Cost-driven OBDA

ISWC 2017
Vienna, Austria

- **Speaker:** Davide Lanti
dlanti@inf.unibz.it

- **Joint Work With:** Guohui Xiao, Diego Calvanese

- **Institution:** Free University of Bozen-Bolzano
An OBDA Scenario (Semantic Web View)

User

Ontology \( \mathcal{T} \)

Virtual Layer

- \( A \text{ owl:subclassOf } B; \)
- \( C \text{ owl:disjointWith } A. \)

Virtual Graph \( \mathcal{G}_{M,D} \)

Mapping \( M \)

Physical Layer

- \( :a/\{id\} \text{ a :A } \leftarrow \text{SELECT id FROM T1} \)
- \( :b/\{id\} \text{ a :B } \leftarrow \text{SELECT id FROM T2} \)

DB \( \mathcal{D} \)

DB \( \mathcal{D} \) exposes Virtual Graph \( \mathcal{G}_{M,D} \)

User queries
An OBDA Scenario (Semantic Web View)

User

Virtual Graph $G_{M,D}$

A `owl:subclassOf` B;
C `owl:disjointWith` A.

Ontology $T$

Virtual Layer

DB $D$

Mapping $M$

Physical Layer

queries

Cost-driven OBDA
Davide Lanti, Guohui Xiao, Diego Calvanese
An OBDA Scenario (Semantic Web View)

User queries

Virtual Graph $G_{M,D}$

Ontology $T$

$A \text{ owl:subclassOf } B;$

$C \text{ owl:disjointWith } A.$

Virtual Layer

exposes

Mapping $M$

Physical Layer

$\begin{array}{c|c}
\text{id} & \text{...} \\
1 & 2 \\
3 & 4 \\
5 & 6 \\
\end{array}$

$\begin{array}{c|c}
\text{id} & \text{...} \\
1 & 2 \\
3 & 4 \\
5 & 6 \\
\end{array}$

$\text{a/1} \text{ a A.}$

$\text{b/1} \text{ a B.}$

$\text{a/2} \text{ a A.}$

$\text{...}$

$\text{a/\{id\}} \text{ a A } \leftarrow \text{SELECT id FROM T1}$

$\text{b/\{id\}} \text{ a B } \leftarrow \text{SELECT id FROM T2}$

Cost-driven OBDA

Davide Lanti, Guohui Xiao, Diego Calvanese
Query Evaluation Process I

OWL 2 QL Ontology $\mathcal{T}$

$A$ owl:subclassOf $B$;
$C$ owl:disjointWith $A$.

Mapping $\mathcal{M}$

$\text{SELECT } id \text{ FROM } T1$

$\text{SELECT } id \text{ FROM } T2$

SPARQL query $q$ → Rewriting → Rewritten Query $q_{rew}$ → Unfolding

Unfolded Query $q_{unf}$

Evaluation

SQL Result

Res. Translation

SPARQL Result

DB $\mathcal{D}$
Query Evaluation Process II

SPARQL query $q$

Rewriting

Rewritten Query $q_{rew}$

Unfolding

Unfolded Query $q_{unf}$

OWL 2 QL Ontology $\mathcal{T}$

A owl:subclassOf B;
C owl:disjointWith A.

Mapping $\mathcal{M}$

\begin{align*}
\text{a/\{id\}} & \quad \text{a :A } \leftarrow \text{SELECT id FROM T1} \\
\text{b/\{id\}} & \quad \text{a :B } \leftarrow \text{SELECT id FROM T2}
\end{align*}

Evaluation

SQL Result

Evaluation

DB $\mathcal{D}$

Cost-driven OBDA
Davide Lanti, Guohui Xiao, Diego Calvanese
Query Evaluation Process III

SPARQL query $q$

**Rewriting**

Rewritten Query $q_{rew}$

**Unfolding**

Unfolded Query $q_{unf}$

Mapping $\mathcal{M}$

OWL 2 QL Ontology $\mathcal{T}$

$A \text{ owl:subclassOf } B;$

$C \text{ owl:disjointWith } A.$

$\text{SELECT } \text{id} \text{ FROM } T1$

$\text{SELECT } \text{id} \text{ FROM } T2$

SQL Result

Evaluation

DB $\mathcal{D}$

Res. Translation

SPARQL Result
Query Evaluation Process IV

OWL 2 QL Ontology $\mathcal{T}$

A owl:subclassOf B;
C owl:disjointWith A.

Mapping $\mathcal{M}$

\[
\begin{align*}
\text{SELECT id} & \text{ FROM T1} \\
\text{SELECT id} & \text{ AS t} \text{ FROM T2}
\end{align*}
\]

Unfolding

\[
\begin{align*}
\text{SELECT id, "b" AS t} & \text{ FROM T2} \\
\text{UNION} & \text{ SELECT id, "b" AS t} \text{ FROM T2}
\end{align*}
\]

Unfolded Query $q_{uf}$

Evaluation

$DB \mathcal{D}$

SPARQL query $q$

Rewriting

Rewritten Query $q_{rew}$

SPARQL Result

Res. Translation

SQL Result

Evaluation

SPARQL Result

Res. Translation

SQL Result

Evaluation

DB $\mathcal{D}$
Query Evaluation Process V

OWL 2 QL Ontology $\mathcal{T}$

$A \text{ owl:subclassOf } B; C \text{ owl:disjointWith } A.$

Mapping $\mathcal{M}$

\begin{align*}
&a/\{id\} \text{ a } A \leftarrow \text{ SELECT id FROM } T1 \\
&b/\{id\} \text{ a } B \leftarrow \text{ SELECT id FROM } T2
\end{align*}

SPARQL query $q$

\{?x a B.\}

Rewritten Query $q_{rew}$

\{?x a B.\} UNION \{?x a A.\}

Unfolding

SELECT id, "b" AS t FROM T2
UNION SELECT id, "b" AS t FROM T2

Evaluation

SQL Result

id t
1 "a"
2 "a"
1 "b"
2 "b"
...

Res. Translation

<table>
<thead>
<tr>
<th>id</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;a&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;b&quot;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

SPARQL Result

Res. Translation

<table>
<thead>
<tr>
<th>id</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;a&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;b&quot;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Unfolded Query $q_{unf}$

SELECT id, "b" AS t FROM T2
UNION SELECT id, "b" AS t FROM T2

Cost-driven OBDA
davide Lanti, Guohui Xiao, Diego Calvanese
Query Evaluation Process VI

OWL 2 QL Ontology $\mathcal{T}$:

\[
A \text{ owl:subclassOf } B; \\
C \text{ owl:disjointWith } A.
\]

Mapping $\mathcal{M}$:

\[
\begin{align*}
:a/\{id\} & \leftarrow \text{SELECT id FROM T1} \\
:b/\{id\} & \leftarrow \text{SELECT id FROM T2}
\end{align*}
\]

Unfolding:

\[
\text{SELECT id, "b" AS t FROM T2}
\]

\[
\text{UNION}
\]

\[
\text{SELECT id, "b" AS t FROM T2}
\]

Unfolded Query $q_{uf}$:

\[
\{?x \text{ a } B.\} \text{ UNION } \{?x \text{ a } A.\}
\]

Evaluation:

<table>
<thead>
<tr>
<th>id</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;a&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;a&quot;</td>
</tr>
<tr>
<td>1</td>
<td>&quot;b&quot;</td>
</tr>
<tr>
<td>2</td>
<td>&quot;b&quot;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

DB $\mathcal{D}$

SPARQL query $q$ → Rewriting → Rewritten Query $q_{rew}$ → Unfolding → Unfolded Query $q_{uf}$ → Evaluation → SPARQL Result $\{?x \text{ a } B.\}$ → Res. Translation → SQL Result → Evaluation → DB $\mathcal{D}$
The Performance Problem

- **Problem:**
  - Performance is an issue in real-world applications of OBDA

- **Current Solutions:**
  - Data-independent Optimizations
    - Structural Optimizations
    - Semantic Query Optimizations
The Performance Problem

Problem:

- Performance is an issue in real-world applications of OBDA

Current Solutions:

- Data-independent Optimizations
  - Structural Optimizations
  - Semantic Query Optimizations

Contribution of this work:

- Data-dependent Optimizations
  - Explore a space of alternatives for unfoldings
  - Choose the best alternative, given the instance
The Performance Problem

Problem:

- Performance is an issue in real-world applications of OBDA

Current Solutions:

- Data-independent Optimizations
  - Structural Optimizations
  - Semantic Query Optimizations

Contribution of this work:

- Data-dependent Optimizations
  - Explore a space of alternatives for unfoldings
  - Choose the best alternative, given the instance
### Unfolding and Structural Optimizations

#### Mapping $\mathcal{M}$

**% High-Level Mappings**  
<table>
<thead>
<tr>
<th>Source View</th>
<th>Target View</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{URI}_1(\vec{x}) \ a \ A$</td>
<td>$V_1(\vec{x})$</td>
</tr>
<tr>
<td>$\text{URI}_2(\vec{x}) \ a \ A$</td>
<td>$V_2(\vec{x})$</td>
</tr>
<tr>
<td>$\text{URI}_1(\vec{x}) \ R \ \text{URI}_3(\vec{y})$</td>
<td>$V_3(\vec{x}, \vec{y})$</td>
</tr>
<tr>
<td>$\text{URI}_1(\vec{x}) \ R \ \text{URI}_4(\vec{y})$</td>
<td>$V_4(\vec{x}, \vec{y})$</td>
</tr>
</tbody>
</table>

**% Low-Level Mappings**  
<table>
<thead>
<tr>
<th>Source View</th>
<th>Source Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1(\vec{x})$</td>
<td>$CQ_1(\vec{x})$</td>
</tr>
<tr>
<td>$V_2(\vec{x})$</td>
<td>$CQ_2(\vec{x})$</td>
</tr>
<tr>
<td>$V_3(\vec{x}, \vec{y})$</td>
<td>$CQ_3(\vec{x}, \vec{y})$</td>
</tr>
<tr>
<td>$V_4(\vec{x}, \vec{y})$</td>
<td>$CQ_4(\vec{x}, \vec{y})$</td>
</tr>
</tbody>
</table>
Unfolding and Structural Optimizations I

**Mapping** $\mathcal{M}$

- **% High-Level Mappings**
  - $\text{URI}_1(\vec{x}) \ a \ A \ \rightsquigarrow \ V_1(\vec{x})$
  - $\text{URI}_2(\vec{x}) \ a \ A \ \rightsquigarrow \ V_2(\vec{x})$
  - $\text{URI}_1(\vec{x}) \ R \ \text{URI}_3(\vec{y}) \ \rightsquigarrow \ V_3(\vec{x}, \vec{y})$
  - $\text{URI}_1(\vec{x}) \ R \ \text{URI}_4(\vec{y}) \ \rightsquigarrow \ V_4(\vec{x}, \vec{y})$

- **% Low-Level Mappings**
  - $V_1(\vec{x}) \ \rightsquigarrow \ CQ_1(\vec{x})$
  - $V_2(\vec{x}) \ \rightsquigarrow \ CQ_2(\vec{x})$
  - $V_3(\vec{x}, \vec{y}) \ \rightsquigarrow \ CQ_3(\vec{x}, \vec{y})$
  - $V_4(\vec{x}, \vec{y}) \ \rightsquigarrow \ CQ_4(\vec{x}, \vec{y})$

**Query q**

```
SELECT ?x ?y WHERE { ?x a A. ?x R ?y. }
```
Query $q$

```
SELECT ?x ?y WHERE { ?x a A. ?x R ?y. }
```

**SELECT CONCAT**

We introduce auxiliary notation:

$$SC_1(V_1) = \text{SELECT CONCAT('URI}_1('', \bar{x}'')' AS x FROM V_1}$$

$$\vdots$$

$$SC_{1,4}(V_4) = \text{SELECT CONCAT('URI}_1('', \bar{x}'')' AS x, CONCAT('URI}_4('', \bar{y}'')' AS y FROM V_4}$$
Unfolding and Structural Optimizations II

Query q

\[
SELECT ?x ?y WHERE \{ ?x a A. ?x R ?y. \}
\]

**SELECT CONCAT**

We introduce auxiliary notation:

\[
SC_1(V_1) = \text{SELECT CONCAT('URI}_1(','\vec{x},')') AS x FROM V_1
\]

\[
\vdots
\]

\[
SC_{1,4}(V_4) = \text{SELECT CONCAT('URI}_4(','\vec{y},')') AS x,
\text{CONCAT('URI}_4(','\vec{y},')') AS y FROM V_4
\]

**Intermediate Unfolding (Similar to [Chebotko et al., 2009])**

\[
(SC_1(V_1) \cup SC_2(V_2)) \bowtie (SC_{1,3}(V_3) \cup SC_{1,4}(V_4))
\]
Unfolding and Structural Optimizations III

**Structural Optimization 1: Transform to UCQ**

<table>
<thead>
<tr>
<th>SC_1(V_1)</th>
<th>×</th>
<th>SC_1,3(V_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC_1(V_1)</td>
<td>×</td>
<td>SC_1,4(V_4)</td>
</tr>
<tr>
<td>SC_2(V_2)</td>
<td>×</td>
<td>SC_1,3(V_3)</td>
</tr>
<tr>
<td>SC_2(V_2)</td>
<td>×</td>
<td>SC_1,4(V_4)</td>
</tr>
</tbody>
</table>

We call the unfolding after structural optimizations UCQ-unfolding.
Unfolding and Structural Optimizations III

**Structural Optimization 1: Transform to UCQ**

| SC₁(V₁)    | SC₁,3(V₃)    |
| SC₁(V₁)    | SC₁,4(V₄)    |
| SC₂(V₂)    | SC₁,3(V₃)    |
| SC₂(V₂)    | SC₁,4(V₄)    |

\[ URI₂ \neq URI₁ \] \(!!\) (Incompatible URIs)

We call the unfolding after structural optimizations UCQ-unfolding.
Unfolding and Structural Optimizations III

**Structural Optimization 1: Transform to UCQ**

<table>
<thead>
<tr>
<th>SC₁(V₁)</th>
<th>SC₁,3(V₃)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC₁(V₁)</td>
<td>SC₁,4(V₄)</td>
</tr>
<tr>
<td>SC₂(V₂)</td>
<td>SC₁,3(V₃)</td>
</tr>
<tr>
<td>SC₂(V₂)</td>
<td>SC₁,4(V₄)</td>
</tr>
</tbody>
</table>

\[ \text{URI}_2 \neq \text{URI}_1 !!! \text{ (Incompatible URIs)} \]

**Structural Optimization 2: Remove joins between incompatible URIs**

<table>
<thead>
<tr>
<th>SC₁(V₁)</th>
<th>SC₁,3(V₃)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC₁(V₁)</td>
<td>SC₁,4(V₄)</td>
</tr>
</tbody>
</table>
Unfolding and Structural Optimizations III

**Structural Optimization 1: Transform to UCQ**

\[
\begin{align*}
SC_1(V_1) & \times SC_{1,3}(V_3) \\
SC_1(V_1) & \times SC_{1,4}(V_4) \\
SC_2(V_2) & \times SC_{1,3}(V_3) \quad \text{URI}_2 \neq \text{URI}_1 \quad \text{!! (Incompatible URIs)} \\
SC_2(V_2) & \times SC_{1,4}(V_4)
\end{align*}
\]

**Structural Optimization 2: Remove joins between incompatible URIs**

\[
\begin{align*}
SC_1(V_1) & \times SC_{1,3}(V_3) \\
SC_1(V_1) & \times SC_{1,4}(V_4)
\end{align*}
\]

**Structural Optimization 3: Remove Joins over templates**

\[
SC_{1,3}(V_1 \bowtie V_3) \cup SC_{1,4}(V_1 \bowtie V_4)
\]
Unfolding and Structural Optimizations III

**Structural Optimization 1: Transform to UCQ**

\[
\begin{align*}
SC_1(V_1) & \times \ SC_{1,3}(V_3) \\
SC_1(V_1) & \times \ SC_{1,4}(V_4) \\
SC_2(V_2) & \times \ SC_{1,3}(V_3) \} \quad & URI_2 \neq URI_1 \ (Incompatible \ URIs) \\
SC_2(V_2) & \times \ SC_{1,4}(V_4) \\
\end{align*}
\]

**Structural Optimization 2: Remove joins between incompatible URIs**

\[
\begin{align*}
SC_1(V_1) & \times \ SC_{1,3}(V_3) \\
SC_1(V_1) & \times \ SC_{1,4}(V_4) \\
\end{align*}
\]

**Structural Optimization 3: Remove Joins over templates**

\[
SC_{1,3}(V_1 \Join V_3) \cup SC_{1,4}(V_1 \Join V_4)
\]

We call the unfolding after structural optimizations **UCQ-unfolding**
### Trade-off Redundancy/Indexing

#### Mapping $\mathcal{M}$

<table>
<thead>
<tr>
<th>% High-Level Mappings</th>
<th>target $\leftrightarrow$ source</th>
</tr>
</thead>
<tbody>
<tr>
<td>${ URI_1(x) \ a \ A \leftarrow V_i(x)</td>
<td>1 \leq i \leq 6 }$</td>
</tr>
<tr>
<td>${ URI_1(x) \ P \ URI_2(y) \leftarrow W_i(x, y)</td>
<td>1 \leq i \leq 6 }$</td>
</tr>
<tr>
<td>${ URI_1(x) \ R \ URI_3(z) \leftarrow Z_i(x, z)</td>
<td>1 \leq i \leq 6 }$</td>
</tr>
</tbody>
</table>
Trade-off Redundancy/Indexing

Mapping $\mathcal{M}$

% High-Level Mappings target $\leftarrow$ source

\begin{align*}
\{ \text{URI}_1(\vec{x}) & \text{ a } A \leftarrow V_i(\vec{x}) \mid 1 \leq i \leq 6 \} \\
\{ \text{URI}_1(\vec{x}) & \text{ P URI}_2(\vec{y}) \leftarrow W_i(\vec{x}, \vec{y}) \mid 1 \leq i \leq 6 \} \\
\{ \text{URI}_1(\vec{x}) & \text{ R URI}_3(\vec{z}) \leftarrow Z_i(\vec{x}, \vec{z}) \mid 1 \leq i \leq 6 \}
\end{align*}

## Trade-off Redundancy/Indexing

### Mapping $\mathcal{M}$

<table>
<thead>
<tr>
<th>% High-Level Mappings</th>
<th>target $\leftarrow$ source</th>
</tr>
</thead>
<tbody>
<tr>
<td>${\text{URI}_1(\bar{x}) \mathbin{\text{a}} A \leftarrow \text{V}_i(\bar{x}) \mid 1 \leq i \leq 6}$</td>
<td></td>
</tr>
<tr>
<td>${\text{URI}_1(\bar{x}) \mathbin{\text{P}} \text{URI}_2(\bar{y}) \leftarrow \text{W}_i(\bar{x}, \bar{y}) \mid 1 \leq i \leq 6}$</td>
<td></td>
</tr>
<tr>
<td>${\text{URI}_1(\bar{x}) \mathbin{\text{R}} \text{URI}_3(\bar{z}) \leftarrow \text{Z}_i(\bar{x}, \bar{z}) \mid 1 \leq i \leq 6}$</td>
<td></td>
</tr>
</tbody>
</table>

- **SELECT DISTINCT * WHERE** \(\{\text{?x rdf:type A. } \text{?x P ?y. } \text{?x R ?z}\}\)

- **UCQ-unfolding:** \(\bigcup_{i,j,k \in \{1, \ldots, 6\}} \text{SC}(\text{V}_i \Join \text{W}_j \Join \text{Z}_k)\)
Trade-off Redundancy/Indexing

**Mapping \( \mathcal{M} \)**

<table>
<thead>
<tr>
<th>% High-Level Mappings</th>
<th>target ( \sim ) source</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ \text{URI}_1(\vec{x}) \ a \ A \ \sim \ V_i(\vec{x}) \mid 1 \leq i \leq 6 }</td>
<td></td>
</tr>
<tr>
<td>{ \text{URI}_1(\vec{x}) \ P \ \text{URI}_2(\vec{y}) \ \sim \ W_i(\vec{x}, \vec{y}) \mid 1 \leq i \leq 6 }</td>
<td></td>
</tr>
<tr>
<td>{ \text{URI}_1(\vec{x}) \ R \ \text{URI}_3(\vec{z}) \ \sim \ Z_i(\vec{x}, \vec{z}) \mid 1 \leq i \leq 6 }</td>
<td></td>
</tr>
</tbody>
</table>

  - **UCQ-unfolding:** \( \bigcup_{i,j,k \in \{1, \ldots, 6\}} \mathcal{SC}(V_i \bowtie W_j \bowtie Z_k) \)
  - \( 6^3 = 216 \) CQs
Trade-off Redundancy/Indexing

**Mapping $\mathcal{M}$**

- **% High-Level Mappings** \( \text{target} \leftarrow \text{source} \)
  
  \[
  \begin{align*}
  &\{ \text{URI}_1(\vec{x}) \text{ a A} \leftarrow V_i(\vec{x}) \mid 1 \leq i \leq 6 \} \\
  &\{ \text{URI}_1(\vec{x}) \text{ P URI}_2(\vec{y}) \leftarrow W_i(\vec{x}, \vec{y}) \mid 1 \leq i \leq 6 \} \\
  &\{ \text{URI}_1(\vec{x}) \text{ R URI}_3(\vec{z}) \leftarrow Z_i(\vec{x}, \vec{z}) \mid 1 \leq i \leq 6 \}
  \end{align*}
  \]

- \( \text{SELECT DISTINCT * WHERE } \{ \text{x rdf:type A. x P y. x R z} \} \)
  
  - **UCQ-unfolding:** \( \bigcup_{i,j,k \in \{1,...,6\}} \text{SC}(V_i \Join W_j \Join Z_k) \)
    
    - \( 6^3 = 216 \) CQs
    
    - Redundancy?
Trade-off Redundancy/Indexing

### Mapping $\mathcal{M}$

<table>
<thead>
<tr>
<th>% High-Level Mappings</th>
<th>$\text{target } \iff \text{source}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${ URI_1(\bar{x}) \ a A \iff V_i(\bar{x}) \</td>
<td>\ 1 \leq i \leq 6 }$</td>
</tr>
<tr>
<td>${ URI_1(\bar{x}) \ P URI_2(\bar{y}) \iff W_i(\bar{x}, \bar{y}) \</td>
<td>\ 1 \leq i \leq 6 }$</td>
</tr>
<tr>
<td>${ URI_1(\bar{x}) \ R URI_3(\bar{z}) \iff Z_i(\bar{x}, \bar{z}) \</td>
<td>\ 1 \leq i \leq 6 }$</td>
</tr>
</tbody>
</table>

- **SELECT DISTINCT * WHERE** $\{ ?x \ \text{rdf:type} \ A. \ ?x \ P \ ?y. \ ?x \ R \ ?z \}$

  - **UCQ-unfolding:** $\bigcup_{i,j,k \in \{1, \ldots, 6\}} SC(V_i \bowtie W_j \bowtie Z_k)$
    - $6^3 = 216 \ CQs$
    - **Redundancy?**

- **SELECT DISTINCT * WHERE** $\{ \{ ?x \ \text{rdf:type} \ A. \ ?x \ P \ ?y \}\ \{ ?x \ \text{rdf:type} \ A. \ ?x \ R \ ?z \} \}$

Cost-driven OBDA
Davide Lanti, Guohui Xiao, Diego Calvanese
Trade-off Redundancy/Indexing

**Mapping \( \mathcal{M} \)**

\% High-Level Mappings target \( \xrightarrow{\sim} \) source

\[
\begin{align*}
\{ \text{URI}_1(\bar{x}) \text{ a } A \sim V_i(\bar{x}) \mid 1 \leq i \leq 6 \} \\
\{ \text{URI}_1(\bar{x}) \text{ P } \text{URI}_2(\bar{y}) \sim W_i(\bar{x}, \bar{y}) \mid 1 \leq i \leq 6 \} \\
\{ \text{URI}_1(\bar{x}) \text{ R } \text{URI}_3(\bar{z}) \sim Z_i(\bar{x}, \bar{z}) \mid 1 \leq i \leq 6 \}
\end{align*}
\]

  
  - **UCQ-unfolding**: \( \bigcup_{i,j,k \in \{1,\ldots,6\}} \text{SC}(V_i \Join W_j \Join Z_k) \)
    
    - \( 6^3 = 216 \) CQs
    
    - Redundancy?

  
  - **Join of UCQ-unfoldings**: \( (\bigcup_{i,j \in \{1,\ldots,6\}} \text{SC}(V_i \Join W_j)) \Join (\bigcup_{i,k \in \{1,\ldots,6\}} \text{SC}(V_i \Join Z_k)) \)
Trade-off Redundancy/Indexing

**Mapping \( \mathcal{M} \)**

% High-Level Mappings target \( \xrightleftharpoons{} \) source

\[
\begin{align*}
&\{ \text{URI}_1(\vec{x}) \ a \ A \ xrightarrow{} V_i(\vec{x}) \ | \ 1 \leq i \leq 6 \} \\
&\{ \text{URI}_1(\vec{x}) \ P \ \text{URI}_2(\vec{y}) \ xrightarrow{} W_i(\vec{x}, \vec{y}) \ | \ 1 \leq i \leq 6 \} \\
&\{ \text{URI}_1(\vec{x}) \ R \ \text{URI}_3(\vec{z}) \ xrightarrow{} Z_i(\vec{x}, \vec{z}) \ | \ 1 \leq i \leq 6 \}
\end{align*}
\]

  - **UCQ-unfolding:** \( \bigcup_{i,j,k \in \{1, \ldots, 6\}} SC(V_i \bowtie W_j \bowtie Z_k) \)
    - \( 6^3 = 216 \) CQs
    - Redundancy?
  - SELECT DISTINCT * WHERE \{ \{ ?x rdf:type A. ?x P ?y \} \ { ?x rdf:type A. ?x R ?z \} \}
    - **Join of UCQ-unfoldings:** \( \bigcup_{i,j \in \{1, \ldots, 6\}} SC(V_i \bowtie W_j) \bowtie \bigcup_{i,k \in \{1, \ldots, 6\}} SC(V_i \bowtie Z_k) \)
    - \( 6^2 + 6^2 = 64 \) CQs
Trade-off Redundancy/Indexing

**Mapping \( \mathcal{M} \)**

% High-Level Mappings \( \text{target} \rightleftharpoons \text{source} \)

\[
\begin{align*}
\{ \text{URI}_1(\vec{x}) \ a \ A \rightleftharpoons V_i(\vec{x}) \mid 1 \leq i \leq 6 \} \\
\{ \text{URI}_1(\vec{x}) \ P \ \text{URI}_2(\vec{y}) \rightleftharpoons W_i(\vec{x}, \vec{y}) \mid 1 \leq i \leq 6 \} \\
\{ \text{URI}_1(\vec{x}) \ R \ \text{URI}_3(\vec{z}) \rightleftharpoons Z_i(\vec{x}, \vec{z}) \mid 1 \leq i \leq 6 \}
\end{align*}
\]

- **SELECT DISTINCT * WHERE** \{ \(?x\) rdf:type A. \(?x\) P \(?y\). \(?x\) R \(?z\) \}
  - **UCQ-unfolding:** \( \bigcup_{i,j,k \in \{1, \ldots, 6\}} SC(V_i \Join W_j \Join Z_k) \)
    - \( 6^3 = 216 \) CQs
    - Redundancy?
- **SELECT DISTINCT * WHERE** \{ \{ \(?x\) rdf:type A. \(?x\) P \(?y\) \}\ ?x\ rdf:type A. \(?x\) R \(?z\) \}
  - **Join of UCQ-unfoldings:** \( (\bigcup_{i,j \in \{1, \ldots, 6\}} SC(V_i \Join W_j)) \Join (\bigcup_{i,k \in \{1, \ldots, 6\}} SC(V_i \Join Z_k)) \)
    - \( 6^2 + 6^2 = 64 \) CQs
    - Indexes?
Key Ideas

- The UCQ form for the unfolding is a “one-fits-all” solution
  - However, other forms could be more efficient over certain data instances
- Ideally, one would want to choose the best form for the given OBDA setting
- This is typical in databases, where cost models are used to select the best among different candidate query execution plans
- [Bursztyn et al., 2015] adapted such cost-based solutions to the OBQA context
- We here study how to adapt such techniques to the OBDA scenario

---

1 OBDA setting: ontology, mappings, database schema and database instance.
2 Ontology-based Query Answering. No mappings.
Aspects to Consider

- Adapting such techniques requires to consider several aspects:
  - Alternative unfoldings and structural optimizations
  - Additional information available in an OBDA input
  - More complex, structured, search space
  - Interaction of the mappings and data with the ontology
Adapting such techniques requires to consider several aspects:

- Alternative unfoldings and structural optimizations
- Additional information available in an OBDA input
- More complex, structured, search space
- Interaction of the mappings and data with the ontology
Collecting Statistics (for Properties)

- Consider mappings $\mathcal{M}$ and a data instance $\mathcal{D}$
- Let $URI_1(\vec{x}) \ P URI_2(\vec{y}) \ \leftarrow \ V(\vec{x}, \vec{y})$ be a mapping in $\mathcal{M}$
- We collect:
  - $|V^\mathcal{D}|$  
  - $|\pi_{\vec{x}}(V^\mathcal{D})|$, $|\pi_{\vec{y}}(V^\mathcal{D})|$  

- If $URI_1(\vec{x}) \ R URI_3(\vec{w}) \ \leftarrow \ W(\vec{x}, \vec{w})$ belongs to $\mathcal{M}$, then we also collect:
  - $|\pi_{\vec{x}}(V^\mathcal{D}) \cap \pi_{\vec{x}}(W^\mathcal{D})|$
Cardinality Estimation

- For a single join:
  \[ f_D(\Join_{\vec{x} = \vec{y}} V \bowtie W) = \frac{|\text{proj}_{\vec{x}} V^D|}{|\pi_{\vec{x}} V^D|} \cdot \frac{|\text{proj}_{\vec{y}} W^D|}{|\pi_{\vec{y}} W^D|} \]

- Statistics Used: (S1), (S2), (S3)

- Assumptions:
  - **Uniformity:** \( P(C = v_1) = P(C = v_2), \) for all \( v_1, v_2 \in C \)
  - **Independence:** \( P(C_1 = v_1 | C_2 = v_2) = P(C_1 = v_1), \) for all \( v_1 \in C_1 \) and \( v_2 \in C_2 \)
Cardinality Estimation for Basic CQs

Given a basic CQ $E'$, $f_D(E')$ estimates the number $|E'\mathcal{D}|$ of distinct results in the evaluation of $E'$ over $\mathcal{D}$. We define it as:

$$f_D(E \bowtie_{V[p]} x = w[q] \cdot y W[q]) = \begin{cases} 
\frac{k_D(V[p] \bowtie_{V[p]} x = w[q] \cdot y W[q]) \cdot |V^D| \cdot |W^D|}{\text{dist}_D(V, V[p] \cdot x = w[q] \cdot y W[q]) \cdot \text{dist}_D(W, W[q] \cdot y W[q])}, & \text{if } E = V \\
\frac{k_D(E \bowtie_{V[p]} x = w[q] \cdot y W[q]) \cdot f_D(E) \cdot |W^D|}{\text{dist}_D(E, V[p] \cdot x = w[q] \cdot y W[q]) \cdot \text{dist}_D(W, V[p] \cdot y W[q])}, & \text{otherwise.}
\end{cases}$$
Cost Model

### Cost for the Unfolding of a UCQ

\[
c(q_{ucq}) = \sum_i c(q_{cq}^i) + c_u(q_{ucq})
\]

- \(c(q_{cq}^i)\) is the cost of evaluating each \(q_{cq}^i\) in \(q_{ucq}\)
- \(c_u(q_{ucq})\) is the cost of removing duplicate results

### Cost for the Unfolding of a Join of UCQs (JUCQ)

\[
c(q_{jucq}) = \sum_i c(q_{ucq}^i) + \sum_{i \neq k} c_{mat}(q_{ucq}^i) + c_{mj}(q_{jucq}) + c_u(q_{jucq})
\]

- \(c(q_{ucq}^i)\) is the cost of evaluating each UCQ component \(q_{ucq}^i\)
- \(\sum_{i \neq k} c_{mat}(q_{ucq}^i)\) is the cost of materializing the intermediate results from \(q_{ucq}^i\)
- \(c_{mj}(q_{jucq})\) is the cost of a merge join over the materialized intermediate results
- \(c_u(q_{jucq})\) is the cost of removing duplicate results
Empirical Evaluation

- **Test suites:** System testing (OBDA extension of the Wisconsin Benchmark) and application-driven (NPD Benchmark)
- **Hardware:** HP Proliant 24 cores@3.47GHz; 106 GB of RAM
- **RDBMS:** PostgreSQL 9.6
Wisconsin Experiment

- Comparison between:
  - SELECT DISTINCT * WHERE {\(?x : mrP1 ?y. \?x : jrP2 ?z. \?x : jrP3 ?w\)} (q\textsubscript{ucq})
  - SELECT DISTINCT \(?x ?y ?z ?w\) WHERE 
    \{ {\?x : mrP1 ?y. \?x : jrP2 ?z} {\?x : mrP1 ?y1. \?x : jrP3 ?w} \} (q\textsubscript{jucq})

- Parameters:
  - \(m\): number of mappings, in \(\{1, \ldots, 6\}\)
  - \(r\): number of redundant mappings\(^3\), in \(\{1, \ldots, m - 1\}\)
  - \(j\): join-selectivity\(^4\), in \(\{5, 10, 15, 20\}\)

- All intra-BGP joins are over indexed values

\(^3\)Mappings which do not contribute to new assertions
\(^4\)% of the number of retrieved rows for each mapping defining the property (each mapping retrieves 200k rows)
Analysis of the Trade-Off Redundancy/Indexing

- **Blue color**: $q^{ucq}$ is better
- **Red color**: $q^{jucq}$ is better
- **Numbers in cells**: $1 - \frac{time(q^{jucq})}{time(q^{ucq})}$

**Figure**: Performance gain of JUCQ compared with UCQ
Cost

Figure: Our cost model vs. evaluation

Figure: PostgreSQL cost model vs. evaluation
Evaluation of PostgreSQL Cardinality Estimation (UCQs)

**Figure**: UCQs: (PostgreSQL estimated cardinality) / (real cardinality)
Evaluation of PostgreSQL Cardinality Estimation (JUCQs)

Figure: JUCQs: (PostgreSQL estimated cardinality) / (real cardinality)
The NPD benchmark is an OBDA-specific benchmark aimed at reproducing the use of an OBDA system in an industrial real world scenario.

### Table: Evaluation over the NPD benchmark

<table>
<thead>
<tr>
<th>SPARQL Query name</th>
<th># triple patterns</th>
<th>Unfolding for UCQs time (s)</th>
<th>Unfolding for JUCQs time (s)</th>
<th># Frags</th>
<th># CQs</th>
</tr>
</thead>
<tbody>
<tr>
<td>q₆</td>
<td>7</td>
<td>2.18</td>
<td>1.20</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>q₁₁</td>
<td>8</td>
<td>3.39</td>
<td>0.40</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>q₁₂</td>
<td>10</td>
<td>6.67</td>
<td>0.47</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>q₃₁</td>
<td>10</td>
<td>54.27</td>
<td>1.58</td>
<td>2</td>
<td>327</td>
</tr>
</tbody>
</table>
Conclusion and Future Work

▶ Contribution:

▶ We studied how to find the most efficient unfolding of a user query
  ▶ We introduced a space of unfolding alternatives
  ▶ We defined structural optimizations for the new alternatives
  ▶ We devised a cost model to search for the best alternative
▶ We performed two empirical evaluations, and observed that:
  ▶ Alternative unfoldings can be orders of magnitude more efficient than traditional UCQ unfoldings
  ▶ The devised cost model is well suited to select the best unfolding

▶ Future work:

▶ Integrate cost estimation with semantic optimizations
▶ Explore more unfolding alternatives
▶ Relax estimation assumptions by integrating our model with existing techniques based on histograms