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

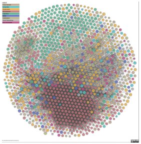
- W3C recommendations
- RDF: graph data model
  - Lightweight incomplete, deductive databases
- SPARQL: powerful SQL-like query language for RDF
  - Interrogates both data and schema/ontology of RDF graphs
  - Requires reasoning to answer queries

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 (`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

Our contributions:

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

Outline

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

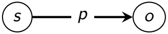
- descriptions are RDF graphs
- the generalization/specialization relation is entailment between RDF graphs

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:
  $(\mathrm{s}, \mathrm{p}, \mathrm{o}) \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})$

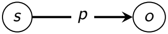


- Built-in property URIs to make RDF statements

RDF statement	Triple
Class assertion	$(\mathrm{s}, \tau, \mathrm{o})$
Property assertion	$(\mathrm{s}, \mathrm{p}, \mathrm{o})$ with $\mathrm{p} \neq \tau$

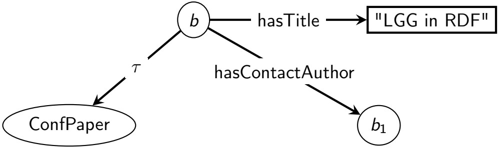
RDF graphs

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



- Built-in property URIs to make RDF statements

RDF statement	Triple
Class assertion	$(\mathrm{s}, \tau, \mathrm{o})$
Property assertion	$(\mathrm{s}, \mathrm{p}, \mathrm{o})$ with $\mathrm{p} \neq \tau$



Adding ontological knowledge to RDF graphs

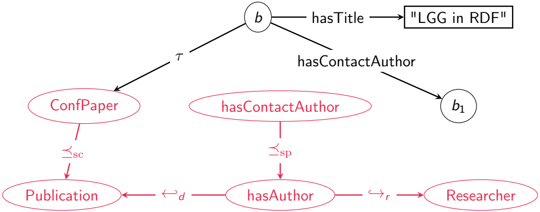
- Built-in property URIs to declare RDF Schema statements, i.e., ontological constraints.

RDFS statement	Triple
Subclass	$(\mathrm{s}, \preceq_{\mathrm{sc}}, \mathrm{o})$
Subproperty	$(\mathrm{s}, \preceq_{\mathrm{sp}}, \mathrm{o})$
Domain typing	$(\mathrm{s}, \hookleftarrow_{d}, \mathrm{o})$
Range typing	$(\mathrm{s}, \hookrightarrow_{r}, \mathrm{o})$

Adding ontological knowledge to RDF graphs

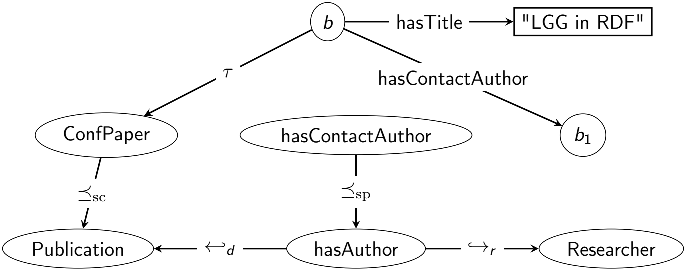
- Built-in property URIs to declare RDF Schema statements, i.e., ontological constraints.

RDFS statement	Triple
Subclass	$(\mathrm{s}, \preceq_{\mathrm{sc}}, \mathrm{o})$
Subproperty	$(\mathrm{s}, \preceq_{\mathrm{sp}}, \mathrm{o})$
Domain typing	$(\mathrm{s}, \hookleftarrow_{d}, \mathrm{o})$
Range typing	$(\mathrm{s}, \hookrightarrow_{r}, \mathrm{o})$



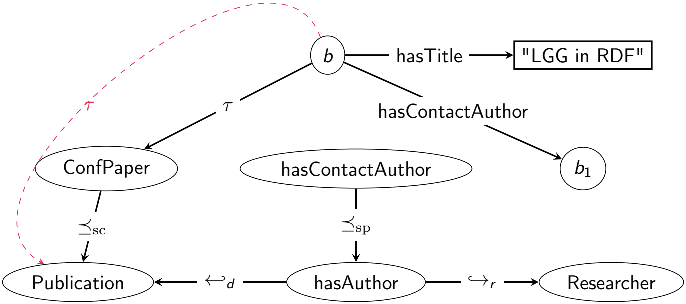
Deriving the implicit triples

Let us consider the following RDF graph:



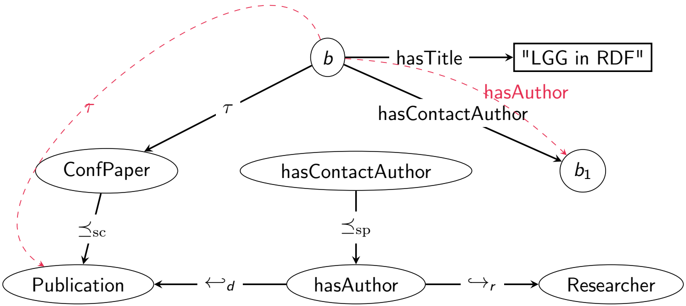
RDF graph $\mathcal{G}$

Deriving the implicit triples



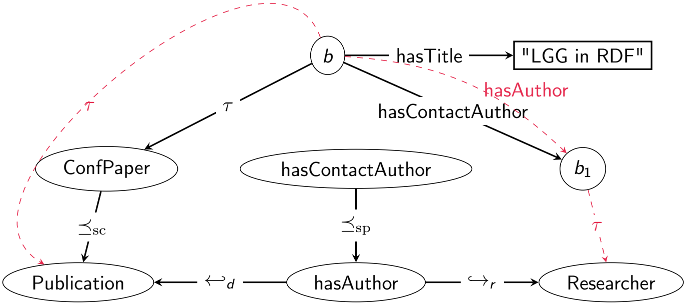
RDF graph $\mathcal{G}$

Deriving the implicit triples



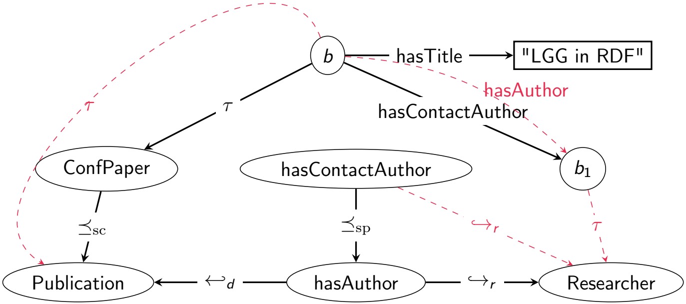
RDF graph $\mathcal{G}$

Deriving the implicit triples



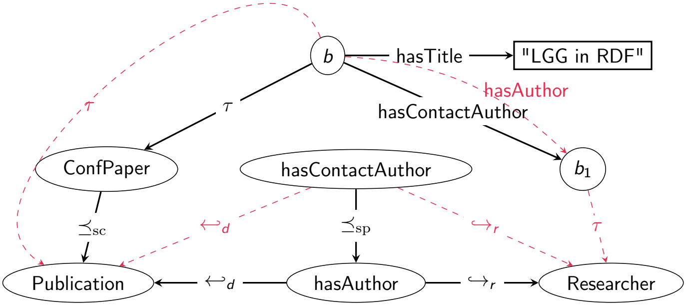
RDF graph $\mathcal{G}$

Deriving the implicit triples



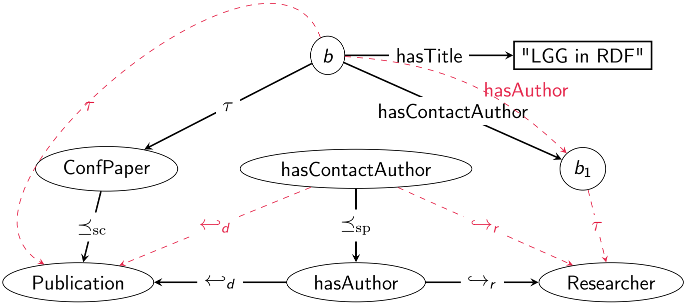
RDF graph $\mathcal{G}$

Deriving the implicit triples



RDF graph $\mathcal{G}$

Deriving the implicit triples



RDF graph $\mathcal{G}$

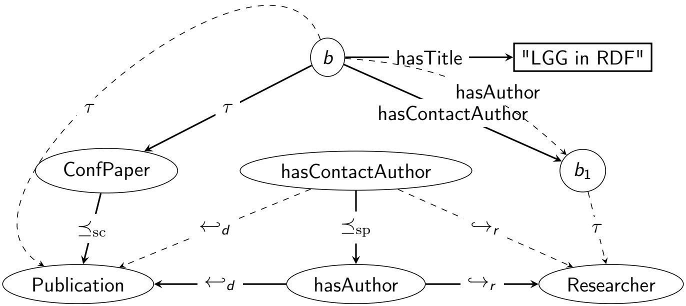
How to derive implicit triples of an RDF graph ?

Entailment rules

Rule [W3C-RDFS, 2014]	Entailment rule
`rdfs2`	$(\mathrm{p}, \hookleftarrow_d, \mathrm{o}), (\mathrm{s}_1, \mathrm{p}, \mathrm{o}_1) \rightarrow (\mathrm{s}_1, \tau, \mathrm{o})$
`rdfs3`	$(\mathrm{p}, \hookrightarrow_r, \mathrm{o}), (\mathrm{s}_1, \mathrm{p}, \mathrm{o}_1) \rightarrow (\mathrm{o}_1, \tau, \mathrm{o})$
`rdfs5`	$(\mathrm{p}_1, \preceq_{\mathrm{sp}}, \mathrm{p}_2), (\mathrm{p}_2, \preceq_{\mathrm{sp}}, \mathrm{p}_3) \rightarrow (\mathrm{p}_1, \preceq_{\mathrm{sp}}, \mathrm{p}_3)$
`rdfs7`	$(\mathrm{p}_1, \preceq_{\mathrm{sp}}, \mathrm{p}_2), (\mathrm{s}, \mathrm{p}_1, \mathrm{o}) \rightarrow (\mathrm{s}, \mathrm{p}_2, \mathrm{o})$
`rdfs9`	$(\mathrm{s}, \preceq_{\mathrm{sc}}, \mathrm{o}), (\mathrm{s}_1, \tau, \mathrm{s}) \rightarrow (\mathrm{s}_1, \tau, \mathrm{o})$
`rdfs11`	$(\mathrm{s}, \preceq_{\mathrm{sc}}, \mathrm{o}), (\mathrm{o}, \preceq_{\mathrm{sc}}, \mathrm{o}_1) \rightarrow (\mathrm{s}, \preceq_{\mathrm{sc}}, \mathrm{o}_1)$
`ext1`	$(\mathrm{p}, \hookleftarrow_d, \mathrm{o}), (\mathrm{o}, \preceq_{\mathrm{sc}}, \mathrm{o}_1) \rightarrow (\mathrm{p}, \hookleftarrow_d, \mathrm{o}_1)$
`ext2`	$(\mathrm{p}, \hookrightarrow_r, \mathrm{o}), (\mathrm{o}, \preceq_{\mathrm{sc}}, \mathrm{o}_1) \rightarrow (\mathrm{p}, \hookrightarrow_r, \mathrm{o}_1)$
`ext3`	$(\mathrm{p}, \preceq_{\mathrm{sp}}, \mathrm{p}_1), (\mathrm{p}_1, \hookleftarrow_d, \mathrm{o}) \rightarrow (\mathrm{p}, \hookleftarrow_d, \mathrm{o})$
`ext4`	$(\mathrm{p}, \preceq_{\mathrm{sp}}, \mathrm{p}_1), (\mathrm{p}_1, \hookrightarrow_r, \mathrm{o}) \rightarrow (\mathrm{p}, \hookrightarrow_r, \mathrm{o})$

Table: Sample RDF entailment rules $\mathcal{R}$.

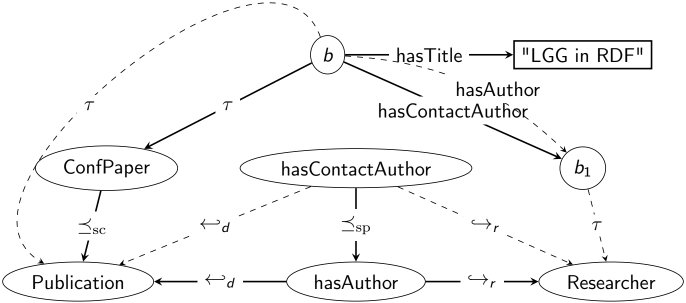
Materializing implicit triples using rules



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

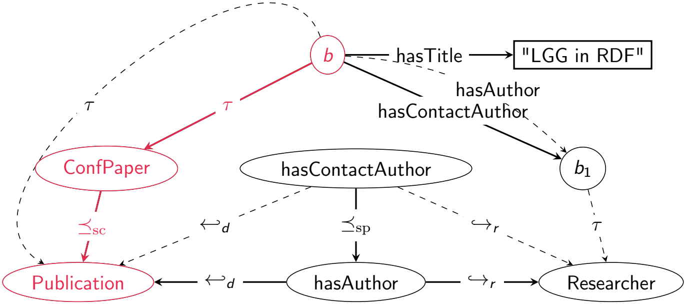
$$\textit{rdfs9} : (\mathtt{s}, \preceq_{sc}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s}) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

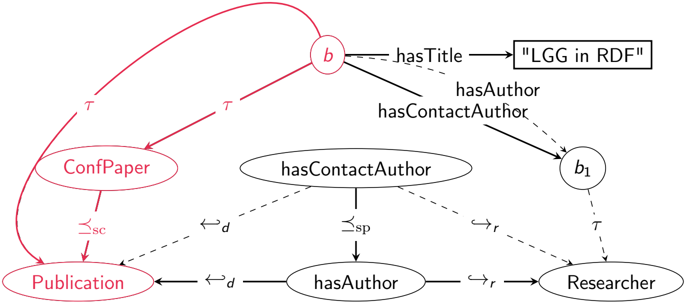
$$rdfs9 : (\mathtt{s}, \preceq_{sc}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s}) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

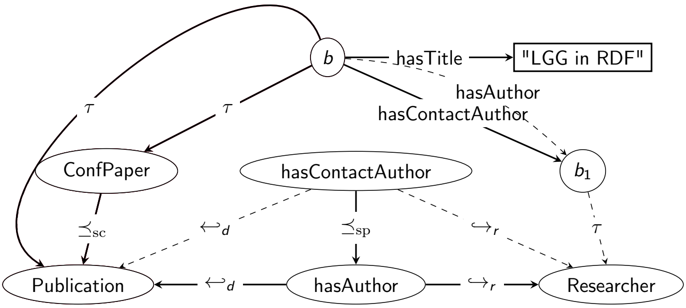
$$rdfs9 : (\mathtt{s}, \preceq_{sc}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s}) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

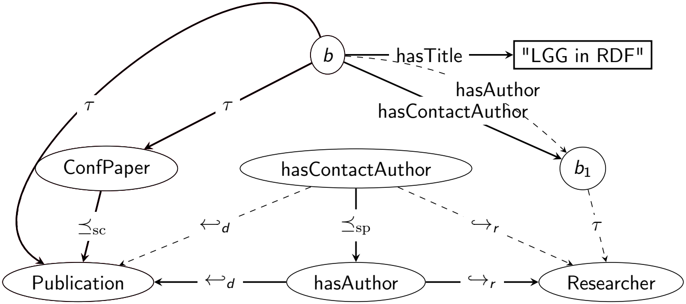
$$\mathit{rdfs9} : (\mathtt{s}, \preceq_{\mathrm{sc}}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s}) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

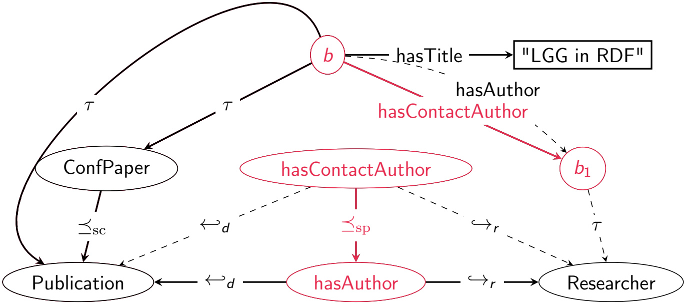
$rdfs7 : (p_1, \preceq_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o)$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

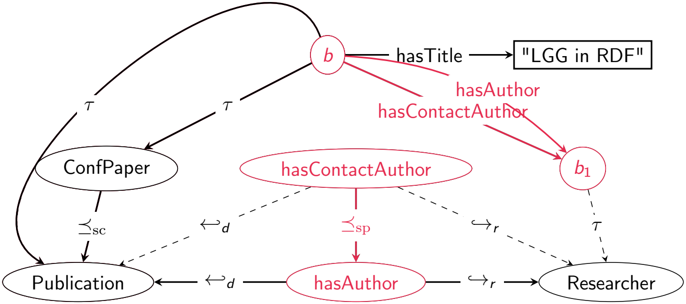
$$\textit{rdfs7} : (\mathrm{p}_1, \preceq_{sp}, \mathrm{p}_2), (\mathrm{s}, \mathrm{p}_1, \mathrm{o}) \rightarrow (\mathrm{s}, \mathrm{p}_2, \mathrm{o})$$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

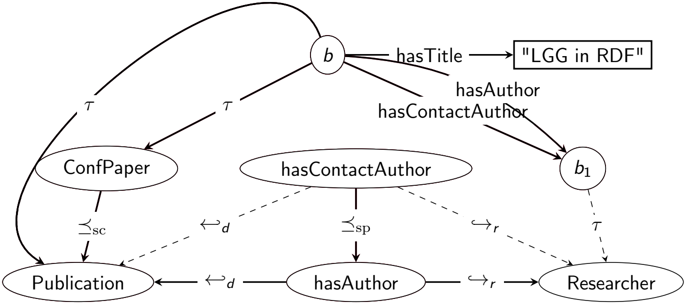
$$\mathit{rdfs7} : (\mathrm{p}_1, \preceq_{\mathrm{sp}}, \mathrm{p}_2), (\mathrm{s}, \mathrm{p}_1, \mathrm{o}) \rightarrow (\mathrm{s}, \mathrm{p}_2, \mathrm{o})$$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

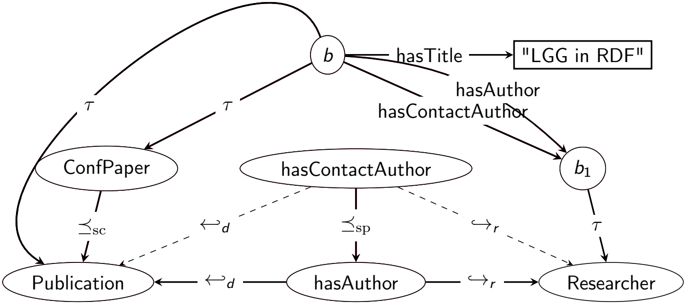
$$rdfs7 : (\mathrm{p}_1, \preceq_{sp}, \mathrm{p}_2), (\mathrm{s}, \mathrm{p}_1, \mathrm{o}) \rightarrow (\mathrm{s}, \mathrm{p}_2, \mathrm{o})$$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

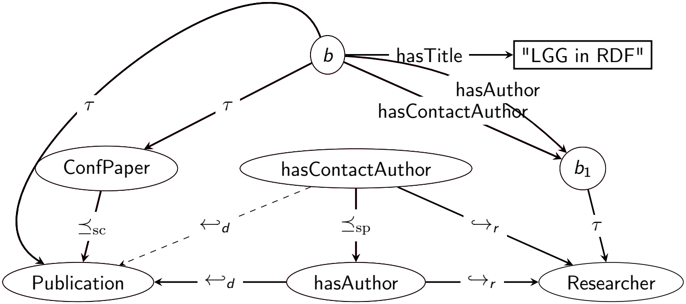
$rdfs3 : (\mathtt{p}, \hookrightarrow_r, \mathtt{o}), (\mathtt{s}_1, \mathtt{p}, \mathtt{o}_1) \rightarrow (\mathtt{o}_1, \tau, \mathtt{o})$



RDF graph $\mathcal{G}$

Materializing implicit triples using rules

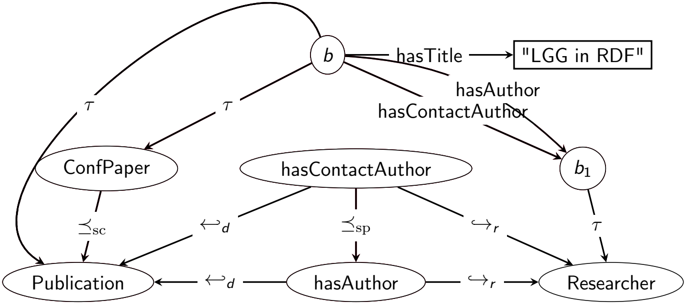
$ext4 : (\mathrm{p}, \preceq_{\mathrm{sp}}, \mathrm{p}_1), (\mathrm{p}_1, \hookrightarrow_r, \mathrm{o}) \rightarrow (\mathrm{p}, \hookrightarrow_r, \mathrm{o})$



RDF graph $\mathcal{G}$

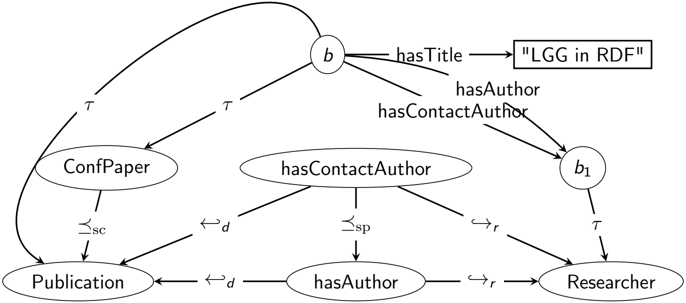
Materializing implicit triples using rules

$ext3 : (\mathrm{p}, \preceq_{sp}, \mathrm{p}_1), (\mathrm{p}_1, \hookleftarrow_d, \mathrm{o}) \rightarrow (\mathrm{p}, \hookleftarrow_d, \mathrm{o})$



RDF graph $\mathcal{G}$

Semantics of an RDF graph



Saturated RDF graph $\mathcal{G}^\infty$

$\mathcal{G}^\infty$ materializes the semantic of $\mathcal{G}$.

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.

lgg in RDF

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

lgg in our SPARQL setting

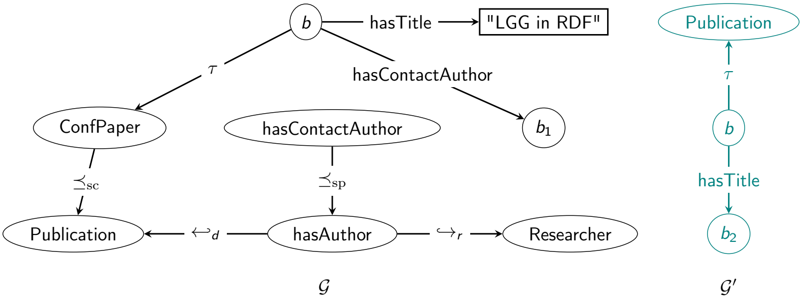
- descriptions are BGP Queries
- the generalization/specialization relation is entailment between BGPQs

Entailment between RDF graphs

$\mathcal{G} \models_{\mathcal{R}} \mathcal{G}' \iff \mathcal{G}^{\infty} \models \mathcal{G}'$

i.e., there exists a graph homomorphism from $\mathcal{G}'$ to $\mathcal{G}^{\infty}$.

$$\mathcal{G} \overset{?}{\models}_{\mathcal{R}} \mathcal{G}'$$

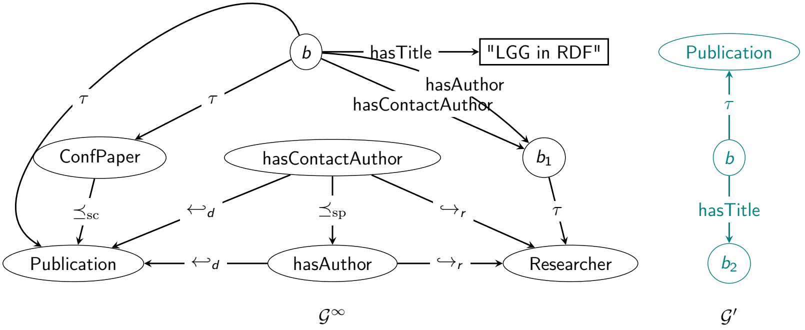


Entailment between RDF graphs

$\mathcal{G} \models_{\mathcal{R}} \mathcal{G}' \iff \mathcal{G}^{\infty} \models \mathcal{G}'$

i.e., there exists a graph homomorphism from $\mathcal{G}'$ to $\mathcal{G}^{\infty}$.

$$\mathcal{G} \models_{\mathcal{R}} \mathcal{G}' \equiv \mathcal{G}^{\infty} \overset{?}{\models} \mathcal{G}'$$

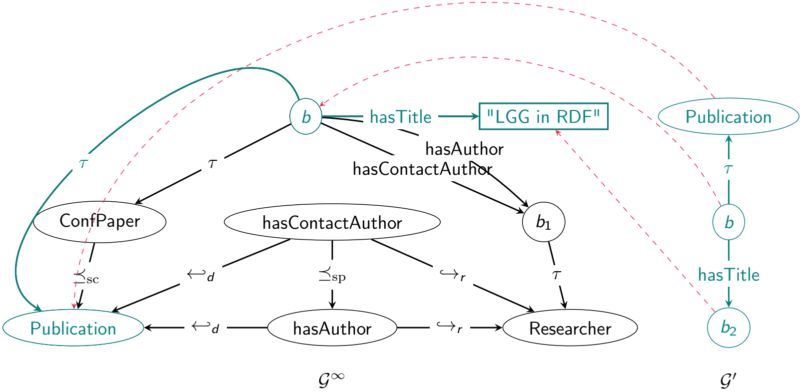


Entailment between RDF graphs

$\mathcal{G} \models_{\mathcal{R}} \mathcal{G}' \iff \mathcal{G}^{\infty} \models \mathcal{G}'$

i.e., there exists a graph homomorphism from $\mathcal{G}'$ to $\mathcal{G}^{\infty}$.

$\mathcal{G} \models_{\mathcal{R}} \mathcal{G}' \equiv \mathcal{G}^{\infty} \models \mathcal{G}'$



$\mathcal{G}$ is more specific than $\mathcal{G}'$!

Towards defining the notion of lgg in SPARQL

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

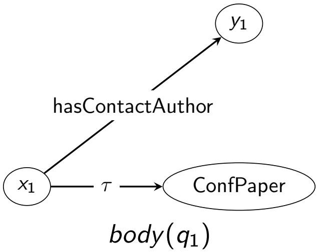
- descriptions are Basic Graph Pattern Queries (BGPQs)
- the generalization/specialization relation is entailment between BGPQs

Basic Graph Pattern Queries (BGPQs)

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

Basic Graph Pattern Queries (BGPQs)

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



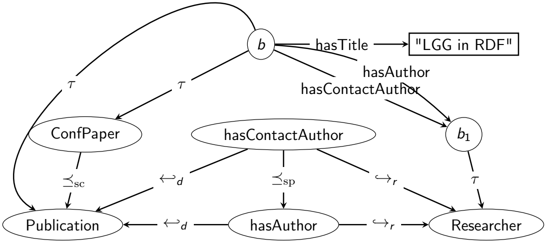
Sample BGPQ $q_1(x_1)$

Entailing and answering queries

Query entailment

$\mathcal{G} \models_{\mathcal{R}} q \iff \mathcal{G}^{\infty} \models q$

$q(x_1, x_2)$

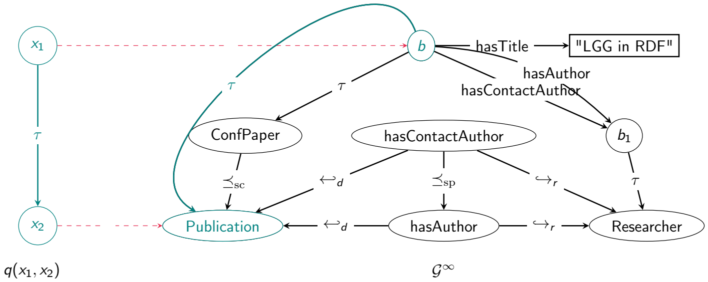


$\mathcal{G}^{\infty}$

Entailing and answering queries

Query entailment

$\mathcal{G} \models_{\mathcal{R}} q \iff \mathcal{G}^{\infty} \models q$

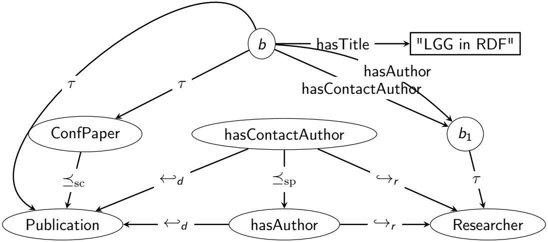


Entailing and answering queries

Query answering

$q(\mathcal{G}) = \{(\bar{x})_\phi \mid \mathcal{G} \models^\phi_\mathcal{R} q\}$

$q(x_1, x_2)$



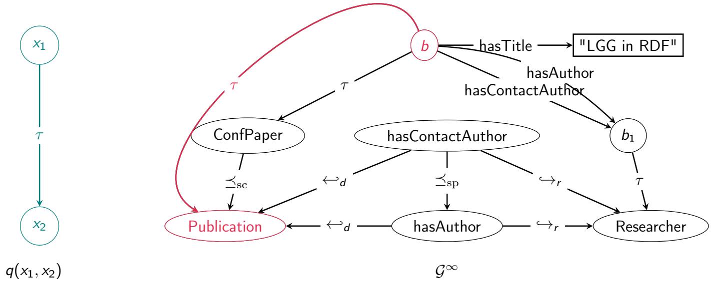
$\mathcal{G}^\infty$

$$q(\mathcal{G}) = \{(b, \mathrm{ConfPaper}), (b, \mathrm{Publication}), (b_1, \mathrm{Researcher})\}$$

Entailing and answering queries

Query answering

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

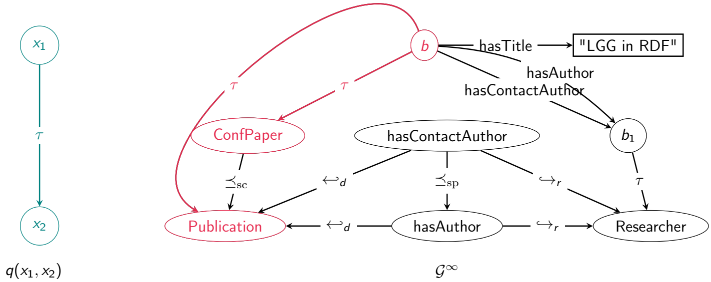


$$q(\mathcal{G}) = \{(b, \text{ConfPaper}), (b, \text{Publication}), (b_1, \text{Researcher})\}$$

Entailing and answering queries

Query answering

$q(\mathcal{G}) = \{(\bar{x})_\phi \mid \mathcal{G} \models^\phi_\mathcal{R} q\}$

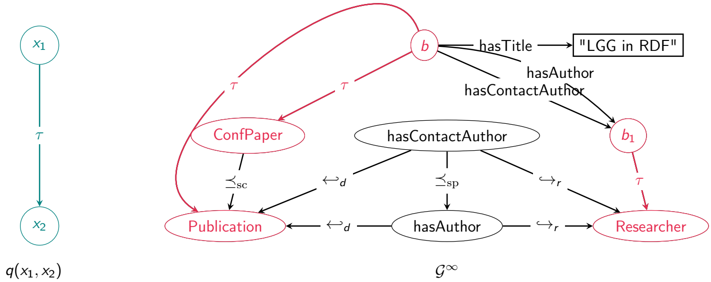


$$q(\mathcal{G}) = \{(b, \mathrm{ConfPaper}), (b, \mathrm{Publication}), (b_1, \mathrm{Researcher})\}$$

Entailing and answering queries

Query answering

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



$$q(\mathcal{G}) = \{(b, \text{ConfPaper}), (b, \text{Publication}), (b_1, \text{Researcher})\}$$

Towards defining the notion of lgg in SPARQL

G. Plotkin

A *least general generalization* (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

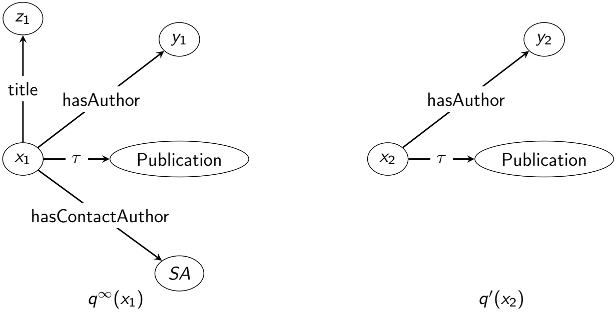
- descriptions are RDF graphs
- the generalization/specialization relation is entailment between RDF graphs

lgg in our SPARQL setting

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

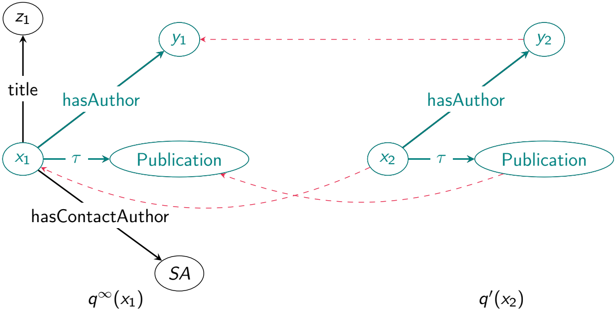
Entailment between BGPQs

$$q \models_{\mathcal{R}} q' \iff q^{\infty} \models q'$$



Entailment between BGPQs

$$q \models_{\mathcal{R}} q' \iff q^{\infty} \models q'$$



Outline

Defining the lgg of RDF graphs

Definition (lgg of RDF graphs)

Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be RDF graphs and $\mathcal{R}$ a set of RDF entailment rules.

- A *generalization* of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ is an RDF graph $\mathcal{G}_g$ such that $\mathcal{G}_i \models_{\mathcal{R}} \mathcal{G}_g$ holds for $1 \leq i \leq n$.
- A *least general generalization* (lgg) of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ is a generalization $\mathcal{G}_{\texttt{lgg}}$ of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ such that for any other generalization $\mathcal{G}_g$ of $\mathcal{G}_1, \ldots, \mathcal{G}_n$, $\mathcal{G}_{\texttt{lgg}} \models_{\mathcal{R}} \mathcal{G}_g$ holds.

Theorem

An lgg of RDF graphs always exists; it is *unique* up to entailment.

Defining the lgg of RDF graphs

Definition (lgg of RDF graphs)

Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be RDF graphs and $\mathcal{R}$ a set of RDF entailment rules.

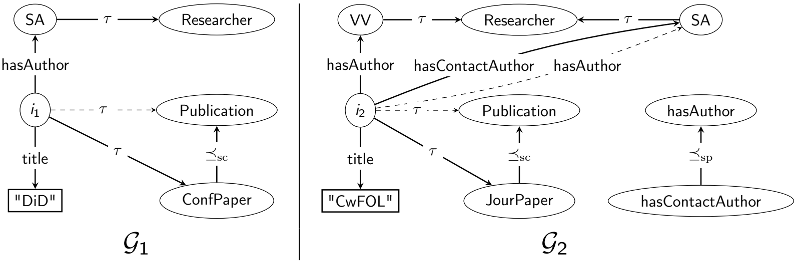
- A *generalization* of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ is an RDF graph $\mathcal{G}_g$ such that $\mathcal{G}_i \models_{\mathcal{R}} \mathcal{G}_g$ holds for $1 \leq i \leq n$.
- A *least general generalization* (lgg) of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ is a generalization $\mathcal{G}_{\texttt{lgg}}$ of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ such that for any other generalization $\mathcal{G}_g$ of $\mathcal{G}_1, \ldots, \mathcal{G}_n$, $\mathcal{G}_{\texttt{lgg}} \models_{\mathcal{R}} \mathcal{G}_g$ holds.

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

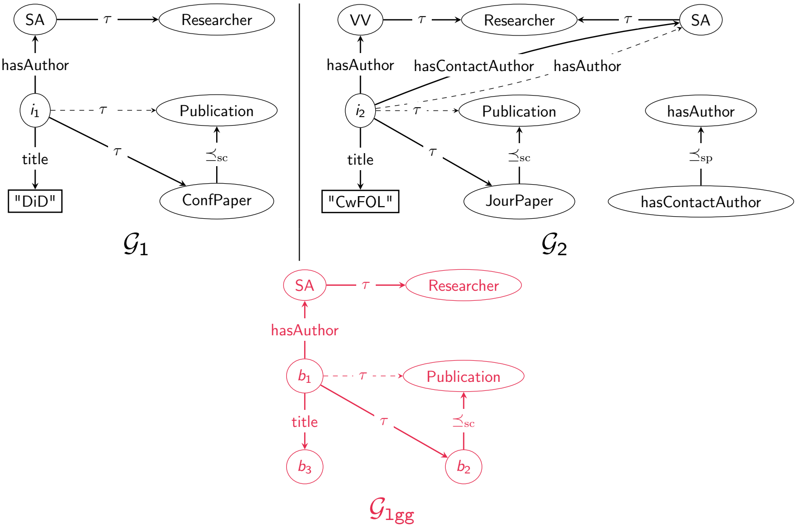
$$
\begin{aligned}
\ell_3(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3) &\equiv_{\mathcal{R}} \ell_2(\ell_2(\mathcal{G}_1, \mathcal{G}_2), \mathcal{G}_3) \\
\ldots & \quad \ldots \\
\ell_n(\mathcal{G}_1, \ldots, \mathcal{G}_n) &\equiv_{\mathcal{R}} \ell_2(\ell_{n-1}(\mathcal{G}_1, \ldots, \mathcal{G}_{n-1}), \mathcal{G}_n) \\
&\equiv_{\mathcal{R}} \ell_2(\ell_2(\cdots \ell_2(\ell_2(\mathcal{G}_1, \mathcal{G}_2), \mathcal{G}_3) \cdots, \mathcal{G}_{n-1}), \mathcal{G}_n)
\end{aligned}
$$

We focus on computing the lgg of two RDF graphs

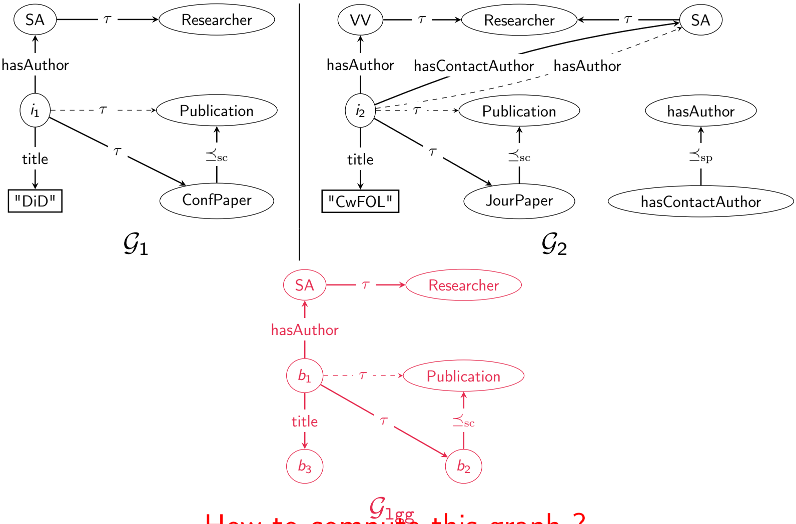
Defining the lgg of RDF graphs



Defining the lgg of RDF graphs



Defining the lgg of RDF graphs



How to compute this graph ?

The cover graph of RDF graphs

Definition (Cover graph)

The *cover graph* $\mathcal{G}$ of two RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ is the RDF graph such that for every property p in both $\mathcal{G}_1$ and $\mathcal{G}_2$:

$$(t_1, p, t_2) \in \mathcal{G}_1 \text{ and } (t_3, p, t_4) \in \mathcal{G}_2 \text{ iff } (\varsigma(t_1, t_3), p, \varsigma(t_2, t_4)) \in \mathcal{G}$$

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

The cover graph of RDF graphs

Definition (Cover graph)

The *cover graph* $\mathcal{G}$ of two RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ is the RDF graph such that for every property p in both $\mathcal{G}_1$ and $\mathcal{G}_2$:

$$(t_1, p, t_2) \in \mathcal{G}_1 \text{ and } (t_3, p, t_4) \in \mathcal{G}_2 \text{ iff } (\varsigma(t_1, t_3), p, \varsigma(t_2, t_4)) \in \mathcal{G}$$

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

Example (Anti-unification)

- $(i1, hasAuthor, SA) \in \mathcal{G}_1$ and $(i2, hasAuthor, SA) \in \mathcal{G}_2$ iff $(b_{i1i2}, hasAuthor, SA) \in \mathcal{G}$
- $(i1, hasAuthor, SA) \in \mathcal{G}_1$ and $(i2, hasContactAuthor, SA) \in \mathcal{G}_2$ but $(b_{i1i2}, b_{hAhCA}, SA) \notin \mathcal{G}$

Cover graph-based lgg

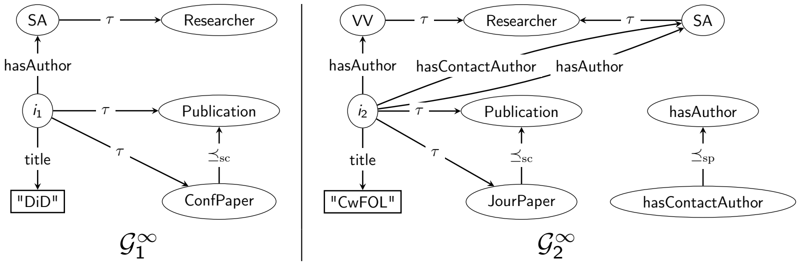
Theorem

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two RDF graphs, and $\mathcal{R}$ a set of RDF entailment rules. The *cover graph* $\mathcal{G}$ of $\mathcal{G}_1^\infty$ and $\mathcal{G}_2^\infty$ is an lgg of $\mathcal{G}_1$ and $\mathcal{G}_2$.

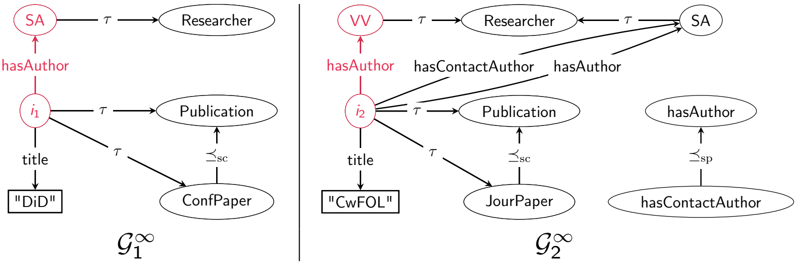
Proposition

An lgg of two RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be computed in $O(|\mathcal{G}_1^\infty| \times |\mathcal{G}_2^\infty|)$ and its size is bounded by $|\mathcal{G}_1^\infty| \times |\mathcal{G}_2^\infty|$.

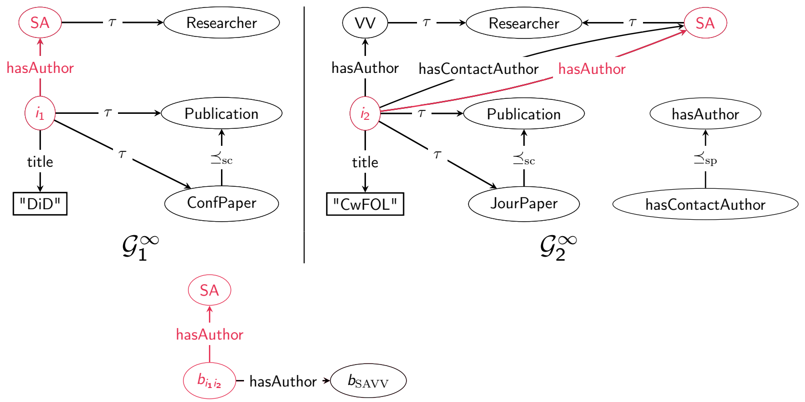
Cover graph-based lgg of RDF graphs



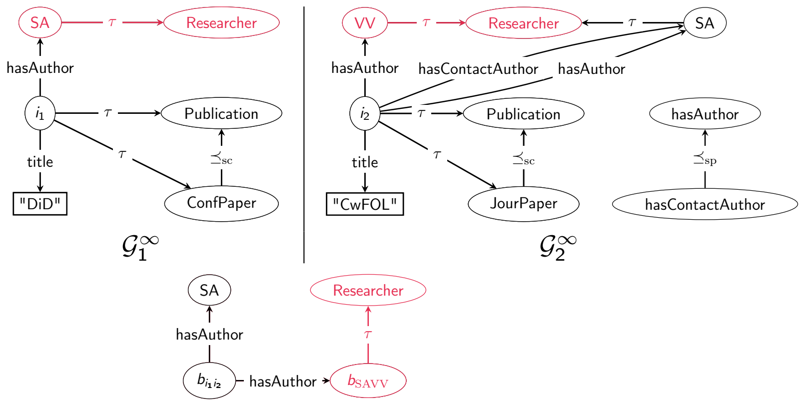
Cover graph-based lgg of RDF graphs



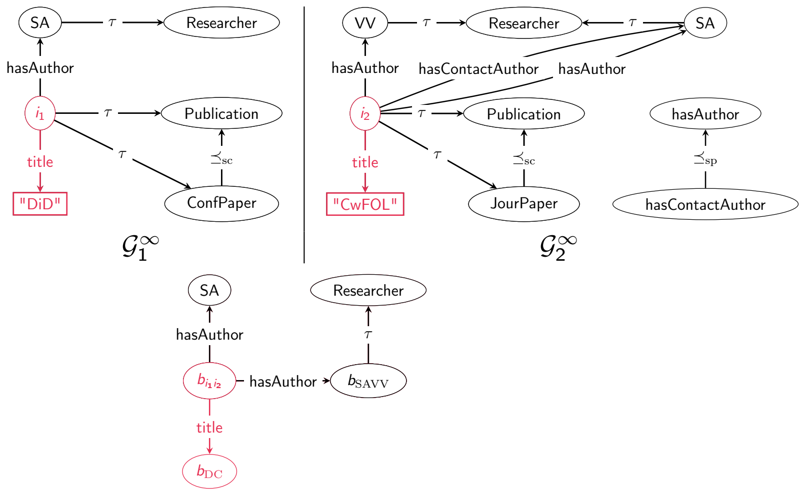
Cover graph-based lgg of RDF graphs



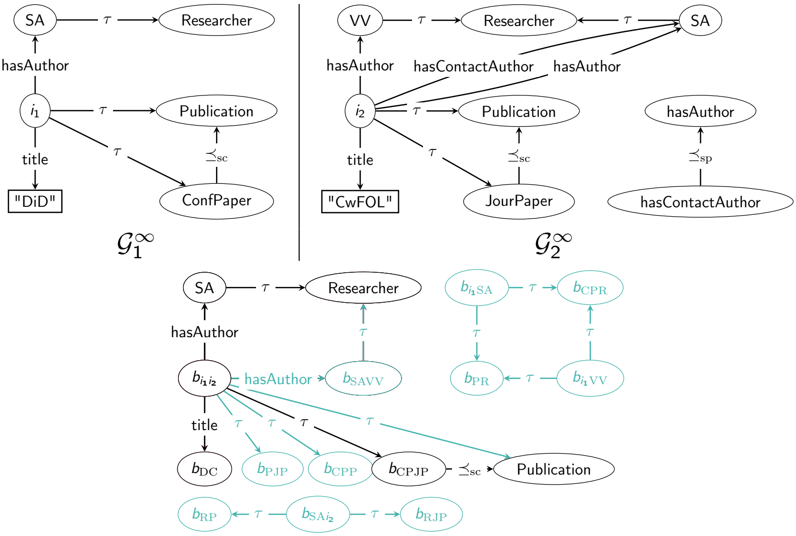
Cover graph-based `lgg` of RDF graphs



Cover graph-based lgg of RDF graphs



Cover graph-based lgg of RDF graphs



Outline

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.

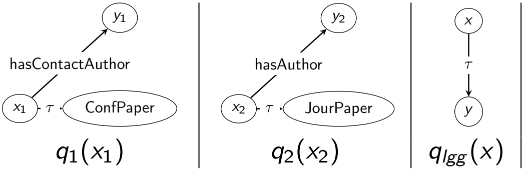
- A *generalization* of $q_1, \ldots, q_n$ is a BGPQ q_g such that $q_i \models_{\mathcal{R}} q_g$ for $1 \leq i \leq n$.
- A *least general generalization* of $q_1, \ldots, q_n$ is a generalization $q_{\texttt{lgg}}$ of $q_1, \ldots, q_n$ such that for any other generalization q_g of $q_1, \ldots, q_n$: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

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 $q_{\texttt{lgg}}$ of $q_1, \ldots, q_n$ such that for any other generalization q_g of $q_1, \ldots, q_n$: $q_{\texttt{lgg}} \models_{\mathcal{R}} q_g$.

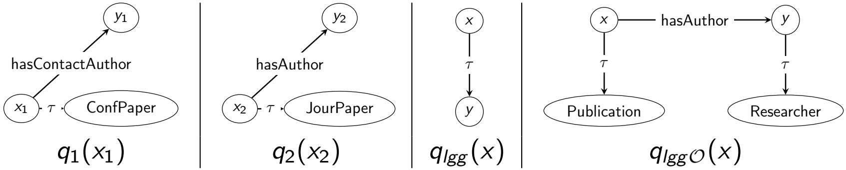


Defining the lgg of queries

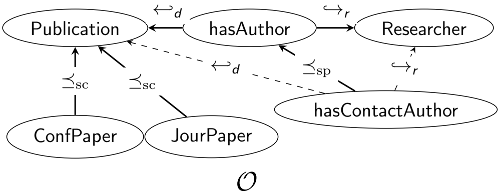
lgg of BGPQs

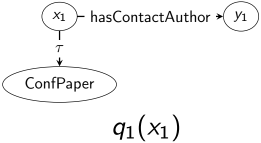
Let $q_1, \ldots, q_n$ be BGPQs with the same arity and $\mathcal{R}$ a set of RDF entailment rules.

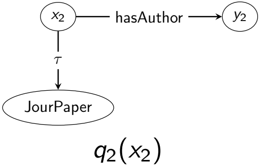
- A *generalization* of $q_1, \ldots, q_n$ is a BGPQ q_g such that $q_i \models_{\mathcal{R}} q_g$ for $1 \leq i \leq n$.
- A *least general generalization* of $q_1, \ldots, q_n$ is a generalization $q_{\mathtt{lgg}}$ of $q_1, \ldots, q_n$ such that for any other generalization q_g of $q_1, \ldots, q_n$: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



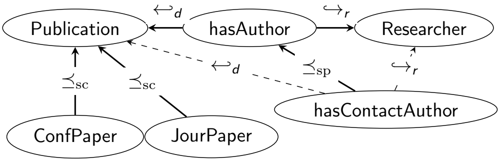
Enriching queries w.r.t. background knowledge



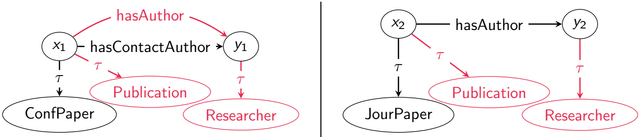




Enriching queries w.r.t. background knowledge



$\mathcal{O}$

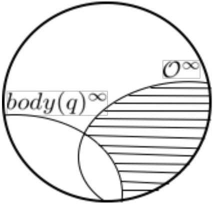


$q_1{}^{\infty}_{\mathcal{O}}(x_1)$ $q_2{}^{\infty}_{\mathcal{O}}(x_2)$

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 $q^{\infty}_{\mathcal{O}}$, is the BGPQ with the same answer variables as q and whose body, noted $body(q^{\infty}_{\mathcal{O}})$, is the maximal subset of $(body(q) \cup \mathcal{O})^{\infty}$ such that for any of its subset $\mathcal{S}$: if $\mathcal{O} \models_{\mathcal{R}} \mathcal{S}$ holds then $body(q) \models_{\mathcal{R}} \mathcal{S}$ holds.



$(body(q) \cup \mathcal{O})^{\infty}$

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 $q_1{}^{\infty}_{\mathcal{O}} \models q_2$ 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

for any graph $\mathcal{G}$ whose set of RDFS constraints is $\mathcal{O}$.

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 $q_{\texttt{lgg}}$ 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. $\mathcal{O}$: $q_{\texttt{lgg}} \models_{\mathcal{R},\mathcal{O}} q_g$.

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.

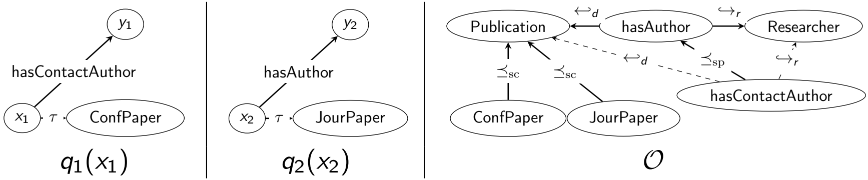
- A *generalization of* $q_1, \ldots, q_n$ *w.r.t.* $\mathcal{O}$ is a BGPQ q_g such that $q_i \models_{\mathcal{R},\mathcal{O}} q_g$ for $1 \leq i \leq n$.
- A *least general generalization of* $q_1, \ldots, q_n$ *w.r.t.* $\mathcal{O}$ is a generalization $q_{\texttt{lgg}}$ of $q_1, \ldots, q_n$ w.r.t.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

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

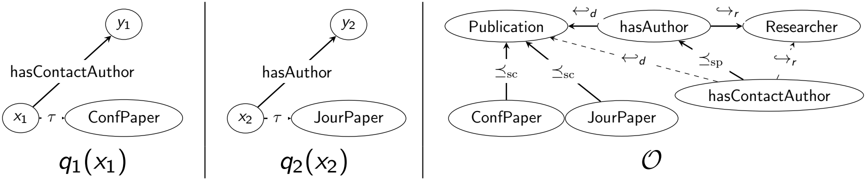
$$
\begin{aligned}
\ell_3(q_1, q_2, q_3) &\equiv_{\mathcal{R},\mathcal{O}} \ell_2(\ell_2(q_1, q_2), q_3) \\
\ldots & \quad \ldots \\
\ell_n(q_1, \ldots, q_n) &\equiv_{\mathcal{R},\mathcal{O}} \ell_2(\ell_{n-1}(q_1, \ldots, q_{n-1}), q_n) \\
&\equiv_{\mathcal{R},\mathcal{O}} \ell_2(\ell_2(\cdots \ell_2(\ell_2(q_1, q_2), q_3) \cdots, q_{n-1}), q_n)
\end{aligned}
$$

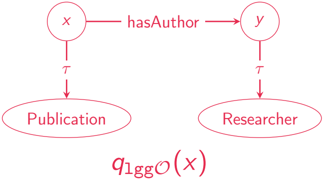
We focus on computing lgg of two BGPQ queries

Defining the lgg of queries

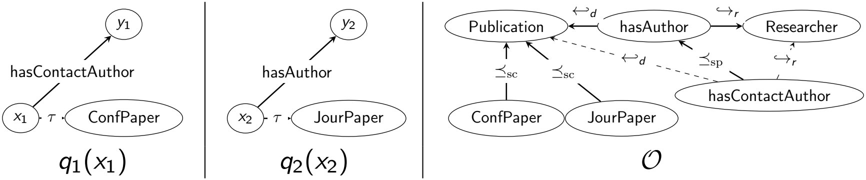


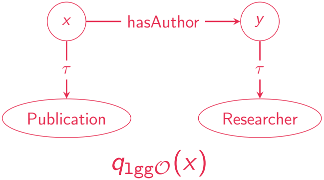
Defining the lgg of queries





Defining the lgg of queries





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

- $head(q_1) = q_1(x_1^1, \dots, x_1^n)$ and $head(q_2) = q_2(x_2^1, \dots, x_2^n)$ iff $head(q) = q(v_{x_1^1 x_2^1}, \dots, v_{x_1^n x_2^n})$
- $(t_1, t_2, t_3) \in body(q_1)$ and $(t_4, t_5, t_6) \in body(q_2)$ iff $(\varsigma(t_1, t_4), \varsigma(t_2, t_5), \varsigma(t_3, t_6)) \in body(q)$ 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

- $head(q_1) = q_1(x_1^1, \dots, x_1^n)$ and $head(q_2) = q_2(x_2^1, \dots, x_2^n)$ iff $head(q) = q(v_{x_1^1 x_2^1}, \dots, v_{x_1^n x_2^n})$
- $(t_1, t_2, t_3) \in body(q_1)$ and $(t_4, t_5, t_6) \in body(q_2)$ iff $(\varsigma(t_1, t_4), \varsigma(t_2, t_5), \varsigma(t_3, t_6)) \in body(q)$ 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.

Example

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

Cover query-based lgg

Theorem

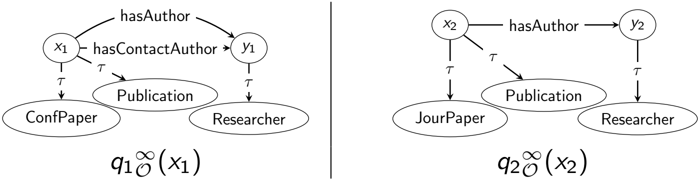
Given a set $\mathcal{R}$ of RDF entailment rules, a set $\mathcal{O}$ of RDFS statements and two BGPQs q_1, q_2 with the same arity,

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

Proposition

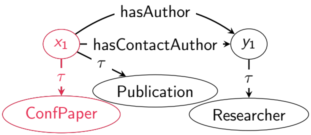
A cover query-based lgg of two BGPQs q_1 and q_2 is computed in $O(|body({q_1}^{\infty}_{\mathcal{O}})| \times |body({q_2}^{\infty}_{\mathcal{O}})|)$ and its size is $|body({q_1}^{\infty}_{\mathcal{O}})| \times |body({q_2}^{\infty}_{\mathcal{O}})|$.

Cover query-based lgg of SPARQL queries

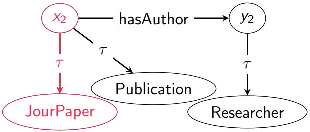


$$q(v_{x1x2})$$

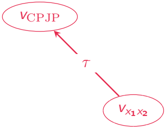
Cover query-based lgg of SPARQL queries



$q_1{}^{\infty}_{\mathcal{O}}(x_1)$

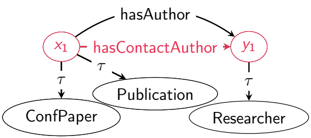


$q_2{}^{\infty}_{\mathcal{O}}(x_2)$

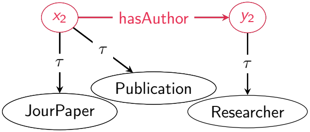


$q(v_{x1x2})$

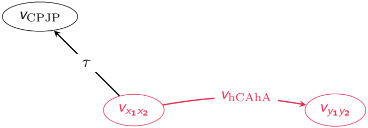
Cover query-based lgg of SPARQL queries



$q_{1\mathcal{O}}^{\infty}(x_1)$

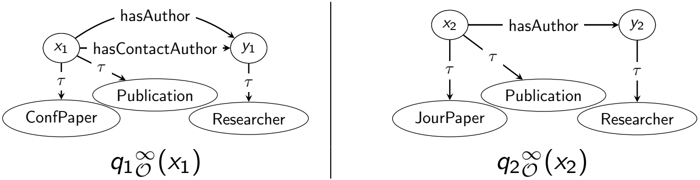


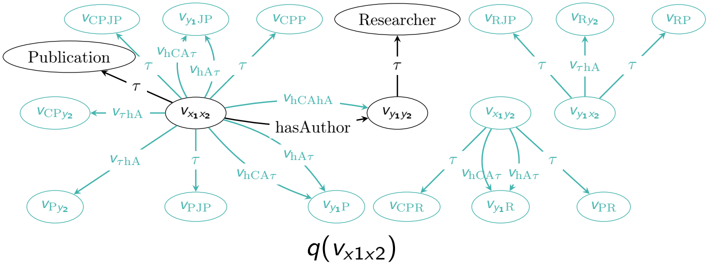
$q_{2\mathcal{O}}^{\infty}(x_2)$



$q(v_{x1x2})$

Cover query-based lgg of SPARQL queries





Experimentation: BGPQs (DBPedia)

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

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

Experimentation: BGPQs (DBPedia)

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

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

DBpedia query $Q_{1\leq i\leq 8}$:	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
Q_i's shape	tree	tree	tree	graph	graph	graph	graph	graph
$\lvert body(Q_i)\rvert$	4	6	4	6	4	6	6	6
Number of URI/variable occurrences in Q_i	7/5	9/9	5/7	7/11	5/7	9/9	9/9	9/9
$\lvert Q_i(\mathcal{G}_{\mathtt{DBpedia}})\rvert$	77	0	41 695	13	6	0	1	0
$\lvert body(Q_i{}^{\infty}_{\mathcal{O}_{\mathtt{DBpedia}}})\rvert$	16	19	19	23	16	23	23	23

Table: Characteristics of our test BGPQs (top) and of their saturations w.r.t. DBpedia constraints (bottom); times are in ms.

Experimentation: lgg of BGPQs (DBPedia)

lgg of 2 DBpedia BGPQs:	$Q_1 Q_2$	$Q_1 Q_3$	$Q_1 Q_4$	$Q_2 Q_3$	$Q_4 Q_5$	$Q_5 Q_6$	$Q_5 Q_7$	$Q_7 Q_8$
Time to compute $q_{\mathtt{lgg}}$	3	3	5	4	4	5	6	5
$\lvert q_{\mathtt{lgg}}(\mathcal{G}_{\mathtt{DBpedia}})\rvert$	477,455	34,747,102	34,901,117	34,747,102	1,977	1,221	35	70
Time to compute $q_{\mathtt{lgg}}^{\mathcal{O}_{\mathtt{DBpedia}}}$	13	14	14	15	15	14	17	18
$\lvert q_{\mathtt{lgg}}^{\mathcal{O}_{\mathtt{DBpedia}}}(\mathcal{G}_{\mathtt{DBpedia}})\rvert$	10,637	7,874,768	456,690	4,537,824	1,701	780	34	36
Gain in precision	97.77	77.33	98.69	86.94	13.96	36.11	2.85	48.57

Table: Characteristics of cover query-based lggs of test queries, w/ or w/o using the DBpedia RDFS constraints; times are in ms.

lgg of 3 DBpedia BGPQs :	$Q_1 Q_2 Q_3$	$Q_1 Q_2 Q_4$	$Q_1 Q_3 Q_4$	$Q_2 Q_3 Q_4$	$Q_4 Q_7 Q_8$	$Q_5 Q_7 Q_8$	$Q_6 Q_7 Q_8$
Time to compute $q_{\mathtt{lgg}}$	5	4	5	6	10	11	12
$\lvert q_{\mathtt{lgg}}(\mathcal{G}_{\mathtt{DBpedia}})\rvert$	34,747,102	34,901,117	34,901,117	34,901,117	70	1,977	4,969
Time to compute $q_{\mathtt{lgg}}^{\mathcal{O}_{\mathtt{DBpedia}}}$	19	20	20	24	27	27	33
$\lvert q_{\mathtt{lgg}}^{\mathcal{O}_{\mathtt{DBpedia}}}(\mathcal{G}_{\mathtt{DBpedia}})\rvert$	7,874,768	615,339	7,874,779	4,537,824	36	1,701	335
Gain in precision	77.33	98.23	77.43	86.99	48.57	13.96	93.25

Table: Characteristics of cover query-based lggs of 3 test queries, w/ or w/o using the DBpedia RDFS constraints; times are in ms.

Outline

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 `lggs` 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 lggs of RDF graphs and entailment rules plus external ontology when learning lggs 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*$_{\mathcal{R}}$ which underpins the OWL2 QL profile of the *Web Ontology Language*, the other Semantic Web data model by W3C.

Thank you !

References I

[Baader et al., 1999] Baader, F., Küsters, R., and Molitor, R. (1999).
Computing least common subsumers in description logics with existential restrictions.
In *IJCAI*.

[Chein and Mugnier, 2009] Chein, M. and Mugnier, M. (2009).
Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs.
Springer.

[Colucci et al., 2016] Colucci, S., Donini, F., Giannini, S., and Sciascio, E. D. (2016).
Defining and computing least common subsumers in RDF.
J. Web Semantics, 39(0).

[Colucci et al., 2013] Colucci, S., Donini, F. M., and Sciascio, E. D. (2013).
Common subsumbers in RDF.
In *AI*IA*.

[Lehmann and Bühmann, 2011] Lehmann, J. and Bühmann, L. (2011).
Autosparql: Let users query your knowledge base.
In *ESWC*.

[Nienhuys-Cheng and de Wolf, 1996] Nienhuys-Cheng, S. and de Wolf, R. (1996).
Least generalizations and greatest specializations of sets of clauses.
J. Artif. Intell. Res.

[Plotkin, 1970] Plotkin, G. D. (1970).
A note on inductive generalization.
Machine Intelligence, 5.

[W3C-RDFS, 2014] W3C-RDFS (2014).
RDF 1.1 semantics.
https://www.w3.org/TR/rdf11-mt/.

[Zarrieß and Turhan, 2013] Zarrieß, B. and Turhan, A. (2013).
Most specific generalizations w.r.t. general EL-TBoxes.
In *IJCAI*.