Perfect Rewriting for Ontology Based Query Answering over Spatial Databases

Ralf Moeller
Hamburg University of Technology

Joint work with
Oezguer Oezcep, Rolf Gruetter and Aleksandar Gudov
Agenda

1. Ontology Based Query Answering (OBQA)

2. GDL-Lite-8 = DL-Lite with extensions to cope with certain aspects of spatial data (RCC8)

3. Query rewriting algorithm for GDL-Lite-8

4. Insights from an implementation and its evaluation
Query answering w.r.t. ontologies

TBox:
MALE ⊆ PERSON
MALE ⊆ ¬FEMALE
∃hasFather ≠ ⊆ MALE
∃hasMother ≠ ⊆ FEMALE

FEMALE ⊆ PERSON
PERSON ⊆ ∃hasFather
PERSON ⊆ ∃hasMother

ABox:
MALE(Bob)
MALE(Paul)
FEMALE(Ann)
hasFather(Ann, Paul)
hasMother(Paul, Mary)

input query:
q(x) ← PERSON(x)

answers to query:
{ Bob, Paul, Ann, Mary }

Example adapted from a presentation by Riccardo Rosati
To Some Extent
Solved by DL Systems…

- ALNHF: CLASSIC (90-96)
- SRIQ(D): RacerPro (99-today)
- SROIQ/OWL 2: Pellet (04-today)
Problems and Solutions

• GIS deal with spatial data
  ➢ Extensions to DLs such as DL-RCC8
    (e.g., PelletSpatial, RacerPro, ...)

• GIS deal with big data
  ➢ (Partitioning approaches)
  ➢ Query rewriting (e.g., ontopQuest, Stardog)
Definition (DL-Lite$^\Box_{F,R}$)

\[ P \in RN \text{ (role symb.)}; \ A \in CN \text{ (concept symb.)}; \ a, b \in Const \text{ (constants)}. \]

\[ R \rightarrow P | P^- \quad B \rightarrow A | \exists R \quad C_l \rightarrow B | C_l \cap B \quad C_r \rightarrow B | \neg B \]

TBox*):
\[ C_l \sqsubseteq C_r, \text{ (funct } R), \ R_1 \sqsubseteq R_2 \]

ABox:
\[ A(a), R(a, b) \]

*) Restriction: If \( R \) occurs in a functionality axiom, then \( R \) and \( R^- \) do not occur on the right-hand side of a role inclusion axiom \( R_1 \sqsubseteq R_2 \).
Rewriting: Example

TBox:
MALE ⊆ PERSON
MALE ⊆ ¬FEMALE
∃hasFather ¬ ⊆ MALE
∃hasMother ¬ ⊆ FEMALE

FEMALE ⊆ PERSON
PERSON ⊆ ∃hasFather
PERSON ⊆ ∃hasMother

input query:
q(x) ← PERSON(x)

rewritten query:
q'(x) ← PERSON(x) ∨ FEMALE(x) ∨ MALE(x) ∨ hasFather(y,x) ∨ hasMother(y,x)

Taken from a presentation by Riccardo Rosati
Example (cntd.)

rewritten query:
q'(x) ← PERSON(x) ∨
  FEMALE(x) ∨
  MALE(x) ∨
  hasFather(y,x) ∨
  hasMother(y,x)

ABox:
MALE(Bob)
MALE(Paul)
FEMALE(Ann)
hasFather(Ann, Paul)
hasMother(Paul, Mary)

answers to query:
{ Bob, Paul, Ann, Mary }
Perfect Reformulation Algorithm for DL-Lite (Backward Chaining)

\[
q(x) \leftarrow \text{PERSON}(x)
\]

\[
q(x) \leftarrow \text{MALE}(x) \quad q(x) \leftarrow \text{FEMALE}(x)
\]

\[
q(x) \leftarrow \text{hasFather}(y,x) \quad q(x) \leftarrow \text{hasMother}(y,x)
\]

how to avoid the infinite chase of the ABox?

**CHASE of the query:**
- inclusions are applied “from right to left”
- this chase always terminates
- this chase is computed independently of the ABox

Taken from a presentation by Riccardo Rosati
Further Problems and Solutions

• Existing DBs store values in n-ary tables ( Virtual Abox and mapping of query atoms to SQL in global-as-view style)

• Query rewriting might cause exponential blowup of the original query (disjunctive normal form), which might lead SQL optimizer into combinatorial explosion ( Optimizations, e.g., in Optique, EU-FP7)

• Rewriting for spatial reasoning

[Oezcep&M ISWC-2012]
Example (1)

• Park with lake (Park+Lake)
• Lake should be reachable from the outside (e.g., for easy access from roads)
\[ \mathcal{B}_8 \] [KR-92]

- **DC(x,y)**: disjointness
- **EC(x,y)**: externally connected
- **TPP(x,y)**: tangential proper part
- **TPPi(x,y)**: tangential proper part inverse
- **PO(x,y)**: partial overlap
- **EQ(x,y)**: equal
- **NTPP(x,y)**: non-tangential proper part
- **NTPPi(x,y)**: non-tangential proper part inverse
Relational Representation (1)

Park+Lake(i)

Diagram:}

- Two nodes labeled "loc" and "tpp" connected by a dotted line.
- Two nodes labeled "hasLake" and "i" connected by a solid line.
- "loc" and "hasLake" connect to "i".
- "tpp" connects to "i".
Relational Representation (1)

Park+Lake(i)

hasLake

loc

tpp

loc

DL

RCC8
Restricted combination with paths from thematic to spatial component (Lutz, Miličić 2007)

$\exists \text{hasLake} \circ \text{loc}, \text{loc}.\text{tpp}(i)$

**Definition (GDL-Lite-8)**

$$R \rightarrow P | P^-$$

$$C_l \rightarrow B | C_l \sqcap B$$

**TBox*)**: $C_l \sqsubseteq C_r, (\text{funct loc}), (\text{funct } R), R_1 \sqsubseteq R_2$

**ABox**: $A(a), R(a, b), \text{loc}(a, a^*), r(a^*, b^*) \text{ for } r \in \text{Rel}_{RCC8}$

*) If (funct $R$) $\in \mathcal{T}$, then $R$ and $R^-$ do not occur on the right-hand side of a role inclusion axiom or in a concept of the form $\exists U_1, U_2.r$. 

[Oezcep and Moeller ISWC-2012]
Example (2)

- Park with playing area (Park4Playing)
Relational Representation (2)

DL

RCC8

Park4Playing(i)

i hasPIAr

loc tpp loc
Combined Representation (KB)

- TBox:
  - Park+Lake ⊆ Park
  - Park4Playing ⊆ Park
  - Park+Lake ⊆ \exists \text{hasLake} \circ \text{loc, loc.tpp}
  - Park4Playing ⊆ \exists \text{hasPLAr} \circ \text{loc, loc.tpp}

- ABox contains Park+Lake(i), Park4Playing(i)
Combined Representation (Models)
Combined Representation (Models)
tpp(x,y) and tppl(y,z)

Resulting Models
## Composition Table

<table>
<thead>
<tr>
<th></th>
<th>DC</th>
<th>EC</th>
<th>PO</th>
<th>TPP</th>
<th>NTPP</th>
<th>TPPI</th>
<th>NTPPi</th>
<th>EQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
</tr>
<tr>
<td>EC</td>
<td>DC</td>
<td>PO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DC EC</td>
<td>DC</td>
</tr>
<tr>
<td>PO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PO</td>
<td>PO EC</td>
<td>PO</td>
</tr>
<tr>
<td>TPP</td>
<td>DC</td>
<td>DC EC</td>
<td></td>
<td></td>
<td>TPPi</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTPP</td>
<td>DC</td>
<td>DC</td>
<td></td>
<td></td>
<td>NTPP</td>
<td>NTPP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPPI</td>
<td>DC</td>
<td>PO</td>
<td></td>
<td></td>
<td>TPPI</td>
<td>NTPPi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTPPi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NTPPi</td>
<td>NTPPi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EQ</td>
<td>DC</td>
<td>EC</td>
<td>PO</td>
<td>TPP</td>
<td>NTPP</td>
<td>TPPI</td>
<td>NTPPi</td>
<td>EQ</td>
</tr>
</tbody>
</table>
Combined Representation (Models)

Park+Lake(i)  hasLake  loc  tpp  { dc, ec, po, tpp, tppi, eq }

Park4Playing(i)  hasPIAr  loc  tpp  tppi

Combined Representation (Models)
Application Scenario

• Large repository of assertions modeling “spatial designs” in an engineering company
• Safety conditions for design support
  – Formalization as queries
  – Non-results are counterexamples
Querying for Safe Designs
Querying for Safe Designs

A

B

C

D

Design OK

not OK
Query Language GCQ$^+$

Example query

$Q(x) = \text{Find all parks } x \text{ with lakes such that } x \text{ contains a playing area that is not contained as island in the lake.}$

- Query formalizable in a special query language GCQ$^+$
  - allows atoms of form $C(x)$ where $C$ is a GDL-Lite-8 concept without $\neg$
  - active domain semantics for atoms of the form $R(x, y)$, $r(x^*, y^*)$ etc.

Query formally

$Q(x) = \text{Park}(x) \land \exists \text{hasLake} \circ \text{loc}, \text{hasPlAr} \circ \text{loc}. \mathcal{B}_{RCC8} \setminus \{\text{ntpp}\}(x)$
The Complete Example

**TBox:**

- $Park + Lake \sqsubseteq Park$
- $Park4Playing \sqsubseteq Park$
- $Park + Lake \sqsubseteq \exists \text{hasLake} \circ loc, loc\text{.tpp}$
- $Park4Playing \sqsubseteq \exists \text{hasPIAr} \circ loc, loc\text{.tpp}$

**ABox contains** $Park + Lake(i), Park4Playing(i)$

**Query formally**

$$Q(x) = Park(x) \land \exists \text{hasLake} \circ loc, \text{hasPIAr} \circ loc. B_{RCC8} \setminus \{\text{ntpp}\}(x)$$
Is Design “i” safe?

$\mathcal{B}_{RCC8} \setminus \{ntpp\}$
Is Design “i” safe?

{ dc, ec, po, tpp, tppi, eq }
Design "i" is safe

\{ dc, ec, po, tpp, tppi, eq \} \subseteq B_{RCC8} \setminus \{ntpp\}
QA with a DL System?

• GDL-Lite-8 is decidable [Lutz-Milicic-2007]

• Why not extending the query rewriting idea to GDL-Lite-8?
The Whole Example

TBox:

\begin{align*}
\text{Park} + \text{Lake} & \sqsubseteq \text{Park} \\
\text{Park4Playing} & \sqsubseteq \text{Park} \\
\text{Park} + \text{Lake} & \sqsubseteq \exists \text{hasLake} \circ \text{loc}, \text{loc.tpp} \\
\text{Park4Playing} & \sqsubseteq \exists \text{hasPIAr} \circ \text{loc}, \text{loc.tpp}
\end{align*}

ABox contains \( \text{Park} + \text{Lake}(i), \text{Park4Playing}(i) \)

Query formally

\[ Q(x) = \text{Park}(x) \land \exists \text{hasLake} \circ \text{loc}, \text{hasPIAr} \circ \text{loc}.B_{RCC8} \setminus \{ntpp\}(x) \]
Rewriting the Example Query

\[ \mathcal{B}_{RCC8} \setminus \{\text{ntpp}\} \]
Rewriting the Example Query

$\mathcal{B}_{RCC8 \setminus \{ntpp\}}$
Rewriting the Example Query

\[ B_{RCC8} \setminus \{ntpp\} \]
Rewriting the Example Query

\[ \mathcal{B}_{RCC8} \setminus \{ntpp\} \]
Rewriting the Example Query

{ dc, ec, po, tpp, tppi, eq } ⊆ \( B_{RCC8} \setminus \{ntpp\} \)
What’s in the Repository?

• Only simple tables:
  – Park+Lake(i)
  – Park4Playing(i)

• Spatial query answering without spatial data!

• Standard SQL can do with UCQ $\rightarrow$ SQL mappings (Unfolding)
System Architecture
Adapted Query Rewriting

**Adapted Perfect Rewriting Algorithm**: an extension of the Perfect Rewriting Algorithm, handling GCQ\(^+\) atoms of the form \(\exists U_1, U_2 . r\) for \(r \in \text{Rel}_{\text{RCC8}}\), by introducing 4 rewriting rules.

E.g.

GCQ\(^+\) Query:

\[
q(x) \leftarrow \text{Park}(x) \& \text{some HAS\_LAKE}^*\text{loc}, \text{HAS\_PLAYGR}^*\text{loc}.
\]

\[
\{dc, ec, po, tpp, tpri, ntppi, eq\}(x)
\]

\[
r \in \text{Rel}_{\text{RCC8}}
\]
Performance Optimization

- **1st rule:** If a GCQ+ atom of the form $\exists R_1 \circ loc, R_2 \circ loc \cdot r_3(x)$ occurs during the rewriting process, then it can be substituted by the conjunct of two new atoms of the form $\exists R_1 \circ loc, loc \cdot r_1(x)$ and $\exists loc, R_2 \circ loc \cdot r_2(x)$ in a new CQ for all $r_1,r_2$, contained in Rel$_{RCC8}$ such that $r_1,r_2 \subseteq r_3$, namely where all possible compositions of the sets $(r_1 \circ r_2)$ from a full composition table are refinements of $r_3$.

- **Disadvantage:** The full composition table has 65025 possible combinations of pairs $(r_1,r_2) \Rightarrow$ exponential blow-up, generating up to 130050 new query atoms for every input query atom in the form $\exists R_1 \circ loc, R_2 \circ loc \cdot r_3(x)$

- **Optimization:** do not search for all $r_1,r_2 \subseteq r_3$, but search for all maximal pairs $r_1,r_2$ such that $r_1,r_2 \subseteq r_3$ and do the reformulation process only w.r.t. these pairs.

  E.g. if $r_1,r_2 \subseteq r_3$, $r_4,r_5 \subseteq r_3$ and $r_4 \subseteq r_1$, $r_5 \subseteq r_2$, $\Rightarrow$ then $r_4,r_5$ is redundant, since $r_1,r_2$ is the maximal pair
Rewriting the Example

\[ Q = \text{Park}(x) \land \\
\exists \text{hasLake} \circ \text{loc}, \text{hasPlAr} \circ \text{loc}.(\mathcal{B}_{RCC8} \setminus \{\text{ntpp}\})(x) \]

\[ Q' = (\exists \text{hasLake} \circ \text{loc}, \text{loc}.\text{tpp})(x) \land \\
(\exists \text{loc}, \text{hasPlAr} \circ \text{loc}.\text{tppi})(x) \]

\[ Q'' = \text{Park+Lake}(x) \land \text{Park4Playing}(x) \]
## Preliminary Evaluation

<table>
<thead>
<tr>
<th>Test parameter</th>
<th>Exp. 1/Result</th>
<th>Exp. 2/Result</th>
<th>Exp. 3/Result</th>
<th>Exp. 4/Result</th>
<th>Exp. 5/Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tbox</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>concepts</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>roles</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>axioms</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>PIs</td>
<td>9</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>FeaturePaths</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Input Query</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>atoms</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>concept atoms</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>role atoms</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FeaturePath atoms</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Perfect Rewr. Algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>adapted</td>
<td>original</td>
<td>adapted</td>
<td>adapted</td>
<td>adapted</td>
</tr>
<tr>
<td>iterations</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>added queries case 1</td>
<td>0</td>
<td>x</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>added queries case 2/3</td>
<td>0</td>
<td>x</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>added queries case 4</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>queries before drop()</td>
<td>5</td>
<td>x</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>queries after drop()</td>
<td>5</td>
<td>x</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>execution time in ms</td>
<td>5</td>
<td>4</td>
<td>47817</td>
<td>47985</td>
<td>48711</td>
</tr>
</tbody>
</table>
Summary and Outlook

- GDL-Lite-8 modeling language
- GCQ+ as a querying language
- First steps towards implementation

- What if there is spatial information in the DB → Wednesday
- GDL-Lite-2, -3 → Wednesday
- TDL-Lite-13 (Allen) seems equally possible
Questions & Answers...
Questions & Answers...

This work was supported by
Deutsche Forschungsgemeinschaft (DFG)
as part of a grant for the project GeoDL

The work is now supported by the European Commision as part of the FP7 project Optique
(http://optique.project.ifi.uio.no)