Learning Commonalities in SPARQL*

Sara El Hassad  François Goasdoué  Hélène Jaudoin

IRISA, Univ. Rennes 1, Lannion, France

BDA 2017 - 14 - 17 November 2017

*this paper has been published in ISWC 2017
Least general generalization (lgg)

- Machine Learning in the early 70’s by Gordon Plotkin
- Knowledge representation domain in the early 90’s
- Recently in semantic web
Introduction

Least general generalization (1gg)
- Machine Learning in the early 70’s by Gordon Plotkin
- Knowledge representation domain in the early 90’s
- Recently in semantic web

Applications of 1gg
- Query optimization: identify candidate views, or potential query sharing
- Query approximation: a set of queries by a single query
- Social network context: recommending users asking for enough related things
Introduction

Least general generalization (lgg)
- Machine Learning in the early 70’s by Gordon Plotkin
- Knowledge representation domain in the early 90’s
- Recently in semantic web

Applications of lgg
- Query optimization: identify candidate views, or potential query sharing
- Query approximation: a set of queries by a single query
- Social network context: recommending users asking for enough related things

Goal
To study the problem in the entire conjunctive fragment of SPARQL setting.
Outline

Introduction

Preliminaries

Finding commonalities between SPARQL conjunctive queries

Experiments

Related work

Conclusion
RDF graphs

- Specification of RDF graphs with 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 state RDF statements

<table>
<thead>
<tr>
<th>RDF statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class assertion</td>
<td>((s, \text{rdf:type}, o))</td>
</tr>
<tr>
<td>Property assertion</td>
<td>((s, p, o)) with (p \neq \text{rdf:type})</td>
</tr>
</tbody>
</table>
RDF graphs

- Specification of RDF graphs with triples:
  \[(s, p, o) \in (U \cup B) \times U \times (U \cup L \cup B)\]

- Built-in property URIs to state RDF statements

<table>
<thead>
<tr>
<th>RDF statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class assertion</td>
<td>((s, \text{rdf:}\text{type}, o))</td>
</tr>
<tr>
<td>Property assertion</td>
<td>((s, p, o)) with (p \neq \text{rdf:}\text{type})</td>
</tr>
</tbody>
</table>

"LGG in RDF" hasTitle "LGG in RDF"

ConfPaper hasContactAuthor \(b_1\)
Adding ontological knowledge to RDF graphs

- Built-in property URIs to state 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>$\langle s, \leq_{sc}, o \rangle$</td>
</tr>
<tr>
<td>Subproperty</td>
<td>$\langle s, \leq_{sp}, o \rangle$</td>
</tr>
<tr>
<td>Domain typing</td>
<td>$\langle s, \leftarrow_{d}, o \rangle$</td>
</tr>
<tr>
<td>Range typing</td>
<td>$\langle s, \rightarrow_{r}, o \rangle$</td>
</tr>
</tbody>
</table>
Adding ontological knowledge to RDF graphs

- Built-in property URIs to state 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, \leftarrow r, o))</td>
</tr>
</tbody>
</table>

![Diagram](image)
Deriving the implicit triples

How to derive implicit triples of an RDF graph?
### Sample set of entailment rules

<table>
<thead>
<tr>
<th>Rule</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, \leftarrow r, o), (s_1, p, o_1) \rightarrow (o_1, \tau, o))</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>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>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, \leftarrow r, o), (o, \preceq_{sc}, o_1) \rightarrow (p, \leftarrow 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, \leftarrow r, o) \rightarrow (p, \leftarrow r, o))</td>
</tr>
</tbody>
</table>

**Table:** Sample RDF entailment rules $\mathcal{R}$
Semantics of RDF graphs

Figure: Saturated RDF graph $G^\infty$
Basic graph pattern queries (BGPQ)

- BGPQ: conjunctive fragment of SPARQL queries, is the counterpart of the select-project-join queries for databases
- \((s, p, o) \in (V \cup U) \times (V \cup U) \times (V \cup U \cup L)\)
Basic graph pattern queries (BGPQ)

- BGPQ: conjunctive fragment of SPARQL queries, is the counterpart of the select-project-join queries for databases
- \((s, p, o) \in (V \cup U) \times (V \cup U) \times (V \cup U \cup L)\)

**Figure: Sample BGPQ**

$$q_1(x_1)$$
Entailing and answering queries

Query entailment
$G |\Rightarrow_R q \iff G^\infty |\models q$

$q(x_1, x_2)$
Entailing and answering queries

Query entailment
$G \models_R q \iff G^\infty \models q$
Entailing and answering queries

Query answering

\[ q(\mathcal{G}) = \{ (\vec{x})_\phi \mid \mathcal{G} \models_R^{\phi} \text{body}(q) \} \]
Entailing and answering queries

Query answering

\[ q(\mathcal{G}) = \{ (\bar{x})_\phi \mid \mathcal{G} \models^R_\mathcal{R} \text{body}(q) \} \]
Entailing between BGPQs

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

\[ q^\infty(x_1) \]

\[ q'(x_2) \]
Entailing between BGPQs

\[ q \models_R q' \iff q^\infty \models q' \]
Outline

Introduction

Preliminaries

Finding commonalities between SPARQL conjunctive queries

Experiments

Related work

Conclusion
Towards defining lgg in SPARQL conjunctive fragment

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 (G.Plotkin).

1gg in our SPARQL setting
- descriptions are BGP Queries
- relation generalization/specialization is entailment between queries
Defining the \( \text{lgg} \) of queries

\( \text{lgg} \) 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 \( q_1, \ldots, q_n \):
  \[ q_{\text{lgg}} \models_{\mathcal{R}} q_g. \]
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 $q_{1gg}$ of $q_1, \ldots, q_n$ such that for any other generalization $q_g$ of $q_1, \ldots, q_n$: $q_{1gg} \models_{\mathcal{R}} q_g$.
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 \( q_{lgg} \) of \( q_1, \ldots, q_n \) such that for any other generalization \( q_g \) of \( q_1, \ldots, q_n \):
  \( q_{lgg} \models_{\mathcal{R}} q_g \).

\[
\begin{align*}
q_1(x_1) & \quad q_2(x_2) & \quad q_{lgg}(b_{x_1x_2}) & \quad q_{lgg} \circ (b_{x_1x_2}) \\
\text{ConfPaper} & \quad \text{JourPaper} & \quad \text{Publication} & \quad \text{Researcher}
\end{align*}
\]
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 $\mathcal{G} \models_{\mathcal{R}} q_1$ holds then $\mathcal{G} \models_{\mathcal{R}} q_2$ holds,
- **Query answering**: $q_1(\mathcal{G}) \subseteq q_2(\mathcal{G})$ holds.
Saturation of queries

BGPQ saturation w.r.t. RDFS constraints

\[
\text{ConfPaper} \xleftarrow{d} \text{Publication} \xrightarrow{r} \text{Researcher} \\
\text{JourPaper} \xleftarrow{d} \text{Publication} \xrightarrow{r} \text{Researcher} \\
\text{hasContactAuthor} \leq_{\text{sc}} \text{hasAuthor} \leq_{\text{sp}} \text{hasAuthor}
\]

\[
(body(q) \cup O)^\infty \quad q_1^\infty(x_1) \quad q_2^\infty(x_2)
\]
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 \( q_{1\text{gg}} \) 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

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

$$
\ell_3(q_1, q_2, q_3) \equiv_{\mathcal{R},\mathcal{O}} \ell_2(\ell_2(q_1, q_2), q_3)
$$

\[ \vdots \]

$$
\ell_n(q_1, \ldots, q_n) \equiv_{\mathcal{R},\mathcal{O}} \ell_2(\ell_{n-1}(q_1, \ldots, q_{n-1}), q_n)
$$

$$
\equiv_{\mathcal{R},\mathcal{O}} \ell_2(\ell_2(\cdots \ell_2(q_1, q_2), q_3) \cdots, q_{n-1}), q_n)
$$

We focus on computing lgg of two BGPQ queries
Defining the 1gg of queries

$q_1(x_1)$

$q_2(x_2)$

How to compute this query?
Defining the lgg of queries

$q_1(x_1)$

\[
\begin{align*}
q_1(x_1) &:= \begin{cases} 
\text{ConfPaper} \leftarrow^d (x_1) & \text{hasContactAuthor} \Rightarrow \text{ConfPaper} \\
\text{JourPaper} \leftarrow^d (x_2) & \text{hasAuthor} \Rightarrow \text{JourPaper}
\end{cases}
\end{align*}
\]

$q_2(x_2)$

\[
\begin{align*}
q_2(x_2) &:= \begin{cases} 
\text{Publication} \leftarrow^d (y_1) & \text{hasAuthor} \Rightarrow \text{Researcher} \\
\text{ConfPaper} \leftarrow^d (x_2) & \text{hasAuthor} \Rightarrow \text{JourPaper}
\end{cases}
\end{align*}
\]

$q_{1gg\mathcal{O}}$

\[
\begin{align*}
q_{1gg\mathcal{O}} &:= \begin{cases} 
\text{Publication} \leftarrow^d (b_{x_1x_2}) & \text{hasAuthor} \Rightarrow \text{Publication} \\
\text{Researcher} \leftarrow^r (b_{y_1y_2}) & \text{hasContactAuthor} \Rightarrow \text{Researcher}
\end{cases}
\end{align*}
\]

How to compute this query?
Defining the lgg 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

- $\text{head}(q_1) = q(x_1^1, \ldots, x_1^n)$ and $\text{head}(q_2) = q(x_2^1, \ldots, x_2^n)$ 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 $(t_7, t_8, t_9) \in \text{body}(q)$ with, for $1 \leq i \leq 3$, $t_{i+6} = t_i$ if $t_i = t_{i+3}$ and $t_i \in \mathcal{U} \cup \mathcal{L}$, otherwise $t_{i+6}$ 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

$q_1^\infty (x_1)$

$q_2^\infty (x_2)$

$q(V_{x1x2})$
The cover of SPARQL queries

\[ q_1^\infty (x_1) \]

\[ q_2^\infty (x_2) \]

\[ q(V_{x_1x_2}) \]
The cover of SPARQL queries

\[ q_1^\infty (x_1) \]

\[ q_2^\infty (x_2) \]

\[ q(V_{x_1x_2}) \]
The cover of SPARQL queries

\[ q_1^\infty(x_1) \]

\[ q_2^\infty(x_2) \]

\[ q(V_{x_1x_2}) \]
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 $q_1^{\infty}, q_2^{\infty}$ exists iff an lgg of $q_1, q_2$ w.r.t. $\mathcal{O}$ exists;
2. the cover query $q$ of $q_1^{\infty}, q_2^{\infty}$ is an lgg of $q_1, q_2$ w.r.t. $\mathcal{O}$.

Corollary

A cover query-based lgg of two BGPQs $q_1$ and $q_2$ is computed in $O(|\text{body}(q_1^{\infty})| \times |\text{body}(q_2^{\infty})|)$ and its size is $|\text{body}(q_1^{\infty})| \times |\text{body}(q_2^{\infty})|$.
1gg of DBPedia queries

\[
q_{1gg} \models q_{1gg}
\]

\[
O_{DBpedia}
\]

<table>
<thead>
<tr>
<th>1gg of:</th>
<th>(Q_1 Q_2)</th>
<th>(Q_1 Q_3)</th>
<th>(Q_1 Q_4)</th>
<th>(Q_2 Q_3)</th>
<th>(Q_4 Q_5)</th>
<th>(Q_5 Q_6)</th>
<th>(Q_5 Q_7)</th>
<th>(Q_7 Q_8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to compute (q_{1gg})</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_{1gg}(G_{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 (O_{DBpedia}) (q_{1gg})</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>O_{DBpedia}(G_{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 1ggs of test queries, w/ or w/o using the DBpedia RDFS constraints; times are in ms.

<table>
<thead>
<tr>
<th>1gg3 of:</th>
<th>(Q_1 Q_2 Q_3)</th>
<th>(Q_1 Q_2 Q_4)</th>
<th>(Q_1 Q_3 Q_4)</th>
<th>(Q_2 Q_3 Q_4)</th>
<th>(Q_4 Q_7 Q_8)</th>
<th>(Q_5 Q_7 Q_8)</th>
<th>(Q_6 Q_7 Q_8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to compute (q_{1gg})</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_{1gg}(G_{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 (O_{DBpedia}) (q_{1gg})</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>O_{DBpedia}(G_{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 1ggs of 3 test queries, w/ or w/o using the DBpedia RDFS constraints; times are in ms.
Outline

Introduction

Preliminaries

Finding commonalities between SPARQL conjunctive queries

Experiments

Related work

Conclusion
Related work

Structural approaches

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

- SPARQL: tree queries
  - [Lehmann and Bühmann, 2011].

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

Approaches independent of the structure

- RDF
  - [Hassad et al., 2017].
  - [Petrova et al., 2017].

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

- First Order Clauses
  - [Nienhuys-Cheng and de Wolf, 1996].
  - [Plotkin, 1970].
We revisited the problem of computing a least general generalization of general BGPQs w.r.t. background knowledge.

We defined **new** entailment relationship between BGPQs w.r.t. background knowledge.

We studied the added-value of considering background knowledge when learning lggs.

**Perspective:**

- Heuristics in order to compute lgg without redundants triples.
- To continue experimentation using queries logs.
Thank you!

Questions?
Computing least common subsumers in description logics with existential restrictions.
In IJCAI.

Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs.
Springer.

Defining and computing least common subsumers in RDF.
J. Web Semantics, 39(0).

Learning commonalities in RDF.

Autosparql: Let users query your knowledge base.
In ESWC.

Least generalizations and greatest specializations of sets of clauses.

Entity comparison in RDF graphs.
In International Semantic Web Conference (ISWC). Springer.

A note on inductive generalization.
Machine Intelligence, 5.

RDF 1.1 semantics.
https://www.w3.org/TR/rdf11-mt/.