Query Rewriting for Horn-$SHIQ$ 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 extensively studied for conjunctive query answering in the $DL$-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-$SHIQ$ ontologies, possibly extended with recursive rules under limited recursion as in $DL+\log$. This setting not only subsumes both $DL$-Lite and $\mathcal{EL}$, but also yields an algorithm for answering (limited) recursive queries over Horn-$SHIQ$ 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-$SHIQ$ ontologies and benign downscaling to $DL$-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 $\text{hasHDI}(T, V)$. 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 $\text{isLocatedIn}$ 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 $\text{isLocatedIn}(X, \text{Mexico})$ 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 (Carichi, 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 $DL$-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, $\text{disadvantagedTerritory}(x, y) \leftarrow \text{hasHDI}(x, \text{low}), \text{country}(y), \text{hasCapital}(y, x), \text{hasHDI}(y, \text{high})$ 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 $DL$-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-$SHIQ$ (the disjunction-free fragment of $SHIQ$). This DL extends $\mathcal{DL}$ 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

Copyright © 2012, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
\( \mathcal{L} \) but not in \( DL-Lite \). Axioms (a), (d) and (e) are not expressible in either of them, but they are expressible in Horn-SHIQ. Despite the increase in expressivity, reasoning in Horn-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-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 \( DL+log \) (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 \( DL-Lite \), 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-SHIQ**

As usual, we assume countably infinite sets \( N_C \supset \{ \top, \bot \} \) and \( N_R \) of atomic concepts and role names, respectively. \( N_R \cup \{ \bot \} \) is the set of roles. If \( r \in N_R \), then \( \text{inv}(r) = r^{-} \) and \( \text{inv}(r^{-}) = r \). Concepts are inductively defined as follows: (a) each \( A \in N_C \) is a concept, and (b) if \( C, D \) are concepts and \( r \) is a role, then \( C \sqcap D, C \sqcup D, \neg C, \forall r.C, \exists r.C, \geq r.C \) and \( \leq r.C \), 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 \( T \) is a finite set of GCLs, RIs and transitivity axioms. We let \( \Gamma_T \) denote the reflexive transitive closure of \( \{ (r, s) \mid r \sqsubseteq s \in T \} \) or \( \text{inv}(r) \sqsubseteq \text{inv}(s) \in T \). A role \( s \) is transitive in \( T \) if \( \text{trans}(s) \in T \) or \( \text{trans}(s^{-}) \in T \).

Table 1: An example ontology and queries

<table>
<thead>
<tr>
<th>village (Carichi)</th>
<th>hasHDI (Carichi, low)</th>
<th>(a) trans (isLocatedIn)</th>
<th>(c) country ( \sqsubseteq \exists \text{hasCapital}. \text{capital} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>state (Chihuahua)</td>
<td>hasHDI (Mexico, high)</td>
<td>(b) hasCapital ( \sqsubseteq ) isLocatedIn</td>
<td>(d) country ( \sqsubseteq \text{hasCapital}. \text{city} )</td>
</tr>
<tr>
<td>country (Mexico)</td>
<td>hasHDI (Islamabad, high)</td>
<td>(e) country ( \sqsubseteq 1 \sqsubseteq \text{isLocatedIn}^{-} \cdot \text{capital} )</td>
<td></td>
</tr>
<tr>
<td>capital (Islamabad)</td>
<td>isLocatedIn (Carichi, Chihuahua)</td>
<td>(q1) disadvantagedTerritory ( (x, y) \leftarrow \text{hasHDI}(x, low), \text{isLocatedIn}(x, y), \text{country}(y), \text{hasHDI}(y, high) )</td>
<td></td>
</tr>
<tr>
<td>country (Pakistan)</td>
<td>isLocatedIn (Islamabad, Pakistan)</td>
<td>(q2) hasDevelopedCapital ( (x) \leftarrow \text{country}(x), \text{hasCapital}(x, y), \text{city}(y), \text{hasHDI}(y, high) )</td>
<td></td>
</tr>
</tbody>
</table>

A rule \( \rho \) is an expression of the form

\[
\text{head} \leftarrow p_1(v_1^1), \ldots, p_m(v_m^m) \tag{1}
\]

where \( \text{head} \) is an atom (the head), \( \{ p_1(v_1^1), \ldots, p_m(v_m^m) \} \) are also atoms (the body atoms, denoted \( \text{body}(\rho) \)), and \( \bar{u}, \bar{v}_1, \ldots, \bar{v}_m \) are tuples of variables. The variables in \( \bar{u} \) are distinguished. A KB \( \mathcal{K} \) 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 $K = (T, A, P)$ is given by extending an interpretation $I$ to symbols in $N_0 \cup N_0$. For any $c \in N_0$ and $p \in N_0$ of arity $n$, we have $c^2 \in \Delta^2$ and $p^T \subseteq (\Delta^2)^n$. A match in $I$ for a rule $\rho$ of the form (1) is a mapping from variables in $\rho$ to elements in $\Delta^2$ such that $\pi(i) \in p^T$ for each body atom $p(i)$ of $\rho$. We define:

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

Finally, given a ground atom $p(\vec{c})$, $K \models p(\vec{c})$ if $(\vec{c})^I \in p^T$ for all models $I$ of $K$. We refer to weak DL-safety (Rosati 2006). A KB $K = (T, A, P)$ is weakly DL-safe if each rule $\rho \in P$ satisfies the next condition: every distinguished variable $x$ of $\rho$ occurs in some body atom $p(i)$ of $\rho$ such that $p \in N_0$. We make the Unique Name Assumption (UNA).

A KB $K = (T, A, P)$ is an ontology (for brevity, we use $O = (T, A)$). A conjunctive query (CQ) $q$ over $O$ is a rule of the form (1) such that $h$ does not occur in $O$. By $\text{ans}(I, q)$ we denote the set of all $\vec{c} \in N_0^{[i]}$ such that there is a match $\pi$ for $q$ with $\pi(u) = (\vec{c})^I$. By $\text{ans}(O, q)$ we denote the answer to $q$ over $O$ defined as the set of all $\vec{c} \in N_0^{[i]}$ such that $\vec{c} \in \text{ans}(I, q)$ for every model $I$ of $O$.

Note that, for a KB $K = (\emptyset, A, P)$, $A \cup P$ 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 $A \cup P$ has a unique least (Herbrand) model $M = (A \cup P)$.

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

**Elimination of Transitivity Axioms** It is handy to eliminate transitive axioms from $SHIQ^I$ 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 $T^*$ be the Horn-$ALCHIQ^I$ TBox obtained from a Horn-$SHIQ^I$ TBox $T$ by (i) adding for every $A \subseteq \forall s.B \in T$ and every transitive role $r$ with $r^E \subseteq s$, the axioms $A \sqsubseteq \exists r.B'$, $B' \sqsubseteq \forall r.B$ and $B' \subseteq B$, where $B'$ 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 $\text{ans}$.

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

(a) if $\alpha = p(\vec{t})$ is a body atom in $q$, where $\alpha \in N_0 \cup N_0$, $p$ is a simple role in $T$, then $\pi(i) \in p^T$, and

(b) if $\alpha = s(x, y)$ with $s$ non-simple, then there exist $d_1 \in \Delta^2, \ldots, d_k \in \Delta^2$ and a transitive role $r^T \subseteq s$ such that $\pi(x) = d_1 = \pi(y)$, and $(\pi(i), \pi(i+1)) \in r^T$ for each $1 \leq i < k$.

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

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

---

| M \subseteq S.(N \sqcap S') & N \subseteq A & R^I_e |
|---|---|---|
| M \subseteq S.(N \sqcap N') & N \subseteq A & R^I_e |
| M \subseteq \exists S\sqcap S'.N & S \subseteq r & R^I_e |
| M \subseteq \exists S(N \sqcap S') & N \subseteq A & R^I_e |
| M \subseteq \exists S.(N \sqcap N') & N \subseteq A & R^I_e |
| M \subseteq \exists S.(N \sqcap r).N & A \subseteq \forall r.B & R^I_e |
| M \subseteq \exists S(N \sqcap r).N & A \subseteq \forall r.B & R^I_e |
| M \subseteq \exists (S \sqcap r).N & A \subseteq \forall r.B & R^I_e |
| M \subseteq \exists S(N \sqcap r).N & A \subseteq \forall r.B & R^I_e |
| M \subseteq \exists S(N \sqcap r).N & A \subseteq \forall r.B & R^I_e |
| M \subseteq \exists (S \sqcap r).N & A \subseteq \forall r.B & R^I_e |
| M \subseteq \exists (S \sqcap r).N & A \subseteq \forall r.B & R^I_e |

**Table 2:** Inference rules. $M^0$, $N^0$, (resp., $S^0$) 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 is the canonical model property (Eiter et al. 2008b; Ortiz, Rudolph, and Simkus 2011; Cali, Gottlob, and Lukasiewicz 2009). In particular, for a consistent Horn-$ALCHIQ^I$ ontology $O = (T, A)$, there exists a model $I$ of $O$ that can be homomorphically mapped into any other model $I'$ of $O$. We show that such an $I$ can be built in three steps:

2. Close $A$ under all but existential axioms of $T$.
3. Extend $A$ by “applying” the existential axioms of $T$.

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-$ALCHIQ^I$ ontologies.

**Definition 3.** Given a Horn-$ALCHIQ^I$ TBox $T$, $\Xi(T)$ is the TBox obtained from $T$ 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-$ALCHIQ^I$ TBox $T$, $cr(T)$ is the Datalog program described in Table 3.

**Definition 5.** Let $T$ be a Horn-$ALCHIQ^I$ TBox and $I$ an interpretation. A GCI $M \subseteq \exists S.N$ is applicable at $e \in \Delta^2$ if

(a) $e \in M^I$. 


B(y) ← A(x), r(x, y) for each A ⊆ ∀r.B ∈ T
B(x) ← A1(x), . . . , An(x) for all A1 ∩ . . . ∩ An ⊆ B ∈ Ξ(T)
(r, x, y) ← r1(x, y), . . . , rn(x, y) for all r1 ∩ . . . ∩ rn ⊆ r ∈ T
⊥(x) ← A(x), r(x, y), B(y), y, y ̸= y2
for each A ⊆ 1.r.B ∈ T
Γ ← A(x), A1(x), . . . , An(x), r(x, y), B(y)
for all A1 ∩ . . . ∩ An ⊆ Ξ(r1 ∩ . . . ∩ rn).B1 ∩ . . . ∩ Bn and
A ⊆ 1.r.B of T such that r = r1 and B = B1 for some
i, j with Γ ∈ {B1(y), B2(y), r1(x, y), . . . , rn(x, y)}

Table 3: Completion rules cr(T) for Horn-ALCHIQ\^n

(b) there is no e′ ∈ Δ^2 with (e, e′) ∈ S^2 and e′ ∈ N^2,
(c) there is no axiom M′ ⊑ Ξ T such that e ∈ (M′)^2 and S′ ⊑ S′′ ⊑ N′ ⊑ N′′.

An interpretation I obtained from T by an application of an
applicable axiom M ⊑ Ξ S.N in e ∈ Δ^2 is defined as:
- Δ^2 = Δ^2 ∪ \{d\} with d a new element not present in Δ^2
(we call d a successor of e),
- For each atomic A ∈ NC and o ∈ Δ^2, we have o ∈ A^2 if
(a) o ∈ Δ^2 and o ∈ A^2; or (b) o = d and A ∈ N.
- For each role name r and o, o′ ∈ Δ^2, we have (o, o′) ∈ r^2 if
(a) o, o′ ∈ Δ^2 and (o, o′) ∈ r^2; or (b) (o, o′) = (e, d) and r ∈ S; or (c) (o, o′) = (d, e) and inv(r) ∈ S.

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

We note that chase(\{I, 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 I. The following statement can
be shown similarly as in (Ortiz, Rudolph, and Simkus 2011).

Proposition 2. Let O = (T, A) be a Horn-ALCHIQ\^n ontology.
Then O is consistent if \(A \cup cr(T)\) is consistent.
Moreover, if O is consistent, then
(a) chase(MM(\(A \cup cr(T))\), Ξ(T)) is a model of O,
(b) chase(MM(\(A \cup cr(T))\), Ξ(T)) can be homomorphically
mapped into any model of O.

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

Rewriting Rules and Programs

The following is immediate from Propositions 1 and 2:

Theorem 3. Let O = (\(T, A\)) be a Horn-SHIQ\^n ontology.
Then \(A \cup cr(T^*)\) is consistent iff O is consistent.
Moreover, if O is consistent, then ans(O, q) = ans^\\(T\)(\(I_0, q\)), where
\(I_0 = \text{chase}(\text{MM}(A \cup cr(T^*)), \text{Ξ}(T^*))\).

Computing ans^\(T\)(\(I_0, q\)) is still tricky because \(I_0\) can
be infinite. So we rewrite q into a set Q of CSQs such that
ans^\(T\)(\(I_0, q\)) = \(\bigcup_{q \in Q} \text{ans}^\\(T\)(\(MM(A \cup cr(T^*)), q\)).

The rewriting of q is done in steps with the following intu-
tion. Suppose q has a \(T\)-match \(r\) in \(I_0\). A rewrite step clips
off some variable \(x\) such that \(\pi(x)\) has no descendant in the
image of \(\pi\), 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 \(T\)-match whenever \(q'\) does.

Definition 6. For a rule \(\rho\) and a Horn-SHIQ\^n TBox T,
we write \(\rho \rightarrow T \rho'\) if \(\rho'\) is obtained from \(\rho\) by the following steps:

(S1) Select an arbitrary non-distinctive variable \(x\) in \(\rho\).
(S2) Replace each role atom \(r(x, y)\) in \(\rho\), where \(y\) is arbitrary
and \(s\) is non-simple, either leave \(\alpha\) untouched or replace
it by two atoms \(r(y, u_1), r(u_2, x)\), where \(u_1\) is a
fresh variable and \(r\) is a transitive role with \(r \subseteq T\).
(S4) Form some partitioning of the set \(\{y ∣ r^\\\(+) r\langle\rangle) ∪ \{x\}\) into sets \(V_x\) and \(V_y\), in such a way that
\(x \in V_x\).
(S5) Select some \(M \subseteq \exists S.N \in \text{Ξ}(T^*)\) such that
(a) \(\{r ∣ r(y, x) \in \text{body}(\rho) \wedge y \in V_p\} \subseteq S,
(b) \{A ∣ A(z) \in \text{body}(\rho) \wedge z \in V_x\} \subseteq N,
(c) for each variable \(z \in V_x\) and atom \(r(z, x)\) in
\(\text{body}(\rho)\) there is a transitive \(s \subseteq T\) such that
i. \(\{s, s^\\(+) \} \subseteq S,
ii. there is an axiom \(M' \subseteq \exists S'.N' \in \text{Ξ}(T^*)\) such that
\(M' \subseteq N\) and \(\{s, s^\\(+) \} \subseteq S'\).
(S6) Drop each atom from \(\rho\) containing a variable \(V_x\).
(S7) Rename each \(y \in V_y\) by \(x\).
(S8) Add the atoms \(\{A(x) ∣ A \in M\} \cup \rho\).

We write \(\rho \rightarrow T \rho'\) if \(\rho'\) can be obtained from \(\rho\) by finitely
many rewrite iterations. We let rew\(T\)(\(\rho\)) = \(\{\rho' ∣ \rho \rightarrow T \rho'\}\).

For a set \(P\) of rules, rew\(T\)(\(P\)) = \(\bigcup_{\rho \in P} \text{rew}_T(\rho)\).

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
\(r \subseteq T\) 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
\(\pi(x)\) can be mapped to the parent \(p\) of \(\pi(x)\).
Now we know that for each atom \(r(y, x)\) the ‘neighbor’ variable \(u\) must mapped to
(a) the parent \(p\) of \(\pi(x)\), or (b) in case \(r\) is non-simple,
then \(\pi(x)\). In (S4) we guess a partition of the neighbor
variables into \(V_x\) and \(V_y\), 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 \(V_y\) (and, for technical
reasons, we rename them to \(x\), which does not occur in
\(\rho\) anymore) in (S7), and add to \(p\) atoms for the concepts on
the left hand side of the selected axiom.
Example 2. In our first example, illustrated in Figure 1a, all roles are simple. Let $\rho : q(x_1) \leftarrow C(x_1), B(x_2), r_1(x_1, x_2), r_2(x_3, x_4)$, and assume $A \sqsubseteq \exists(r_1 \cap r_2 \cap r_3). B \in \Xi(T^*)$. In (S1) we select the non-distinguished variable $x_2$. Next, in (S2), we replace $r_2(x_3, x_4)$ 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), $A \sqsubseteq \exists(r_1 \cap r_1 \cap r_2). B$. 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 $\rho(x_1) \in T$. As shown in Figure 1b, 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 $V_p = \{u, x_1, x_4\}$. 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 $T$, $cr(T) = cr(T^*) \cup \{r(x, z)\} \quad r \text{ is transitive in } T\}.$

Now we can show the following:

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

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

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

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 $M \sqsubseteq \exists \Xi N$ in (S5). Suppose $\pi_{q_i}$ is a $\mathcal{T}$-match for $q_i$ in $\mathcal{I}_O$ with $\bar{u} = \pi_{q_i}(\bar{x})$, i.e. $\bar{u} \in \text{ans}^T(\mathcal{I}_O, q_i)$. Let $d = \pi_{q_i}(x)$. Due to step (S8) in the rewriting and the fact that $\mathcal{I}_O$ is a model of $O$, we have $d \in (\exists \Xi N)^{\mathcal{I}_O}$. Then there is $d' \in \Delta^{\mathcal{I}_O}$ such that $(d, d') \in S^{\mathcal{I}_O}$ and $d' \in N^{\mathcal{I}_O}$. One can show that the mapping $\pi_{q_i-1}$ defined as follows is a $\mathcal{T}$-match for $q_{i-1}$ in $\mathcal{I}_O$:

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

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

The query $q_2$ is obtained by selecting for (S1) an $x$ such that $\pi_{q_1}(x) \notin N$ (which exists because $\text{deg}(\pi_{q_1}) > 0$) and there is no variable $y$ other than $q_1$, with $\pi_{q_1}(x)$ a prefix of $\pi_q(x)$ (that is, $x$ is a leaf in the subforest of $\mathcal{I}_O$ induced by the image of $\pi_{q_1}$). Let $d_1 = \pi_{q_1}(x)$, and let $d_2$ be the parent of $d_1$, i.e. $d_2 = d_1 \cdot n$ for some integer $n$. For Step (S3), we choose to rewrite the atoms $\Gamma = \{s(y, x) \in q_1 \mid \pi_{q_1}(y) \neq d_1 \land \pi_{q_1}(y) \neq d_2\}$. By definition of $\mathcal{T}$-match, for each such atom there is a transitive role $r \subseteq \mathcal{I}_O$ such that there is an $r$-path from $\pi_{q_1}(y)$ to $\pi_{q_1}(x)$; this $r$ can be used in the fresh atoms. Let $V_1$ be the set of fresh variables introduced in this step. For Step (S4), let $V_x = \{z \in \text{var}(q_1) \mid \pi_{q_1}(z) = d_2\}$ and $V_y = \{z \in \text{var}(q_1) \mid \pi_{q_1}(z) = d_2\}$. We know from the construction of $\mathcal{I}_O$ that $d_2$ was introduced by an application of an axiom $ax = M \sqsubseteq \exists \Xi N \in \Xi(T^*)$ such that $d_2 \in M^{\mathcal{I}_O}$; we choose this axiom for Step (S5). It is not hard to show that (S5.a-c) hold. Finally, a $\mathcal{T}$-match $\pi_{q_2}$ for $q_2$ in $\mathcal{I}_O$ such that $\bar{u} = \pi_{q_2}(\bar{x})$ and $\text{deg}(\pi_{q_2}) < \text{deg}(\pi_{q_1})$ is obtained from $\pi_{q_1}$ by setting $\pi_{q_2}(z) = \pi_{q_1}(z)$ for all $z \in q_2$ with $z \neq x$, and $\pi_{q_2}(x) = d_2$. $\Box$

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

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

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

To show the “only if” direction, again the only interesting case is where \( \text{cr}(T) \cup \text{rew}_T(P) \cup A \models \alpha \), and show that \( I = \text{chase}(A', \Xi(T')) \) is a model of \((T, A, P)\) such that \( I \not\models \alpha \) for all \( \alpha \) such that \( \text{cr}(T) \cup \text{rew}_T(P) \cup A \not\models \alpha \).

This reduction yields a worst-case optimal algorithm.

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

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

We recall that \( \Xi(T') \) can be computed in exponential time in size of \( T \) and is independent from \( A \). The program rewriting \( \text{rew}_T(P) \) is finite and computable in time exponential in the size of \( T \) and \( P \); rules in \( \text{rew}_T(\rho) \), where \( \rho \in P \), use only relation names and variables that occur in \( \rho \) and \( T \) (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 \( T \) and \( P \), and thus the number of rules in \( \text{rew}_T(P) \) is at most exponential in the size of \( T \) and \( P \). The size of \( \text{rew}_T(P) \) is constant when data complexity is considered. Furthermore, the grounding of \( \text{cr}(T) \cup \text{rew}_T(P) \cup A \) is exponential in the size of \( K \), but polynomial for fixed \( T \) and \( P \). 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 \( \Xi(T') \) and \( \text{rew}_T(P) \) 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-SHIQ ontology \( \mathcal{O} = (T, A) \), which works as follows: (1) we eliminate transitivity axioms in \( T \) to get \( T' \); (2) we saturate \( T' \) into

| \( \mathcal{O} \) | \( \text{Q0(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q1(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q2(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q3(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q4(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q5(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q6(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |

**Table 4: Additional queries in rewriting evaluation**

| \( \mathcal{O} \) | \( \text{Q7(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q8(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q9(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q10(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |
| \( \mathcal{O} \) | \( \text{Q11(x, z)} \rightarrow \text{Device(x, z)}, \text{Address(x, y, z)} \), \( \text{ReadingDevice(y)} \) |

**Table 5: Downscaling query rewriting evaluation**

\( \Xi(T') \) using the calculus in Table 2; (3) we obtain \( \text{rew}_T(\rho) \) by exhaustively applying to \( \rho \) the rewriting step in Definition 6 using the axioms \( \Xi(T') \); (4) we put together \( A \), the completion rules \( \text{cr}(T) \), and \( \text{rew}_T(P) \) into a Datalog program \( P \); and (5) we evaluate the program \( P \).

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-SHIQ (under the standard semantics of first-order logic), and in expressiveness subsumes similar DL-Lite and 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 \( \mathcal{O} = (T, A) \) and a query \( q \) in SPARQL syntax as input. Initial preprocessing involves normalization of \( \mathcal{O} \) and checking that \( T \) 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), Chant(z), isHeadOf(z, x), like(z, y)
Q7(x, y) ← Postdoc(x), hasAlumnus(y, x)
Q8(x, y) ← GraduateCourse(x, y), isTaughtBy(y, z), isHeadOf(x, z)
Q9(x) ← PeopleWithManyHobbies(x), isMemberOf(x, y)
Q10(x) ← SportsLover(x), isHeadOf(x, y), ResearchGroup(y)

Table 6: Queries over Horn-SHIQ ontology

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

Table 7: Experiment with UOBM Horn-SHIQ ontology

1. Downscaling test. We compared CLIPPER with other query rewriting systems for 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 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-SHIQ. To test CLIPPER on a full Horn-SHIQ ontology, we modified the UOBM ontology (Ma et al. 2006), which is in SHOIN(D), by dropping or strengthening (in case of disjunctions) non-Horn-SHIQ TBox axioms; the final ontology has 171 TBox axioms. We used ABoxes A1, 1 ≤ i ≤ 4, with 20k, 80k, 140k and 200k assertions. The test queries in Table 6 were tailored to require reasoning with Horn-SHIQ constructs unavailable in DL-Lite and 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 A1/ A2/ A3/ A4). 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 DL-Lite, many query rewriting techniques have been developed and implemented, e.g. (Perez-Urbina et al. 2009, Rosati and Almatelli 2010, short by et al. 2011, Gottlob et al. 2011), usually aiming at an optimized rewriting size. Some of them also go beyond DL-Lite; e.g., Perez-Urbina et al. cover EL, while Gottlob et al. consider Datalog. 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 DL-Lite and a (recursive) Datalog program for DLs of the 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 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 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 SHOIQ and SROIQ, but it does not cover CQs.

To our knowledge, CQ answering for Horn-SHIQ and beyond has not been implemented before. Algorithms for full SHIQ were first given in (Glimm et al. 2008, Calvanese et al. 2007), for Horn-SHIQ in (Eiter et al. 2008b), and for Horn-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-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 A1 ≤ r.B, while for a full Horn-SHIQ TBox T an axiomatization of the equality predicate can be used to replace the constraints in cr(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-SHIQ ontologies.

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

References


Eiter, T.; Gottlob, G.; Ortiz, M.; and Simkus, M. 2008b. Query answering in the description logic Horn-SHIQ. In JELIA’08, 166–179. Springer.


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.


