PRACTICAL ASPECTS OF QUERY REWRITING FOR OWL 2

Héctor Pérez-Urbina, Ian Horrocks, and Boris Motik

Oxford University
Computing Laboratory

OWL: Experiences and Directions Workshop
October 2009
DATA ACCESS WITH OWL 2 QL

\( T = \{ Q(x) \leftarrow \text{teaches}(x, y), Q(x) \leftarrow \text{Teacher}(x), Q(x) \leftarrow \text{Professor}(x), Q(x) \leftarrow \text{hasTutor}(y, x) \} \)

\( \text{sql}(Q_T) = \text{SELECT name FROM PROFESSOR UNION SELECT Tutor FROM Student} \)
DATA ACCESS WITH OWL 2 QL

\[ Q(x) \leftarrow \text{teaches}(x, y) \]

Teacher \sqsubseteq \exists \text{teaches}
Professor \sqsubseteq \text{Teacher}
\exists \text{hasTutor} \sqsubseteq \text{Professor}

PROFESSOR(name, office, phone)
STUDENT(name, major, tutor)
DATA ACCESS WITH OWL 2 QL

$Q(x) \leftarrow teaches(x, y)$

Teacher $\sqsubseteq \exists.teaches$
Professor $\sqsubseteq$ Teacher
$\exists.hasTutor^{-}$ $\sqsubseteq$ Professor

PROFESSOR($name, office, phone$)
STUDENT($name, major, tutor$)

Professor $\mapsto$ SELECT name
FROM PROFESSOR

hasTutor $\mapsto$ SELECT name,tutor
FROM STUDENT
**DATA ACCESS WITH OWL 2 QL**

\[ Q(x) \leftarrow \text{teaches}(x, y) \]

Teacher \[ \sqsubseteq \exists \text{teaches} \]
Professor \[ \sqsubseteq \text{Teacher} \]
\[ \exists \text{hasTutor}^- \sqsubseteq \text{Professor} \]

PROFESSOR(*name*, *office*, *phone*)
STUDENT(*name*, *major*, *tutor*)

Professor \[ \mapsto \begin{align*}
& \text{SELECT name} \\
& \text{FROM PROFESSOR}
\end{align*} \]

hasTutor \[ \mapsto \begin{align*}
& \text{SELECT name, tutor} \\
& \text{FROM STUDENT}
\end{align*} \]

\[ Q \]

\[ \mathcal{T} \]

\[ Q_{\mathcal{T}} \]

\[ \mathcal{M} \]

\[ \text{Transform } Q_{\mathcal{T}} \text{ to SQL} \]

\[ \text{Evaluate } \text{sql}(Q_{\mathcal{T}}) \]

\[ \text{ans}(Q, \langle \mathcal{T}, \text{DB} \rangle) \]

Héctor Pérez-Urbina, Ian Horrocks, and Boris Motik
Data Access with OWL 2 QL

\[ Q(x) \leftarrow \text{teaches}(x, y) \]

Teacher \sqsubseteq \exists \text{teaches}
Professor \sqsubseteq \text{Teacher}
\exists \text{hasTutor}^- \sqsubseteq \text{Professor}

CGLLR rewriting algorithm by Calvanese et al.

\[ Q = \{ Q(x) \leftarrow \text{teaches}(x, y), Q(x) \leftarrow \text{Teacher}(x), Q(x) \leftarrow \text{Professor}(x), Q(x) \leftarrow \text{hasTutor}(y, x) \} \]

\[ \text{sql}(Q_T) = \text{SELECT name FROM PROFESSOR \cup SELECT Tutor FROM Student} \]

Héctor Pérez-Urbina, Ian Horrocks, and Boris Motik
DATA ACCESS WITH OWL 2 QL

\[ Q(x) \leftarrow teaches(x, y) \]

Teacher ⊑ ∃teaches
Professor ⊑ Teacher
∃hasTutor ⊑ Professor

\[ Q_T = \{ Q(x) \leftarrow teaches(x, y), \]
\[ Q(x) \leftarrow Teacher(x), \]
\[ Q(x) \leftarrow Professor(x), \]
\[ Q(x) \leftarrow hasTutor(y, x) \} \]

\[ \text{sql}(Q_T) = \text{SELECT name FROM PROFESSOR UNION SELECT Tutor FROM Student} \]

Héctor Pérez-Urbina, Ian Horrocks, and Boris Motik
Practical Aspects of Query Rewriting for OWL 2
Data Access with OWL 2 QL

\[ Q(x) \leftarrow \text{teaches}(x, y) \]

Teacher ⊑ ∃teaches
Professor ⊑ Teacher
∃hasTutor ⊑ Professor

\[ Q_T = \{ Q(x) \leftarrow \text{teaches}(x, y), \\
    Q(x) \leftarrow \text{Teacher}(x), \\
    Q(x) \leftarrow \text{Professor}(x), \\
    Q(x) \leftarrow \text{hasTutor}(y, x) \} \]

\[ \text{sql}(Q_T) = \text{SELECT name} \\
    \text{FROM PROFESSOR} \\
    \text{UNION} \\
    \text{SELECT Tutor} \\
    \text{FROM Student} \]

Héctor Pérez-Urbina, Ian Horrocks, and Boris Motik
Practical Aspects of Query Rewriting for OWL 2
SO, ARE WE DONE HERE?
SO, ARE WE DONE HERE?

- $Q_T$ is worst-case exponential w.r.t. $Q$ and $T$
  - Costly to compute
  - Costly (or impossible) to evaluate

Applications may require constructs that go beyond QL.

Student $\sqsubseteq \exists$

\text{hasSupervisor}$\sqsubseteq \text{GraduateStudent}$

Existential

$\exists$

\text{studies.Course}$\sqsubseteq \text{Student}$

OxfordStudent $\sqsubseteq \exists$

\text{studiesAt}.\{OxfordUniversity\}$

RQR (R\text{ESOLUTION - B}ASED \text{QUERY R}\text{EWRITING})

Handles ELHIO $\neg$ (most of OWL 2 EL)

$QT$ might be a datalog query

"Pay-as-you-go" behavior: extends and generalizes CGLLR
So, are we done here?

- $Q_T$ is worst-case exponential w.r.t. $Q$ and $T$
  - Costly to compute
  - Costly (or impossible) to evaluate
- Applications may require constructs that go beyond QL
  - $\text{Student} \sqcap \exists \text{hasSupervisor} \sqsubseteq \text{GraduateStudent}$
  - $\exists \text{studies.Course} \sqsubseteq \text{Student}$
  - $\text{OxfordStudent} \sqsubseteq \exists \text{studiesAt.}\{\text{OxfordUniversity}\}$

RQR (Resolution-Based Query Rewriting)

Handles ELHIO $\neg$ (most of OWL 2 EL)

$QT$ might be a datalog query

"Pay-as-you-go" behavior: extends and generalizes CGLLR
SO, ARE WE DONE HERE?

- $Q_T$ is worst-case exponential w.r.t. $Q$ and $T$
  - Costly to compute
  - Costly (or impossible) to evaluate
- Applications may require constructs that go beyond QL
  - Student $\sqcap \exists$hasSupervisor $\sqsubseteq$ GraduateStudent
  - $\exists$studies.Course $\sqsubseteq$ Student
  - OxfordStudent $\sqsubseteq \exists$studiesAt.\{OxfordUniversity\}

RQR (Resolution-based Query Rewriting)

- Handles $\text{ELHIOT}^-$ (most of OWL 2 EL)
So, are we done here?

- $Q_T$ is worst-case exponential w.r.t. $Q$ and $T$
  - Costly to compute
  - Costly (or impossible) to evaluate

- Applications may require constructs that go beyond QL
  - Student $\sqcap \exists \text{hasSupervisor} \sqsubseteq \text{GraduateStudent}$
  - $\exists \text{studies.Course} \sqsubseteq \text{Student}$
  - OxfordStudent $\sqsubseteq \exists \text{studiesAt.\{}\text{OxfordUniversity}\}$

---

**RQR (Resolution-based Query Rewriting)**

- Handles $ELHIO^-$ (most of OWL 2 EL)
- $Q_T$ might be a datalog query
- “Pay-as-you-go” behavior: extends and generalizes CGLLR
EVALUATION

---

**NUMBER OF INFERENCES**

**Overall**

<table>
<thead>
<tr>
<th>REQUIEM (RQR) C (CGLLR)</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>70,846</td>
</tr>
<tr>
<td></td>
<td>343,813</td>
</tr>
</tbody>
</table>

**Average**

<table>
<thead>
<tr>
<th>REQUIEM (RQR) C (CGLLR)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2,200</td>
</tr>
<tr>
<td></td>
<td>12,066</td>
</tr>
</tbody>
</table>

REQUIEM: 73% smaller, 0% larger, and 27% equal

---

**NUMBER OF QUERIES**

**Overall**

<table>
<thead>
<tr>
<th>REQUIEM (RQR) C (CGLLR)</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10,931</td>
</tr>
<tr>
<td></td>
<td>75,301</td>
</tr>
</tbody>
</table>

**Average**

<table>
<thead>
<tr>
<th>REQUIEM (RQR) C (CGLLR)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>289</td>
</tr>
<tr>
<td></td>
<td>2,682</td>
</tr>
</tbody>
</table>

REQUIEM: 83% smaller, 0% larger, and 17% equal
## Evaluation

<table>
<thead>
<tr>
<th>Number of Inferences ≈ Time</th>
<th>REQUIEM (RQR)</th>
<th>C (CGLLR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>70,846</td>
<td>343,813</td>
</tr>
<tr>
<td>Average</td>
<td>2,200</td>
<td>12,066</td>
</tr>
</tbody>
</table>

- REQUIEM: 73% smaller, 0% larger, and 27% equal
# Evaluation

## Number of Inferences ≈ Time

<table>
<thead>
<tr>
<th></th>
<th>REQUIEM (RQR)</th>
<th>C (CGLLR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>70,846</td>
<td>343,813</td>
</tr>
<tr>
<td>Average</td>
<td>2,200</td>
<td>12,066</td>
</tr>
</tbody>
</table>

- **REQUIEM**: 73% smaller, 0% larger, and 27% equal

## Number of Queries

<table>
<thead>
<tr>
<th></th>
<th>REQUIEM (RQR)</th>
<th>C (CGLLR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>10,931</td>
<td>75,301</td>
</tr>
<tr>
<td>Average</td>
<td>289</td>
<td>2,682</td>
</tr>
</tbody>
</table>

- **REQUIEM**: 83% smaller, 0% larger, and 17% equal
Optimizing the Rewritings

**Query Subsumption**

\[ Q_1 = Q(x) \leftarrow \text{teaches}(x, y) \]
\[ Q_2 = Q(x) \leftarrow \text{teaches}(x, y) \land \text{Student}(y) \]
OPTIMIZING THE REWRITINGS

QUERY SUBSUMPTION

\[ Q_1 = Q(x) \leftarrow \text{teaches}(x, y) \]
\[ Q_2 = Q(x) \leftarrow \text{teaches}(x, y) \land \text{Student}(y) \]

\( Q_1 \) subsumes \( Q_2 \)
**Optimizing the Rewritings**

**Query Subsumption**

\[
Q_1 = Q(x) \leftarrow \text{teaches}(x, y) \\
Q_2 = Q(x) \leftarrow \text{teaches}(x, y) \land \text{Student}(y)
\]

\[Q_1 \text{ subsumes } Q_2\]

- Discard **subsumed** queries a posteriori
- **Significant** reduction in the size of the rewritings
**Optimizing the Rewritings**

**Query Subsumption**

\[ Q_1 = Q(x) \leftarrow \text{teaches}(x, y) \]
\[ Q_2 = Q(x) \leftarrow \text{teaches}(x, y) \land \text{Student}(y) \]

\( Q_1 \) subsumes \( Q_2 \)

- Discard *subsumed* queries a posteriori
- Significant reduction in the size of the rewritings
- On the fly: *forward/backward* subsumption
- Straightforwardly applicable to RQR
CONSIDERATIONS

- $Q_T$ is not guaranteed to be a UCQ: use of deductive database systems
## Considerations

- $Q_T$ is not guaranteed to be a UCQ: use of **deductive** database systems
- Additional **optimizations**
  - Empty EDB predicates pruning
  - Dependency graph pruning
CONSIDERATIONS

- $Q_T$ is not guaranteed to be a UCQ: use of deductive database systems
- Additional optimizations
  - Empty EDB predicates pruning
  - Dependency graph pruning
- Greedy unfolding

Good performance w.r.t. time and size of the rewritings
Greedy unfolding produces UCQs in many cases
GOING BEYOND QL

CONSIDERATIONS

- $Q_T$ is not guaranteed to be a UCQ: use of deductive database systems
- Additional optimizations
  - Empty EDB predicates pruning
  - Dependency graph pruning
- Greedy unfolding

EVALUATION

- Good performance w.r.t. time and size of the rewritings
- Greedy unfolding produces UCQs in many cases
CONCLUSIONS AND FUTURE WORK

CONCLUSIONS

- RQR: significantly smaller rewritings in significantly fewer steps than existing algorithms
- Amenable to various straightforward optimizations
- Use of databases for realistic OWL 2 EL ontologies
- Open source implementation: REQUIEM

1 http://www.comlab.ox.ac.uk/projects/requiem/
**Conclusions**

- **RQR:** significantly smaller rewritings in significantly fewer steps than existing algorithms
- Amenable to various straightforward optimizations
- Use of databases for realistic OWL 2 EL ontologies
- **Open source** implementation: REQUIEM\(^1\)

**Future Work**

- Evaluate REQUIEM with databases

---

\(^1\)[http://www.comlab.ox.ac.uk/projects/requiem/]