Towards Practical Query Answering for Horn-$\mathcal{SHIQ}$

Thomas Eiter$^1$ Magdalena Ortiz$^1$ Mantas Šimkus$^1$
Trung-Kien Tran $^2$ Guohui Xiao$^1$

$^1$Institute of Information Systems, Vienna University of Technology
$^2$STARLab, Vrije Universiteit Brussel

25th International Workshop on Description Logics (DL 2012)
Query Answering in DLs

Ontology Based Data Access is a key application of DLs. Hence, query answering in DLs a crucial problem.

Evaluate a conjunctive query $q$ over an ABox $A$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$.
Query Answering in DLs

Ontology Based Data Access is a key application of DLs. Hence, query answering in DLs is a crucial problem.

Evaluate a conjunctive query $q$ over an ABox $A$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$.

$$\text{hasDevelopedCapital}(x) \leftarrow \text{country}(x), \text{hasCapital}(x, y), \text{city}(y), \text{hasHDI}(y, \text{high})$$

$$A$$

- country($\text{Brazil}$)
- capital($\text{Brasilia}$)
- isLocatedIn($\text{Brasilia}, \text{RegiãoCentroOeste}$)
- isLocatedIn($\text{RegiãoCentroOeste}, \text{Brazil}$)
- hasHDI($\text{Brasilia}, \text{high}$)
Ontology Based Data Access is a key application of DLs. Hence, query answering in DLs a crucial problem.

Evaluate a conjunctive query $q$ over an ABox $A$, taking into account the constraints expressed by a DL TBox $T$.

$$\text{hasDevelopedCapital}(x) \leftarrow \text{country}(x), \text{hasCapital}(x, y), \text{city}(y), \text{hasHDI}(y, \text{high})$$

$$\text{country}(\text{Brazil})$$

$$\text{hasHDI}(\text{Brasilia}, \text{high})$$

$A$

- $\text{country}(\text{Brazil})$
- $\text{capital}(\text{Brasilia})$
- $\text{isLocatedIn}(\text{Brasilia}, \text{RegiãoCentroOeste})$
- $\text{isLocatedIn}(\text{RegiãoCentroOeste}, \text{Brazil})$
- $\text{hasHDI}(\text{Brasilia}, \text{high})$
Query Answering in DLs

Ontology Based Data Access is a key application of DLs
Hence, query answering in DLs a crucial problem

Evaluate a conjunctive query $q$ over an ABox $A$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$

\begin{align*}
\text{hasDevelopedCapital}(x) &\leftarrow \text{country}(x), \text{hasCapital}(x, y), \text{city}(y), \text{hasHDI}(y, \text{high}) \\
\text{country}(\text{Brazil}) &\quad \text{hasHDI}(\text{Brasilia, high}) \\
\text{hasCapital}(\text{Brazil, Brasilia}), \text{city}(\text{Brasilia}),
\end{align*}

\begin{align*}
\mathcal{A} &\quad \mathcal{T} \\
\text{country}(\text{Brazil}) &\quad \text{trans}(\text{isLocatedIn}) \\
\text{capital}(\text{Brasilia}) &\quad \text{country} \sqsubseteq \exists \text{hasCapital}.\text{capital} \\
\text{isLocatedIn}(\text{Brasilia, RegiãoCentroOeste}) &\quad \text{hasCapital} \sqsubseteq \text{isLocatedIn} \\
\text{isLocatedIn}(\text{RegiãoCentroOeste, Brazil}) &\quad \text{country} \sqsubseteq \leq 1 \text{isLocatedIn}^{-}.\text{capital} \\
\text{hasHDI}(\text{Brasilia, high}) &\quad \text{country} \sqsubseteq \forall \text{hasCapital}.\text{city}
\end{align*}
1. Motivation

Query Answering in DLs

Ontology Based Data Access is a key application of DLs. Hence, query answering in DLs is a crucial problem.

Evaluate a conjunctive query $q$ over an ABox $A$, taking into account the constraints expressed by a DL TBox $\mathcal{T}$.

$$\text{hasDevelopedCapital}(x) \leftarrow \text{country}(x), \text{hasCapital}(x, y), \text{city}(y), \text{hasHDI}(y, \text{high})$$

$$\text{hasDevelopedCapital}($Brazil$) \leftarrow \text{country}($Brazil$)$$

$$\text{hasHDI}($Brasilia$, \text{high})$$

$$\text{hasCapital}($Brazil$, $Brasilia$), \text{city}($Brasilia$),$$

### $A$

- country($Brazil$)
- capital($Brasilia$)
- isLocatedIn($Brasilia$, RegiãoCentroOeste)
- isLocatedIn(RegiãoCentroOeste, Brazil)
- hasHDI($Brasilia$, high)

### $\mathcal{T}$

- trans(isLocatedIn)
- country $\sqsubseteq \exists$ hasCapital.capital
- hasCapital $\sqsubseteq$ isLocatedIn
- country $\sqsubseteq \leq 1$ isLocatedIn$^\neg$.capital
- country $\sqsubseteq \forall$ hasCapital.city
State of the art

For lightweight DLs, successful query answering using database technologies

For DL-Lite

- Query rewriting compiles $q$ and $\mathcal{T}$ into a UCQ/FO query $q^\mathcal{T}$
- $q^\mathcal{T}$ can be evaluated over $\mathcal{A}$ only with off-the-shelf RDBMSs
- Many papers on better and shorter rewritings

For $\mathcal{EL}$

- Query rewriting into Datalog (e.g., Requiem), no FO rewritability
- Alternative: the combined approach
  - TBox partially materialized in $\mathcal{A}$ (polynomial expansion)
  - $q$ rewritten into a FO query over the expanded data
  - evaluation possible with off-the-shelf RDBMSs
State of the art (cont’d)

For more expressive DLs

- (full) CQ answering not supported by reasoners
- algorithms don’t seem implementable
- many bad complexity results
  - $ALCI$ and $SH$ are 2-ExpTime-hard in combined complexity
  - already $AL$ intractable in data complexity
State of the art (cont’d)

For more expressive DLs

- (full) CQ answering not supported by reasoners
- algorithms don’t seem implementable
- many bad complexity results
  - $ALCI$ and $SH$ are 2-ExpTime-hard in combined complexity
  - already $AL$ intractable in data complexity ➡ bad news!
1. Motivation

State of the art (cont’d)

For more expressive DLs

- (full) CQ answering not supported by reasoners
- algorithms don’t seem implementable
- many bad complexity results
  - \( ALCI \) and \( SH \) are 2-ExpTime-hard in combined complexity
  - already \( AL \) intractable in data complexity \( \Rightarrow \) bad news!

- Is query answering beyond \( DL\text{-Lite} \) and \( EL \) practicable?

- Can we realize it using existing efficient technologies? (RDBMS, Datalog engines, etc.)
Query Answering in Horn-\textit{SHIQ}

Horn fragments of DLs like \textit{SHIQ} seem promising candidates

- Horn-\textit{SHIQ} is \textit{tractable in data complexity} (PTime-complete)
- The combined complexity is not higher than for standard reasoning
  \[\text{ExpTime-complete}\]
- It has useful features not present in \textit{EL} and \textit{DL-Lite}

\[
\begin{align*}
\text{trans}(\text{isLocatedIn}) & \quad \text{country} \sqsubseteq \forall \text{hasCapital}.\text{city} \\
\text{country} \sqsubseteq & \quad \leq 1 \text{isLocatedIn}^- \text{.capital}
\end{align*}
\]
Query Answering in Horn-$SHIQ$

Horn fragments of DLs like $SHIQ$ seem promising candidates

- Horn-$SHIQ$ is **tractable in data complexity** (PTime-complete)
- The combined complexity is not higher than for standard reasoning
  - ExpTime-complete
- It has useful features not present in $EL$ and $DL$-Lite

\[
\text{trans}(\text{isLocatedIn}) \quad \text{country} \sqsubseteq \forall \text{hasCapital} \cdot \text{city} \quad \text{country} \sqsubseteq \leq 1 \text{isLocatedIn}^\ast \cdot \text{capital}
\]

- These features make the problem significantly more complex
- Incorporating them into existing techniques is not trivial
Query Answering for Horn-\textit{SHIQ}

Our Contribution

We present a novel query rewriting algorithm for Horn-\textit{SHIQ} and a prototype implementation that shows promising results.

- We rewrite $q$ into a UCQ $\text{rew}_T(q)$ (depends on the TBox $T$)
- The (non-existential) axioms of $T$ are rewritten into \textit{Datalog rules} $\text{cr}(T)$
- Answering $q$ over $(T, A)$ amounts to evaluating the Datalog program $A \cup \text{cr}(T) \cup \text{rew}_T(q)$
  - We can also evaluate $\text{rew}_T(q)$ over the completion of $A$ (with no additional unnamed objects)
- $\text{rew}_T(q)$ can be exponential, but has manageable size for real queries and ontologies
The rewriting algorithm

Main idea:

- Eliminate query variables that can be matched at unnamed objects
  - Query matches have tree-shaped parts
  - We clip off the variables $x$ that can be leaves
  - Replace them by constraints $D(y)$ on their parent variables $y$
  - The added atoms $D(y)$ ensure the existence of a match for $x$

- In the resulting queries all variables are matched to named objects
One step of query rewriting

\[ q(x_1) \leftarrow r(x_1, x_2), r(x_1, x_4), r(x_2, x_3), s(x_3, x_4), A(x_1), B(x_4), B'(x_2), C(x_3) \]
One step of query rewriting

1. Select the non-distinguished variable $x_3$
One step of query rewriting

1. Select the non-distinguished variable $x_3$
2. Ensure that $x_3$ has only incoming edges
   - replace $r(x, y)$ by $r^-(y, x)$ as needed
One step of query rewriting

1. Select the non-distinguished variable $x_3$
2. Ensure that $x_3$ has only incoming edges
   - replace $r(x, y)$ by $r^-(y, x)$ as needed
3. Merge the predecessors
   - if $x_3$ is a leaf of a tree, they must be mapped together
2. Our Contribution

One step of query rewriting

1. Select the non-distinguished variable $x_3$
2. Ensure that $x_3$ has only incoming edges
   - replace $r(x, y)$ by $r^-(y, x)$ as needed
3. Merge the predecessors
   - if $x_3$ is a leaf of a tree, they must be mapped together
4. Find an axiom that enforces an $(r \sqcap s^-)$-child that is $C$
   - fail if $\mathcal{T}$ does not imply such an axiom
One step of query rewriting

1. Select the non-distinguished variable $x_3$

2. Ensure that $x_3$ has only incoming edges
   - replace $r(x, y)$ by $r^-(y, x)$ as needed

3. Merge the predecessors
   - if $x_3$ is a leaf of a tree, they must be mapped together

4. Find an axiom that enforces an $(r \cap s^-)$-child that is $C$
   - fail if $\mathcal{T}$ does not imply such an axiom

5. Drop $x_3$ and add $D(x_2)$
2. Our Contribution

One step of query rewriting

1. Select the non-distinguished variable $x_3$

2. Ensure that $x_3$ has only incoming edges
   - replace $r(x, y)$ by $r^-(y, x)$ as needed

3. Merge the predecessors
   - if $x_3$ is a leaf of a tree, they must be mapped together

4. Find an axiom that enforces an $(r \cap s^-)$-child that is $C$
   - fail if $\mathcal{T}$ does not imply such an axiom

5. Drop $x_3$ and add $D(x_2)$
Another step of query rewriting

The query using the axiom is rewritten to

\[ A \sqsubseteq \exists r_2. A_3 \]

To handle transitive roles in the query:
- introduce a new variable between the eliminated variable and some of its predecessors
- eliminate sets of variables (due to variables connected in the query that may be mapped to the same object)
Another step of query rewriting

The query using the axiom is rewritten to

\[ A_1 x_1 \quad A_2 x_2 \quad A_3 x_3 \quad A_4 x_4 \]

\[ r_1 \quad r_2 \quad r_3 \]

\[ A_1 \sqsubseteq \exists r_2.A_3 \]

To handle **transitive roles** in the query:

- introduce a new variable between the eliminated variable and some of its predecessors
- eliminate **sets of variables** (due to variables connected in the query that may be mapped to the same object)
TBox Saturation

- We compute in advance a set $\Xi(T)$ of relevant axioms
  - Tailored resolution calculus for Horn-$\mathcal{ALCHIQ}$
  - Adaptation of existing consequence driven procedures for satisfiability

**Example Rules**

\[
\begin{align*}
M \sqsubseteq \exists S.(N \cap N') & \quad N \sqsubseteq A \\ R^c_c & \\
M \sqsubseteq \exists S.(N \cap N' \cap A) & \\
\end{align*}
\]

\[
\begin{align*}
M \sqsubseteq \exists(S \cap \text{inv}(r)).(N \cap A) & \quad A \sqsubseteq \forall r.B \\
M \sqsubseteq B & \quad R^c_v
\end{align*}
\]

- The rewriting step simply searches for an axiom in $\Xi(T)$
2. Our Contribution

The Query Answering Algorithm

**Algorithm 1**: Answering CQs via Query Rewriting

| Input: normal Horn-\textit{SHIQ} KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, Conjunctive Query $q$ |
| Output: query answers |

$$\Xi(\mathcal{T}) \leftarrow \text{Saturate}(\mathcal{T});$$

$$\text{rew}_\mathcal{T}(q) \leftarrow \text{Rewrite}(q, \Xi(\mathcal{T}));$$

$$\text{cr}(\mathcal{T}) \leftarrow \text{CompletionRules}(\mathcal{T});$$

$$\mathcal{P} \leftarrow \mathcal{A} \cup \text{cr}(\mathcal{T}) \cup \text{rew}_\mathcal{T}(q);$$

$$\text{ans} \leftarrow \{\vec{u} \mid q(\vec{u}) \in \text{MinimalModel}(\mathcal{P})\};$$

\[\triangleright \text{ call Datalog reasoner}\]

The completion rules $\text{cr}(\mathcal{T})$ are straightforward, e.g.

$$B(y) \leftarrow A(x), r(x, y)$$  \hspace{2cm} \text{for each } A \sqsubseteq \forall r.B \in \mathcal{T}$$

$$r(x, y) \leftarrow r_1(x, y), \ldots, r_n(x, y)$$  \hspace{2cm} \text{for each } r_1 \sqcap \ldots \sqcap r_n \sqsubseteq r \in \mathcal{T}$$
We have implemented a prototype system called **Clipper**
(http://www.kr.tuwien.ac.at/research/systems/clipper)

- **Ontology Preprocessing**
- **Saturation**
- **Query Rewriting**
- **Datalog Translation**
- **Datalog Engine** (DLV, Clingo)

1. CQ with simple roles only
2. Normalized TBox
3. Existential axioms in $\Xi(T)$
4. Axioms for ABox completion
5. $P \leftarrow A \cup cr(T) \cup \text{rew}_T(q)$
We carried out the following experiments:

- **Downscaling test**
  - We compared Clipper with state of the art query rewriting systems over $\mathcal{DL}$-Lite ontologies
We carried out the following experiments:

- **Downscaling test**
  - We compared Clipper with state of the art query rewriting systems over \textit{DL-Lite} ontologies

- **Full Horn-\textit{SHIQ} test**
  - We tested Clipper over a full Horn-\textit{SHIQ} ontology
3. Experiments

**Downscaling test**

- We used the ontologies of the Requiem test suite: Adolena (A), Stock Exchange (S), Vicodi (V) and University (U).
- We added 2 queries to the 5 existing ones for each ontology.
- We compared Clipper with Presto and Requiem.
- We compared rewriting times and size of the rewritten queries.
- For U, we also evaluated the resulting Datalog program over 4 ABoxes, 67k to 320k.
## 3. Experiments

### Downscaling test (2)

<table>
<thead>
<tr>
<th>Q</th>
<th>Query</th>
<th># Rules/CQs</th>
<th>Rewriting time, ms (avg. eval. time, DLV)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RequiemG</td>
<td>Presto</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>27</td>
<td>53</td>
<td>42</td>
</tr>
<tr>
<td>Q2</td>
<td>50</td>
<td>32</td>
<td>31</td>
</tr>
<tr>
<td>Q3</td>
<td>104</td>
<td>32</td>
<td>31</td>
</tr>
<tr>
<td>Q4</td>
<td>224</td>
<td>43</td>
<td>36</td>
</tr>
<tr>
<td>Q5</td>
<td>624</td>
<td>37</td>
<td>36</td>
</tr>
<tr>
<td>Q6</td>
<td>364</td>
<td>35</td>
<td>30</td>
</tr>
<tr>
<td>Q7</td>
<td>2548</td>
<td>43</td>
<td>32</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>6</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Q2</td>
<td>2</td>
<td>3</td>
<td>22</td>
</tr>
<tr>
<td>Q3</td>
<td>4</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>Q4</td>
<td>4</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>Q5</td>
<td>8</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Q6</td>
<td>4</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Q7</td>
<td>8</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>U</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Q2</td>
<td>1</td>
<td>2</td>
<td>45</td>
</tr>
<tr>
<td>Q3</td>
<td>4</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>Q4</td>
<td>2</td>
<td>56</td>
<td>63</td>
</tr>
<tr>
<td>Q5</td>
<td>10</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Q6</td>
<td>10</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>Q7</td>
<td>960</td>
<td>24</td>
<td>19</td>
</tr>
</tbody>
</table>
Full Horn-$SHIQCQ$ test

- We modified the $SHOIN(D)$ ontology UOBM
- We dropped or strengthened (in case of disjunctions) non-Horn-$SHIQCQ$ TBox axioms
- The final ontology has 196 TBox axioms
- We used ABoxes with 20k, 80k, 140k and 200k assertions.
- 10 small, hand-tailored queries, with an average of 5 atoms
### Full Horn-$\mathcal{SHIQ}$ test (2)

<table>
<thead>
<tr>
<th>Query</th>
<th># Rules</th>
<th>Rewriting Time (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>
Conclusions

- We have proposed a query rewriting technique for Horn-$SHIQ$
- It allows us to reduce CQs answering over Horn-$SHIQ$ to evaluating a Datalog program over the ABox
Conclusions

- We have proposed a query rewriting technique for Horn-\(SHIQ\).
- It allows us to reduce CQs answering over Horn-\(SHIQ\) to evaluating a Datalog program over the ABox.
- The prototype system Clipper shows promising results.
Conclusions

- We have proposed a query rewriting technique for Horn-SHIQ.
- It allows us to reduce CQs answering over Horn-SHIQ to evaluating a Datalog program over the ABox.
- The prototype system Clipper shows promising results.
  - Version with transitive roles may be available soon!
Conclusions

- We have proposed a query rewriting technique for Horn-$SHIQ$
- It allows us to reduce CQs answering over Horn-$SHIQ$ to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results
  ► Version with transitive roles may be available soon!
- Possible extensions:
  - weakly DL-safe rules (algorithm developed, implementation pending)
  - other DLs: regular $EL^{++}$, Horn-$SRIQ$; datatypes
  - other query languages: regular path queries
Conclusions

- We have proposed a query rewriting technique for Horn-\(SHIQ\)
- It allows us to reduce CQs answering over Horn-\(SHIQ\) to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results
  - Version with transitive roles may be available soon!
- Possible extensions:
  - weakly DL-safe rules (algorithm developed, implementation pending)
  - other DLs: regular \(\mathcal{EL}^{++}\), Horn-\(SRIQ\); datatypes
  - other query languages: regular path queries
- Far more testing needed
Conclusions

- We have proposed a query rewriting technique for Horn-\(SHIQ\)
- It allows us to reduce CQs answering over Horn-\(SHIQ\) to evaluating a Datalog program over the ABox
- The prototype system Clipper shows promising results
  - Version with transitive roles may be available soon!
- Possible extensions:
  - weakly DL-safe rules (algorithm developed, implementation pending)
  - other DLs: regular \(\mathcal{EL}^{++}\), Horn-\(SRIQ\); datatypes
  - other query languages: regular path queries
- Far more testing needed
  - but lack of realistic test cases is a big issue!