Query Rewriting for Horn-$\mathcal{SHIQ}$ plus Rules

Thomas Eiter[1], **Magdalena Ortiz**[1], **Mantas Šimkus**[1], **Trung-Kien Tran**[2] and **Guohui Xiao**[1]

[1]Institute of Information Systems, Vienna University of Technology
{*eiter—ortiz—xiao*}@*kr.tuwien.ac.at* *simkus@dbai.tuwien.ac.at*

[2]STARLab, Vrije Universiteit Brussel
truntran@vub.ac.be

Abstract

Query answering over Description Logic (DL) ontologies has become a vibrant field of research. Efficient realizations often exploit database technology and rewrite a given query to an equivalent SQL or Datalog query over a database associated with the ontology. This approach has been intensively studied for conjunctive query answering in the $\mathcal{DL}$-*Lite* and $\mathcal{EL}$ families, but is much less explored for more expressive DLs and queries. We present a rewriting-based algorithm for conjunctive query answering over Horn-$\mathcal{SHIQ}$ ontologies, possibly extended with recursive rules under limited recursion as in $\mathcal{DL}$+*log*. This setting not only subsumes both $\mathcal{DL}$-*Lite* and $\mathcal{EL}$, but also yields an algorithm for answering (limited) recursive queries over Horn-$\mathcal{SHIQ}$ ontologies (an undecidable problem for full recursive queries). A prototype implementation shows its potential for applications, as experiments exhibit efficient query answering over full Horn-$\mathcal{SHIQ}$ ontologies and benign downscaling to $\mathcal{DL}$-*Lite*, where it is competitive with comparable state of the art systems.

Introduction

Description Logics (DLs) are the primary tool for representing and reasoning about knowledge given by an *ontology*. They are mostly fragments of first-order logic with a clear-cut semantics, convenient syntax and decidable reasoning, performed by quite efficient algorithms. This has led to important applications of DLs in areas like Ontology Based Data Access (OBDA), Data Integration and the Semantic Web, where the OWL standard is heavily based on DLs.

An important reasoning task in DLs is query answering similar as in databases, where a database-style query is evaluated over an ontology, viewing it as an enriched database.

Example 1. Consider the following sociopolitical ontology. The Human Development Index (HDI) of certain territories T, whose value V may be low, medium, high or very high (as in the UN Development Programme) is given by facts $\mathrm{hasHDI}(T, V)$. Further facts classify territories as cities, countries, etc. and relate their locations. The facts are shown in the two left columns of Table 1. The axioms (a)–(e) on the right hand side provide a terminology (in DL syntax) stating that: (a) the $\mathrm{isLocatedIn}$ relation is transitive; (b) the capital of a territory is located in that territory; (c) every country has a capital; (d) only cities can be capitals; and (e) only one capital can be located in each country. The query q_1 can be used to retrieve disadvantaged territories that lie in countries with high HDI but have a low HDI themselves. Observe that if we evaluate q_1 over the database (i.e., the facts), it returns no answer: indeed, Mexico is the only country with high HDI, and there is no fact $\mathrm{islocatedIn}(X, \mathit{Mexico})$ such that territory X has low HDI. However, if we evaluate q_1 over the full ontology, we can infer from axiom (a) that Carichi is located in Mexico, and return $(\mathit{Carichi}, \mathit{Mexico})$ as an answer. The query q_2, which retrieves countries whose capital city has a high HDI, would also have an empty answer over the database, but from the axioms (b)–(e) we can infer that Brasilia is the capital of Brazil and Islamabad the capital of Pakistan, and return both countries as an answer to the query.

To supply this reasoning service, a number of challenges must be faced. *Conjunctive queries* (CQs) have typically much higher complexity than standard reasoning in a DL, and recursive Datalog queries are undecidable even in very weak DLs, including the ones considered here (Levy and Rousset 1998). For reasoning with large instance data, *translating* queries into database query languages has proved to be efficient. Calvanese et al. (2007) introduced a *query rewriting* technique for the $\mathcal{DL}$-*Lite* family of DLs, where the terminological information is incorporated into the query in such a way that it can be straight evaluated over the database facts. For example, a rewriting of query q_1 in Table 1 should include, among other queries,

$$\mathrm{disadvantagedTerritory}(x, y) \leftarrow \mathrm{hasHDI}(x, \mathit{low}), \mathrm{country}(y), \mathrm{hasCapital}(y, x), \mathrm{hasHDI}(y, \mathit{high})$$

which adds all tuples (x, y) to the query answer that can be inferred using axiom (b). Such rewriting approaches have been developed for answering CQs in DLs of the $\mathcal{DL}$-*Lite* family, and to a lesser extent for $\mathcal{EL}$, but they are practically unexplored for more expressive DLs and queries (see Related Work for details).

In this paper we present a rewriting-based method for query answering over ontologies in Horn-$\mathcal{SHIQ}$ (the disjunction-free fragment of $\mathcal{SHIQ}$). This DL extends $\mathcal{DL}$-*Lite* and $\mathcal{EL}$, two prominent DLs closely related to the OWL 2 QL and the OWL 2 EL profiles, respectively, which offer different expressiveness while allowing for tractable reasoning. For example, axiom (b) is allowed in most DLs of the $\mathcal{DL}$-*Lite* family but not in $\mathcal{EL}$, while (c) is allowed in

village(*Carichi*)	hasHDI(*Carichi*, *low*)	(a) *trans*(isLocatedIn)	(c) country $\sqsubseteq \exists$hasCapital.capital
state(*Chihuahua*)	hasHDI(*Mexico*, *high*)	(b) hasCapital $\sqsubseteq$ isLocatedIn$^-$	(d) country $\sqsubseteq \forall$hasCapital.city
country(*Mexico*)	hasHDI(*Islamabad*, *high*)		(e) country $\sqsubseteq \leqslant 1$ isLocatedIn$^-$.capital
capital(*Islamabad*)	hasHDI(*Brasilia*, *high*)		
country(*Pakistan*)	isLocatedIn(*Carichi*, *Chihuahua*)	(q_1) disadvantagedTerritory$(x, y) \leftarrow$	hasHDI(x, low), isLocatedIn(x, y), country(y), hasHDI$(y, high)$
capital(*Brasilia*)	isLocatedIn(*Chihuahua*, *Mexico*)		
country(*Brazil*)	isLocatedIn(*Islamabad*, *Pakistan*)	(q_2) hasDevelopedCapital$(x) \leftarrow$	country(x), hasCapital(x, y), city(y), hasHDI$(y, high)$
	isLocatedIn(*Brasilia*, *Brazil*)		

Table 1: An example ontology and queries

$\mathcal{EL}$ but not in $\mathcal{DL}$-*Lite*. Axioms (a), (d) and (e) are not expressible in either of them, but they are expressible in Horn-$\mathcal{SHIQ}$. Despite the increase in expressivity, reasoning in Horn-$\mathcal{SHIQ}$ is still tractable in data complexity.

In this paper, we make the following contributions:

- We provide a practical algorithm for rewriting queries over Horn-$\mathcal{SHIQ}$ ontologies. It first applies a special resolution calculus, and then rewrites the query w.r.t. the saturated TBox into a Datalog program ready for evaluation over any ABox. It runs in polynomial time in data complexity, and thus is worst-case optimal.
- It can handle CQs and the more general *weakly DL-safe* Datalog queries in the style of $\mathcal{DL}$+*log* (Rosati 2006), where only existentially quantified variables may be bound to 'anonymous' domain elements implied by axioms.
- It is, to our knowledge, the first rewriting algorithm that supports transitive roles, which are considered relevant in practice (Sattler 2000), although challenging for query answering (Glimm et al. 2006, Eiter et al. 2009). It simultaneously allows for full existential quantification, inverse roles, and number restrictions, covering and extending the OWL2 profiles QL and RL, as well as a large fragment of EL.
- A prototype implementation for CQ answering (without transitive roles) shows that our approach behaves well in practice. In experiments it worked efficiently and it scaled down nicely to $\mathcal{DL}$-*Lite*, where it is competitive with state of the art query rewriting systems.

Due to space constraints, we only provide proof sketches of the central results, and refer the reader to (Eiter et al. 2012b) for more details.

Description Logic Horn-$\mathcal{SHIQ}$

As usual, we assume countably infinite sets $\mathsf{N_C} \supset \{\top, \bot\}$ and $\mathsf{N_R}$ of *atomic concepts* and *role names*, respectively. $\mathsf{N_R} \cup \{r^- \mid r \in \mathsf{N_R}\}$ is the set of *roles*. If $r \in \mathsf{N_R}$, then $\mathsf{inv}(r) = r^-$ and $\mathsf{inv}(r^-) = r$. *Concepts* are inductively defined as follows: (a) each $A \in \mathsf{N_C}$ is a concept, and (b) if C, D are concepts and r is a role, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall r.C$, $\exists r.C$, $\geqslant n\, r.C$ and $\leqslant n\, r.C$, for $n \geq 1$, are concepts. An expression $C \sqsubseteq D$, where C, D are concepts, is a *general concept inclusion axiom (GCI)*. An expression $r \sqsubseteq s$, where r, s are roles, is a *role inclusion (RI)*. A *transitivity axiom* is an expression $trans(r)$, where r is a role. A TBox $\mathcal{T}$ is a finite set of GCIs, RIs and transitivity axioms. We let $\sqsubseteq^*_{\mathcal{T}}$ denote the reflexive transitive closure of $\{(r, s) \mid r \sqsubseteq s \in \mathcal{T} \text{ or } \mathsf{inv}(r) \sqsubseteq \mathsf{inv}(s) \in \mathcal{T}\}$. A role s is *transitive* in $\mathcal{T}$ if $trans(s) \in \mathcal{T}$ or $trans(s^-) \in \mathcal{T}$. A role s is *simple* in $\mathcal{T}$ if there is no transitive r in $\mathcal{T}$ s.t. $r \sqsubseteq^*_{\mathcal{T}} s$. $\mathcal{T}$ is a $\mathcal{SHIQ}$ terminology if roles in concepts of the form $\geqslant n\, r.C$ and $\leqslant n\, r.C$ are simple. The semantics for TBoxes is given by *interpretations* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$. We write $\mathcal{I} \models \mathcal{T}$ if $\mathcal{I}$ is a *model* of $\mathcal{T}$. See (Baader et al. 2007) for more details.

A TBox $\mathcal{T}$ is a Horn-$\mathcal{SHIQ}$ TBox (in normalized form), if each GCI in $\mathcal{T}$ takes one the following forms:

(F1)	$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$,	(F3)	$A_1 \sqsubseteq \forall r.B$,
(F2)	$A_1 \sqsubseteq \exists r.B$,	(F4)	$A_1 \sqsubseteq \leqslant 1\, r.B$,

where $A_1, \ldots, A_n, B$ are concept names and r is a role. Axioms (F2) are called *existential*. W.l.o.g. we treat here only Horn-$\mathcal{SHIQ}$ TBoxes in normalized form; our results generalize to full Horn-$\mathcal{SHIQ}$ by means of TBox *normalization*; see e.g. (Kazakov 2009; Krötzsch, Rudolph, and Hitzler 2007) for a definition and normalization procedures.

An Horn-$\mathcal{ALCHIQ}$ TBox is a Horn-$\mathcal{SHIQ}$ TBox with no transitivity axioms. Horn-$\mathcal{ALCHIQ}^{\sqcap}$ TBoxes are obtained by allowing *role conjunction* $r_1 \sqcap r_2$, where r_1, r_2 are roles (we use it for a similar purpose as Glimm et al. (2008)). In any interpretation $\mathcal{I}$, $(r_1 \sqcap r_2)^{\mathcal{I}} = r_1^{\mathcal{I}} \cap r_2^{\mathcal{I}}$. We let $\mathsf{inv}(r_1 \sqcap r_2) = \mathsf{inv}(r_1) \sqcap \mathsf{inv}(r_2)$ and assume w.l.o.g. that for each role inclusion $r \sqsubseteq s$ of an Horn-$\mathcal{ALCHIQ}^{\sqcap}$ TBox $\mathcal{T}$, (i) $\mathsf{inv}(r) \sqsubseteq \mathsf{inv}(s) \in \mathcal{T}$, and (ii) $s \in \{p, p^-\}$ for a role name p. For a set W and a concept or role conjunction $\Gamma = \gamma_1 \sqcap \ldots \sqcap \gamma_m$, we write $\Gamma \subseteq W$ for $\{\gamma_1, \ldots, \gamma_m\} \subseteq W$.

Ontologies and Knowledge Bases

Following (Levy and Rousset 1998) we now define *knowledge bases (KBs)*. Let $\mathsf{N_I}$, $\mathsf{N_V}$ and $\mathsf{N_D}$ be countable infinite sets of *constants* (or, *individuals*), *variables* and *Datalog relations*, respectively; we assume these sets as well as $\mathsf{N_C}$ and $\mathsf{N_R}$ are all mutually disjoint. Each $\sigma \in \mathsf{N_D}$ has an associated non-negative integer *arity*. An *atom* is an expression $p(\vec{t})$, where $\vec{t} \in (\mathsf{N_I})^n \cup (\mathsf{N_V})^n$, and (i) $p \in \mathsf{N_C}$ and $n = 1$, (ii) $p \in \mathsf{N_R}$ and $n = 2$, or (iii) $p \in \mathsf{N_D}$ and n is the arity of p. If $\vec{t} \in (\mathsf{N_I})^n$, then $p(\vec{t})$ is *ground*. Ground atoms $A(a)$ and $r(a, b)$, where $A \in \mathsf{N_C}$ and r is a role, are *concept* and *role assertions*, respectively. An ABox $\mathcal{A}$ is a finite set of ground atoms. A rule ρ is an expression of the form

$$h(\vec{u}) \leftarrow p_1(\vec{v_1}), \ldots, p_m(\vec{v_m}) \qquad (1)$$

where $h(\vec{u})$ is an atom (the *head*), $\{p_1(\vec{v_1}), \ldots, p_m(\vec{v_m})\}$ are also atoms (the *body* atoms, denoted $body(\rho)$), and $\vec{u}, \vec{v_1}, \ldots, \vec{v_m}$ are tuples of variables. The variables in $\vec{u}$ are *distinguished*. A *KB* is a tuple $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{P})$, where $\mathcal{T}$ is a TBox, $\mathcal{A}$ is an ABox, and $\mathcal{P}$ is a set of rules (a *program*).

The semantics for a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{P})$ is given by extending an interpretation $\mathcal{I}$ to symbols in $\mathsf{N_I} \cup \mathsf{N_D}$. For any $c \in \mathsf{N_I}$ and $p \in \mathsf{N_D}$ of arity n, we have $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and $p^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$. A *match* in $\mathcal{I}$ for a rule ρ of the form (1) is a mapping from variables in ρ to elements in $\Delta^{\mathcal{I}}$ such that $\pi(\vec{t}) \in p^{\mathcal{I}}$ for each body atom $p(\vec{t})$ of ρ. We define:

(a) $\mathcal{I} \models \rho$ if $\pi(\vec{u}) \in h^{\mathcal{I}}$ for every match π for ρ in $\mathcal{I}$,
(b) $\mathcal{I} \models \mathcal{P}$ if $\mathcal{I} \models \rho$ for each $\rho \in \mathcal{P}$,
(c) $\mathcal{I} \models \mathcal{A}$ if $(\vec{c})^{\mathcal{I}} \in p^{\mathcal{I}}$ for all $p(\vec{c}) \in \mathcal{A}$,
(d) $\mathcal{I} \models \mathcal{K}$ if $\mathcal{I} \models \mathcal{T}$, $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \models \mathcal{P}$.

Finally, given a ground atom $p(\vec{c})$, $\mathcal{K} \models p(\vec{c})$ if $(\vec{c})^{\mathcal{I}} \in p^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$. We recall *weak DL-safety* (Rosati 2006). A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{P})$ is weakly DL-safe if each rule $\rho \in \mathcal{P}$ satisfies the next condition: every distinguished variable x of ρ occurs in some body atom $p(\vec{t})$ of ρ such that $p \in \mathsf{N_D}$. We make the *Unique Name Assumption (UNA)*.

A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \emptyset)$ is an *ontology* (for brevity, we use $\mathcal{O} = (\mathcal{T}, \mathcal{A})$). A *conjunctive query (CQ)* q over $\mathcal{O}$ is a rule of the form (1) such that h does not occur in $\mathcal{O}$. By $ans(\mathcal{I}, q)$ we denote the set of all $\vec{c} \in \mathsf{N_I}^{|\vec{u}|}$ such that there is a match π for q with $\pi(\vec{u}) = (\vec{c})^{\mathcal{I}}$. By $ans(\mathcal{O}, q)$ we denote the *answer to* q *over* $\mathcal{O}$ defined as the set of all $\vec{c} \in \mathsf{N_I}^{|\vec{u}|}$ such that $\vec{c} \in ans(\mathcal{I}, q)$ for every model $\mathcal{I}$ of $\mathcal{O}$.

Note that, for a KB $\mathcal{K} = (\emptyset, \mathcal{A}, \mathcal{P})$, $\mathcal{A} \cup \mathcal{P}$ is an ordinary Datalog program with constraints (cf. (Dantsin et al. 2001)). By *models* of Datalog programs, we mean Herbrand models, and we recall that a consistent $\mathcal{A} \cup \mathcal{P}$ has a unique least (Herbrand) model $MM(\mathcal{A} \cup \mathcal{P})$.

We also consider programs $\mathcal{P}$ with atoms $r^-(x, y)$, $r \in \mathsf{N_R}$. Its semantics is given by the semantics of the program $\mathcal{P}'$ obtained by replacing in $\mathcal{P}$ each $r^-(x, y)$ by $r(y, x)$.

Elimination of Transitivity Axioms It is handy to eliminate transitivity axioms from $\mathcal{SHIQ}$ TBoxes (see, e.g., Hustadt et al. (2007)). We use the transformation from (Kazakov 2009), which ensures the resulting TBox is in normal form.

Definition 1. Let $\mathcal{T}^*$ be the Horn-$\mathcal{ALCHIQ}$ TBox obtained from a Horn-$\mathcal{SHIQ}$ TBox $\mathcal{T}$ by (i) adding for every $A \sqsubseteq \forall s.B \in \mathcal{T}$ and every transitive role r with $r \sqsubseteq^*_{\mathcal{T}} s$, the axioms $A \sqsubseteq \forall r.B^r$, $B^r \sqsubseteq \forall r.B^r$ and $B^r \sqsubseteq B$, where B^r is a fresh concept name; and (ii) removing all transitivity axioms.

As the transformation does not preserve answers to CQs, we relax the notion of match.

Definition 2. Let $\mathcal{T}$ be a Horn-$\mathcal{SHIQ}$ TBox. A $\mathcal{T}$*-match* for a query q in an interpretation $\mathcal{I}$ is a mapping π from variables of q to elements in $\Delta^{\mathcal{I}}$ such that

(a) if $\alpha = p(\vec{t})$ is a body atom in q, where $p \in \mathsf{N_C} \cup \mathsf{N_D}$ or p is a simple role in $\mathcal{T}$, then $\pi(\vec{t}) \in p^{\mathcal{I}}$, and
(b) if $\alpha = s(x, y)$ with s non-simple, then there exist $d_1 \in \Delta^{\mathcal{I}}, \ldots, d_k \in \Delta^{\mathcal{I}}$ and a transitive $r \sqsubseteq^*_{\mathcal{T}} s$ s.t. $d_1 = \pi(x)$, $d_k = \pi(y)$, and $(d_i, d_{i+1}) \in r^{\mathcal{I}}$ for all each $1 \leq i < k$.

The sets $ans^{\mathcal{T}}(\mathcal{I}, q)$ and $ans^{\mathcal{T}}(\mathcal{O}, q)$ are defined as $ans(\mathcal{I}, q)$ and $ans(\mathcal{O}, q)$ but using $\mathcal{T}$-matches instead of matches. The next claim follows from known techniques, see e.g. (Eiter, Ortiz, and Simkus 2012) for a similar result.

Proposition 1. *For any Horn-$\mathcal{SHIQ}$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ and CQ q, we have $ans(\mathcal{O}, q) = ans^{\mathcal{T}}((\mathcal{T}^*, \mathcal{A}), q)$.*

$$\frac{M \sqsubseteq \exists S.(N \sqcap N') \quad N \sqsubseteq A}{M \sqsubseteq \exists S.(N \sqcap N' \sqcap A)}\ \mathbf{R}^c_{\sqsubseteq}$$

$$\frac{M \sqsubseteq \exists (S \sqcap S').N \quad S \sqsubseteq r}{M \sqsubseteq \exists (S \sqcap S' \sqcap r).N}\ \mathbf{R}^r_{\sqsubseteq}$$

$$\frac{M \sqsubseteq \exists S.(N \sqcap \bot)}{M \sqsubseteq \bot}\ \mathbf{R}_{\bot}$$

$$\frac{M \sqsubseteq \exists (S \sqcap r).N \quad A \sqsubseteq \forall r.B}{M \sqcap A \sqsubseteq \exists (S \sqcap r).(N \sqcap B)}\ \mathbf{R}_{\forall}$$

$$\frac{M \sqsubseteq \exists (S \sqcap \mathsf{inv}(r)).(N \sqcap A) \quad A \sqsubseteq \forall r.B}{M \sqsubseteq B}\ \mathbf{R}^-_{\forall}$$

$$\frac{\begin{array}{l} M \sqsubseteq \exists (S \sqcap r).(N \sqcap B) \quad A \sqsubseteq\ \leqslant 1\, r.B \\ M' \sqsubseteq \exists (S' \sqcap r).(N' \sqcap B) \end{array}}{M \sqcap M' \sqcap A \sqsubseteq \exists (S \sqcap S' \sqcap r).(N \sqcap N')}\ \mathbf{R}_{\leq}$$

$$\frac{\begin{array}{l} M \sqsubseteq \exists (S \sqcap \mathsf{inv}(r)).(N_1 \sqcap N_2 \sqcap A) \quad A \sqsubseteq\ \leqslant 1\, r.B \\ N_1 \sqcap A \sqsubseteq \exists (S' \sqcap r).(N' \sqcap B \sqcap C) \end{array}}{M \sqcap B \sqsubseteq C \quad M \sqcap B \sqsubseteq \exists (S \sqcap \mathsf{inv}(S' \sqcap r)).(N_1 \sqcap N_2 \sqcap A)}\ \mathbf{R}^-_{\leq}$$

Table 2: Inference rules. $M^{(\prime)}$, $N^{(\prime)}$, (resp., $S^{(\prime)}$) are conjunctions of atomic concepts (roles); A, B are atomic concepts

Canonical Models

A stepping stone for tailoring query answering methods for Horn DLs and languages like $Datalog^{\pm}$ is the *canonical model property* (Eiter et al. 2008b; Ortiz, Rudolph, and Simkus 2011; Calì, Gottlob, and Lukasiewicz 2009). In particular, for a consistent Horn-$\mathcal{ALCHIQ}^{\sqcap}$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$, there exists a model $\mathcal{I}$ of $\mathcal{O}$ that can be homomorphically mapped into any other model $\mathcal{I}'$ of $\mathcal{O}$. We show that such an $\mathcal{I}$ can be built in three steps:

(1) Close $\mathcal{T}$ under specially tailored inferences rules.
(2) Close $\mathcal{A}$ under all but existential axioms of $\mathcal{T}$.
(3) Extend $\mathcal{A}$ by "applying" the existential axioms of $\mathcal{T}$.

For Step (1), we tailor from the inference rules in (Kazakov 2009; Ortiz, Rudolph, and Simkus 2010) a calculus to support model building for Horn-$\mathcal{ALCHIQ}^{\sqcap}$ ontologies.

Definition 3. Given a Horn-$\mathcal{ALCHIQ}^{\sqcap}$ TBox $\mathcal{T}$, $\Xi(\mathcal{T})$ is the TBox obtained from $\mathcal{T}$ by exhaustively applying the inference rules in Table 2.

For Step (2), we use Datalog rules that express the semantics of GCIs, ignoring existential axioms.

Definition 4. Given a Horn-$\mathcal{ALCHIQ}^{\sqcap}$ TBox $\mathcal{T}$, $\mathsf{cr}(\mathcal{T})$ is the Datalog program described in Table 3.

Given a consistent Horn-$\mathcal{ALCHIQ}^{\sqcap}$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$, the least model $\mathcal{J}$ of the Datalog program $\mathsf{cr}(\mathcal{T}) \cup \mathcal{A}$ is almost a canonical model of $\mathcal{O}$; however, existential axioms may be violated. We deal with this in Step (3), by extending $\mathcal{J}$ with new domain elements as required by axioms $A \sqsubseteq \exists r.N$ in $\Xi(\mathcal{T})$. This step is akin to the database *chase* (Maier and Mendelzon 1979), where fresh values and tuples are introduced to satisfy a given set of dependencies.

Definition 5. Let $\mathcal{T}$ be a Horn-$\mathcal{ALCHIQ}^{\sqcap}$ TBox and $\mathcal{I}$ an interpretation. A GCI $M \sqsubseteq \exists S.N$ is *applicable at* $e \in \Delta^{\mathcal{I}}$ if

(a) $e \in M^{\mathcal{I}}$,

$B(y) \leftarrow A(x), r(x,y)$ for each $A \sqsubseteq \forall r.B \in \mathcal{T}$
$B(x) \leftarrow A_1(x), \ldots, A_n(x)$ for all $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \in \Xi(\mathcal{T})$
$r(x,y) \leftarrow r_1(x,y), \ldots, r_n(x,y)$ for all $r_1 \sqcap \ldots \sqcap r_n \sqsubseteq r \in \mathcal{T}$
$\bot(x) \leftarrow A(x), r(x,y_1), r(x,y_2), B(y_1), B(y_2), y_1 \neq y_2$ for each $A \sqsubseteq \leqslant 1\, r.B \in \mathcal{T}$
$\Gamma \leftarrow A(x), A_1(x), \ldots, A_n(x), r(x,y), B(y)$ for all $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq \exists(r_1 \sqcap \ldots \sqcap r_m).B_1 \sqcap \ldots \sqcap B_k$ and $A \sqsubseteq \leqslant 1\, r.B$ of $\Xi(\mathcal{T})$ such that $r = r_i$ and $B = B_j$ for some i, j with $\Gamma \in \{B_1(y), \ldots, B_k(y), r_1(x,y), \ldots, r_k(x,y)\}$

Table 3: Completion rules $\mathsf{cr}(\mathcal{T})$ for Horn-$\mathcal{ALCHIQ}^{\sqcap}$

(b) there is no $e' \in \Delta^{\mathcal{I}}$ with $(e, e') \in S^{\mathcal{I}}$ and $e' \in N^{\mathcal{I}}$,

(c) there is no axiom $M' \sqsubseteq \exists S'.N' \in \mathcal{T}$ such that $e \in (M')^{\mathcal{I}}$, $S \subseteq S'$, $N \subseteq N'$, and $S \subset S'$ or $N \subset N'$.

An interpretation $\mathcal{J}$ obtained from $\mathcal{I}$ by an *application* of an applicable axiom $M \sqsubseteq \exists S.N$ at $e \in \Delta^{\mathcal{I}}$ is defined as:

- $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}} \cup \{d\}$ with d a new element not present in $\Delta^{\mathcal{I}}$ (we call d a *successor* of e),
- For each atomic $A \in \mathsf{N_C}$ and $o \in \Delta^{\mathcal{J}}$, we have $o \in A^{\mathcal{J}}$ if (a) $o \in \Delta^{\mathcal{I}}$ and $o \in A^{\mathcal{I}}$; or (b) $o = d$ and $A \in N$.
- For each role name r and $o, o' \in \Delta^{\mathcal{J}}$, we have $(o, o') \in r^{\mathcal{J}}$ if (a) $o, o' \in \Delta^{\mathcal{I}}$ and $(o, o') \in r^{\mathcal{I}}$; or (b) $(o, o') = (e, d)$ and $r \in S$; or (c) $(o, o') = (d, e)$ and $\mathsf{inv}(r) \in S$.

We denote by *chase*$(\mathcal{I}, \mathcal{T})$ a possibly infinite interpretation obtained from $\mathcal{I}$ by applying the existential axioms in $\mathcal{T}$. We require the application to be *fair*: the application of an applicable axiom can not be infinitely postponed.

We note that *chase*$(\mathcal{I}, \mathcal{T})$ is unique up to renaming of domain elements. As usual in DLs, it can be seen as a 'forest': application of existential axioms simply attaches 'trees' to a possibly arbitrarily shaped $\mathcal{I}$. The following statement can be shown similarly as in (Ortiz, Rudolph, and Simkus 2011).

Proposition 2. *Let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ be a Horn-$\mathcal{ALCHIQ}^{\sqcap}$ ontology. Then $\mathcal{O}$ is consistent iff $\mathcal{A} \cup \mathsf{cr}(\mathcal{T})$ consistent. Moreover, if $\mathcal{O}$ is consistent, then*

(a) chase$(MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T})), \Xi(\mathcal{T}))$ is a model of $\mathcal{O}$, and

(b) chase$(MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T})), \Xi(\mathcal{T}))$ can be homomorphically mapped into any model of $\mathcal{O}$.

In database terms, this means that checking consistency of $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ reduces to evaluating the (plain) Datalog query $\mathsf{cr}(\mathcal{T})$ over the database $\mathcal{A}$. Note that $\Xi(\mathcal{T})$ can be computed in exponential time in size of $\mathcal{T}$: the calculus only infers axioms of the form $M \sqsubseteq B$ and $M \sqsubseteq \exists S.N$, where M, N are conjunctions of atomic concepts, B is atomic and S is a conjunction of roles. The number of such axiom is single exponential in the size of $\mathcal{T}$.

Rewriting Rules and Programs

The following is immediate from Propositions 1 and 2:

Theorem 3. *Let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ be a Horn-$\mathcal{SHIQ}$ ontology. Then $\mathcal{A} \cup \mathsf{cr}(\mathcal{T}^*)$ is consistent iff $\mathcal{O}$ is consistent. Moreover, if $\mathcal{O}$ is consistent, then $ans(\mathcal{O}, q) = ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q)$, where $\mathcal{I}_{\mathcal{O}} = chase(MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T}^*), \Xi(\mathcal{T}^*))$.*

Computing $ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q)$ is still tricky because $\mathcal{I}_{\mathcal{O}}$ can be infinite. So we rewrite q into a set Q of CQs such that $ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q) = \bigcup_{q' \in Q} ans^{\mathcal{T}}(MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T}^*), q')$. This yields an algorithm for answering q over $\mathcal{O}$, using only the finite $MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T}^*))$.

The rewriting of q is done in steps with the following intuition. Suppose q has a $\mathcal{T}$-match π in $\mathcal{I}_{\mathcal{O}}$. A rewrite step clips off some variable x such that $\pi(x)$ has no descendant in the image of π, merges the variables that are mapped to the predecessor of $\pi(x)$, and adds concept atoms to the resulting q' that ensure that q has a $\mathcal{T}$-match whenever q' does.

Definition 6. For a rule ρ and a Horn-$\mathcal{SHIQ}$ TBox $\mathcal{T}$, we write $\rho \rightarrow_{\mathcal{T}} \rho'$ if ρ' is obtained from ρ by the following steps:

(S1) Select an arbitrary non-distinguished variable x in ρ.

(S2) Replace each role atom $r(x, y)$ in ρ, where y is arbitrary, by the atom $\mathsf{inv}(r)(y, x)$.

(S3) For each atom $\alpha = s(y, x)$ in ρ, where y is arbitrary and s is non-simple, either leave α untouched or replace it by two atoms $r(y, u_\alpha), r(u_\alpha, x)$, where u_α is a fresh variable and r is a transitive role with $r \sqsubseteq^*_{\mathcal{T}} s$.

(S4) Form some partitioning of the set $\{y \mid \exists r : r(y, x) \in body(\rho)\} \cup \{x\}$ into sets V_x and V_p, in such a way that $x \in V_x$ and V_x has no distinguished variables.

(S5) Select some $M \sqsubseteq \exists S.N \in \Xi(\mathcal{T}^*)$ such that
 (a) $\{r \mid r(y, x) \in body(\rho) \wedge y \in V_p\} \subseteq S$,
 (b) $\{A \mid A(z) \in body(\rho) \wedge z \in V_x\} \subseteq N$, and
 (c) for each variable $z \in V_x$ and atom $r(z, x)$ in $body(\rho)$ there is a transitive $s \sqsubseteq^*_{\mathcal{T}} r$ such that
  i. $\{s, s^-\} \subseteq S$, or
  ii. there is an axiom $M' \sqsubseteq \exists S'.N' \in \Xi(\mathcal{T}^*)$ such that $M' \subseteq N$ and $\{s, s^-\} \subseteq S'$.

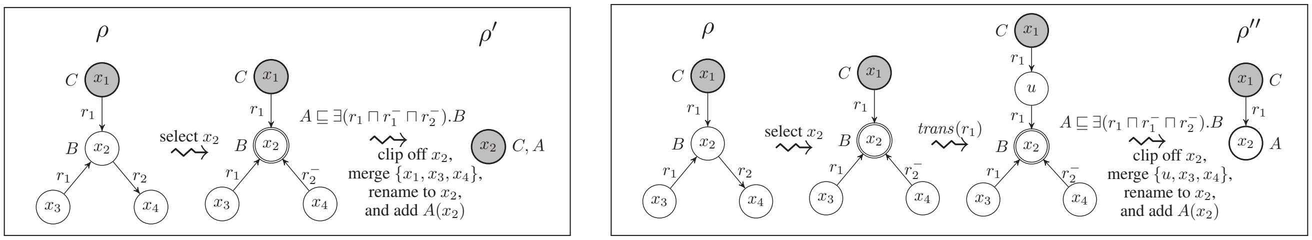
(S6) Drop each atom from ρ containing a variable from V_x.

(S7) Rename each $y \in V_p$ of ρ by x.

(S8) Add the atoms $\{A(x) \mid A \in M\}$ to ρ.

We write $\rho \rightarrow^*_{\mathcal{T}} \rho'$ if ρ' can be obtained from ρ by finitely many rewrite iterations. We let $\mathsf{rew}_{\mathcal{T}}(\rho) = \{\rho' \mid \rho \rightarrow^*_{\mathcal{T}} \rho'\}$. For a set $\mathcal{P}$ of rules, $\mathsf{rew}_{\mathcal{T}}(\mathcal{P}) = \bigcup_{\rho \in \mathcal{P}} \mathsf{rew}_{\mathcal{T}}(\rho)$.

In (S1) we guess the variable x and, for technical reasons, in (S2) we invert all atoms of the form $r(x, y)$. For all atoms $s(y, x)$ where s is not simple, there must exist a transitive $r \sqsubseteq^*_{\mathcal{T}} s$ such that there is an r-path from $\pi(y)$ to $\pi(x)$. For the atoms where this path has length at least 2, we use in (S3) the role r and introduce an 'intermediate' variable u that can be mapped to the parent p of $\pi(x)$. Now we know that for each atom $r(y, x)$ the 'neighbor' variable y must mapped to (a) the parent p of $\pi(x)$, or (b) in case r is non-simple, possibly to $\pi(x)$. In (S4) we guess a partition of the neighbor variables into V_p and V_x, which correspond to cases (a) and (b), respectively. In (S5) we select some axiom that would cause the existence of $\pi(x)$ when applied at p. Then we can clip off x in (S6), merge all variables of V_p (and, for technical reasons, we rename them to x, which does not occur in ρ anymore) in (S7), and add to ρ atoms for the concepts on the left hand side of the selected axiom.



(a) Example 2: Query rewriting with simple roles

(b) Example 3: Query rewriting with transitive roles

Figure 1: Examples of query rewriting

Example 2. In our first example, illustrated in Figure 1a, all roles are simple. Let $\rho: q(x_1) \leftarrow C(x_1), B(x_2), r_1(x_1, x_2), r_1(x_3, x_2), r_2(x_2, x_4)$, and assume $A \sqsubseteq \exists(r_1 \sqcap r_1^- \sqcap r_2^-).B \in \Xi(\mathcal{T}^*)$. In (S1) we select the non-distinguished variable x_2. Next, in (S2), we replace $r_2(x_2, x_4)$ by $r_2^-(x_4, x_2)$. Since all roles are simple, we do nothing in (S3). In (S4) we choose $V_{x_2} = \{x_2\}$ and $V_p = \{x_1, x_3, x_4\}$, and in (S5), $A \sqsubseteq \exists(r_1 \sqcap r_1^- \sqcap r_2^-).B$. Then we clip off x_2 in (S6), merge all variables in V_p and rename them to x_2 in (S7), and add $A(x_2)$ in (S8), to obtain $\rho' : q(x_2) \leftarrow C(x_2), A(x_2)$.

Example 3 (ctd). Now assume that $trans(r_1) \in \mathcal{T}$. As shown in Figure 3, in (S3) we choose to replace $r_1(x_1, x_2)$ with $r_1(x_1, u)$ and $r_1(u, x_2)$. In (S4) we choose $V_{x_2} = \{x_2\}$ and $V_p = \{u, x_3, x_4\}$. Then we proceed similarly as above to obtain $\rho'' : q(x_1) \leftarrow C(x_1), r_1(x_1, x_2), A(x_2)$.

This rewriting also provides a reduction of query answering to reasoning in Datalog, provided that the completion rules take transitivity into account.

Definition 7. For a Horn-$\mathcal{SHIQ}$ TBox $\mathcal{T}$, $\mathsf{cr}(\mathcal{T}) = \mathsf{cr}(\mathcal{T}^*) \cup \{r(x, z) \leftarrow r(x, y), r(y, z) \mid r \text{ is transitive in } \mathcal{T}\}$.

Now we can show the following:

Theorem 4. *Assume a consistent Horn-$\mathcal{SHIQ}$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ and a conjunctive query q. Then $ans(\mathcal{O}, q) = \bigcup_{q' \in \mathsf{rew}_{\mathcal{T}}(q)} ans(MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T})), q')$.*

Proof. (Sketch) Using standard techniques, it is easy to show $ans(MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T})), q) = ans^{\mathcal{T}}(MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T}^*)), q)$ for every ABox $\mathcal{A}$ and CQ q. By this and Theorem 3, we only need to show $ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q) = ans^{\mathcal{T}}(\mathcal{J}, \mathsf{rew}_{\mathcal{T}}(q))$, where $\mathcal{J} = MM(\mathcal{A} \cup \mathsf{cr}(\mathcal{T}^*))$ and $\mathcal{I}_{\mathcal{O}} = chase(\mathcal{J}, \Xi(\mathcal{T}^*))$.

To show $ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q) \supseteq ans^{\mathcal{T}}(\mathcal{J}, \mathsf{rew}_{\mathcal{T}}(q))$, assume $\vec{u} \in ans^{\mathcal{T}}(\mathcal{J}, \mathsf{rew}_{\mathcal{T}}(q))$. Then there is a query $q' \in \mathsf{rew}_{\mathcal{T}}(q)$ and a $\mathcal{T}$-match $\pi_{q'}$ for q' in $\mathcal{J}$ such that $\vec{u} = \pi_{q'}(\vec{x})$. By the construction of $\mathcal{I}_{\mathcal{O}}$, we also have $\vec{u} \in ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q')$. If $q' \neq q$ (otherwise we are done), there is $n > 0$ such that $q_0 \to_{\mathcal{T}} q_1, \cdots, q_{n-1} \to_{\mathcal{T}} q_n$ with $q_0 = q$ and $q_n = q'$. Thus to prove the claim it suffices to show that $\vec{u} \in ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q_i)$ implies $\vec{u} \in ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q_{i-1})$, where $0 < i \leq n$.

Assume that q_i was obtained from q_{i-1} by a rewriting step, where some variable x was chosen in (S1), some V_x and V_p in (S4), and some axiom $M \sqsubseteq \exists S.N$ in (S5). Suppose π_{q_i} is a $\mathcal{T}$-match for q_i in $\mathcal{I}_{\mathcal{O}}$ with $\vec{u} = \pi_{q'}(\vec{x})$, i.e. $\vec{u} \in ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q_i)$. Let $d = \pi_{q_i}(x)$. Due to step (S8) in the rewriting and the fact that $\mathcal{I}_{\mathcal{O}}$ is a model of $\mathcal{O}$, we have $d \in (\exists S.N)^{\mathcal{I}_{\mathcal{O}}}$. Then there is $d' \in \Delta^{\mathcal{I}_{\mathcal{O}}}$ such that $(d, d') \in S^{\mathcal{I}_{\mathcal{O}}}$ and $d' \in N^{\mathcal{I}_{\mathcal{O}}}$. One can show that the mapping $\pi_{q_{i-1}}$ defined as follows is a $\mathcal{T}$-match for q_{i-1} in $\mathcal{I}_{\mathcal{O}}$: $\pi_{q_{i-1}}(z) = d'$ for all $z \in V_x$, $\pi_{q_{i-1}}(u) = d$ for all $u \in V_p$, and $\pi_{q_{i-1}}(z) = \pi_{q_i}(z)$ for the remaining variables z.

To show $ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q) \subseteq ans^{\mathcal{T}}(\mathcal{J}, \mathsf{rew}_{\mathcal{T}}(q))$ we prescribe the naming of fresh domain elements introduced when chasing $\mathcal{J}$ w.r.t. $\Xi(\mathcal{T}^*)$, and use an element $e \cdot n$ for some integer n for each successor of an element e. For $d \in \Delta^{\mathcal{J}}$ we let $|d| = 0$, and for $w \cdot n \in \Delta^{\mathcal{I}_{\mathcal{O}}}$ we let $|w \cdot n| = |w| + 1$. Assume a tuple $\vec{u} \in ans^{\mathcal{T}}(\mathcal{I}_{\mathcal{O}}, q)$. By definition, there is $\mathcal{T}$-match π_q for q in $\mathcal{I}_{\mathcal{O}}$ such that $\vec{u} = \pi_q(\vec{x})$. We have to show that there exists $q' \in \mathsf{rew}_{\mathcal{T}}(q)$ and a $\mathcal{T}$-match $\pi_{q'}$ for q' in $\mathcal{J}$ such that $\vec{u} = \pi_{q'}(\vec{x})$.

For a $\mathcal{T}$-match π' in $\mathcal{I}_{\mathcal{O}}$, let $deg(\pi') = \sum_{y \in rng(\pi')} |\pi'(y)|$. Then, given that $q \in \mathsf{rew}_{\mathcal{T}}(q)$, to prove the claim it suffices to prove the following statement: if $q_1 \in \mathsf{rew}_{\mathcal{T}}(q)$ has a $\mathcal{T}$-match π_{q_1} for q_1 in $\mathcal{I}_{\mathcal{O}}$ such that $\vec{u} = \pi_{q_1}(\vec{x})$ and $deg(\pi_{q_1}) > 0$, then there exists $q_2 \in \mathsf{rew}_{\mathcal{T}}(q)$ that has a $\mathcal{T}$-match π_{q_2} for q_2 in $\mathcal{I}_{\mathcal{O}}$ such that $\vec{u} = \pi_{q_2}(\vec{x})$ and $deg(\pi_{q_2}) < deg(\pi_{q_1})$.

The query q_2 is obtained by selecting for (S1) an x such that $\pi_{q_1}(x) \notin \mathsf{N_I}$ (which exists because $deg(\pi_{q_1}) > 0$) and there is no variable x' of q_1 with $\pi_{q_1}(x)$ a prefix of $\pi_{q_1}(x')$ (that is, x is a leaf in the subforest of $\mathcal{I}_{\mathcal{O}}$ induced by the image of π_{q_1}). Let $d_x = \pi_{q_1}(x)$, and let d_p be the parent of d_x, i.e. $d_x = d_p \cdot n$ for some integer n. For Step (S3), we choose to rewrite the atoms $\Gamma = \{s(y, x) \in q_1 \mid \pi_{q_1}(y) \neq d_x \wedge \pi_{q_1}(y) \neq d_p\}$. By definition of $\mathcal{T}$-match, for each such atom there is a transitive role $r_s \sqsubseteq^*_{\mathcal{T}} s$ such that there is an r_s-path from $\pi_{q_1}(y)$ to $\pi_{q_1}(x)$; this r_s can be used in the fresh atoms. Let V_t be the set of fresh variables introduced in this step. For Step (S4), let $V_x = \{z \in var(q_1) \mid \pi_{q_1}(z) = d_x\}$ and $V_p = \{z \in var(q_1) \mid \pi_{q_1}(z) = d_p\} \cup V_t$. We know from the construction of $\mathcal{I}_{\mathcal{O}}$ that d_x was introduced by an application of an axiom $ax = M \sqsubseteq \exists S.N \in \Xi(\mathcal{T}^*)$ such that $d_p \in M^{\mathcal{I}_{\mathcal{O}}}$; we choose this axiom for Step (S5). It is not hard to show that (S5.a–c) hold. Finally, a $\mathcal{T}$-match π_{q_2} for q_2 in $\mathcal{I}_{\mathcal{O}}$ such that $\vec{u} = \pi_{q_2}(\vec{x})$ and $deg(\pi_{q_2}) < deg(\pi_{q_1})$ is obtained from π_{q_1} by setting $\pi_{q_2}(z) = \pi_{q_1}(z)$ for all z of q_2 with $z \neq x$, and $\pi_{q_2}(x) = d_p$. □

By the above, we can answer q over $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ by posing $\mathsf{rew}_{\mathcal{T}}(q)$ over the Datalog program $\mathcal{A} \cup \mathsf{cr}(\mathcal{T})$. The method also applies to KBs $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{P})$, where $\mathcal{T}$ is in Horn-$\mathcal{SHIQ}$ and $\mathcal{P}$ is weakly DL-safe. The ground atomic consequences of $\mathcal{K}$ can be collected by fixed-point computation: until no new consequences are derived, pose rules in $\mathcal{P}$ as CQs over $(\mathcal{T}, \mathcal{A})$ and put the obtained answers into $\mathcal{A}$. If we employ the rewriting in Definition 6, this computation can be achieved using a plain Datalog program.

Theorem 5. *For a ground atom α over a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{P})$ where $\mathcal{T}$ is a Horn-$\mathcal{SHIQ}$ TBox and $\mathcal{P}$ is weakly DL-safe, we have $(\mathcal{T}, \mathcal{A}, \mathcal{P}) \models \alpha$ iff $\mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(\mathcal{P}) \cup \mathcal{A} \models \alpha$.*

Proof. (Sketch) Let $\mathcal{P}' = \mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(\mathcal{P})$. For the "if" direction, the only interesting case is when $(\mathcal{T}^*, \mathcal{A})$ is consistent. In this case it suffices to show that the rules of $\mathcal{P}'$ applied on $\mathcal{A}$ derive only atoms that are consequences of $(\mathcal{T}, \mathcal{A}, \mathcal{P})$. This is straightforward for all rules in $\mathsf{cr}(\mathcal{T})$, since the rules are already logical consequences of $\mathcal{T}$. For the rules in $\rho' \in \mathsf{rew}_{\mathcal{T}}(\mathcal{P})$ it is a consequence of Theorem 4.

To show the "only if" direction, again the only interesting case is where $\mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(\mathcal{P}) \cup \mathcal{A}$ is consistent. In this case, one can consider the set $\mathcal{A}'$ of all ground α such that $\mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(\mathcal{P}) \cup \mathcal{A} \models \alpha$, and show that $\mathcal{I} = chase(\mathcal{A}', \Xi(\mathcal{T}^*))$ is a model of $(\mathcal{T}, \mathcal{A}, \mathcal{P})$ such that $\mathcal{I} \not\models \alpha$ for all α such that $\mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(\mathcal{P}) \cup \mathcal{A} \not\models \alpha$. ☐

This reduction yields a worst-case optimal algorithm.

Theorem 6. *For a ground atom α over a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{P})$ where $\mathcal{T}$ is a Horn-$\mathcal{SHIQ}$ TBox and $\mathcal{P}$ is weakly DL-safe, checking $(\mathcal{T}, \mathcal{A}, \mathcal{P}) \models \alpha$ is* ExpTime*-complete in general, and* PTime*-complete in data complexity.*

Proof. (Sketch) By Theorem 5, checking $(\mathcal{T}, \mathcal{A}, \mathcal{P}) \models \alpha$ is equivalent to deciding $\mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(\mathcal{P}) \cup \mathcal{A} \models \alpha$. We analyze the computational cost of the latter check.

We recall that $\Xi(\mathcal{T}^*)$ can be computed in exponential time in size of $\mathcal{T}$ and is independent from $\mathcal{A}$. The program rewriting $\mathsf{rew}_{\mathcal{T}}(\mathcal{P})$ is finite and computable in time exponential in the size of $\mathcal{T}$ and $\mathcal{P}$: rules in $\mathsf{rew}_{\mathcal{T}}(\rho)$, where $\rho \in \mathcal{P}$, use only relation names and variables that occur in ρ and $\mathcal{T}$ (fresh variables introduced in (S3) are eliminated in (S6) and (S7)). Hence, the size of each rule resulting from a rewrite step is of size polynomial in the size of $\mathcal{T}$ and $\mathcal{P}$, and thus the number of rules in $\mathsf{rew}_{\mathcal{T}}(\mathcal{P})$ is at most exponential in the size of $\mathcal{T}$ and $\mathcal{P}$. The size of $\mathsf{rew}_{\mathcal{T}}(\mathcal{P})$ is constant when data complexity is considered. Furthermore, the *grounding* of $\mathsf{cr}(\mathcal{T}) \cup \mathsf{rew}_{\mathcal{T}}(\mathcal{P}) \cup \mathcal{A}$ is exponential in the size of $\mathcal{K}$, but polynomial for fixed $\mathcal{T}$ and $\mathcal{P}$. By the complexity of Datalog, it follows that the algorithm resulting from Theorem 5 is exponential in combined but polynomial in data complexity. These results are worst-case optimal, and apply already to plain conjunctive queries (Eiter et al. 2008b). ☐

Both $\Xi(\mathcal{T}^*)$ and $\mathsf{rew}_{\mathcal{T}}(\mathcal{P})$ can be of exponential size, but this worst-case complexity is only exhibited by some 'hard' instances. Our experimental results in the next Section show that both sets are of manageable size for many ontologies.

Implementation and Experiments

The results of the previous Section directly give an algorithm for answering a CQ q over a given Horn-$\mathcal{SHIQ}$ ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$, which works as follows: (1) we eliminate transitivity axioms in $\mathcal{T}$ to get $\mathcal{T}^*$; (2) we saturate $\mathcal{T}^*$ into $\Xi(\mathcal{T}^*)$ using the calculus in Table 2; (3) we obtain $\mathsf{rew}_{\mathcal{T}}(\rho)$ by exhaustively applying to q the rewriting step in Definition 6 using the axioms $\Xi(\mathcal{T}^*)$; (4) we put together $\mathcal{A}$, the completion rules $\mathsf{cr}(\mathcal{T})$, and $\mathsf{rew}_{\mathcal{T}}(\rho)$ into a Datalog program $\mathcal{P}$; and (5) we evaluate the program $\mathcal{P}$.

A	$Q6(x)$ ←	Device(x), assistsWith(x, y), ReadingDevice(y)
	$Q7(x)$ ←	Device(x), assistsWith(x, y), ReadingDevice(y), assistsWith(y, z), SpeechAbility(z)
S	$Q6(x,z)$ ←	Investor(x), hasStock(x,y), Stock(y), Company(z), hasStock(z,y)
	$Q7(x, z, w)$ ←	Investor(x), hasStock(x, y), Stock(y), isListedIn(y, z), StockExchangeList(z), Company(w), hasStock(w, y)
U	$Q6(x, y)$ ←	Professor(x), teacherOf(x, y), GraduateCourse(y)
	$Q7(x, z)$ ←	Faculty(y), Professor(z), Student(x), memberOf(x, y), worksFor(z,y)
V	$Q6(x, y, z)$ ←	Person(x), hasRole(x, y), Leader(y), exists(y, z)
	$Q7(x,y,z,v)$ ←	Person(x), hasRole(x, y), Leader(y), exists(y, z), TemporalInterval(z), related(x, v), Country(v)

Table 4: Additional queries in rewriting evaluation

		# Rules			Time (ms)		
		ReqG	Presto	Clipper	ReqG	Presto	Clipper
A	Q1	27	53	42	281	45	50
	Q2	50	32	31	184	46	62
	Q3	104	32	31	292	27	65
	Q4	224	43	36	523	32	71
	Q5	624	37	36	1177	25	70
	Q6	364	35	30	523	31	65
	Q7	2548	43	32	7741	61	64
S	Q1	6	7	10	14	7	19
	Q2	2	3	22	263	9	22
	Q3	4	4	9	1717	10	21
	Q4	4	4	24	1611	9	23
	Q5	8	5	10	18941	10	22
	Q6	4	8	5	204	11	21
	Q7	8	6	7	1733	11	17
V	Q1	15	16	15	13	8	73
	Q2	10	3	10	16	10	58
	Q3	72	28	26	77	12	63
	Q4	185	44	41	261	17	71
	Q5	30	16	8	99	15	44
	Q6	18	22	18	27	11	69
	Q7	180	34	27	359	12	105
U	Q1	2	4	2	14 (1247)	12 (1252)	27 (1255)
	Q2	1	2	45	201 (1247)	23 (1262)	36 (1637)
	Q3	4	8	17	477 (2055)	26 (2172)	29 (1890)
	Q4	2	56	63	2431 (1260)	20 (1235)	28 (1735)
	Q5	10	8	16	7216 (1267)	26 (1305)	36 (1372)
	Q6	10	13	10	13 (1272)	14 (1260)	27 (1262)
	Q7	960	24	19	1890 (1730)	15 (1310)	35 (1322)

Table 5: Downscaling query rewriting evaluation

To evaluate the feasibility of this algorithm, we have implemented a prototype system Clipper[1] for answering CQs containing only simple roles. To the best of our knowledge, it is the first such system for full Horn-$\mathcal{SHIQ}$ (under the standard semantics of first-order logic), and in expressiveness subsumes similar $\mathcal{DL}$-*Lite* and $\mathcal{EL}$ reasoning engines.

Clipper is implemented in Java and uses OWLAPI 3.2.2 (Horridge and Bechhofer 2011) to manage ontologies. It accepts an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ and a query q in SPARQL syntax as input. Initial preprocessing involves normalization of $\mathcal{O}$ and checking that $\mathcal{T}$ is Horn. Then it applies steps (1) – (5) above (with some minor optimizations). For the Datalog evaluation in step (5), it uses DLV-20101014 (Leone et al. 2006) or Clingo 3.0.3 (Gebser et al. 2011).

Experiments We tested our Clipper system on a Pentium Core2 Duo 2.00GHZ with 2GB RAM under Ubuntu 10.04 and 512MB heap size for the Java VM. We conducted the following experiments.

[1] http://www.kr.tuwien.ac.at/research/systems/clipper/

Q1(x)←worksFor(x, y), isAffiliatedOrganizationOf(y, z)
Q2(x,z)←LeisureStudent(x), takesCourse(x,y), isTaughtBy(y,z), SportsLover(z)
Q3(x, y)←enrollIn(x, y), hasDegreeFrom(x, y)
Q4(x, z)←Student(x), hasDegreeFrom(x, y), Professor(z), worksFor(z, y)
Q5(x)←Postdoc(x), worksFor(x, y), University(y), hasAlumnus(y, x)
Q6(x)←Person(x), like(x, y), Chair(z), isHeadOf(z, v), like(z, y)
Q7(x, y)←Postdoc(x), hasAlumnus(y, x)
Q8(x, y)←GraduateCourse(x), isTaughtBy(x, y), isHeadOf(y, z)
Q9(x)←PeopleWithManyHobbies(x), isMemberOf(x, y)
Q10(x)←SportsLover(x), isHeadOf(x, y), ResearchGroup(y)

Table 6: Queries over Horn-$\mathcal{SHIQ}$ ontology

	# Rules	Rewriting (ms)	Datalog (DLV) Time (ms)
Q1	2	68	80 / 320 / 560 / 830
Q2	3	63	90 / 330 / 560 / 830
Q3	9	96	90 / 320 / 570 / 810
Q4	172	143	230 / 830 / 1430 / 1580
Q5	16	91	90 / 330 / 570 / 820
Q6	255	177	250 / 890 / 1530 / 1800
Q7	8	89	80 / 320 / 570 / 820
Q8	175	146	230 / 830 / 1430 / 1580
Q9	175	145	230 / 820 / 1400 / 1600
Q10	2	64	80 / 330 / 570 / 830

Table 7: Experiment with UOBM Horn-$\mathcal{SHIQ}$ ontology

1. Downscaling test. We compared CLIPPER with other query rewriting systems for $\mathcal{DL}$-*Lite*, viz. REQUIEM (Perez-Urbina et al. 2009) and PRESTO (Rosati and Almatelli 2010), and found that it is competitive and scales down well on $\mathcal{DL}$-*Lite* ontologies. We used the ontologies ADOLENA (A), STOCK-EXCHANGE (S), VICODI (V) , and UNIVERSITY (U), and queries (Q1–Q5) from the REQUIEM test suite, which have been widely used for system tests. In addition, we considered the queries in Table 4.

The number of rewritten queries and rewriting time are shown in Table 5 (loading and preprocessing times are excluded). CLIPPER and PRESTO generated in most cases rule sets of comparable size, and in short time; in a few cases PRESTO generated significantly less rules than CLIPPER. REQUIEM, in its G-version (which generates optimized rule sets), generated in several cases significantly more rules. This is largely explained by rule unfolding to produce CQs; still in many cases, in particular for S and U except Q7, the final result is small (but needed considerably more time).

For UNIVERSITY, we evaluated the rewritten queries over 4 different ABoxes with 67k to 320k assertions using DLV (the other ontologies in the suite don't have ABoxes). Interestingly, in all cases the execution times for the three rewritings were very similar; the average runtime of each query on the 4 ABoxes is shown in brackets.

2. Full Horn-$\mathcal{SHIQ}$. To test CLIPPER on a full Horn-$\mathcal{SHIQ}$ ontology, we modified the UOBM ontology (Ma et al. 2006), which is in $\mathcal{SHOIN}(\mathbf{D})$, by dropping or strengthening (in case of disjunctions) non-Horn-$\mathcal{SHIQ}$ TBox axioms; the final ontology has 171 TBox axioms. We used ABoxes $\mathcal{A}_i$, $1 \leq i \leq 4$, with 20k, 80k, 140k and 200k assertions. The test queries in Table 6 were tailored to require reasoning with Horn-$\mathcal{SHIQ}$ constructs unavailable in $\mathcal{DL}$-*Lite* and $\mathcal{EL}$. Table 7 presents the number of rewritten queries, rewriting time, and DLV running time of Datalog programs. The results show that CLIPPER answered all queries in reasonable time and scaled well (time printed $\mathcal{A}_1$/ $\mathcal{A}_2$ / $\mathcal{A}_3$ /$\mathcal{A}_4$). The rewriting times for all the queries are small and close. The high number of rules generated for Q4, Q6, Q8, and Q9 is due to many different possibilities to derive atoms in the query; e.g., for Professor(z) and worksFor(z,y) in Q4. However, the evaluation still performs well (it stays within a small factor).

Related Work and Conclusion

Since Calvanese et al. (2007) introduced query rewriting in their seminal work on $\mathcal{DL}$-*Lite*, many query rewriting techniques have been developed and implemented, e.g. (Perez-Urbina et al. 2009, Rosati and Almatelli 2010, Chortaras et al. 2011, Gottlob et al. 2011), usually aiming at an optimized rewriting size. Some of them also go beyond $\mathcal{DL}$-*Lite*; e.g. Perez-Urbina et al. cover $\mathcal{ELHI}$, while Gottlob et al. consider $Datalog^{\pm}$. Most approaches rewrite a query into a (union of) CQs; Rosati and Almatelli generate a non-recursive Datalog program, while Perez-Urbina et al. produce a CQ for $\mathcal{DL}$-*Lite* and a (recursive) Datalog program for DLs of the $\mathcal{EL}$ family. Our approach rewrites a CQ into a union of CQs, but generates (possible recursive) Datalog rules to capture the TBox.

Our technique resembles Rosati's (2007) for CQs in $\mathcal{EL}$, which incorporates the input CQ into the TBox *before* saturation, after which the TBox is translated into Datalog. This is best-case exponential, which we avoid (a rewrite step occurs only if the TBox has an applicable existential axiom).

Rewriting approaches for more expressive DLs are less common. A notable exception is Hustadt et al.'s (2007) translation of $\mathcal{SHIQ}$ terminologies into disjunctive Datalog, which is implemented in the KAON2 reasoner. It can be used to answer queries over arbitrary ABoxes, but supports only instance queries. An extension to CQs (without transitive roles) (Hustadt et al., 2004) is not implemented. Ortiz et al. (2010) use a Datalog rewriting to establish complexity bounds of standard reasoning in the Horn fragments of $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$, but it does not cover CQs.

To our knowledge, CQ answering for Horn-$\mathcal{SHIQ}$ and beyond has not been implemented before. Algorithms for full $\mathcal{SHIQ}$ were first given in (Glimm et al. 2008, Calvanese et al. 2007), for Horn-$\mathcal{SHIQ}$ in (Eiter et al. 2008b), and for Horn-$\mathcal{SHOIQ}$ in (Ortiz et al. 2011). They are of theoretical interest (to prove complexity results) but not suited for implementation due to prohibitive sources of complexity.

Outlook We presented a rewriting-based algorithm for CQ answering over Horn-$\mathcal{SHIQ}$ ontologies, possibly extended with weakly DL-safe rules. Our results can be generalized to the case without UNA using standard techniques: the current algorithm works correctly for TBoxes without axioms of the form $A \sqsubseteq\ \leqslant 1\, r.B$, while for a full Horn-$\mathcal{SHIQ}$ TBox $\mathcal{T}$ an axiomatization of the equality predicate can be used to replace the constraints in $\mathsf{cr}(\mathcal{T})$.

Our prototype implementation shows potential for practical applications, and further optimizations will improve it. Future versions of CLIPPER will support transitive roles in queries, and KBs with sets of weakly DL-safe rules.

An interesting application of our method is reasoning with *DL-programs*, which loosely couple rules and ontologies (Eiter et al. 2008a). The *inline evaluation* framework translates ontologies into rules to avoid the overhead caused by using two interacting reasoners (Heymans, Eiter, and Xiao 2010; Eiter et al. 2012a). The techniques in this paper can be faithfully integrated into it to efficiently evaluate DL-programs involving Horn-$\mathcal{SHIQ}$ ontologies.

Acknowledgements. This work was supported by the Austrian Science Fund (FWF) grants P20840 and T515.

References

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P., eds. 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, second edition.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2009. Datalog±: a unified approach to ontologies and integrity constraints. In *ICDT '09*, 14–30. New York, NY, USA: ACM.

Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning* 39(3):385–429.

Calvanese, D.; Eiter, T.; and Ortiz, M. 2007. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *AAAI'07*, 391–396.

Chortaras, A.; Trivela, D.; and Stamou, G. 2011. Optimized query rewriting for OWL 2 QL. In *CADE'11*, 192–206. Berlin, Heidelberg: Springer-Verlag.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys* 33(3):374–425.

Eiter, T.; Ianni, G.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2008a. Combining answer set programming with description logics for the Semantic Web. *Artificial Intelligence* 172(12-13):1495–1539.

Eiter, T.; Gottlob, G.; Ortiz, M.; and Simkus, M. 2008b. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *JELIA'08*, 166–179. Springer.

Eiter, T.; Lutz, C.; Ortiz, M.; and Simkus, M. 2009. Query answering in description logics with transitive roles. In *IJCAI'09*, 759–764.

Eiter, T.; Krennwallner, T.; Schneider, P.; and Xiao, G. 2012a. Uniform Evaluation of Nonmonotonic DL-Programs. In *FoIKS'12*, 1–22. Springer.

Eiter, T.; Ortiz, M.; Šimkus, M.; Tran, T.; and Xiao, G. 2012b. Query rewriting for Horn-SHIQ plus rules. Technical Report INFSYS RR-1843-12-04, TU Vienna. http://www.kr.tuwien.ac.at/research/reports/rr1204.pdf.

Eiter, T.; Ortiz, M.; and Simkus, M. 2012. Conjunctive query answering in the description logic SH using knots. *J. Comput. Syst. Sci.* 78(1):47–85.

Gebser, M.; Kaufmann, B.; Kaminski, R.; Ostrowski, M.; Schaub, T.; and Schneider, M. 2011. Potassco: The potsdam answer set solving collection. *AI Commun.* 24(2):107–124.

Glimm, B.; Lutz, C.; Horrocks, I.; and Sattler, U. 2008. Conjunctive query answering for the description logic SHIQ. *J. Artif. Intell. Res. (JAIR)* 31:157–204.

Glimm, B.; Horrocks, I.; and Sattler, U. 2006. Conjunctive query answering for description logics with transitive roles. In *DL'06*. CEUR-WS.org.

Gottlob, G.; Orsi, G.; and Pieris, A. 2011. Ontological queries: Rewriting and optimization. In *ICDE'11*, 2 –13.

Heymans, S.; Eiter, T.; and Xiao, G. 2010. Tractable reasoning with dl-programs over datalog-rewritable description logics. In *ECAI'10*, 35–40. IOS Press.

Horridge, M., and Bechhofer, S. 2011. The OWL API: A java API for OWL ontologies. *Semantic Web* 2(1):11–21.

Hustadt, U.; Motik, B.; and Sattler, U. 2004. A decomposition rule for decision procedures by resolution-based calculi. In *LPAR'04*, 21–35. Springer.

Hustadt, U.; Motik, B.; and Sattler, U. 2007. Reasoning in description logics by a reduction to disjunctive datalog. *J. Autom. Reasoning* 39(3):351–384.

Kazakov, Y. 2009. Consequence-driven reasoning for Horn SHIQ ontologies. In *IJCAI'09*, 2040–2045.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2007. Complexity boundaries for Horn description logics. In *AAAI'07*, 452–457. AAAI Press.

Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; and Scarcello, F. 2006. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.* 7(3):499–562.

Levy, A. Y., and Rousset, M.-C. 1998. Combining Horn rules and description logics in CARIN. *Artif. Intell.* 104:165–209.

Ma, L.; Yang, Y.; Qiu, Z.; Xie, G. T.; Pan, Y.; and Liu, S. 2006. Towards a complete OWL ontology benchmark. In *ESWC'06*, 125–139. Springer.

Maier, D., and Mendelzon, A. 1979. Testing implications of data dependencies. *ACM Transactions on Database Systems* 4:455–469.

Ortiz, M.; Rudolph, S.; and Simkus, M. 2010. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In *KR'10*. AAAI Press.

Ortiz, M.; Rudolph, S.; and Simkus, M. 2011. Query answering in the Horn fragments of the description logics SHOIQ and SROIQ. In *IJCAI'11*, 1039–1044.

Pérez-Urbina, H.; Motik, B.; and Horrocks, I. 2009. A comparison of query rewriting techniques for DL-Lite. In *DL'09*. CEUR-WS.org.

Rosati, R., and Almatelli, A. 2010. Improving query answering over DL-Lite ontologies. In *KR'10*.

Rosati, R. 2006. DL+log: Tight integration of description logics and disjunctive datalog. In *KR'06*, 68–78.

Rosati, R. 2007. On conjunctive query answering in EL. In *DL'07*. CEUR Electronic Workshop Proceedings.

Sattler, U. 2000. Description logics for the representation of aggregated objects. In *ECAI'09*, 239–243. IOS Press.