \to A(o_{R,A})$. Furthermore, the translation of $\mathcal{K}$ into datalog is given by $\mathsf{D}(\mathcal{K}) = \mathsf{D}(\mathcal{T}) \cup \top(\mathcal{T}) \cup \mathcal{A}$.*

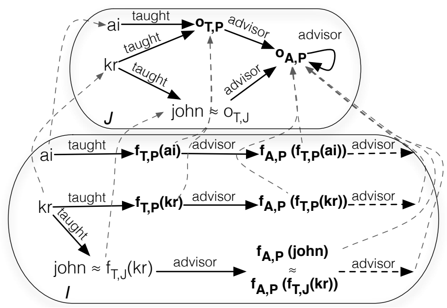

Fig. 1. Representing the Models of $\Xi(\mathcal{K})$.

Example 1. Let $\mathcal{T}$ be the following $\mathcal{ELHO}^r_\bot$ TBox:

$$\begin{array}{rcl} KRC \sqsubseteq \exists taught.JProf & JProf \sqsubseteq \{john\} & KRC \sqsubseteq Course \\ Course \sqsubseteq \exists taught.Prof & \exists taught.\top \sqsubseteq Course & \mathsf{range}(taught, Prof) \\ Prof \sqsubseteq \exists advisor.Prof & \{kr\} \sqsubseteq KRC & \end{array}$$

Then, $\mathsf{D}(\mathcal{T})$ contains the following clauses:

$$\begin{array}{lll} KRC(x) \to taught(x, o_{T,J}) & Prof(x) \to Prof(o_{A,P}) & KRC(kr) \\ KRC(x) \to JProf(o_{T,J}) & Prof(x) \to advisor(x, o_{A,P}) & KRC(x) \to Course(x) \\ Course(x) \to taught(x, o_{T,P}) & Prof(x) \to Prof(o_{A,P}) & taught(x,y) \to Prof(y) \\ Course(x) \to Prof(o_{T,P}) & JProf(x) \to x \approx john & \\ Prof(x) \to advisor(x, o_{A,P}) & taught(x,y) \to Course(x) & \end{array}$$ ◊

The following result readily follows from the definition of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$.

Proposition 1. *Program $\mathsf{D}(\mathcal{K})$ can be computed in time linear in the size of $\mathcal{K}$.*

Next, we prove that the datalog program $\mathsf{D}(\mathcal{K})$ can be used to decide the satisfiability of $\mathcal{K}$. To this end, we define a function δ that maps each term w in the Herbrand universe of $\Xi(\mathcal{K})$ to the Herbrand universe of $\mathsf{D}(\mathcal{K})$ as follows:

$$\delta(w) = \begin{cases} w & \text{if } w \in N_I, \\ o_{R,A} & \text{if } w \text{ is of the form } w = f_{R,A}(w'). \end{cases}$$

Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Mapping δ establishes a tight relationship between I and J as illustrated in the following example.

Example 2. Let $\mathcal{A} = \{Course(ai)\}$, let $\mathcal{T}$ be as in Example 1, and let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. Figure 1 shows a graphical representation of the minimal Herbrand models I and J of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. The grey dotted lines show how δ relates the terms in I to the terms in J. For the sake of clarity, Figure 1 does not show the reflexivity of $\approx$. ◊

Mapping δ is a homomorphism from I to J.

Lemma 1. *Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Mapping δ satisfies the following three properties for all terms w' and w, each $B \in N_C \cup \{\top, \bot\}$, and each $R \in N_R$.*

1. *$B(w) \in I$ implies $B(\delta(w)) \in J$.*
2. *$R(w', w) \in I$ implies $R(\delta(w'), \delta(w)) \in J$.*
3. *$w' \approx w \in I$ implies $\delta(w') \approx \delta(w) \in J$.*

For a similar result in the other direction, we need a couple of definitions. Let H be an arbitrary Herbrand model. Then, $\mathsf{dom}(H)$ is the set containing each term w that occurs in H in at least one fact with a predicate in $N_C \cup \{\top, \bot\} \cup N_R$; note that, by this definition, we have $w \notin \mathsf{dom}(H)$ whenever w occurs in H only in assertions involving the $\approx$ predicate. Furthermore, aux_H is the set of all terms $w \in \mathsf{dom}(H)$ such that, for each term w' with $w \approx w' \in H$, we have $w' \notin N_I$. We say that the terms in aux_H are 'true' auxiliary terms—that is, they are not equal to an individual in N_I. In Figure 1, bold terms are 'true' auxiliary terms in I and J.

Lemma 2. *Let I and J be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively. Mapping δ satisfies the following five properties for all terms w_1 and w_2 in $\mathsf{dom}(I)$, each $B \in N_C \cup \{\top, \bot\}$, and each $R \in N_R$.*

1. *$B(\delta(w_1)) \in J$ implies that $B(w_1) \in I$.*
2. *$R(\delta(w_1), \delta(w_2)) \in J$ and $\delta(w_2) \notin \mathsf{aux}_J$ imply that $R(w_1, w_2) \in I$.*
3. *$R(\delta(w_1), \delta(w_2)) \in J$ and $\delta(w_2) \in \mathsf{aux}_J$ imply that $\delta(w_2)$ is of the form $o_{P,A}$, that $R(w_1, f_{P,A}(w_1)) \in I$, and that a term w'_1 exists such that $R(w'_1, w_2) \in I$.*
4. *$\delta(w_1) \approx \delta(w_2) \in J$ and $\delta(w_2) \notin \mathsf{aux}_J$ imply that $w_1 \approx w_2 \in I$.*
5. *For each term u occurring in J, term $w \in \mathsf{dom}(I)$ exists such that $\delta(w) = u$.*

Lemmas 1 and 2 allow us to decide the satisfiability of $\mathcal{K}$ by answering a simple query over $\mathsf{D}(\mathcal{K})$, as shown in Proposition 2. The complexity claim is due to the fact that each clause in $\mathsf{D}(\mathcal{K})$ contains a bounded number of variables [17].

Proposition 2. *For $\mathcal{K}$ an arbitrary $\mathcal{ELHO}^r_\bot$ knowledge base, $\Xi(\mathcal{K}) \models \exists y.\bot(y)$ if and only if $\mathsf{D}(\mathcal{K}) \models \exists y.\bot(y)$. Furthermore, the satisfiability of $\mathcal{K}$ can be checked in time polynomial in the size of $\mathcal{K}$.*

4 Answering Conjunctive Queries

In this section, we fix a satisfiable $\mathcal{ELHO}^r_\bot$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a conjunctive query $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$. Furthermore, we fix I and J to be the minimal Herbrand models of $\Xi(\mathcal{K})$ and $\mathsf{D}(\mathcal{K})$, respectively.

While $\mathsf{D}(\mathcal{K})$ can be used to decide the satisfiability of $\mathcal{K}$, the following example shows that $\mathsf{D}(\mathcal{K})$ cannot be used directly to compute the answers to q.

Example 3. Let $\mathcal{K}$ be as in Example 2, and let q_1, q_2, and q_3 be the following CQs:

$$q_1 = taught(x_1, x_2)$$

$$q_2 = \exists y_1, y_2, y_3.\ taught(x_1, y_1) \wedge taught(x_2, y_2) \wedge advisor(y_1, y_3) \wedge advisor(y_2, y_3)$$

$$q_3 = \exists y.\ advisor(y, y)$$

Furthermore, let τ_i be the following substitutions:

$$\tau_1 = \{x_1 \mapsto kr,\ x_2 \mapsto o_{T,P}\}$$
$$\tau_2 = \{x_1 \mapsto kr,\ x_2 \mapsto ai,\ y_1 \mapsto o_{T,P},\ y_2 \mapsto o_{T,P},\ y_3 \mapsto o_{A,P}\}$$
$$\tau_3 = \{y \mapsto o_{A,P}\}$$

Finally, let each π_i be the projection of τ_i to the answer variables of q_i. Using Figure 1, one can readily check that $\mathsf{D}(\mathcal{K}) \models \tau_i(q_i)$, but $\Xi(\mathcal{K}) \not\models \pi_i(q_i)$, for each $1 \leq i \leq 3$. $\diamondsuit$

This can be explained by observing that J is a homomorphic image of I. Now homomorphisms preserve CQ answers (i.e., $\Xi(\mathcal{K}) \models \pi(q)$ implies $\mathsf{D}(\mathcal{K}) \models \pi(q)$), but they can also introduce unsound answers (i.e., $\mathsf{D}(\mathcal{K}) \models \pi(q)$ does not necessarily imply $\Xi(\mathcal{K}) \models \pi(q)$). This gives rise to the following notion of spurious answers.

Definition 2. *A substitution τ with $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$ and $\mathsf{D}(\mathcal{K}) \models \tau(q)$ is a* spurious answer *to q if $\tau|_{\vec{x}}$ is not a certain answer to q over $\Xi(\mathcal{K})$.*

Based on these observations, we answer q over $\mathcal{K}$ in two steps: first, we evaluate q over $\mathsf{D}(\mathcal{K})$ and thus obtain an overestimation of the certain answers to q over $\Xi(\mathcal{K})$; second, for each substitution τ obtained in the first step, we eliminate spurious answers using a special function isSpur. We next formally introduce this function. We first present all relevant definitions, after which we discuss the intuitions. As we shall see, each query in Example 3 illustrates a distinct source of spuriousness that our function needs to deal with.

Definition 3. *Let τ be a substitution s.t. $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$ and $\mathsf{D}(\mathcal{K}) \models \tau(q)$. Relation $\sim\, \subseteq N_T(q) \times N_T(q)$ for q, τ, and $\mathsf{D}(\mathcal{K})$ is the smallest reflexive, symmetric, and transitive relation closed under the* fork *rule, where $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$ is the set containing each individual u from $\mathsf{D}(\mathcal{K})$ for which no individual $c \in N_I$ exists such that $\mathsf{D}(\mathcal{K}) \models u \approx c$.*

$$(\mathsf{fork})\ \frac{s' \sim t'}{s \sim t}\quad R(s, s') \text{ and } P(t, t') \text{ occur in } q\text{, and } \tau(s') \in \mathsf{aux}_{\mathsf{D}(\mathcal{K})}$$

Please note that the definition $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$ is actually a reformulation of the definition of aux_J, but based on the consequences of $\mathsf{D}(\mathcal{K})$ rather than the facts in J.

Relation $\sim$ is reflexive, symmetric, and transitive, so it is an equivalence relation, which allows us to normalise each term $t \in N_T(q)$ to a representative of its equivalence class using the mapping γ defined below. We then construct a graph G_{aux} that checks whether substitution τ matches 'true' auxiliary individuals in a way that cannot be converted to a match over 'true' auxiliary terms in I.

Definition 4. *Let τ and $\sim$ be as specified in Definition 3. Function $\gamma : N_T(q) \mapsto N_T(q)$ maps each term $t \in N_T(q)$ to an arbitrary, but fixed representative $\gamma(t)$ of the equivalence class of $\sim$ that contains t. Furthermore, the directed graph $G_{\mathsf{aux}} = \langle V_{\mathsf{aux}}, E_{\mathsf{aux}} \rangle$ is defined as follows.*

- *Set V_{aux} contains a vertex $\gamma(t)$ for each term $t \in N_T(q)$ such that $\tau(t) \in \mathsf{aux}_{\mathsf{D}(\mathcal{K})}$.*
- *Set E_{aux} contains an edge $\langle \gamma(s), \gamma(t) \rangle$ for each atom of the form $R(s, t)$ in q such that $\{\gamma(s),\ \gamma(t)\} \subseteq V_{\mathsf{aux}}$.*

Query q is aux-cyclic w.r.t. τ and $\mathsf{D}(\mathcal{K})$ *if G_{aux} contains a cycle; otherwise, q is* aux-acyclic w.r.t. τ and $\mathsf{D}(\mathcal{K})$.

We are now ready to define our function that checks whether a substitution τ is a spurious answer.

Definition 5. *Let τ and $\sim$ be as specified in Definition 3. Function* $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau)$ *returns* $\mathtt{t}$ *if and only if at least one of the following conditions hold.*

(a) Variable $x \in \vec{x}$ exists such that $\tau(x) \notin N_I$.
(b) Terms s and t occurring in q exist such that $s \sim t$ and $\mathsf{D}(\mathcal{K}) \not\models \tau(s) \approx \tau(t)$.
(c) Query q is aux-cyclic w.r.t. τ and $\mathsf{D}(\mathcal{K})$.

We next discuss the intuition behind our definitions. We ground our discussion in minimal Herbrand models I and J, but our technique does not depend on such models: all conditions are stated as entailments that can be checked using an arbitrary sound and complete technique. Since $\mathcal{K}$ is an $\mathcal{ELHO}^r_\bot$ knowledge base, model I is *forest-shaped*: roughly speaking, the role assertions in I that involve at least one functional term are of the form $R(w_1, f_{R,A}(w_1))$ or $R(w_1, a)$ for $a \in N_I$; thus, I can be viewed as a family of directed trees whose roots are the individuals in N_I and whose edges point from parents to children or to the individuals in N_I. This is illustrated in Figure 1, whose lower part shows the the forest-model of the knowledge base from Example 3. Note that assertions of the form $R(w_1, a)$ are introduced via equality reasoning.

Now let τ be a substitution such that $\mathsf{D}(\mathcal{K}) \models \tau(q)$, and let $\pi = \tau|_{\vec{x}}$. If τ is not a spurious answer, it should be possible to convert τ into a substitution π^* such that $\pi = \pi^*|_{\vec{x}}$ and $\pi^*(q) \subseteq I$. Using the queries from Example 3, we next identify three reasons why this may not be possible.

First, τ may map an answer variable of q to an auxiliary individual, so by the definition π cannot be a certain answer to q; condition (a) of Definition 5 identifies such cases. Query q_1 and substitution τ_1 from Example 3 illustrate such a situation: $\tau_2(x_2) = o_{T,P}$ and $o_{T,P}$ is a 'true' auxiliary individual, so π_1 is not a certain answer to q_1.

The remaining two problems arise because model J is not forest-shaped, so τ might map q into J in a way that cannot be converted into a substitution π^* that maps q into I.

The second problem is best explained using substitution τ_2 and query q_2 from Example 3. Query q_2 contains a 'fork' $advisor(y_1, y_3) \wedge advisor(y_2, y_3)$. Now, substitution τ_2 maps y_3 to 'true' auxiliary individual $o_{A,P}$ which represents 'true' auxiliary terms $f_{A,P}(f_{T,P}(ai))$, $f_{A,P}(f_{T,P}(kr))$, and so on. Since I is forest-shaped, a match π_2^* for q in I obtained from τ_2 would need to map y_3 to one of these terms; let us assume that $\pi_2^*(y_3)$ is mapped to $f_{A,P}(f_{T,P}(ai))$. Since I is forest-shaped and $f_{A,P}(f_{T,P}(ai))$ is a 'true' auxiliary term, this means that both y_1 and y_2 must be mapped to the same term (in both J and I). This is captured by the (fork) rule: in our example, the rule derives $y_1 \sim y_2$, and condition (b) of Definition 5 checks whether τ_2 maps y_1 and y_2 in a way that satisfies this constraint. Note that, due to role hierarchies, the rule needs to be applied to atoms $R(s, s')$ and $P(t, t')$ with $R \neq P$. Moreover, such constraints must be propagated further up the query. In our example, due to $y_1 \sim y_2$, atoms $taught(x_1, y_1) \wedge taught(x_2, y_2)$ in q_2 also constitute a 'fork', so the rule derives $x_1 \sim x_2$; this allows condition (b) of Definition 5 to correctly identify τ_2 as spurious.

Table 2. Size of the materialisations.

	Individuals (% in $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$)		Unary facts (% over $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$)		Binary facts (% over $\mathsf{aux}_{\mathsf{D}(\mathcal{K})}$)	
L-5	100848		169079		296941	
Mat.	100868	(0.01)	309350	(0.01)	632489	(49.2)
L-10	202387		339746		598695	
Mat.	202407	(0.01)	621158	(0.01)	1277575	(49.3)
L-20	426144		714692		1259936	
Mat.	426164	(0.01)	1304815	(0.01)	2691766	(49.3)
SEM	17945		17945		47248	
Mat.	17953	(0.04)	25608	(0.03)	76590	(38.3)

The third problem is best explained using substitution τ_3 and query q_3 from Example 3. Model J contains a 'loop' on individual $o_{A,P}$, which allows τ_3 to map q_3 into J. In contrast, model I is forest-shaped, and so the 'true' auxiliary terms that correspond to $o_{A,P}$ do not form loops. Condition (c) of Definition 5 detects such situations using the graph G_{aux}. The vertices of G_{aux} correspond to the terms of q that are matched to 'true' auxiliary individuals (mapping γ simply ensures that equal terms are represented as one vertex), and edges of G_{aux} correspond to the role atoms in q. Hence, if G_{aux} is cyclic, then the substitution π^* obtained from τ would need to match the query q over a cycle of 'true' auxiliary terms, which is impossible since I is forest-shaped.

Unlike the known combined approaches, our approach does not extend q with conditions that detect spurious answers. Due to nominals, the relevant equality constraints have a recursive nature, and they depend on both the substitution τ and on the previously derived constraints. Consequently, filtering in our approach is realised as postprocessing; furthermore, to ensure correctness of our filtering condition, auxiliary individuals must depend on both a role and an atomic concept. The following theorem proves the correctness of our approach.

Theorem 1. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a satisfiable $\mathcal{ELHO}^r_\bot$ KB, let $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be a CQ, and let $\pi : \vec{x} \mapsto N_I$ be a candidate answer for q. Then, $\Xi(\mathcal{K}) \models \pi(q)$ iff a substitution τ exists such that $\mathsf{dom}(\tau) = \vec{x} \cup \vec{y}$, $\tau|_{\vec{x}} = \pi$, $\mathsf{D}(\mathcal{K}) \models \tau(q)$, and $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau) = \mathsf{f}$.*

Furthermore, $\mathsf{isSpur}(q, \mathsf{D}(\mathcal{K}), \tau)$ can be evaluated in polynomial time, so the main source of complexity in our approach is in deciding whether $\mathsf{D}(\mathcal{K}) \models \tau(q)$ holds. This gives rise to the following result.

Theorem 2. *Deciding whether $\mathcal{K} \models \pi(q)$ can be implemented in nondeterministic polynomial time w.r.t. the size of $\mathcal{K}$ and q, and in polynomial time w.r.t. the size of $\mathcal{A}$.*

5 Evaluation

To gain insight into the practical applicability of our approach, we implemented our technique in a prototypical system. The system uses HermiT, a widely used ontology reasoner, as a datalog engine in order to materialise the consequences of $\mathsf{D}(\mathcal{K})$ and evaluate q. The system has been implemented in Java, and we ran our experiments on a MacBook Pro with 4GB of RAM and an Intel Core 2 Duo 2.4 Ghz processor. We used two ontologies in our evaluation, details of which are given below. The ontologies, the queries, and the prototype are available at http://www.cs.ox.ac.uk/isg/tools/KARMA/.

The LSTW benchmark [18] consists of an OWL 2 QL version of the LUBM ontology [19], queries $q_1^l, \ldots, q_{11}^l$, and a data generator. The LSTW ontology extends the standard LUBM ontology with several axioms of type 6 (see Table 1). To obtain an $\mathcal{ELHO}_\bot^r$ ontology, we removed inverse roles and datatypes, added 11 axioms using 9 freshly introduced nominals, and added one axiom of type 4 (see Table 1). These additional axioms resemble the ones in Example 1, and they were designed to test equality reasoning. The resulting signature consists of 132 concepts, 32 roles, and 9 nominals, and the ontology contains 180 axioms. From the 11 LSTW queries, we did not consider queries q_4^l, q_6^l, q_7^l, and q_{11}^l because their result sets were empty: q_4^l relies on existential quantification over inverse roles, and the other three are empty already w.r.t. the original LSTW ontology. Query q_2^l is similar to query q_2 from Example 3, and it was designed to produce only spurious answers and thus stress the system. We generated data sets with 5, 10 and 20 universities. For each data set, we denote with L-i the knowledge base consisting of our $\mathcal{ELHO}_\bot^r$ ontology and the ABox for i universities (see Table 2).

SEMINTEC is an ontology about financial services developed within the SEMINTEC project at the University of Poznan. To obtain an $\mathcal{ELHO}_\bot^r$ ontology, we removed inverse roles, role functionality axioms, and universal restrictions, added nine axioms of type 6 (see Table 1), and added six axioms using 4 freshly introduced nominals. The resulting ontology signature consists of 60 concepts, 16 roles, and 4 nominals, and the ontology contains 173 axioms. Queries q_1^s–q_5^s are tree-shaped queries used in the SEMINTEC project, and we developed queries q_6^s–q_9^s ourselves. Query q_6^s resembles query q_2^l from LSTW, and queries q_8^s and q_9^s were designed to retrieve a large number of answers containing auxiliary individuals, thus stressing condition (a) of Definition 5. Finally, the SEMINTEC ontology comes with a data set consisting of approximately 65,000 facts concerning 18,000 individuals (see row SEM in Table 2).

The practicality of our approach, we believe, is determined mainly by the following two factors. First, the number of facts involving auxiliary individuals introduced during the materialisation step should not be 'too large'. Table 2 shows the materialisation results: the first column shows the number of individuals before and after materialisation and the percentage of 'true' auxiliary individuals, the second column shows the number of unary facts before and after materialisation and the percentage of facts involving a 'true' auxiliary individual, and the third column does the same for binary facts. As one can see, for each data set, the materialisation introduces few 'true' auxiliary individuals, and the number of facts at most doubles. The number of unary facts involving a 'true' auxiliary individual does not change with the size of the input data set, whereas the number of such binary facts increases by a constant factor. This is because, in clauses of type 6, atoms $A(o_{R,A})$ do not contain a variable, whereas atoms $R(x, o_{R,A})$ do.

Second, evaluating q over $\mathsf{D}(\mathcal{K})$ should not produce too many spurious answers. Table 3 shows the total number of answers for each query—that is, the number of answers obtained by evaluating the query over $\mathsf{D}(\mathcal{K})$; moreover, the table shows what percentage

Table 3. Total number of answers and ratio spurious to answers. In Table LSTW, the ratio is stable for each data set. The bottom table displays results for the SEMINTEC ontology.

LSTW	q_1^l Tot (%)	q_2^l Tot (%)	q_3^l Tot (%)	q_5^l Tot (%)	q_8^l Tot (%)	q_9^l Tot (%)	q_{10}^l Tot (%)
L-5	116K	3.7M	10	28K	13K	1K	12K
L-10	233K (4.0)	32M (100.0)	22 (0.0)	57K (0.0)	26K (26.0)	2K (0.0)	25K (74.5)
L-20	487K	170M	43	121K	55K	4K	53K

q_1^s Tot (%)	q_2^s Tot (%)	q_3^s Tot (%)	q_4^s Tot (%)	q_5^s Tot (%)	q_6^s Tot (%)	q_7^s Tot (%)	q_8^s Tot (%)	q_9^s Tot (%)
7 (0.0)	53 (0.0)	16 (0.0)	12 (0.0)	31 (0.0)	838K (55.4)	5K (0.0)	5K (54.3)	13K (33.3)

of these answers are spurious. Queries q_2^l, q_{10}^l, q_6^s, and q_8^s retrieve a significant percentage of spurious answers. However, only query q_2^l has proven to be challenging for our system due to the large number of retrieved answers, with an evaluation time of about 40 minutes over the largest knowledge base (L-20). Surprisingly, q_1^l also performed rather poorly despite a low number of spurious answers, with an evaluation time of about 20 minutes for L-20. All other queries were evaluated in at most a few seconds, thus suggesting that queries q_1^l and q_2^l are problematical mainly because HermiT does not implement query optimisation algorithms typically used in relational databases.

6 Conclusion

We presented the first combined technique for answering conjunctive queries over DL ontologies that include nominals. A preliminary evaluation suggests the following. First, the number of materialised facts over 'true' anonymous individuals increases by a constant factor with the size of the data. Second, query evaluation results have shown that, while some cases may be challenging, in most cases the percentage of answers that are spurious is manageable. Hence, our technique provides a practical CQ answering algorithm for a large fragment of OWL 2 EL.

We anticipate several directions for our future work. First, we would like to investigate the use of top-down query evaluation techniques, such as magic sets [20] or SLG resolution [21]. Second, tighter integration of the detection of spurious answers with the query evaluation algorithms should make it possible to eagerly detect spurious answers (i.e., before the query is fully evaluated). Lutz et al. [18] already implemented a filtering condition as a user-defined function in a database, but it is unclear to what extent such an implementation can be used to optimise query evaluation. Finally, we would like to extend our approach to all of OWL 2 EL.

Acknowledgements

This work was supported by the Royal Society; Alcatel-Lucent; the EU FP7 project OPTIQUE; and the EPSRC projects ExODA, MASI^3, and QueRe.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2007) ISBN 9780511717383.
2. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. Journal of Web Semantics **6**(4) (2008) 309–322
3. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$. Journal of Artificial Intelligence Research **31** (2008) 151–198
4. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope. In Kaelbling, L.P., Saffiotti, A., eds.: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinburgh, UK, Morgan Kaufmann Publishers (July 30–August 5 2005) 364–369
5. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. Journal of Automated Reasoning **9**(3) (2007) 385–429
6. Krötzsch, M., Rudolph, S., Hitzler, P.: Conjunctive queries for a tractable fragment of OWL 1.1. In Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P., eds.: Proceedings of the 6th International Semantic Web Conference (ISWC'07). Volume 4825 of LNCS., Springer (2007) 310–323
7. Ortiz, M., Rudolph, S., Simkus, M.: Query Answering in the Horn Fragments of the Description Logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In Walsh, T., ed.: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), Barcelona, Spain, AAAI Press (July 16–22 2011) 1039–1044
8. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite Family and Relations. Journal of Artificial Intelligence Research **36** (2009) 1–69
9. Pérez-Urbina, H., Motik, B., Horrocks, I.: Tractable Query Answering and Rewriting under Description Logic Constraints. Journal of Applied Logic **8**(2) (2010) 186–209
10. Eiter, T., Ortiz, M., Simkus, M., Tran, T.K., Xiao, G.: Query Rewriting for Horn-SHIQ Plus Rules. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, (AAAI 2012), AAAI Press (2012)
11. Rosati, R.: On Conjunctive Query Answering in $\mathcal{EL}$. In Calvanese, D., Franconi, E., Haarslev, V., Lembo, D., Motik, B., Turhan, A.Y., Tessaris, S., eds.: Proceedings of the 20th International Workshop on Description Logics (DL-2007). CEUR Workshop Proceedings, CEUR-WS.org (2007)
12. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The Combined Approach to Ontology-Based Data Access. In Walsh, T., ed.: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), AAAI Press (2011)
13. Lutz, C., Toman, D., Wolter, F.: Conjunctive Query Answering in the Description Logic $\mathcal{EL}$ Using a Relational Database System. In Boutilier, C., ed.: Proceedings of the 21st International Joint Conference on Artificial Intelligence, (IJCAI 2009), AAAI Press (2009) 2070–2075
14. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable rules for OWL 2. In Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K., eds.: Proceedings of the 7th International Semantic Web Conference (ISWC'08). Volume 5318 of LNCS., Springer (2008) 649–664
15. Stefanoni, G., Motik, B., Horrocks, I.: Introducing Nominals to the Combined Query Answering Approaches for EL. CoRR **abs/1303.7430** (2013)
16. Fitting, M.: First-order logic and automated theorem proving (2nd ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA (1996)

17. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. ACM Computing Surveys **33**(3) (2001) 374–425
18. Lutz, C., Seylan, I., Toman, D., Wolter, F.: The Combined Approach to OBDA: Taming Role Hierarchies using Filters. In Fokoue, A., Liebig, T., Goodman, E., Weaver, J., Urbani, J., Mizell, D., eds.: Proceedings of the Joint Workshop on Scalable and High-Performance Semantic Web Systems (SSWS+HPCSW 2012). Volume 943 of CEUR Workshop Proceedings., CEUR-WS.org (2012) 16–31
19. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. Journal of Web Semantics **3**(2–3) (2005) 158–182
20. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
21. Chen, W., Warren, D.S.: Query evaluation under the well-founded semantics. In: Proceedings of the 12th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. PODS '93, New York, NY, USA, ACM (1993) 168–179