Learning Commonalities in RDF and SPARQL

François Goasdoué (fg@irisa.fr)

Joint work with:

Sara El Hassad

Hélène Jaudoin

Reasoning on Data (RoD), Thursday 27th, 2019
RDF/SPARQL and data management

RDF/SPARQL: the prominent standards for the Semantic Web

- W3C recommendations
- RDF: graph data model
  - Lightweight **incomplete, deductive databases**
- SPARQL: powerful SQL-like query language for RDF
  - Interrogates both data and schema/ontology of RDF graphs
  - Requires reasoning to answer queries

RDF/SPARQL raises a timely data management challenge

- Efficient query answering in the presence of updates

RDF/SPARQL is widely adopted for semantic-rich data applications

- Linked Open Data:
Learning commonalities and data management

Learning commonalities: a variety of data management applications

- **Exploration**
  - Identification of common data and query patterns
  - Clustering of datasets and queries

- **Optimization**
  - Multi-Query Optimization
  - View selection

- **Recommendation**
  - User-to-user suggestions
  - Search suggestions
Learning commonalities in RDF and SPARQL

Least general generalization (lgg), a.k.a. least common subsumer

- Machine Learning (ILP) since the early 70’s
  - Clauses
- Knowledge Representation since the early 90’s
  - Description logics
- Semantic Web [Lehmann and Bühmann, 2011], [Colucci et al., 2013], [Colucci et al., 2016]
  - RDF: rooted RDF graphs, purely structural approaches
  - SPARQL: tree queries, purely structural approaches
Learning commonalities in RDF and SPARQL

Least general generalization (1gg), a.k.a. least common subsumer

- Machine Learning (ILP) since the early 70’s
  - Clauses
- Knowledge Representation since the early 90’s
  - Description logics
- Semantic Web [Lehmann and Bühmann, 2011], [Colucci et al., 2013], [Colucci et al., 2016]
  - RDF: rooted RDF graphs, purely structural approaches
  - SPARQL: tree queries, purely structural approaches

Our contributions:

1. 1gg of RDF graphs w.r.t. the entire RDF standard [ESWC17,ILP17]
2. 1gg of SPARQL conjunctive queries w.r.t. ontological knowledge [BDA17,ESWC17,ISWC17]
Outline

1. Introduction
2. Preliminaries
3. Lgg in RDF
   - Defining the lgg in RDF
   - Computing the lgg in RDF
4. Lgg in SPARQL
   - Defining the lgg in SPARQL
   - Computing the lgg in SPARQL
   - Experimental results
5. Related work
6. Conclusion & Perspectives
Towards defining the notion of lgg in RDF

G. Plotkin

A least general generalization (lgg) of \( n \) descriptions \( d_1, \ldots, d_n \) is a most specific description \( d \) generalizing every \( d_1 \leq i \leq n \) for some generalization/specialization relation between descriptions.

**lgg in RDF**
- descriptions are RDF graphs
- the generalization/specialization relation is entailment between RDF graphs

**lgg in our SPARQL setting**
- descriptions are Basic Graph Pattern Queries (BGPQs)
- the generalization/specialization relation is entailment between BGPQs
RDF graphs

- RDF graphs are made of triples:
  \[(s, p, o) \in (U \cup B) \times U \times (U \cup L \cup B)\]
- Built-in property URIs to make RDF statements

**RDF statement** | **Triple**
--- | ---
Class assertion | \((s, \tau, o)\)
Property assertion | \((s, p, o)\) with \(p \neq \tau\)
RDF graphs

- RDF graphs are made of triples:
  \[(s, p, o) \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})\]

- Built-in property URIs to make RDF statements

<table>
<thead>
<tr>
<th>RDF statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class assertion</td>
<td>((s, \tau, o))</td>
</tr>
<tr>
<td>Property assertion</td>
<td>((s, p, o)) with (p \neq \tau)</td>
</tr>
</tbody>
</table>

![RDF Graph Example](image-url)
Built-in property URIs to declare RDF Schema statements, i.e., ontological constraints.

<table>
<thead>
<tr>
<th>RDFS statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>((s, \preceq_{sc}, o))</td>
</tr>
<tr>
<td>Subproperty</td>
<td>((s, \preceq_{sp}, o))</td>
</tr>
<tr>
<td>Domain typing</td>
<td>((s, \rightarrow_{d}, o))</td>
</tr>
<tr>
<td>Range typing</td>
<td>((s, \rightarrow_{r}, o))</td>
</tr>
</tbody>
</table>
Adding ontological knowledge to RDF graphs

- Built-in property URIs to declare RDF Schema statements, i.e., ontological constraints.

<table>
<thead>
<tr>
<th>RDFS statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>$(s, \preceq_{sc}, o)$</td>
</tr>
<tr>
<td>Subproperty</td>
<td>$(s, \preceq_{sp}, o)$</td>
</tr>
<tr>
<td>Domain typing</td>
<td>$(s, \leftarrow_d, o)$</td>
</tr>
<tr>
<td>Range typing</td>
<td>$(s, \rightarrow_r, o)$</td>
</tr>
</tbody>
</table>

```
ConfPaper
  \preceq_{sc}
  Publication
  \leftarrow_d
  hasAuthor
  \rightarrow_r
  Researcher

b
  \tau
  "LGG in RDF"

b_1
  hasContactAuthor

hasContactAuthor
```
Let us consider the following RDF graph:
Deriving the implicit triples

RDF graph $G$

ConfPaper

Publication

hasTitle

"LGG in RDF"

hasContactAuthor

Researcher

hasContactAuthor

hasAuthor

hasAuthor

$\tau$

$\tau$

$\leq_{sc}$

$\leq_{sp}$

$d$

$r$
Deriving the implicit triples

RDF graph $\mathcal{G}$
Deriving the implicit triples

RDF graph $\mathcal{G}$

- ConfPaper
  - Publication
  - hasContactAuthor
  - hasTitle
- Researcher
  - hasAuthor
- $b$
  - hasTitle
  - hasAuthor
  - hasContactAuthor
- $b_1$
  - $\leq_{sc}$
  - $\leq_{sp}$
  - $\leq_{d}$
  - $\leq_{r}$

How to derive implicit triples of an RDF graph?
Deriving the implicit triples

RDF graph $G$
Deriving the implicit triples

RDF graph $G$

ConfPaper

Publication

Researcher

$\tau$

$\tau$

$\tau$

hasTitle

"LGG in RDF"

hasContactAuthor

hasAuthor

Publication

hasContactAuthor

hasAuthor

Researcher

hasTitle

"LGG in RDF"

hasContactAuthor

hasAuthor

RDF graph $G$
Deriving the implicit triples

How to derive implicit triples of an RDF graph?
### Entailment rules

<table>
<thead>
<tr>
<th>Rule [W3C-RDFS, 2014]</th>
<th>Entailment rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs2</td>
<td>((p, \leftarrow_d, o), (s_1, p, o_1) \rightarrow (s_1, \tau, o))</td>
</tr>
<tr>
<td>rdfs3</td>
<td>((p, \rightarrow_r, o), (s_1, p, o_1) \rightarrow (o_1, \tau, o))</td>
</tr>
<tr>
<td>rdfs5</td>
<td>((p_1, \preceq_{sp}, p_2), (p_2, \preceq_{sp}, p_3) \rightarrow (p_1, \preceq_{sp}, p_3))</td>
</tr>
<tr>
<td>rdfs7</td>
<td>((p_1, \preceq_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o))</td>
</tr>
<tr>
<td>rdfs9</td>
<td>((s, \preceq_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o))</td>
</tr>
<tr>
<td>rdfs11</td>
<td>((s, \preceq_{sc}, o), (o, \preceq_{sc}, o_1) \rightarrow (s, \preceq_{sc}, o_1))</td>
</tr>
<tr>
<td>ext1</td>
<td>((p, \leftarrow_d, o), (o, \preceq_{sc}, o_1) \rightarrow (p, \leftarrow_d, o_1))</td>
</tr>
<tr>
<td>ext2</td>
<td>((p, \rightarrow_r, o), (o, \preceq_{sc}, o_1) \rightarrow (p, \rightarrow_r, o_1))</td>
</tr>
<tr>
<td>ext3</td>
<td>((p, \preceq_{sp}, p_1), (p_1, \leftarrow_d, o) \rightarrow (p, \leftarrow_d, o))</td>
</tr>
<tr>
<td>ext4</td>
<td>((p, \preceq_{sp}, p_1), (p_1, \rightarrow_r, o) \rightarrow (p, \rightarrow_r, o))</td>
</tr>
</tbody>
</table>

Table: Sample RDF entailment rules \( \mathcal{R} \).
Materializing implicit triples using rules

RDF graph $G$
Materializing implicit triples using rules

\[ \text{rdfs9} : (s, \preceq_{\text{sc}}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o) \]
Materializing implicit triples using rules

\[ \text{rdfs9} : (s, \preceq_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o) \]
Materializing implicit triples using rules

\[ \text{rdfs9} : (s, \preceq_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o) \]
Materializing implicit triples using rules

\[ \text{rdfs9 : } (s, \preceq_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o) \]

RDF graph \( \mathcal{G} \)
Materializing implicit triples using rules

\[ \text{rdfs7 : } (p_1, \preceq_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o) \]
Materializing implicit triples using rules

\[
rdfs7 : (p_1, \lesssim_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o)
\]
Materializing implicit triples using rules

\[ rdfs7 : (p_1, \lessdot_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o) \]
Materializing implicit triples using rules

\[
\text{rdfs7} : (p_1, \preceq_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o)
\]

RDF graph $G$
Materializing implicit implicit triples using rules

\[ \text{rdfs3 : } (p, \leftarrow_r, o), (s_1, p, o_1) \rightarrow (o_1, \tau, o) \]
Materializing implicit triples using rules

\[
ext4 : (p, _{sp} p_1), (p_1, _r o) \rightarrow (p, _r o)
\]
Materializing implicit triples using rules

\[
\text{ext3} : (p, \preceq_{sp}, p_1), (p_1, \leftrightarrow d, o) \rightarrow (p, \leftrightarrow d, o)
\]
Semantics of an RDF graph

Saturated RDF graph $G^\infty$

$G^\infty$ materializes the semantic of $G$. 
Towards defining the notion of lgg in RDF

G. Plotkin

A least general generalization (lgg) of \( n \) descriptions \( d_1, \ldots, d_n \) is a most specific description \( d \) generalizing every \( d_{1 \leq i \leq n} \) for some generalization/specialization relation between descriptions.

1gg in RDF

- descriptions are RDF graphs
- the generalization/specialization relation is entailment between RDF graphs

1gg in our SPARQL setting

- descriptions are BGP Queries
- the generalization/specialization relation is entailment between BGPQs
Entailment between RDF graphs

\[ G \models_{R} G' \iff G^\infty \models G' \]

i.e., there exists a graph homomorphism from \( G' \) to \( G^\infty \).

\[ G \models_{R} G' \]

\[ \tau \]

\[ b \]

\[ \text{hasTitle} \]

\[ "LGG in RDF" \]

\[ \text{hasContactAuthor} \]

\[ \text{ConfPaper} \]

\[ \tau \]

\[ \text{Publication} \]

\[ \preceq_{sc} \]

\[ \text{hasAuthor} \]

\[ \leftrightarrow_{d} \]

\[ \text{Researcher} \]

\[ \tau \]

\[ b \]

\[ \text{hasTitle} \]

\[ \text{b}_1 \]

\[ \text{b}_2 \]

\[ G \]

\[ G' \]
Entailment between RDF graphs

\[ G \models_R G' \iff G^\infty \models G' \]

i.e., there exists a graph homomorphism from \( G' \) to \( G^\infty \).

\[ G \models_R G' \equiv G^\infty \models G' \]
Entailment between RDF graphs

\( G \models R G' \iff G^\infty \models G' \)

i.e., there exists a graph homomorphism from \( G' \) to \( G^\infty \).

\( G\models R G' \equiv G^\infty \models G' \)

\( G \) is more specific than \( G' \)!
Towards defining the notion of lgg in SPARQL

G. Plotkin

A least general generalization (1gg) of $n$ descriptions $d_1, \ldots, d_n$ is a most specific description $d$ generalizing every $d_{1 \leq i \leq n}$ for some generalization/specialization relation between descriptions.

1gg in RDF

- descriptions are RDF graphs
- the generalization/specialization relation is entailment between RDF graphs

1gg in our SPARQL setting

- descriptions are Basic Graph Pattern Queries (BGPQs)
- the generalization/specialization relation is entailment between BGPQs
Basic Graph Pattern Queries (BGPQs)

- BGPQs: SPARQL conjunctive queries, i.e., select-project-join queries
- \((s, p, o) \in (V \cup U) \times (V \cup U) \times (V \cup U \cup L)\)
Basic Graph Pattern Queries (BGPQs)

- BGPQs: SPARQL conjunctive queries, i.e., select-project-join queries
- \((s, p, o) \in (\mathcal{V} \cup \mathcal{U}) \times (\mathcal{V} \cup \mathcal{U}) \times (\mathcal{V} \cup \mathcal{U} \cup \mathcal{L})\)

Sample BGPQ \(q_1(x_1)\)
Entailing and answering queries

Query entailment
\[ \mathcal{G} \models_R q \iff \mathcal{G}^\infty \models q \]

\[ q(x_1, x_2) \]
Query entailment
\( \mathcal{G} \models_{\mathcal{R}} q \iff \mathcal{G}^\infty \models q \)
Entailing and answering queries

Query answering

\[ q(\mathcal{G}) = \{ (\bar{x})_\phi \mid \mathcal{G} \models_{\mathcal{R}} q \} \]

\[ q(x_1, x_2) \]

\[ q(\mathcal{G}) = \{ (b, \text{ConfPaper}), (b, \text{Publication}), (b_1, \text{Researcher}) \} \]
Entailing and answering queries

Query answering

\[ q(G) = \{ (\bar{x})_{\phi} \mid G \models_R q \} \]

\[ q(x_1, x_2) \]

\[ q(G) = \{ (b, \text{ConfPaper}), (b, \text{Publication}), (b_1, \text{Researcher}) \} \]
Entailing and answering queries

Query answering

\[ q(G) = \{(\bar{x})_\phi \mid G \models_R q\} \]

\[ q(x_1, x_2) \]

\[ q(G) = \{(b, \text{ConfPaper}), (b, \text{Publication}), (b_1, \text{Researcher})\} \]
Query answering

\[ q(G) = \{ (\bar{x})_\phi \mid G \models^R \phi \} \]

\[ q(x_1, x_2) \]

\[ q(G) = \{(b, \text{ConfPaper}), (b, \text{Publication}), (b_1, \text{Researcher})\} \]
Towards defining the notion of lgg in SPARQL

G. Plotkin

A least general generalization (1gg) of $n$ descriptions $d_1, \ldots, d_n$ is a most specific description $d$ generalizing every $d_1 \leq i \leq d_n$ for some generalization/specialization relation between descriptions.

1gg in RDF

- descriptions are RDF graphs
- the generalization/specialization relation is entailment between RDF graphs

1gg in our SPARQL setting

- descriptions are Basic Graph Pattern Queries (BGPQs)
- the generalization/specialization relation is entailment between BGPQs
Entailment between BGPQs

\[ q \models^R q' \iff q^\infty \models q' \]

\begin{align*}
\text{Publication} & \xrightarrow{\tau} \text{Publication} \\
\text{hasAuthor} & \xrightarrow{} \text{hasAuthor} \\
\text{hasContactAuthor} & \xrightarrow{} \text{hasContactAuthor} \\
q^\infty(x_1) & \xrightarrow{} q'(x_2)
\end{align*}
Entailment between BGPQs

\[ q \models R q' \iff q^{\infty} \models q' \]
Outline

1. Introduction
2. Preliminaries
3. Lgg in RDF
4. Lgg in SPARQL
5. Related work
6. Conclusion & Perspectives
Defining the lgg of RDF graphs

Definition (1gg of RDF graphs)

Let $G_1, \ldots, G_n$ be RDF graphs and $\mathcal{R}$ a set of RDF entailment rules.

- A generalization of $G_1, \ldots, G_n$ is an RDF graph $G_g$ such that $G_i \models_{\mathcal{R}} G_g$ holds for $1 \leq i \leq n$.

- A least general generalization (1gg) of $G_1, \ldots, G_n$ is a generalization $G_{1gg}$ of $G_1, \ldots, G_n$ such that for any other generalization $G_g$ of $G_1, \ldots, G_n$, $G_{1gg} \models_{\mathcal{R}} G_g$ holds.

Theorem

An 1gg of RDF graphs always exists; it is unique up to entailment.
Defining the lgg of RDF graphs

Definition (1gg of RDF graphs)

Let $G_1, \ldots, G_n$ be RDF graphs and $\mathcal{R}$ a set of RDF entailment rules.

- A generalization of $G_1, \ldots, G_n$ is an RDF graph $G_g$ such that $G_i \models_{\mathcal{R}} G_g$ holds for $1 \leq i \leq n$.

- A least general generalization (1gg) of $G_1, \ldots, G_n$ is a generalization $G_{1\text{gg}}$ of $G_1, \ldots, G_n$ such that for any other generalization $G_g$ of $G_1, \ldots, G_n$, $G_{1\text{gg}} \models_{\mathcal{R}} G_g$ holds.

Result: lgg of n RDF graphs vs lgg of two RDF graphs

$$\ell_3(G_1, G_2, G_3) \equiv_{\mathcal{R}} \ell_2(\ell_2(G_1, G_2), G_3)$$
$$\ldots \quad \ldots$$
$$\ell_n(G_1, \ldots, G_n) \equiv_{\mathcal{R}} \ell_2(\ell_{n-1}(G_1, \ldots, G_{n-1}), G_n)$$
$$\equiv_{\mathcal{R}} \ell_2(\ell_2(\ldots \ell_2(\ell_2(G_1, G_2), G_3) \ldots , G_{n-1}), G_n)$$

We focus on computing the lgg of two RDF graphs
Defining the lgg of RDF graphs

\[ G_1 \]

\[ G_2 \]
Defining the 1gg of RDF graphs

\[ G_1 \]

\[ G_2 \]

\[ G_{1gg} \]
Defining the $l_{gg}$ of RDF graphs

How to compute this graph?
The cover graph of RDF graphs

Definition (Cover graph)

The cover graph $G$ of two RDF graphs $G_1$ and $G_2$ is the RDF graph such that for every property $p$ in both $G_1$ and $G_2$:

$$(t_1, p, t_2) \in G_1 \text{ and } (t_3, p, t_4) \in G_2 \iff (\varsigma(t_1, t_3), p, \varsigma(t_2, t_4)) \in G$$

with $\varsigma(t_1, t_3) = t_1$ if $t_1 = t_3$ and $t_1 \in U \cup L$, else $\varsigma(t_1, t_3)$ is the blank node $b_{t_1t_3}$, and, similarly $\varsigma(t_2, t_4) = t_2$ if $t_2 = t_4$ and $t_2 \in U \cup L$, else $\varsigma(t_2, t_4)$ is the blank node $b_{t_2t_4}$. 

Example (Anti-unification)

$$(i_1, \text{hasAuthor}, \text{SA}) \in G_1 \text{ and } (i_2, \text{hasAuthor}, \text{SA}) \in G_2 \iff (b_{i_1i_2}, \text{hasAuthor}, \text{SA}) \in G$$
The cover graph of RDF graphs

Definition (Cover graph)

The cover graph \( G \) of two RDF graphs \( G_1 \) and \( G_2 \) is the RDF graph such that for every property \( p \) in both \( G_1 \) and \( G_2 \):

\[(t_1, p, t_2) \in G_1 \text{ and } (t_3, p, t_4) \in G_2 \iff (\varsigma(t_1, t_3), p, \varsigma(t_2, t_4)) \in G\]

with \( \varsigma(t_1, t_3) = t_1 \) if \( t_1 = t_3 \) and \( t_1 \in U \cup L \), else \( \varsigma(t_1, t_3) \) is the blank node \( b_{t_1 t_3} \), and, similarly \( \varsigma(t_2, t_4) = t_2 \) if \( t_2 = t_4 \) and \( t_2 \in U \cup L \), else \( \varsigma(t_2, t_4) \) is the blank node \( b_{t_2 t_4} \).

Example (Anti-unification)

- \((i_1, hasAuthor, SA) \in G_1 \) and \((i_2, hasAuthor, SA) \in G_2 \) iff \((b_{i_1 i_2}, hasAuthor, SA) \in G\)
- \((i_1, hasAuthor, SA) \in G_1 \) and \((i_2, hasContactAuthor, SA) \in G_2 \) but \((b_{i_1 i_2}, b_{hAhCA}, SA) \notin G\)
Theorem

Let $G_1$ and $G_2$ be two RDF graphs, and $\mathcal{R}$ a set of RDF entailment rules. The cover graph $G$ of $G_1^\infty$ and $G_2^\infty$ is an lgg of $G_1$ and $G_2$.

Proposition

An lgg of two RDF graphs $G_1$ and $G_2$ can be computed in $O(|G_1^\infty| \times |G_2^\infty|)$ and its size is bounded by $|G_1^\infty| \times |G_2^\infty|$. 
Cover graph-based 1gg of RDF graphs
Cover graph-based Lgg of RDF graphs
Cover graph-based lgg of RDF graphs

\( G_1^{\infty} \) and \( G_2^{\infty} \) are shown with nodes representing concepts such as Researcher, Publication, and Identifier, connected by edges indicating relationships like hasAuthor, title, and hasContactAuthor. The diagrams illustrate the structure of RDF graphs, with specific instances like "DiD" and "CwFOL".
Cover graph-based Lgg of RDF graphs
Cover graph-based 1gg of RDF graphs
Cover graph-based $\mathcal{L}$gg of RDF graphs

$G_1^\infty$

$G_2^\infty$
Defining the lgg of queries

1gg of BGPQs

Let $q_1, \ldots, q_n$ be BGPQs with the same arity and $\mathcal{R}$ a set of RDF entailment rules.

- A generalization of $q_1, \ldots, q_n$ is a BGPQ $q_g$ such that $q_i \models_{\mathcal{R}} q_g$ for $1 \leq i \leq n$.
- A least general generalization of $q_1, \ldots, q_n$ is a generalization $q_{\text{lgg}}$ of $q_1, \ldots, q_n$ such that for any other generalization $q_g$ of $q_1, \ldots, q_n$: $q_{\text{lgg}} \models_{\mathcal{R}} q_g$. 
Defining the lgg of queries

lgg of BGPQs

Let $q_1, \ldots, q_n$ be BGPQs with the same arity and $\mathcal{R}$ a set of RDF entailment rules.

- A generalization of $q_1, \ldots, q_n$ is a BGPQ $q_g$ such that $q_i \models_\mathcal{R} q_g$ for $1 \leq i \leq n$.
- A least general generalization of $q_1, \ldots, q_n$ is a generalization $q_{\text{lgg}}$ of $q_1, \ldots, q_n$ such that for any other generalization $q_g$ of $q_1, \ldots, q_n$: $q_{\text{lgg}} \models_\mathcal{R} q_g$. 

\[
\begin{align*}
q_1(x_1) &\quad \text{hasContactAuthor} \quad y_1 \\
q_2(x_2) &\quad \text{hasAuthor} \quad y_2 \\
q_{\text{lgg}}(x) &\quad \tau \\
\end{align*}
\]
Defining the \( lgg \) of queries

**\( lgg \) of BGPQs**

Let \( q_1, \ldots, q_n \) be BGPQs with the same arity and \( \mathcal{R} \) a set of RDF entailment rules.

- A *generalization* of \( q_1, \ldots, q_n \) is a BGPQ \( q_g \) such that \( q_i \models_{\mathcal{R}} q_g \) for \( 1 \leq i \leq n \).
- A *least general generalization* of \( q_1, \ldots, q_n \) is a generalization \( q_{lgg} \) of \( q_1, \ldots, q_n \) such that for any other generalization \( q_g \) of \( q_1, \ldots, q_n \):
  \[ q_{lgg} \models_{\mathcal{R}} q_g. \]
Enriching queries w.r.t. background knowledge

\[ q_1(x_1) \quad \text{and} \quad q_2(x_2) \]
Enriching queries w.r.t. background knowledge
Saturation of a query

BGPQ saturation w.r.t. RDFS constraints

Let $\mathcal{R}$ be a set of RDF entailment rules, $\mathcal{O}$ a set of RDFS statements, and $q$ a BGPQ. The saturation of $q$ w.r.t. $\mathcal{O}$, noted $q^\infty$, is the BGPQ with the same answer variables as $q$ and whose body, noted $\text{body}(q^\infty)$, is the maximal subset of $(\text{body}(q) \cup \mathcal{O})^\infty$ such that for any of its subset $S$: if $\mathcal{O} \vdash_{\mathcal{R}} S$ holds then $\text{body}(q) \vdash_{\mathcal{R}} S$ holds.
Entailment between BGPQs w.r.t. background knowledge

Entailment between BGPQs w.r.t. \( \mathcal{R}, \mathcal{O} \)

Given a set \( \mathcal{R} \) of RDF entailment rules, a set \( \mathcal{O} \) of RDFS statements, and two BGPQs \( q_1 \) and \( q_2 \) with the same arity, \( q_1 \) entails \( q_2 \) w.r.t. \( \mathcal{O} \), denoted \( q_1 \models_{\mathcal{R}, \mathcal{O}} q_2 \), iff \( q_1^{\infty} \models q_2 \) holds.

Well-founded relation: \( q_1 \models_{\mathcal{R}, \mathcal{O}} q_2 \)
- **Query entailment**: if \( G \models_{\mathcal{R}} q_1 \) holds then \( G \models_{\mathcal{R}} q_2 \) holds,
- **Query answering**: \( q_1(G) \subseteq q_2(G) \) holds

for any graph \( G \) whose set of RDFS constraints is \( \mathcal{O} \).
Defining the lgg of queries w.r.t. background knowledge

Definition (1gg of BGPQs w.r.t. RDFS constraints)

Let \( \mathcal{R} \) be a set of RDF entailment rules, \( \mathcal{O} \) a set of RDFS statements, and \( q_1, \ldots, q_n \) \( n \) BGPQs with the same arity.

- A generalization of \( q_1, \ldots, q_n \) w.r.t. \( \mathcal{O} \) is a BGPQ \( q_g \) such that \( q_i \models_{\mathcal{R},\mathcal{O}} q_g \) for \( 1 \leq i \leq n \).

- A least general generalization of \( q_1, \ldots, q_n \) w.r.t. \( \mathcal{O} \) is a generalization \( q_{1\text{gg}} \) of \( q_1, \ldots, q_n \) w.r.t. \( \mathcal{O} \) such that for any other generalization \( q_g \) of \( q_1, \ldots, q_n \) w.r.t. \( \mathcal{O} \): \( q_{1\text{gg}} \models_{\mathcal{R},\mathcal{O}} q_g \).

Theorem

An lgg of BGPQs w.r.t. RDFS statements may not exist for some set of RDF entailment rules; when it exists, it is unique up to entailment (\( \models_{\mathcal{R},\mathcal{O}} \)).
Defining the lgg of queries w.r.t. background knowledge

Definition (lgg of BGPQs w.r.t. RDFS constraints)

Let $\mathcal{R}$ be a set of RDF entailment rules, $\mathcal{O}$ a set of RDFS statements, and $q_1, \ldots, q_n$ $n$ BGPQs with the same arity.

- A generalization of $q_1, \ldots, q_n$ w.r.t. $\mathcal{O}$ is a BGPQ $q_g$ such that $q_i \models_{\mathcal{R}, \mathcal{O}} q_g$ for $1 \leq i \leq n$.

- A least general generalization of $q_1, \ldots, q_n$ w.r.t. $\mathcal{O}$ is a generalization $q_{\text{lgg}}$ of $q_1, \ldots, q_n$ w.r.t. $\mathcal{O}$ such that for any other generalization $q_g$ of $q_1, \ldots, q_n$ w.r.t. $\mathcal{O}$: $q_{\text{lgg}} \models_{\mathcal{R}, \mathcal{O}} q_g$.

Result: lgg of $n$ BGPQ queries vs lgg of two BGPQ queries

\[
\ell_3(q_1, q_2, q_3) \equiv_{\mathcal{R}, \mathcal{O}} \ell_2(\ell_2(q_1, q_2), q_3) \\
\vdots \quad \quad \vdots \\
\ell_n(q_1, \ldots, q_n) \equiv_{\mathcal{R}, \mathcal{O}} \ell_2(\ell_{n-1}(q_1, \ldots, q_{n-1}), q_n) \\
\quad \equiv_{\mathcal{R}, \mathcal{O}} \ell_2(\ell_2(\cdots \ell_2(\ell_2(q_1, q_2), q_3) \cdots, q_{n-1}), q_n)
\]

We focus on computing lgg of two BGPQ queries
Defining the \( \text{lgg} \) of queries

\[
q_1(x_1) \quad q_2(x_2) \quad q\text{}\lgg(x)
\]
Defining the \( \lgg \) of queries

\[
q_1(x_1) = \text{ConfPaper}(x_1) \quad \text{hasContactAuthor} \quad y_1
\]

\[
q_2(x_2) = \text{JourPaper}(x_2) \quad \text{hasAuthor} \quad y_2
\]

\[
\bigcirc \quad \text{Publication} \quad \text{hasAuthor} \quad \text{Researcher} \quad \text{hasContactAuthor}
\]

\[
q_{\lgg \bigcirc}(x) = x \quad \text{hasAuthor} \quad y \quad \tau
\]

\[
\text{Publication} \quad \tau
\]

\[
\text{Researcher} \quad \tau
\]
Defining the $l_{gg}$ of queries

$q_1(x_1)$

$q_2(x_2)$

How to compute this query?
The cover of SPARQL queries

Definition (Cover query)

Let $q_1, q_2$ be two BGPQs with the same arity $n$. If there exists the BGPQ $q$ such that

- $\text{head}(q_1) = q_1(x_1^1, \ldots, x_1^n)$ and $\text{head}(q_2) = q_2(x_2^1, \ldots, x_2^n)$ iff $\text{head}(q) = q(v_{x_1^1 x_2^1}, \ldots, v_{x_1^n x_2^n})$

- $(t_1, t_2, t_3) \in \text{body}(q_1)$ and $(t_4, t_5, t_6) \in \text{body}(q_2)$ iff $(\varsigma(t_1, t_4), \varsigma(t_2, t_5), \varsigma(t_3, t_6)) \in \text{body}(q)$ with, for $1 \leq i \leq 3$, $\varsigma(t_i, t_{i+3}) = t_i$ if $t_i = t_{i+3}$ and $t_i \in \mathcal{U} \cup \mathcal{L}$, otherwise $\varsigma(t_i, t_{i+3})$ is the variable $v_{t_i t_{i+3}}$

then $q$ is the cover query of $q_1, q_2$. 
The cover of SPARQL queries

Definition (Cover query)

Let $q_1, q_2$ be two BGPQs with the same arity $n$. If there exists the BGPQ $q$ such that

- $\text{head}(q_1) = q_1(x_1^1, \ldots, x_1^n)$ and $\text{head}(q_2) = q_2(x_2^1, \ldots, x_2^n)$ iff $\text{head}(q) = q(v_{x_1^1}x_2^1, \ldots, v_{x_1^n}x_2^n)$

- $(t_1, t_2, t_3) \in \text{body}(q_1)$ and $(t_4, t_5, t_6) \in \text{body}(q_2)$ iff $(\varsigma(t_1, t_4), \varsigma(t_2, t_5), \varsigma(t_3, t_6)) \in \text{body}(q)$ with, for $1 \leq i \leq 3$, $\varsigma(t_i, t_{i+3}) = t_i$ if $t_i = t_{i+3}$ and $t_i \in \mathcal{U} \cup \mathcal{L}$, otherwise $\varsigma(t_i, t_{i+3})$ is the variable $v_{t_i}\bar{t}_{i+3}$

then $q$ is the cover query of $q_1, q_2$.

Example

- $(x_1, \text{hasContactAuthor}, y_1) \in \text{body}(q_1)$ and $(x_2, \text{hasAuthor}, y_2) \in \text{body}(q_2)$ iff $(v_{x_1}x_2, v_{\text{hCAhA}}, v_{y_1}y_2) \in \text{body}(q)$
Cover query-based lgg

Theorem

Given a set $\mathcal{R}$ of RDF entailment rules, a set $\mathcal{O}$ of RDFS statements and two BGPQs $q_1, q_2$ with the same arity,

1. the cover query $q$ of $q_1^{\mathcal{O}}, q_2^{\mathcal{O}}$ exists iff an lgg of $q_1, q_2$ w.r.t. $\mathcal{O}$ exists;
2. the cover query $q$ of $q_1^{\mathcal{O}}, q_2^{\mathcal{O}}$ is an lgg of $q_1, q_2$ w.r.t. $\mathcal{O}$.

Proposition

A cover query-based lgg of two BGPQs $q_1$ and $q_2$ is computed in $O(|\text{body}(q_1^{\mathcal{O}})| \times |\text{body}(q_2^{\mathcal{O}})|)$ and its size is $|\text{body}(q_1^{\mathcal{O}})| \times |\text{body}(q_2^{\mathcal{O}})|$. 
Cover query-based lgg of SPARQL queries

$q_1^{\infty}(x_1)$

$q_2^{\infty}(x_2)$

$q(v_{x_1x_2})$
Cover query-based LGG of SPARQL queries

\[ q_1^\infty(x_1) \]

\[ q_2^\infty(x_2) \]

\[ q(v_{x_1x_2}) \]
Cover query-based $\text{lgg}$ of SPARQL queries

$q_1^\infty (x_1)$

$q_2^\infty (x_2)$

$q(v_{x_1x_2})$
Cover query-based Lgg of SPARQL queries

\[ q_1^\infty(x_1) \]

\[ q_2^\infty(x_2) \]

\[ q(v_{x_1x_2}) \]
Goal

- How much more precise Lggs are when entailment between BGPQs w.r.t. background knowledge ($\models_{\mathcal{R},\mathcal{O}}$) are utilized instead of just simple entailment ($\models$).

Result

$$q_{1 \leq i \leq n}, q_i \models_{\mathcal{R}} q_{1_{Lgg}} \models_{\mathcal{R},\mathcal{O}} \models_{\mathcal{R}} q_{1_{Lgg}}$$
Goal

- How much more precise $\text{lgg}$s are when entailment between BGPQs w.r.t. background knowledge ($\models_{\mathcal{R},\mathcal{O}}$) are utilized instead of just simple entailment ($\models$).

Result

$$q_{1 \leq i \leq n}, q_i \models_{\mathcal{R}} q_{\text{lgg}} \models_{\mathcal{R},\mathcal{O}} q_{\text{lgg}}$$

<table>
<thead>
<tr>
<th>DBpedia query $Q_{1 \leq i \leq 8}$</th>
<th>$Q_1$</th>
<th>$Q_2$</th>
<th>$Q_3$</th>
<th>$Q_4$</th>
<th>$Q_5$</th>
<th>$Q_6$</th>
<th>$Q_7$</th>
<th>$Q_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_i$'s shape</td>
<td>tree</td>
<td>tree</td>
<td>tree</td>
<td>graph</td>
<td>graph</td>
<td>graph</td>
<td>graph</td>
<td>graph</td>
</tr>
<tr>
<td>$</td>
<td>body(Q_i)</td>
<td>$</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Number of URI/variable occurrences in $Q_i$</td>
<td>7/5</td>
<td>9/9</td>
<td>5/7</td>
<td>7/11</td>
<td>5/7</td>
<td>9/9</td>
<td>9/9</td>
<td>9/9</td>
</tr>
<tr>
<td>$</td>
<td>Q_i(\mathcal{G}_{\text{DBpedia}})</td>
<td>$</td>
<td>77</td>
<td>0</td>
<td>41695</td>
<td>13</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>$</td>
<td>body(Q_i(\mathcal{G}_{\text{DBpedia}}))</td>
<td>$</td>
<td>16</td>
<td>19</td>
<td>19</td>
<td>23</td>
<td>16</td>
<td>23</td>
</tr>
</tbody>
</table>

Table: Characteristics of our test BGPQs (top) and of their saturations w.r.t. DBpedia constraints (bottom); times are in ms.
Experimentation: \textit{lgg} of BGPQs (DBPedia)

<table>
<thead>
<tr>
<th>\text{lgg} of 2 DBpedia BGPQs:</th>
<th>$Q_1Q_2$</th>
<th>$Q_1Q_3$</th>
<th>$Q_1Q_4$</th>
<th>$Q_2Q_3$</th>
<th>$Q_4Q_5$</th>
<th>$Q_5Q_6$</th>
<th>$Q_5Q_7$</th>
<th>$Q_7Q_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to compute $q_{1\text{lgg}}$</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>$</td>
<td>q_{1\text{lgg}}(G_{\text{DBpedia}})</td>
<td>$</td>
<td>477,455</td>
<td>34,747,102</td>
<td>34,901,117</td>
<td>34,747,102</td>
<td>1,977</td>
<td>1,221</td>
</tr>
<tr>
<td>Time to compute $q_{1\text{lgg}}^{O_{\text{DBpedia}}}$</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>15</td>
<td>14</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>$</td>
<td>q_{1\text{lgg}}^{O_{\text{DBpedia}}}(G_{\text{DBpedia}})</td>
<td>$</td>
<td>10,637</td>
<td>7,874,768</td>
<td>456,690</td>
<td>4,537,824</td>
<td>1,701</td>
<td>780</td>
</tr>
<tr>
<td>Gain in precision</td>
<td>97.77</td>
<td>77.33</td>
<td>98.69</td>
<td>86.94</td>
<td>13.96</td>
<td>36.11</td>
<td>2.85</td>
<td>48.57</td>
</tr>
</tbody>
</table>

Table: Characteristics of cover query-based \textit{lgg}s of test queries, w/ or w/o using the DBpedia RDFS constraints; times are in ms.

<table>
<thead>
<tr>
<th>\text{lgg} of 3 DBpedia BGPQs:</th>
<th>$Q_1Q_2Q_3$</th>
<th>$Q_1Q_2Q_4$</th>
<th>$Q_1Q_3Q_4$</th>
<th>$Q_2Q_3Q_4$</th>
<th>$Q_4Q_7Q_8$</th>
<th>$Q_5Q_7Q_8$</th>
<th>$Q_6Q_7Q_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to compute $q_{1\text{lgg}}$</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>$</td>
<td>q_{1\text{lgg}}(G_{\text{DBpedia}})</td>
<td>$</td>
<td>34,747,102</td>
<td>34,901,117</td>
<td>34,901,117</td>
<td>34,901,117</td>
<td>70</td>
</tr>
<tr>
<td>Time to compute $q_{1\text{lgg}}^{O_{\text{DBpedia}}}$</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td>24</td>
<td>27</td>
<td>27</td>
<td>33</td>
</tr>
<tr>
<td>$</td>
<td>q_{1\text{lgg}}^{O_{\text{DBpedia}}}(G_{\text{DBpedia}})</td>
<td>$</td>
<td>7,874,768</td>
<td>615,339</td>
<td>7,874,779</td>
<td>4,537,824</td>
<td>36</td>
</tr>
<tr>
<td>Gain in precision</td>
<td>77.33</td>
<td>98.23</td>
<td>77.43</td>
<td>86.99</td>
<td>48.57</td>
<td>13.96</td>
<td>93.25</td>
</tr>
</tbody>
</table>

Table: Characteristics of cover query-based \textit{lgg}s of 3 test queries, w/ or w/o using the DBpedia RDFS constraints; times are in ms.
Outline

1. Introduction
2. Preliminaries
3. Lgg in RDF
   - Defining the lgg in RDF
   - Computing the lgg in RDF
4. Lgg in SPARQL
   - Defining the lgg in SPARQL
   - Computing the lgg in SPARQL
   - Experimental results
5. Related work
6. Conclusion & Perspectives
Related work

Structural approaches

- **Description Logics**
  - [Baader et al., 1999].
  - [Zarrieß and Turhan, 2013].

- **RDF: Rooted graphs, ignore RDF entailment**
  - [Colucci et al., 2016].

- **SPARQL : tree queries, ignore RDF entailment**
  - [Lehmann and Bühmann, 2011].

Approaches independent of the structure

- **First Order Clauses**
  - [Plotkin, 1970].
  - [Nienhuys-Cheng and de Wolf, 1996].

- **Conceptual Graphs**
  - [Chein and Mugnier, 2009].
Conclusion

Our contributions on learning commonalities in RDF and SPARQL

- We revisited the problem of computing a least general generalization in the entire setting of RDF & SPARQL conjunctive queries.
- We defined a new entailment relationship between BGPQs w.r.t. background knowledge.
- We devise algorithms to compute lgggs of conjunctive queries and small-to-huge RDF graphs:
  - In-memory
  - Data management system
  - MapReduce
- We studied the added-value of considering entailment rules when learning lgggs of RDF graphs and entailment rules plus external ontology when learning lgggs of BGPQs, using synthetic LUBM data and real DBpedia data.
Perspectives

Learning commonalities in DL-Lite

- We study the problem of learning the lgg of KBs or queries w.r.t. an ontology, in the setting of the $DL$-$Lite_R$ which underpins the OWL2 QL profile of the Web Ontology Language, the other Semantic Web data model by W3C.
Thank you!
References I


