The Combined Approach to Query Answering Beyond the OWL 2 Profiles*

Cristina Feier[1], David Carral[2], Giorgio Stefanoni[1], Bernardo Cuenca Grau[1], Ian Horrocks[1]

[1] Department of Computer Science
University of Oxford, Oxford UK
firstname.lastname@cs.ox.ac.uk

[2] Department of Computer Science
Wright State University, Dayton US
carral.2@wright.edu

Abstract

Combined approaches have become a successful technique for CQ answering over ontologies. Existing algorithms, however, are restricted to the logics underpinning the OWL 2 profiles. Our goal is to make combined approaches applicable to a wider range of ontologies. We focus on RSA: a class of Horn ontologies that extends the profiles while ensuring tractability of standard reasoning. We show that CQ answering over RSA ontologies without role composition is feasible in NP. Our reasoning procedure generalises the combined approach for $\mathcal{ELHO}$ and DL-Lite$_\mathcal{R}$ using an encoding of CQ answering into fact entailment w.r.t. a logic program with function symbols and stratified negation. Our results have significant practical implications since many out-of-profile Horn ontologies are RSA.

1 Introduction

Answering conjunctive queries (CQs) over ontology-enriched datasets is a core reasoning task for many applications. CQ answering is computationally expensive: for expressive description logic ontology languages it is at least doubly exponential in combined complexity [Lutz, 2007], and it remains single exponential even when restricted to Horn ontology languages [Ortiz *et al.*, 2011].

In recent years, there has been a growing interest in ontology languages with favourable computational properties, such as $\mathcal{EL}$ [Baader *et al.*, 2005], DL-Lite [Calvanese *et al.*, 2007] and the rule language datalog, which provide the foundation for the EL, QL and RL profiles of OWL 2, respectively [Motik *et al.*, 2009]. Standard reasoning tasks (e.g., satisfiability checking) are tractable for all three profiles. CQ answering is NP-complete (in combined complexity) for the QL and RL profiles, and PSPACE-complete for OWL 2 EL [Stefanoni *et al.*, 2014]; PSPACE-hardness of CQ answering in EL is due to role composition axioms and the complexity further drops to NP if these are restricted to express role transitivity and reflexivity [Stefanoni and Motik, 2015]. Furthermore, in all these cases CQ answering is tractable in data complexity. Such complexity bounds are rather benign, and this has spurred the development of practical algorithms.

A technique that is receiving increasing attention is the *combined approach* [Lutz *et al.*, 2009; Kontchakov *et al.*, 2010; 2011; Lutz *et al.*, 2013; Stefanoni *et al.*, 2013], which can be summarised as follows. First, the data is augmented in a query-independent way to build (in polynomial time) a canonical interpretation. Although this interpretation cannot be homomorphically embedded into each model (and might not be a model itself), it can be exploited for CQ answering in two equivalent ways. In the approach by [Kontchakov *et al.*, 2010] the query is first rewritten and then evaluated against the interpretation. Alternatively, in [Stefanoni *et al.*, 2013; Lutz *et al.*, 2013] the query is first evaluated over the interpretation and unsound answers are discarded by means of a *filtration* process. With the exception of [Gottlob *et al.*, 2014] and [Thomazo and Rudolph, 2014] who focus on decidable classes of existential rules, algorithms based on the combined approach are restricted to (fragments of) the OWL 2 profiles.

Our goal is to push the boundaries of the logics underpinning the OWL 2 profiles while retaining their nice complexity for CQ answering. Furthermore, we aim to devise algorithms that seamlessly extend the combined approach and which can be applied to a wide range of ontologies.

Recently, a class of Horn ontologies has been proposed [Carral *et al.*, 2014a; 2014b] that extends the profiles and cannot be captured by known decidable classes of existential rules, while ensuring tractability of standard reasoning tasks. The idea is to allow the use of all language constructs in the profiles, while establishing polynomially checkable conditions that preclude their harmful interaction. Ontologies satisfying these conditions are referred to as *role safety acyclic* (RSA). The roles in an RSA ontology are partitioned into *safe* and *unsafe* depending on the way they are used, where the latter ones are involved in potentially harmful interactions which could increase complexity; then, an *acyclicity* condition is imposed on unsafe roles to ensure tractability. A recent evaluation revealed that over 60% of out-of-profile Horn ontologies are RSA [Carral *et al.*, 2014b].

In this paper, we investigate CQ answering over RSA ontologies and show its feasibility in NP. This result has significant implications in practice as it shows that CQ answering over a wide range of out-of-profile ontologies is no harder (in combined complexity) than over a database. Our procedure

*Work supported by the Royal Society, the EPSRC grants Score!, DBOnto and MaSI[3], the NSF award 1017255 "III: Small: TROn: Tractable Reasoning with Ontologies" and "La Caixa" Foundation.

generalises the combined approach for $\mathcal{ELHO}$ [Stefanoni *et al.*, 2013] and DL-Lite$_\mathcal{R}$ [Lutz *et al.*, 2013] in a seamless way by means of a declarative encoding of CQ answering into fact entailment w.r.t. a logic program (LP) with function symbols and stratified negation. The least Herbrand model of this program can be computed in time polynomial in the ontology size and exponential in query size. We have implemented our encoding using the LP engine DLV [Leone *et al.*, 2006] and tested its feasibility with encouraging results.

The proofs of all our results are delegated to an extended technical report (http://tinyurl.com/pqmxa5u).

2 Preliminaries

Logic Programs We use the standard notions of constants, terms and atoms in first-order logic (FO). A *literal* is an atom a or its negation $not\ a$. A rule r is an expression of the form $\varphi(\vec{x},\vec{z}) \to \psi(\vec{x})$ with $\varphi(\vec{x},\vec{z})$ a conjunction of literals with variables $\vec{x} \cup \vec{z}$, and $\psi(\vec{x})$ a non-empty conjunction of atoms over $\vec{x}$.[1] We denote with $vars(r)$ the set $\vec{x} \cup \vec{z}$. With $head(r)$ we denote the set of atoms in ψ, $body^+(r)$ is the set of atoms in φ, and $body^-(r)$ is the set of atoms which occur negated in r. Rule r is safe iff $vars(r)$ all occur in $body^+(r)$. We consider only safe rules. Rule r is *definite* if $body^-(r)$ is empty and it is *datalog* if it is definite and function-free. A *fact* is a rule with empty body and head consisting of a single function-free atom.

A program $\mathcal{P}$ is a finite set of rules. Let $preds(X)$ denote the predicates in X, with X either a set of atoms or a program. A *stratification* of program $\mathcal{P}$ is a function $str : preds(\mathcal{P}) \to \{1,\ldots,k\}$, where $k \le |preds(\mathcal{P})|$, s.t. for every $r \in \mathcal{P}$ and $P \in preds(head(r))$ it holds that:

- for every $Q \in preds(body^+(r))$: $str(Q) \le str(P)$
- for every $Q \in preds(body^-(r))$: $str(Q) < str(P)$

The *stratification partition* of $\mathcal{P}$ induced by str is the sequence $(\mathcal{P}_1,\ldots,\mathcal{P}_k)$, with each $\mathcal{P}_i$ consisting of all rules $r \in \mathcal{P}$ s.t. $max_{a\in head(r)}(str(pred(a))) = i$. The programs $\mathcal{P}_i$ are the *strata* of P. A program is *stratified* if it admits a stratification. All definite programs are stratified.

Stratified programs have a least Herbrand model (LHM), which is constructed using the immediate consequence operator $T_\mathcal{P}$. Let $\mathbf{U}$ and $\mathbf{B}$ be the Herbrand universe and base of $\mathcal{P}$, and let $S \subseteq \mathbf{B}$. Then, $T_\mathcal{P}(S)$ consists of all facts in $head(r)\sigma$ with $r \in \mathcal{P}$ and σ a substitution from $vars(r)$ to $\mathbf{U}$ satisfying $body^+(r)\sigma \subseteq S$ and $body^-(r)\sigma \cap S = \emptyset$. The powers of $T_\mathcal{P}$ are as follows: $T^0_\mathcal{P}(S) = S$, $T^{n+1}_\mathcal{P}(S) = T_\mathcal{P}(T^n_\mathcal{P}(S))$, and $T^\omega_\mathcal{P}(S) = \bigcup_{i=0}^\infty T^i_\mathcal{P}(S)$. Let str be a stratification of $\mathcal{P}$, and let $(\mathcal{P}_1,\ldots,\mathcal{P}_k)$ be its stratification partition. Also, let $U_1 = T^\omega_{\mathcal{P}_1}(\emptyset)$ and for each $1 \le i \le k$ let $U_{i+1} = T^\omega_{\mathcal{P}_{i+1}}(U_i)$. Then, the LHM of $\mathcal{P}$ is U_k and is denoted $M[\mathcal{P}]$. A program $\mathcal{P}$ entails a positive existential sentence α ($\mathcal{P} \models \alpha$) if $M[\mathcal{P}]$ seen as a FO structure satisfies α.

We use LPs to encode FO theories. For this, we introduce rules axiomatising the built-in semantics of the equality ($\approx$) and truth ($\top$) predicates. For a finite signature Σ, we denote with $\mathcal{F}^\top_\Sigma$ the smallest set with a rule

$$p(x_1, x_2, \ldots, x_n) \to \top(x_1) \wedge \top(x_2) \wedge \ldots \wedge \top(x_n)$$

for each n-ary predicate p in Σ, and with $\mathcal{F}^\approx_\Sigma$ the usual axiomatisation of $\approx$ as a congruence over Σ. For an LP $\mathcal{P}$, we denote with $\mathcal{P}^{\approx,\top}$ the extension of $\mathcal{P}$ to $\mathcal{P} \cup \mathcal{F}^\top_\Sigma \cup \mathcal{F}^\approx_\Sigma$ with Σ the signature of $\mathcal{P}$.

[1]We assume rule heads non-empty, and allow multiple atoms.

	Ax./constr. α	Definite LP rules $\pi(\alpha)$
(R1)	R^-	$R(x,y) \to R^-(y,x)$; $R^-(y,x) \to R(x,y)$
(R2)	$R \sqsubseteq S$	$R(x,y) \to S(x,y)$
(T1)	$\sqcap_{i=1}^n A_i \sqsubseteq B$	$\bigwedge_{i=1}^n A_i(x) \to B(x)$
(T2)	$A \sqsubseteq \{a\}$	$A(x) \to x \approx a$
(T3)	$\exists R.A \sqsubseteq B$	$R(x,y) \wedge A(y) \to B(x)$
(T4)	$A \sqsubseteq \le 1 R.B$	$A(x) \wedge R(x,y) \wedge B(y) \wedge R(x,z) \wedge B(z) \to y \approx z$
(T5)	$A \sqsubseteq \exists R.B$	$A(x) \to R(x, f^A_{R,B}(x)) \wedge B(f^A_{R,B}(x)))$
(A1)	$A(a)$	$\to A(a)$
(A2)	$R(a,b)$	$\to R(a,b)$

Table 1: Translation from Horn ontologies into rules.

Ontologies and Queries We define Horn-$\mathcal{ALCHOIQ}$ and specify its semantics via translation to definite programs. W.l.o.g. we consider ontologies in a normal form close to that in [Ortiz *et al.*, 2010]. Let N_C, N_R and N_I be countable pairwise disjoint sets of concept names, role names and individuals. We assume $\{\top,\bot\} \subseteq N_\mathsf{C}$. A *role* is an element of $N_\mathsf{R} \cup \{R^- | R \in N_\mathsf{R}\}$, where the roles in the latter set are called *inverse roles*. The function $\mathsf{Inv}(\cdot)$ is defined as follows, where $R \in N_\mathsf{R}$: $\mathsf{Inv}(R) = R^-$ and $\mathsf{Inv}(R^-) = R$. An *RBox* $\mathcal{R}$ is a finite set of axioms (R2) in Table 1, where R and S are roles and $\sqsubseteq^*_\mathcal{R}$ is the minimal reflexive-transitive relation over roles s.t. $\mathsf{Inv}(R) \sqsubseteq^*_\mathcal{R} \mathsf{Inv}(S)$ and $R \sqsubseteq^*_\mathcal{R} S$ hold if $R \sqsubseteq S \in \mathcal{R}$. A *TBox* $\mathcal{T}$ is a finite set of axioms (T1)-(T5) where $A, B \in N_\mathsf{C}$ and R is a role.[2] An *ABox* $\mathcal{A}$ is a finite set of axioms of the form (A1) and (A2), with $A \in N_\mathsf{C}$ and $R \in N_\mathsf{R}$. An *ontology* is a finite set of axioms $\mathcal{O} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{A}$.

OWL 2 specifies the EL, QL, and RL profiles; these are fragments of Horn-$\mathcal{ALCHOIQ}$ with the exception of property chain axioms and transitivity, which we do not consider here. An ontology is: *(i) EL* if it does not contain inverse roles or axioms (T4); *(i) RL* if it does not contain axioms (T5); and *(i) QL* if it does not contain axioms (T2) or (T4), each axiom (T1) satisfies $n = 1$, and each axiom (T3) satisfies $A = \top$.

A *conjunctive query (CQ)* Q is a formula $\exists\vec{y}.\psi(\vec{x},\vec{y})$ with $\psi(\vec{x},\vec{y})$ a conjunction of function-free atoms over $\vec{x} \cup \vec{y}$, where $\vec{x}$ are the *answer variables*. We denote with $terms(Q)$ the set of terms in Q. Queries with no answer variables are *Boolean* (BCQs) and for convenience are written as a set of atoms.

We define the semantics by a mapping π into definite rules as in Table 1: $\pi(\mathcal{O}) = \{\pi(\alpha) \mid \alpha \in \mathcal{O}\}$ [3]. An ontology $\mathcal{O}$ is satisfiable if $\pi(\mathcal{O})^{\approx,\top} \not\models \exists y.\bot(y)$. A tuple of constants $\vec{c}$ is an *answer* to Q if $\mathcal{O}$ is unsatisfiable, or $\pi(\mathcal{O})^{\approx,\top} \models \exists\vec{y}.\psi(\vec{c},\vec{y})$. The set of answers is written $cert(Q,\mathcal{O})$. This semantics is equivalent to the usual one.

[2]Axioms $A \sqsubseteq \ge n\, R.B$ can be simulated by (T1) and (T5).

[3]By abuse of notation we say that $R^- \in \mathcal{O}$ whenever R^- occurs in $\mathcal{O}$.

3 Reasoning over RSA Ontologies

CQ answering is EXPTIME-complete for Horn-$\mathcal{ALCHOIQ}$ ontologies [Ortiz *et al.*, 2010], and the EXPTIME lower bound holds already for satisfiability checking. Intractability is due to *and-branching*: owing to the interaction between axioms in Table 1 of type (T5) with either axioms (T3) and (R1), or axioms (T4) an ontology may only be satisfied by large (possibly infinite) models which cannot be succinctly represented.

RSA is a class of ontologies where all axioms in Table 1 are allowed, but their interaction is restricted s.t. model size can be polynomially bounded [Carral *et al.*, 2014b]. We next recapitulate RSA ontologies and their properties; in the remainder, we fix an arbitrary Horn-$\mathcal{ALCHOIQ}$ ontology $\mathcal{O}$.

Roles in $\mathcal{O}$ are divided into *safe* and *unsafe*. The intuition is that unsafe roles may participate in harmful interactions.

Definition 1 *A role R in $\mathcal{O}$ is* unsafe *if it occurs in axioms (T5), and there is a role S s.t. either of the following holds:*

1. $R \sqsubseteq^*_{\mathcal{R}}$ $\mathsf{Inv}(S)$ *and S occurs in an axiom (T3) with left-hand-side concept $\exists S.A$ where $A \neq \top$.*
2. *S is in an axiom (T4) and $R \sqsubseteq^*_{\mathcal{R}} S$ or $R \sqsubseteq^*_{\mathcal{R}} \mathsf{Inv}(S)$.*

A role R in $\mathcal{O}$ is safe*, if it is not unsafe.*

It follows from Definition 1 that RL, QL, and EL ontologies contain only safe roles.

Example 1 *Let $\mathcal{O}_{\mathsf{Ex}}$ be the (out-of-profile) ontology with the following axioms, which we will use as a running example:*

$A(a)$	(1)	$\exists S.A \sqsubseteq D$	(4)	$R \sqsubseteq T^-$	(7)
$A \sqsubseteq D$	(2)	$D \sqsubseteq \exists R.B$	(5)	$S \sqsubseteq T$	(8)
$A \sqsubseteq \exists S^-.C$	(3)	$B \sqsubseteq \exists S.D$	(6)		

Roles R, S, T, and T^- are safe; however, S^- is unsafe as it occurs in an axiom (T5) while S occurs in an axiom (T3).

The distinction between safe and unsafe roles makes it possible to strengthen the translation π in Table 1 while preserving satisfiability and entailment of unary facts. The translation of axioms (T5) with R safe can be realised by replacing the functional term $f^A_{R,B}(x)$ with a Skolem constant $v^A_{R,B}$ unique to A, R and B. The modified transformation typically leads to a smaller LHM: if all roles are safe then $\mathcal{O}$ is mapped to a datalog program whose LHM is of size polynomial in the size of $\mathcal{O}$.

Definition 2 *Let $v^A_{R,B}$ be a fresh constant for each pair of concepts A, B and each safe role R in $\mathcal{O}$. The function π_{safe} maps each $\alpha \in \mathcal{O}$ to*

1. *a rule $A(x) \rightarrow R(x, v^A_{R,B}) \wedge B(v^A_{R,B})$ if α is of type (T5) with R safe; and*
2. *a rule $\pi(\alpha)$ otherwise.*

Let $\mathcal{P} = \{\pi_{\mathsf{safe}}(\alpha) \mid \alpha \in \mathcal{O}\}$; then, we define $\mathcal{P}_{\mathcal{O}} = \mathcal{P}^{\approx,\top}$.

Example 2 *Mapping π_{safe} differs from π on ax. (5), and (6). For instance, (5) yields $D(x) \rightarrow R(x, v^D_{R,B}) \wedge B(v^D_{R,B})$.*

The properties of $\mathcal{P}_{\mathcal{O}}$ are given by the following theorem.

Theorem 1 *[Carral* et al.*, 2014b, Theorem 2] Ontology $\mathcal{O}$ is satisfiable iff $\mathcal{P}_{\mathcal{O}} \not\models \exists y.\bot(y)$. If $\mathcal{O}$ is satisfiable, then $\mathcal{O} \models A(c)$ iff $A(c) \in M[\mathcal{P}_{\mathcal{O}}]$ for each unary predicate A and constant c in $\mathcal{O}$.*

If $\mathcal{O}$ has unsafe roles the model $M[\mathcal{P}_{\mathcal{O}}]$ might be infinite. We next define a datalog program $\mathcal{P}_{\mathsf{RSA}}$ by introducing Skolem constants for all axioms (T5) in $\mathcal{O}$. $\mathcal{P}_{\mathsf{RSA}}$ introduces also a predicate PE which 'tracks' all binary facts generated by the application of Skolemised rules over unsafe roles. A unary predicate U is initialised with the constants associated to unsafe roles and a rule $\mathsf{U}(x) \wedge \mathsf{PE}(x,y) \wedge \mathsf{U}(y) \rightarrow \mathsf{E}(x,y)$ stores the PE-facts originating from unsafe roles using a predicate E. Then, $M[\mathcal{P}_{\mathcal{O}}]$ is of polynomial size when the graph induced by the extension of E is an oriented forest (i.e., a DAG whose underlying undirected graph is a forest). When this condition is fulfilled together with some additional conditions which preclude harmful interactions between equality-generating axioms and inverse roles, we say that $\mathcal{O}$ is RSA.

Definition 3 *Let PE and E be fresh binary predicates, let U be a fresh unary predicate, and let $u^A_{R,B}$ be a fresh constant for each concept A, B and each role R in $\mathcal{O}$. Function π_{RSA} maps each $\alpha \in \mathcal{O}$ to*

- *$A(x) \rightarrow R(x, u^A_{R,B}) \wedge B(u^A_{R,B}) \wedge \mathsf{PE}(x, u^A_{R,B})$ if α is of type (T5) and*
- *$\pi(\alpha)$, otherwise.*

The program $\mathcal{P}_{\mathsf{RSA}}$ consists of $\pi_{\mathsf{RSA}}(\alpha)$ for each $\alpha \in \mathcal{O}$, a rule $\mathsf{U}(x) \wedge \mathsf{PE}(x,y) \wedge \mathsf{U}(y) \rightarrow \mathsf{E}(x,y)$, and a fact $\mathsf{U}(u^A_{R,B})$ for each $u^A_{R,B}$ with R unsafe.

Let M_{RSA} be the LHM of $\mathcal{P}_{\mathsf{RSA}}{}^{\approx,\top}$. Then, $G_{\mathcal{O}}$ is the digraph with an edge (c,d) for each $\mathsf{E}(c,d)$ in M_{RSA}. Ontology $\mathcal{O}$ is equality-safe if:

- *for each pair of atoms $w \approx t$ (with w and t distinct) and $R(t, u^A_{R,B})$ in M_{RSA} and each role S s.t. $R \sqsubseteq \mathsf{Inv}(S)$, it holds that S does not occur in an axiom (T4); and*
- *for each pair of atoms $R(a, u^A_{R,B})$, $S(u^A_{R,B}, a)$ in M_{RSA}, with $a \in N_I$, there does not exist a role T such that both $R \sqsubseteq^*_{\mathcal{R}} T$ and $S \sqsubseteq^*_{\mathcal{R}} \mathsf{Inv}(T)$ hold.*

We say that $\mathcal{O}$ is RSA if it is equality-safe and $G_{\mathcal{O}}$ is an oriented forest.

The fact that $G_{\mathcal{O}}$ is a DAG ensures that the LHM $M[\mathcal{P}_{\mathcal{O}}]$ is finite, whereas the lack of 'diamond-shaped' subgraphs in $G_{\mathcal{O}}$ guarantees polynomiality of $M[\mathcal{P}_{\mathcal{O}}]$. The safety condition on $\approx$ ensures that RSA ontologies enjoy a special form of forest-model property that we exploit for CQ answering. Every ontology in QL (which is equality-free), RL (where $\mathcal{P}_{\mathsf{RSA}}$ has no Skolem constants) and EL (no inverse roles) is RSA.

Theorem 2 *[Carral* et al.*, 2014b, Theorem 3] If $\mathcal{O}$ is RSA, then $|M[\mathcal{P}_{\mathcal{O}}]|$ is polynomial in $|\mathcal{O}|$.*

Tractability of standard reasoning for RSA ontologies follows from Theorems 1, 2. It can be checked that $\mathcal{O}_{\mathsf{Ex}}$ is RSA.

4 Answering Queries over RSA Ontologies

We next present our combined approach with filtration to CQ answering over RSA ontologies, which generalises existing techniques for DL-Lite$_{\mathcal{R}}$ and $\mathcal{ELHO}$.

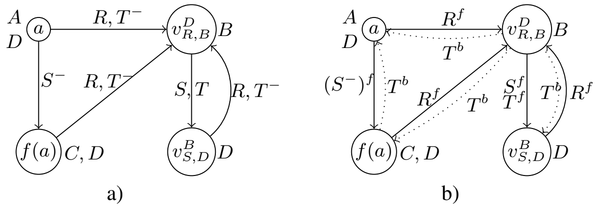

Figure 1: Original (a) and annotated (b) model for $\mathcal{O}_{\mathsf{Ex}}$

In Section 4.1 we take the LHM for RSA ontologies given in Section 3 as a starting point and extend it to a more convenient canonical model over an extended signature. In order to deal with the presence of inverse roles in RSA ontologies, the extended model captures the 'directionality' of binary atoms; this will allow us to subsequently extend the filtration approach from [Stefanoni *et al.*, 2013] in a seamless way. The canonical model is captured declaratively as the LHM of an LP program over the extended signature.

As usual in combined approaches, this model is not universal (it cannot be homomorphically embedded into every other model of the ontology) and the evaluation of CQs may lead to spurious, i.e. unsound answers. In Section 4.2, we specify our filtration approach for RSA ontologies as the LHM of a stratified program. In the following, we fix an arbitrary RSA ontology $\mathcal{O} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{A}$ and an input CQ Q, which we use to parameterise all our technical results.

4.1 Constructing the Canonical Model

The LHM $M[\mathcal{P}_{\mathcal{O}}]$ in Section 3 is a model of $\mathcal{O}$ that preserves entailment of unary facts. It generalises the canonical model in [Stefanoni *et al.*, 2013], which is specified as the LHM of a datalog program obtained by Skolemising all axioms (T5). However, RSA ontologies allow for unsafe roles and hence $M[\mathcal{P}_{\mathcal{O}}]$ may contain also functional terms.

A main source for spurious matches when evaluating Q over the canonical model of an EL ontology is the presence of 'forks'—confluent chains of binary atoms— in the query which map to 'forks' in the model over Skolem constants. This is also problematical in our setting since RSA ontologies have the forest-model property.

Example 3 *Fig. 1 a) depicts the LHM $M[\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}]$ of $\mathcal{O}_{\mathsf{Ex}}$ (the function $f_{S,C}$ is abbreviated with f). We see models as digraphs where the direction of edges reflects the satisfaction of axioms (T5). Consider $Q_1 = \{A(y_1), R(y_1, y_2), R(y_3, y_2)\}$. Substitution $(y_1 \to a, y_2 \to v^D_{R,B}, y_3 \to v^B_{S,D})$ is a spurious match of Q_1 as it relies on edges $(a, v^D_{R,B})$ and $(v^B_{S,D}, v^D_{R,B}))$ in $M[\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}]$, which form a fork over $v^D_{R,B}$.*

In EL, only queries which themselves contain forks can be mapped to forks in the model. This is, however, no longer the case for RSA ontologies, where forks in the model can lead to spurious answers even for linearly-shaped queries due to the presence of inverse roles.

Example 4 *Let $Q_2 = \{A(y_1), R(y_1, y_2), T(y_2, y_3)\}$. Substitution $(y_1 \to a, y_2 \to v^D_{R,B}, y_3 \to f(a))$ is a spurious match for Q_2 as it relies on the fork $(a, v^D_{R,B})$, $(f(a), v^D_{R,B}))$ in $M[\mathcal{P}_{\mathcal{O}_{\mathsf{Ex}}}]$. This is due to axiom $R \sqsubseteq T^-$ which causes a linear match over R and T to become a fork over R and T^-.*

To identify such situations, we compute a canonical model over an extended signature that contains fresh roles R^f and R^b for each role R. Annotations f (forward) and b (backwards) are intended to reflect the directionality of binary atoms in the model, where binary atoms created to satisfy an axiom (T5) are annotated with f. To realise this intuition declaratively, we modify the rules in $\mathcal{P}_{\mathcal{O}}$ for axioms (T5) as follows. If R is safe, then we introduce the rule $A(x) \to R^f(x, v^A_{R,B}) \wedge B(v^A_{R,B})$; if it is unsafe, we introduce rule $A(x) \to R^f(x, f^A_{R,B}(x)) \wedge B(f^A_{R,B}(x))$ instead.

Roles inherit the direction of their subroles, while roles and their inverses have opposite direction. This is captured with the following rules for $* \in \{f, b\}$: *(i)* $R^*(x, y) \to S^*(x, y)$ for each $R \sqsubseteq S$ in $\mathcal{O}$; *(ii)* $R^f(x, y) \to \mathsf{Inv}(R)^b(y, x)$ and $R^b(x, y) \to \mathsf{Inv}(R)^f(y, x)$ for each role R; and *(iii)* $R^*(x, y) \to R(x, y)$ for each role R. Rules *(ii)* are included only if $\mathcal{O}$ has inverse roles, and rules *(iii)* 'copy' annotated atoms to atoms over the original predicate. Fig. 1 b) depicts the annotated model for $P_{\mathcal{O}_{\mathsf{Ex}}}$ where solid (resp. dotted) lines represent 'forward' (resp. 'backwards') atoms.

Fig. 2 depicts the ways in which query matches may spuriously rely on a fork in an annotated model. Nodes represent the images in the model of the query terms; solid lines indicate the annotated atoms responsible for the match; and dashed lines depict the underpinning fork. The images of s and t must not be equal; additionally, y cannot be mapped to (a term identified to) a constant in $\mathcal{O}$. For instance, the match in Ex. 4 is spurious as it corresponds to pattern (b) in Fig. 2.

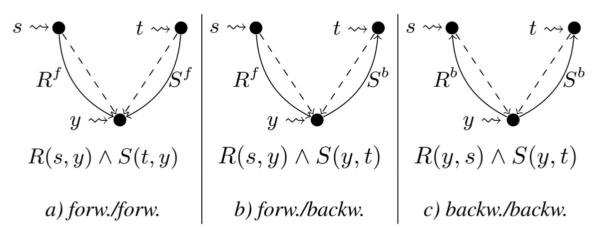

Figure 2: Forks in the presence of inverse roles

Unfortunately, the annotated model can present ambiguity: it is possible for both atoms $R^f(s, t)$ and $R^b(s, t)$ to hold.

Example 5 *Consider Q_2 from Ex. 4. An alternative match is $(y_1 \to a, y_2 \to v^D_{R,B}, y_3 \to v^B_{S,D})$, where $T^f(v^D_{R,B}, v^B_{S,D})$ and $T^b(v^D_{R,B}, v^B_{S,D})$ hold in the annotated model. The reason is that $S^f(v^D_{R,B}, v^B_{S,D})$ and $R^f(v^B_{S,D}, v^D_{R,B})$ form a cycle in the model and $S \sqsubseteq^*_{\mathcal{R}} T$ and $R \sqsubseteq^*_{\mathcal{R}} T^-$ hold in $\mathcal{O}_{\mathsf{Ex}}$.*

Such ambiguity is problematic for the subsequent filtration step. To disambiguate, we use a solution similar to the technique in [Lutz *et al.*, 2013] for DL-Lite$_{\mathcal{R}}$, where the idea is to unfold certain cycles of length one (self-loops) and two in the canonical model by introducing additional auxiliary constants. We unfold self-loops to cycles of length three while cycles of length two are unfolded to cycles of length four.

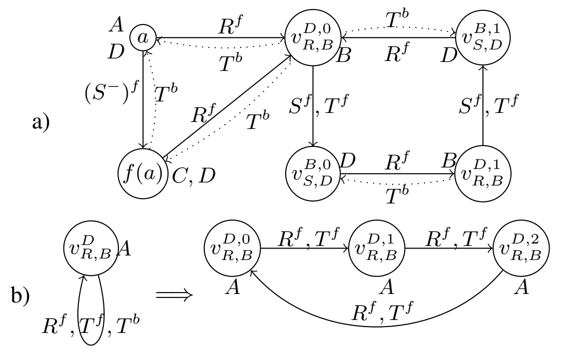

Figure 3: Model expansion in the presence of loops/cycles

symbols/ax. in $\mathcal{O}$	logic programming rules
ax. α not (T5)	$\pi(\alpha)$
$R \sqsubseteq S$, $* \in \{f, b\}$	$R^*(x,y) \to S^*(x,y)$
R role, $* \in \{f, b\}$	$R^*(x,y) \to R(x,y)$ $R^f(x,y) \to \mathsf{Inv}(R)^b(y,x)$ $R^b(x,y) \to \mathsf{Inv}(R)^f(y,x)$
ax. (T5), R unsafe	$A(x) \to R^f(x, f^A_{R,B}(x)) \wedge B(f^A_{R,B}(x))$
ax. (T5), R safe	$A(x) \wedge \mathsf{notIn}(x, \mathsf{unfold}(A,R,B)) \to$ $R^f(x, v^{A,0}_{R,B}) \wedge B(v^{A,0}_{R,B})$
	if $R \in \mathsf{confl}(R)$, for every $i = 0, 1$: $A(v^{A,i}_{R,B}) \to R^f(v^{A,i}_{R,B}, v^{A,i+1}_{R,B}) \wedge B(v^{A,i+1}_{R,B})$
	for every $x \in \mathsf{cycle}(A,R,B)$: $A(x) \to R^f(x, v^{A,1}_{R,B}) \wedge B(v^{A,1}_{R,B})$

Table 2: Rules in the program $E_\mathcal{O}$

Example 6 *Fig. 3 a) shows the model expansion for ontology $\mathcal{O}_{\mathsf{Ex}}$. Note that ambiguities are resolved. Fig. 3 b) shows the unfolding of a generic self-loop over a safe role R for which T exists s.t. $R \sqsubseteq T$ and $R \sqsubseteq \mathsf{Inv}(T)$ hold.*

We now specify a program that yields the required model.

Definition 4 *Let $\mathsf{confl}(R)$ be the set of roles S s.t. $R \sqsubseteq^*_\mathcal{R} T$ and $S \sqsubseteq^*_\mathcal{R} \mathsf{Inv}(T)$ for some T. Let $\prec$ be a strict total order on triples (A, R, B), with R safe and A, B concept names in $\mathcal{O}$. For each (A, R, B), let $v^{A,0}_{R,B}$, $v^{A,1}_{R,B}$, and $v^{A,2}_{R,B}$ be fresh constants; let $\mathsf{self}(A,R,B)$ be the smallest set containing $v^{A,0}_{R,B}$ and $v^{A,1}_{R,B}$ if $R \in \mathsf{confl}(R)$; and let $\mathsf{cycle}(A,R,B)$ be the smallest set of terms containing, for each $S \in \mathsf{confl}(R)$,*

- *$v^{D,0}_{S,C}$ if $(A,R,B) \prec (D,S,C)$;*
- *$v^{D,1}_{S,C}$ if $(D,S,C) \prec (A,R,B)$; and*
- *$f^D_{S,C}(v^{A,0}_{R,B})$ and each $f^F_{T,E}(v^{A,0}_{R,B})$ s.t. $u^D_{S,C} \approx u^F_{T,E}$ is in M_{RSA}, if S is unsafe.*

Finally, $\mathsf{unfold}(A,R,B) = \mathsf{self}(A,R,B) \cup \mathsf{cycle}(A,R,B)$.

Let R^f and R^b be fresh binary predicates for each role R in $\mathcal{O}$, let NI be a fresh unary predicate, and notIn be a built-in predicate which holds when the first argument is an element of the set given as second argument. Let $\mathcal{P}$ be the smallest program with a rule $\to \mathsf{NI}(a)$ for each constant a and all rules in Table 2. We define $E_\mathcal{O} = \mathcal{P}^{\approx,\top}$.

(1) $\psi(\vec{x},\vec{y}) \to \mathsf{QM}(\vec{x},\vec{y})$
(2) $\to named(a)$ for each constant a in $\mathcal{O}$
(3a) $\mathsf{QM}(\vec{x},\vec{y})$, $not\ \mathsf{NI}(y_i) \to \mathsf{id}(\vec{x},\vec{y},i,i)$, for each $1 \le i \le \lvert\vec{y}\rvert$ **(3b)** $\mathsf{id}(\vec{x},\vec{y},u,v) \to \mathsf{id}(\vec{x},\vec{y},v,u)$ **(3c)** $\mathsf{id}(\vec{x},\vec{y},u,v) \wedge \mathsf{id}(\vec{x},\vec{y},v,w) \to \mathsf{id}(\vec{x},\vec{y},u,w)$

for all $R(s,y_i)$, $S(t,y_j)$ in Q with $y_i, y_j \in \vec{y}$
(4a) $R^f(s,y_i) \wedge S^f(t,y_j) \wedge \mathsf{id}(\vec{x},\vec{y},i,j) \wedge not\ s \approx t \to \mathit{fk}(\vec{x},\vec{y})$
for all $R(s,y_i)$, $S(y_j,t)$ in Q with $y_i, y_j \in \vec{y}$:
(4b) $R^f(s,y_i) \wedge S^b(y_j,t) \wedge \mathsf{id}(\vec{x},\vec{y},i,j) \wedge not\ s \approx t \to \mathit{fk}(\vec{x},\vec{y})$
for all $R(y_i,s)$, $S(y_j,t)$ in Q with $y_i, y_j \in \vec{y}$:
(4c) $R^b(y_i,s) \wedge S^b(y_j,t) \wedge \mathsf{id}(\vec{x},\vec{y},i,j) \wedge not\ s \approx t \to \mathit{fk}(\vec{x},\vec{y})$

for all $R(y_i,y_j)$, $S(y_k,y_l)$ in Q with $y_i,y_j,y_k,y_l \in \vec{y}$: **(5a)** $R^f(y_i,y_j) \wedge S^f(y_k,y_l) \wedge \mathsf{id}(\vec{x},\vec{y},j,l) \wedge$ $\wedge y_i \approx y_k \wedge not\ \mathsf{NI}(y_i) \to \mathsf{id}(\vec{x},\vec{y},i,k)$ **(5b)** $R^f(y_i,y_j) \wedge S^b(y_k,y_l) \wedge \mathsf{id}(\vec{x},\vec{y},j,k) \wedge$ $\wedge y_i \approx y_l \wedge not\ \mathsf{NI}(y_i) \to \mathsf{id}(\vec{x},\vec{y},i,l)$ **(5c)** $R^b(y_i,y_j) \wedge S^b(y_l,y_k) \wedge \mathsf{id}(\vec{x},\vec{y},i,l) \wedge$ $\wedge y_j \approx y_k \wedge not\ \mathsf{NI}(y_j) \to \mathsf{id}(\vec{x},\vec{y},j,k)$
for each $R(y_i,y_j)$ in Q with $y_i,y_j \in \vec{y}$, and $* \in \{f,b\}$: **(6)** $R^*(y_i,y_j) \wedge \mathsf{id}(\vec{x},\vec{y},i,v) \wedge \mathsf{id}(\vec{x},\vec{y},j,w) \to AQ^*(\vec{x},\vec{y},v,w)$
for each $* \in \{f,b\}$: **(7a)** $AQ^*(\vec{x},\vec{y},u,v) \to TQ^*(\vec{x},\vec{y},u,v)$ **(7b)** $AQ^*(\vec{x},\vec{y},u,v) \wedge TQ^*(\vec{x},\vec{y},v,w) \to TQ^*(\vec{x},\vec{y},u,w)$
(8a) $\mathsf{QM}(\vec{x},\vec{y}) \wedge not\ named(x) \to \mathsf{sp}(\vec{x},\vec{y})$, for each $x \in \vec{x}$ **(8b)** $\mathit{fk}(\vec{x},\vec{y}) \to \mathsf{sp}(\vec{x},\vec{y})$ **(8c)** $TQ^*(\vec{x},\vec{y},v,v) \to \mathsf{sp}(\vec{x},\vec{y})$, for each $* \in \{f,b\}$
(9) $\mathsf{QM}(\vec{x},\vec{y}) \wedge not\ \mathsf{sp}(\vec{x},\vec{y}) \to \mathsf{Ans}(\vec{x})$

Table 3: Rules in $\mathcal{P}_Q$. Variables u, v, w from U are distinct.

The set $\mathsf{confl}(R)$ contains the roles that may cause ambiguity in conjunction with R. The ordering $\prec$ determines how cycles are unfolded using auxiliary constants. Each axiom $A \sqsubseteq \exists R.B$ with R safe is Skolemised by default using $v^{A,0}_{R,B}$, except when the axiom applies to an auxiliary constant in $\mathsf{unfold}(R,B)$ where we use $v^{A,1}_{R,B}$ or $v^{A,2}_{R,B}$ instead. The key properties of $E_\mathcal{O}$ are given next.

Theorem 3 *The following holds: (i) $M[E_\mathcal{O}]$ is polynomial in $\lvert\mathcal{O}\rvert$ (ii) $\mathcal{O}$ is satisfiable iff $E_\mathcal{O} \not\models \exists y.\bot(y)$ (iii) if $\mathcal{O}$ is satisfiable, $\mathcal{O} \models A(c)$ iff $A(c) \in M[E_\mathcal{O}]$ and (iv) there are no terms s, t and role R s.t. $E_\mathcal{O} \models R^f(s,t) \wedge R^b(s,t)$.*

4.2 Filtering Unsound Answers

We now define a program $\mathcal{P}_Q$ that can be used to eliminate all spurious matches of Q over the annotated model of $\mathcal{O}$. The rules of the program are summarised in Table 3. In what follows, we refer to all functional terms and Skolem constants in the model that are not equal to a constant in $\mathcal{O}$ as *anonymous*.

Matches where an answer variable is not mapped to a constant in $\mathcal{O}$ are spurious. We introduce a predicate $named$ and populate it with such constants (rules **(2)**); then, we flag answers as spurious using a rule with negation (rules **(8a)**).

To detect forks we introduce a predicate fk, whose definition in datalog will encode the patterns in Fig. 2 (rules **(4)**). If terms s and t in Fig. 2 are existential variables mapping to the same anonymous term, further forks might be recursively induced due to the identity of s and t.

Example 7 *Let $Q_3 = \{A(y_1), R(y_1,y_2), T(y_2,y_3), C(y_4), R(y_4,y_5), S(y_5,y_3)\}$ be a BCQ over $\mathcal{O}_{\mathsf{Ex}}$. Substitution $(y_1 \to a, y_2 \to v^{D,0}_{R,B}, y_3 \to v^{B,0}_{S,D}, y_4 \to f(a), y_5 \to v^{D,0}_{R,B})$ is* *the only match over the model in Fig. 3a). The identity of y_2, y_5 induces a fork on the match of $R(y_1, y_2)$ and $R(y_4, y_5)$.*

We track identities in the model relative to a match using a fresh predicate id. It is initialised as the minimal congruence relation over the positions of the existential variables in the query which are mapped to anonymous terms (rules **(3)**). Identity is recursively propagated using rules **(5)**. Matches involving forks are marked as spurious by rule **(8b)**.

Spurious matches can also be caused by cycles in the model and query satisfying certain requirements. First, the positions of existential variables of the query must be cyclic when considering also the id relation. Second, the match must involve only anonymous terms. Finally, all binary atoms must have the same directionality.

Example 8 *Consider the following BCQs over $\mathcal{O}_{Ex}$:*

$$Q_4 = \{S(y_1, y_2), R(y_2, y_3), S(y_3, y_4), R(y_4, y_1)\}$$
$$Q_5 = \{T(y_1, y_2), S(y_2, y_3), R(y_3, y_1)\}$$
$$Q_6 = \{S(y_1, y_2), R(y_2, y_3), S(y_3, y_4), R(y_4, y_5)\}$$

Then, $(y_1 \to v^{D,0}_{R,B}, y_2 \to v^{B,0}_{S,D}, y_3 \to v^{D,1}_{R,B}, y_4 \to v^{B,1}_{S,D})$ is a match of Q_4 inducing a cycle: all binary atoms are mapped 'forward' and the cycle involves only anonymous terms. In contrast, match $(y_1 \to v^{D,0}_{R,B}, y_2 \to f(a), y_3 \to a)$ over Q_5 does not satisfy the requirements as it involves constant a and the atoms do not have the same directionality. Note that Q_4 and Q_5 are cyclic. In contrast, Q_6 is not cyclic; thus, although the match $(y_1 \to v^{D,0}_{R,B}, y_2 \to v^{B,0}_{S,D}, y_3 \to v^{D,1}_{R,B}, y_4 \to v^{B,1}_{S,D}, y_5 \to v^{D,0}_{R,B})$ involves a cycle in the model with the required properties, it is not spurious.

Such cycles are recognised by rules **(6)** and **(7)**. Rule **(6)** defines potential arcs in the cycle with their directionality, and the position of each relevant existential variable. These are recorded using predicates AQ^* with $* \in \{f, b\}$. Rules **(7)** detect the cycles recursively using predicates TQ^*. Matches involving cycles are marked as spurious by rules **(8c)**. All correct answers are collected by rule **(9)** using predicate Ans.

We now define program $\mathcal{P}_Q$ and its extension $\mathcal{P}_{\mathcal{O},Q}$ with $E_{\mathcal{O}}$ in Def. 4, which can be exploited to answer Q w.r.t. $\mathcal{O}$.

Definition 5 *Let $Q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ be a CQ, let* QM, sp, *and fk be fresh predicates of arity $|\vec{x}| + |\vec{y}|$, let* id, *AQ^*, and TQ^*, with $* \in \{f, b\}$, be fresh predicates of arity $|\vec{x}| + |\vec{y}| + 2$, let* Ans *be a fresh predicate of arity $|\vec{x}|$, let $named$ be a fresh unary predicate, and let U be a set of fresh variables s.t. $|U| \geq |\vec{y}|$. Then, $\mathcal{P}_Q$ is the smallest program with all rules in Table 3, and $\mathcal{P}_{\mathcal{O},Q}$ is defined as $E_{\mathcal{O}} \cup \mathcal{P}_Q$.*

Note that $\approx$ is not axiomatised for the symbols in $\mathcal{P}_Q$ that do not occur in $E_{\mathcal{O}}$. In this way, we can distinguish between the constants in $\mathcal{O}$ (recorded by predicate $named$ in $\mathcal{P}_Q$) and their closure under equality (recorded by NI in $E_{\mathcal{O}}$).

Theorem 4 *(i) $\mathcal{P}_{\mathcal{O},Q}$ is stratified; (ii) $M[\mathcal{P}_{\mathcal{O},Q}]$ is polynomial in $|\mathcal{O}|$ and exponential in $|Q|$; and (iii) if $\mathcal{O}$ is satisfiable, $\vec{x} \in cert(Q, \mathcal{O})$ iff $\mathcal{P}_{\mathcal{O},Q} \models$ Ans$(\vec{x})$.*

Theorem 4 suggests a worst-case exponential algorithm that, given $\mathcal{O}$ and Q, materialises $\mathcal{P}_{\mathcal{O},Q}$ and returns the extension of predicate Ans. This procedure can be modified to obtain a 'guess and check' algorithm applicable to BCQs. This algorithm first materialises $E_{\mathcal{O}}$ in polynomial time; then, it guesses a match σ to Q over the materialisation; finally, it materialises $(\mathcal{P}_{\mathcal{O},Q})\sigma$, where variables $\vec{x}$ and $\vec{y}$ are grounded by σ. The latter step can also be shown to be tractable.

Facts: M1	Model: M2/M3	q_1: M4/M5/M6	q_2: M4/M5/M6	q_3: M4/M5/M6
$10{\cdot}10^3$	1s / $51{\cdot}10^3$	1s / 2 / 0%	1s / 0 / 0%	1s / 18 / 28%
$49{\cdot}10^3$	4s / $246{\cdot}10^3$	3s / 7 / 0%	3s / 0 / 0%	3s / 89 / 26%
$98{\cdot}10^3$	9s / $487{\cdot}10^3$	7s / 9 / 0%	6s / 1 / 0%	6s / 193 / 23%
$146{\cdot}10^3$	11s / $726{\cdot}10^3$	13s / 14 / 0%	12s / 1 / 0%	10s / 273 / 22%

Figure 5: Results for Uniprot.

Theorem 5 *Checking whether $\mathcal{O} \models Q$ with $\mathcal{O}$ an RSA ontology and Q a BCQ is* NP*-complete in combined complexity.*

5 Proof of Concept

We have implemented our approach using the DLVsystem,[4] which supports function symbols and stratified negation. For testing, we have used the LUBM ontology [Guo *et al.*, 2005] (which contains only safe roles) and the Horn fragments of the Reactome and Uniprot (which are RSA, but contain also unsafe roles).[5] LUBM comes with a data generator; Reactome and Uniprot come with large datasets, which we have sampled. All test queries are given in the appendix. In each experiment, we measured (M1) size (number of facts) of the given data; (M2) materialisation times for the canonical model; (M3) model size; (M4) query processing times (i.e., time needed to materialise $\mathcal{P}_Q$); (M5) number of candidate query answers, i.e. (unfiltered) answers of the query over the canonical model; and (M6) percentage of spurious answers. All experiments were performed on a MacBook Pro laptop with 8GB RAM and an Intel Core 2.4 GHz processor.

Fig. 4 and 5 summarise our results. The relevant measures M1-M5 are indicated in the headings of the tables. We can see that computation times for the models scale linearly in data size. Model size is at most 6 times larger than the original data, which we see as a reasonable growth factor in practice. As usual in combined approaches (e.g., see [Stefanoni *et al.*, 2013]), query processing times depend on the number of candidate answers; thus, the applicability of the combined approach largely depends on the ratio between spurious and correct answers. Queries q_1-q_2 in Reactome and Uniprot are realistic queries given as examples in the EBI website. Neither of these queries lead to spurious answers, and processing times scale linearly with data size. No query in the LUBM benchmark leads to spurious answers (e.g., LUBM queries q_3 and q_4 in Fig. 4b). Thus, we manually crafted one additional query for Reactome and Uniprot (q_3 in both cases) and two for LUBM (queries q_1 and q_2), which lead to a high percentage of spurious answers. Although these queries are challenging, we can observe that the proportion of spurious answers (and thus growth in processing times) remains constant with increasing data size. Finally, note that query q_1 in LUBM retrieves the highest number of candidate

[4]http://www.dlvsystem.com/dlv/

[5]http://www.ebi.ac.uk/rdf/platform

Facts M1	Model M2/M3	q_1 M4/M5/M6	q_2 M4/M5/M6	q_3 M4/M5/M6
$54 \cdot 10^3$	8s / $242 \cdot 10^3$	6s / 10 / 0%	5s / 11 / 0%	6s / 50 / 48%
$107 \cdot 10^3$	16s / $485 \cdot 10^3$	14s / 11 / 0%	14s / 17 / 0%	12s / 122 / 38%
$159 \cdot 10^3$	21s / $728 \cdot 10^3$	42s / 17 / 0%	44s / 23 / 0%	36s/ 216 / 35%
$212 \cdot 10^3$	19s / $970 \cdot 10^3$	19s / 21 / 0%	15s/ 24 / 0%	14s/ 299 / 34%

(a) Reactome.

Facts Facts:M1	Model Model:M2/M3	q_1 q_1:M4/M5/M6	q_2:M4/M5/M6 q_2:M4/M5/M6	q_3:M4/M5/M6 q_3:M4/M5/M6	q_4:M4/M5/M6 q_4:M4/M5/M6
$37 \cdot 10^3$	4s / $213 \cdot 10^3$	11s / 2350 / 86%	4s / 650/ 96%	4s / 1580/ 0%	5s (1743/0%)
$75 \cdot 10^3$	6s / $395 \cdot 10^3$	45s / 9340/ 85%	8s / 1640/ 97%	9s / 7925/ 0%	8s / 5969/ 0%
$113 \cdot 10^3$	8s / $550 \cdot 10^3$	108s / 24901/ 83%	13s / 2352/ 98%	13s / 18661/ 0%	13s / 10870/ 0%
$150 \cdot 10^3$	11s / $682 \cdot 10^3$	188s / 52196/ 83%	17s / 2550/ 98%	18s / 32370/ 0%	24s / 15076/ 0%
$188 \cdot 10^3$	12s / $795 \cdot 10^3$	305s / 91366/ 82%	31s / 2550/ 98%	40s / 49555/ 0%	38s / 18517/ 0%
$226 \cdot 10^3$	14s / $894 \cdot 10^3$	390s / 148340/ 80%	39s / 2550/ 98%	46s / 72438/ 0%	40s / 20404/ 0%

(b) LUBM.

Figure 4: Results for Reactome and LUBM.

answers and is thus the most challenging query. Our prototype and all test data, ontologies and queries are available at http://tinyurl.com/qcolx3w.

6 Conclusions and Future Work

We have presented an extension to the combined approaches to query answering that can be applied to a wide range of out-of-profile Horn ontologies. Our theoretical results unify and extend existing techniques for $\mathcal{ELHO}$ and DL-Lite$_\mathcal{R}$ in a seamless and elegant way. Our preliminary experiments indicate the feasibility of our approach in practice.

We anticipate several directions for future work. First, we have not considered logics with transitive roles. Recently, it was shown that CQ answering over EL ontologies with transitive roles is feasible in NP [Stefanoni and Motik, 2015]. We believe that our techniques can be extended in a similar way. Finally, we would like to optimise our encoding into LP and conduct a more extensive evaluation.

References

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *IJCAI*, pages 364–369, 2005.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Automated Reasoning (JAR)*, 39(3):385–429, 2007.

[Carral *et al.*, 2014a] David Carral, Cristina Feier, Bernardo Cuenca Grau, Pascal Hitzler, and Ian Horrocks. $\mathcal{EL}$-ifying ontologies. In *IJCAR*, 2014.

[Carral *et al.*, 2014b] David Carral, Cristina Feier, Bernardo Cuenca Grau, Pascal Hitzler, and Ian Horrocks. Pushing the boundaries of tractable ontology reasoning. In *ISWC*, 2014.

[Gottlob *et al.*, 2014] Georg Gottlob, Marco Manna, and Andreas Pieris. Polynomial combined rewritings for existential rules. In *KR*, 2014.

[Guo *et al.*, 2005] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *J. Web Semantics*, 3(2-3):158–182, 2005.

[Kontchakov *et al.*, 2010] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in DL-Lite. In *KR*, 2010.

[Kontchakov *et al.*, 2011] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In *IJCAI*, pages 2656–2661, 2011.

[Leone *et al.*, 2006] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.*, 7(3):499–562, 2006.

[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*, pages 2070–2075, 2009.

[Lutz *et al.*, 2013] Carsten Lutz, Inanç Seylan, David Toman, and Frank Wolter. The combined approach to OBDA: Taming role hierarchies using filters. In *ISWC*, pages 314–330, 2013.

[Lutz, 2007] Carsten Lutz. Inverse roles make conjunctive queries hard. In *DL*, 2007.

[Motik *et al.*, 2009] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz, editors. *OWL 2 Web Ontology Language: Profiles*. W3C Recommendation, 2009.

[Ortiz *et al.*, 2010] Magdalena Ortiz, Sebastian Rudolph, and Mantas Simkus. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In *KR*, 2010.

[Ortiz *et al.*, 2011] Magdalena Ortiz, Sebastian Rudolph, and Mantas Simkus. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *IJCAI*, pages 1039–1044, 2011.

[Stefanoni and Motik, 2015] Giorgio Stefanoni and Boris Motik. Answering conjunctive queries over $\mathcal{EL}$ knowledge bases with transitive and reflexive roles. In *AAAI*, 2015.

[Stefanoni *et al.*, 2013] Giorgio Stefanoni, Boris Motik, and Ian Horrocks. Introducing nominals to the combined query answering approaches for $\mathcal{EL}$. In *AAAI*, 2013.

[Stefanoni *et al.*, 2014] Giorgio Stefanoni, Boris Motik, Markus Krötzsch, and Sebastian Rudolph. The complexity of answering conjunctive and navigational queries over OWL 2 EL knowledge bases. *J. Artif. Intell. Res. (JAIR)*, 51:645–705, 2014.

[Thomazo and Rudolph, 2014] Michaël Thomazo and Sebastian Rudolph. Mixing materialization and query rewriting for existential rules. In *ECAI*, pages 897–902, 2014.