



ENSSAT
LANNION



Learning Commonalities in RDF

Sara El Hassad François Goasdoué Hélène Jaudoin

IRISA, Univ. Rennes 1, Lannion, France

ILP 2017 - 5th September 2017

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

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

- Social context : lgg of users descriptions (profiles)
- Research of common graph patterns between datasets
- Linked Data Cloud : links between datasets

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

- Social context : lgg of users descriptions (profiles)
- Research of common graph patterns between datasets
- Linked Data Cloud : links between datasets

Goal

To study the problem in the setting of the *entire* RDF standard

Outline

Introduction

The Resource Description Framework

Finding commonalities between RDF graphs

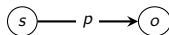
Related work

Conclusion

RDF graphs

- Specification of RDF graphs with triples :

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



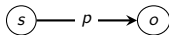
- Built-in property URIs to state RDF statements

RDF statement	Triple
Class assertion	$(s, \text{rdf:type}, o)$
Property assertion	(s, p, o) with $p \neq \text{rdf:type}$

RDF graphs

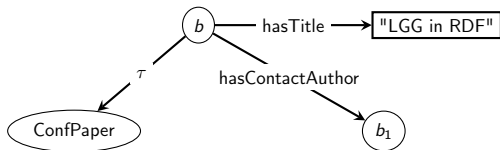
- Specification of RDF graphs with triples :

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



- Built-in property URLs to state RDF statements

RDF statement	Triple
Class assertion	$(s, \text{rdf:type}, o)$
Property assertion	(s, p, o) with $p \neq \text{rdf:type}$



Adding ontological knowledge to RDF graphs

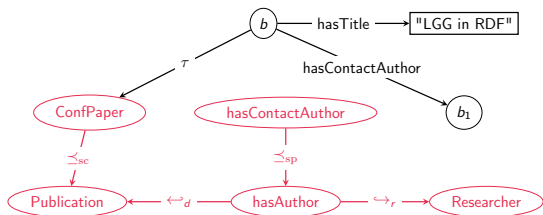
- Built-in property URLs to state RDF Schema statements, i.e., ontological constraints.

RDFS statement	Triple
Subclass	(s, \preceq_{sc}, o)
Subproperty	(s, \preceq_{sp}, o)
Domain typing	(s, \leftarrow_d, o)
Range typing	(s, \rightarrow_r, o)

Adding ontological knowledge to RDF graphs

- Built-in property URLs to state RDF Schema statements, i.e., ontological constraints.

RDFS statement	Triple
Subclass	(s, \preceq_{sc}, o)
Subproperty	(s, \preceq_{sp}, o)
Domain typing	(s, \leftarrow_d, o)
Range typing	(s, \rightarrow_r, o)



Deriving the implicit triples

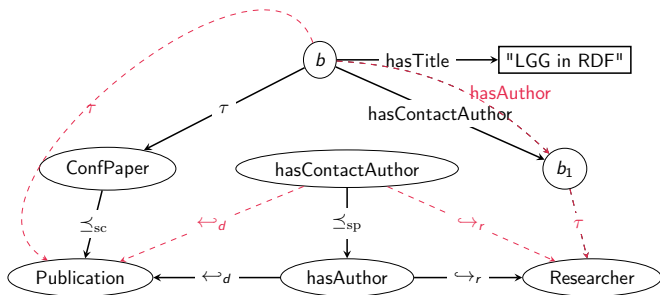


Figure: RDF graph \mathcal{G}

How to derive implicit triples of an RDF graph ?

Sample set of entailment rules

Rule [1]	Entailment rule
rdfs2	$(p, \leftrightarrow_d, o), (s_1, p, o_1) \rightarrow (s_1, \tau, o)$
rdfs3	$(p, \leftrightarrow_r, o), (s_1, p, o_1) \rightarrow (o_1, \tau, o)$
rdfs5	$(p_1, \preceq_{sp}, p_2), (p_2, \preceq_{sp}, p_3) \rightarrow (p_1, \preceq_{sp}, p_3)$
rdfs7	$(p_1, \preceq_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o)$
rdfs9	$(s, \preceq_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o)$
rdfs11	$(s, \preceq_{sc}, o), (o, \preceq_{sc}, o_1) \rightarrow (s, \preceq_{sc}, o_1)$
ext1	$(p, \leftrightarrow_d, o), (o, \preceq_{sc}, o_1) \rightarrow (p, \leftrightarrow_d, o_1)$
ext2	$(p, \leftrightarrow_r, o), (o, \preceq_{sc}, o_1) \rightarrow (p, \leftrightarrow_r, o_1)$
ext3	$(p, \preceq_{sp}, p_1), (p_1, \leftrightarrow_d, o) \rightarrow (p, \leftrightarrow_d, o)$
ext4	$(p, \preceq_{sp}, p_1), (p_1, \leftrightarrow_r, o) \rightarrow (p, \leftrightarrow_r, o)$

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

Semantics of RDF graphs

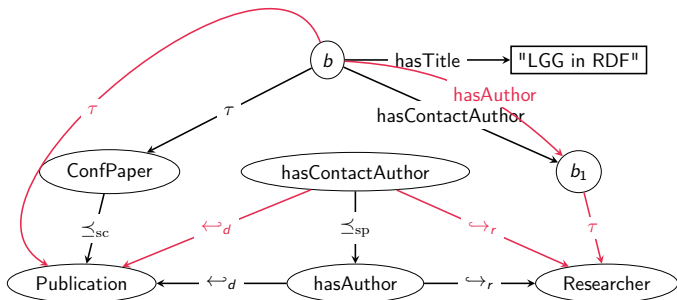


Figure: Saturated RDF graph \mathcal{G}^∞

Entailment between RDF graphs

Let \mathcal{G} and \mathcal{G}' be two graphs RDF and \mathcal{R} a set of RDF entailment rules. There exists a relationship to compare \mathcal{G} and \mathcal{G}' called *entailment between graphs*

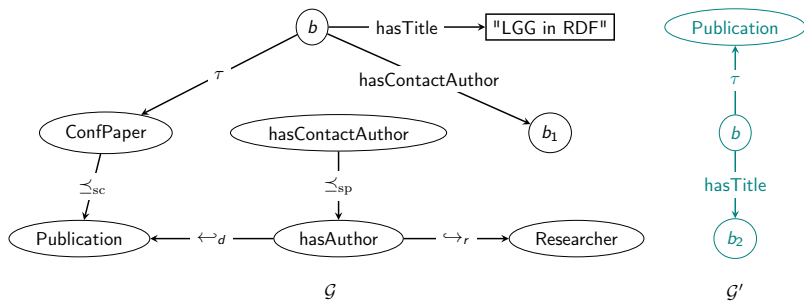
\mathcal{G} is more specific than \mathcal{G}' :

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

There must exist an embedding of \mathcal{G}' in \mathcal{G}^{∞} .

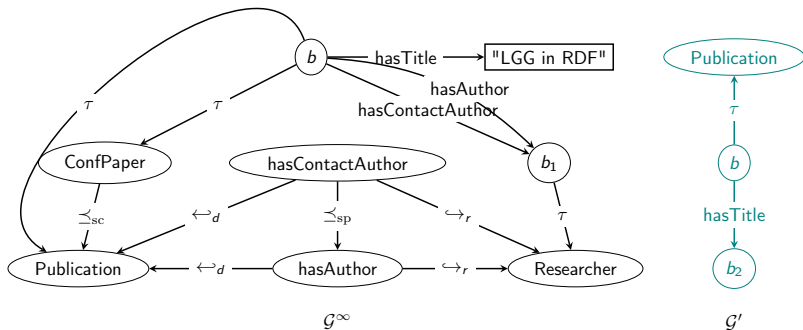
Entailment between RDF graphs

$$\mathcal{G} \stackrel{?}{\models_{\mathcal{R}}} \mathcal{G}'$$



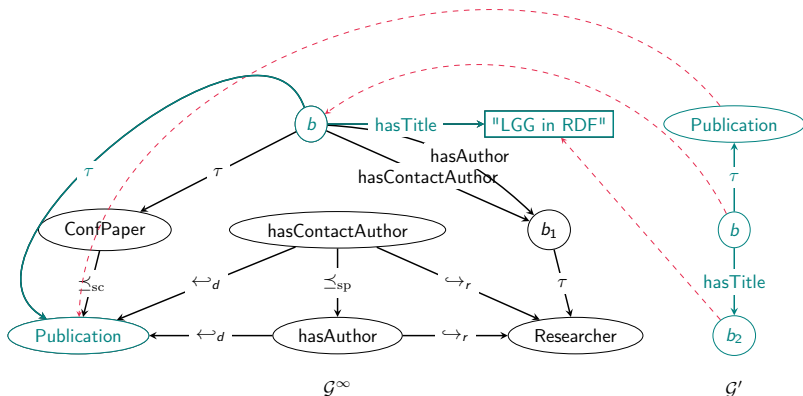
Entailment between RDF graphs

$$g^\infty \stackrel{?}{\models} g'$$



Entailment between RDF graphs

$$G^\infty \stackrel{?}{\models} G'$$



RDF graph G is more specific than RDF graph G'

Outline

Introduction

The Resource Description Framework

Finding commonalities between RDF graphs

Related work

Conclusion

Towards an lgg definition in RDF

A *least general generalization* (lgg) of n descriptions d_1, \dots, 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).

lgg in RDF

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

Defining the lgg of RDF graphs

Definition (lgg of RDF graphs)

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

- A *generalization* of $\mathcal{G}_1, \dots, \mathcal{G}_n$ 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 $\mathcal{G}_1, \dots, \mathcal{G}_n$ is a generalization \mathcal{G}_{lgg} of $\mathcal{G}_1, \dots, \mathcal{G}_n$ such that for any other generalization \mathcal{G}_g of $\mathcal{G}_1, \dots, \mathcal{G}_n$, $\mathcal{G}_{\text{lgg}} \models_{\mathcal{R}} \mathcal{G}_g$ holds.

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

$$\ell_3(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3) \equiv_{\mathcal{R}} \ell_2(\ell_2(\mathcal{G}_1, \mathcal{G}_2), \mathcal{G}_3)$$

...

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

Defining the lgg of RDF graphs

Definition (lgg of RDF graphs)

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

- A *generalization* of $\mathcal{G}_1, \dots, \mathcal{G}_n$ 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 $\mathcal{G}_1, \dots, \mathcal{G}_n$ is a generalization \mathcal{G}_{lgg} of $\mathcal{G}_1, \dots, \mathcal{G}_n$ such that for any other generalization \mathcal{G}_g of $\mathcal{G}_1, \dots, \mathcal{G}_n$, $\mathcal{G}_{\text{lgg}} \models_{\mathcal{R}} \mathcal{G}_g$ holds.

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

$$l_3(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3) \equiv_{\mathcal{R}} l_2(l_2(\mathcal{G}_1, \mathcal{G}_2), \mathcal{G}_3)$$

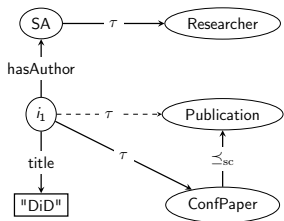
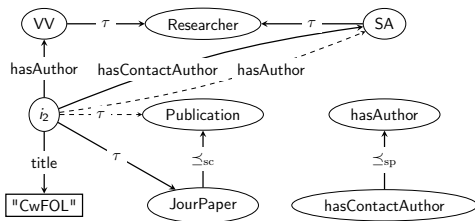
...

$$l_n(\mathcal{G}_1, \dots, \mathcal{G}_n) \equiv_{\mathcal{R}} l_2(l_{n-1}(\mathcal{G}_1, \dots, \mathcal{G}_{n-1}), \mathcal{G}_n)$$

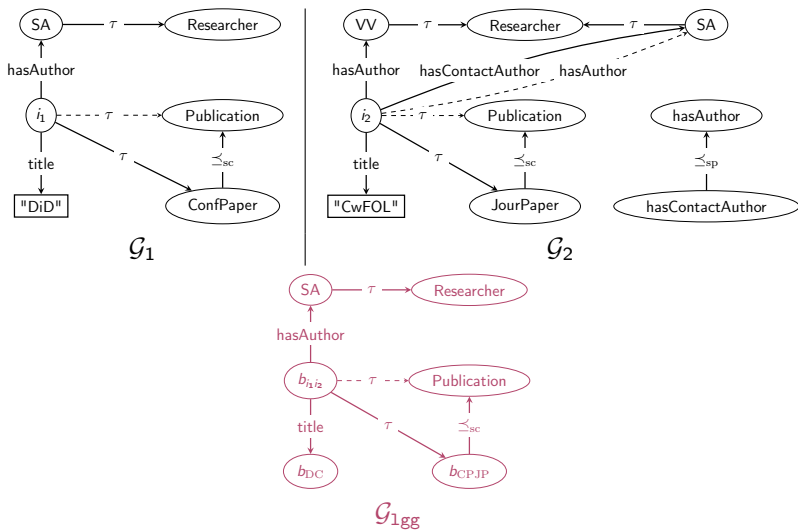
$$\equiv_{\mathcal{R}} l_2(l_2(\dots l_2(l_2(\mathcal{G}_1, \mathcal{G}_2), \mathcal{G}_3) \dots), \mathcal{G}_{n-1}), \mathcal{G}_n)$$

We focus on computing lgg of two RDF graphs

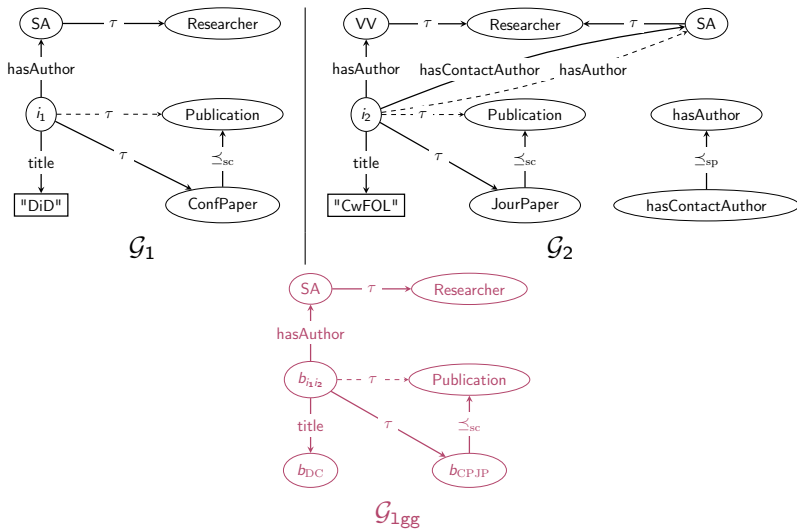
Defining the lgg of RDF graphs

 \mathcal{G}_1  \mathcal{G}_2

Defining the 1gg of RDF graphs



Defining the 1_{gg} of RDF graphs



How to compute this graph ?

The cover graph of RDF graphs

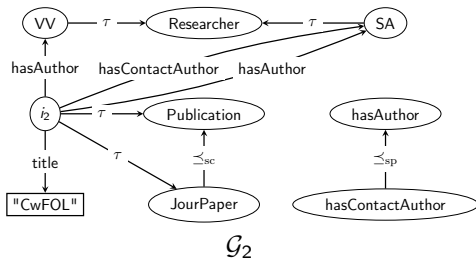
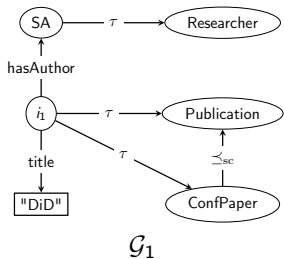
Definition (Cover graph)

The *cover graph* \mathcal{G} of two RDF graph \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 :

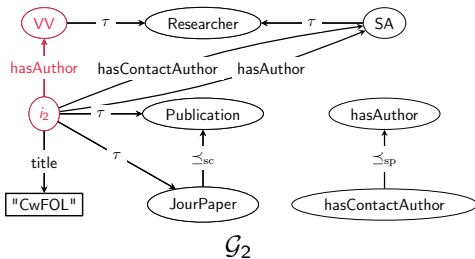
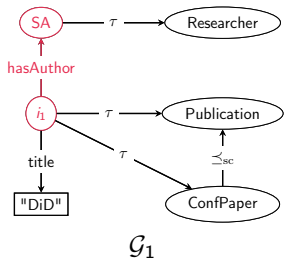
$$(t_1, p, t_2) \in \mathcal{G}_1 \text{ and } (t_3, p, t_4) \in \mathcal{G}_2 \text{ iff } (t_5, p, t_6) \in \mathcal{G}$$

with $t_5 = t_1$ if $t_1 = t_3$ and $t_1 \in \mathcal{U} \cup \mathcal{L}$, else t_5 is the blank node $b_{t_1 t_3}$, and, similarly $t_6 = t_2$ if $t_2 = t_4$ and $t_2 \in \mathcal{U} \cup \mathcal{L}$, else t_6 is the blank node $b_{t_2 t_4}$.

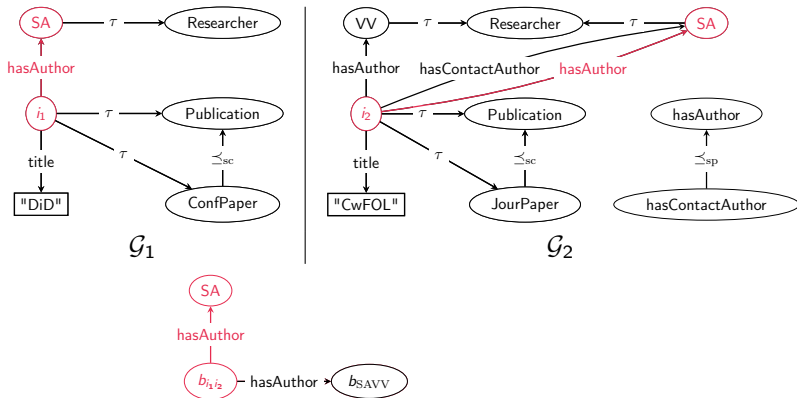
The cover graph of RDF graphs



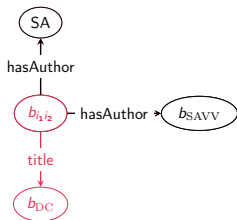
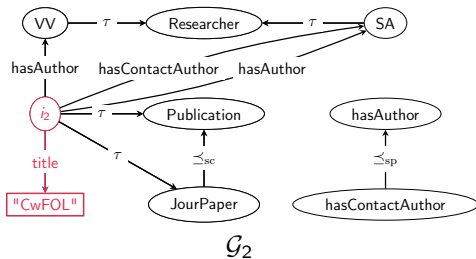
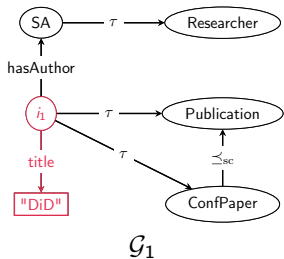
The cover graph of RDF graphs



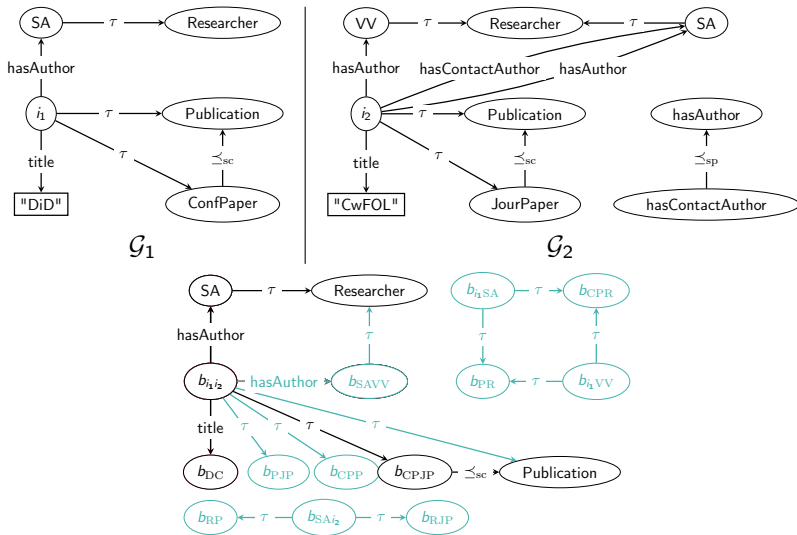
The cover graph of RDF graphs



The cover graph of RDF graphs



The cover graph of RDF graphs



Cover graph vs 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^∞ and \mathcal{G}_2^∞ is an lgg of \mathcal{G}_1 and \mathcal{G}_2 .

Cover graph vs 1gg

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^∞ and \mathcal{G}_2^∞ is an 1gg of \mathcal{G}_1 and \mathcal{G}_2 .

Corollary

An 1gg 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 $|\mathcal{G}_1^\infty| \times |\mathcal{G}_2^\infty|$.

Outline

Introduction

The Resource Description Framework

Finding commonalities between RDF graphs

Related work

Conclusion

Related work

Structural approaches

- Description Logics \mathcal{EL}
 - F. Baader and al. : *Computing least common subsumers in description logics with existential restrictions*. In IJCAI, 1999.
 - B. ZarrieB and al. : *Most specific generalizations w.r.t. general EL-TBoxes*. In IJCAI, 2013.
- RDF
 - SPARQL : tree queries
 - J. Lehmann and L. Buhmann. *Autosparql : Let users query your knowledge base*. In ESWC, 2011.
 - Rooted graphs, ignore RDF entailment :
 - S. Colucci and al. : *Defining and computing least common subsumers in RDF*. J. Web Semantics, 39(0), 2016.

Approaches independent of the structure

- Conceptual Graphs
 - M. Chein and M. Mugnier. *Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs*. Springer, 2009.
- Machine Learning
 - Gordon D. Plotkin. *A Note on Inductive Generalization*. Machine Intelligence, 1970.

Conclusion

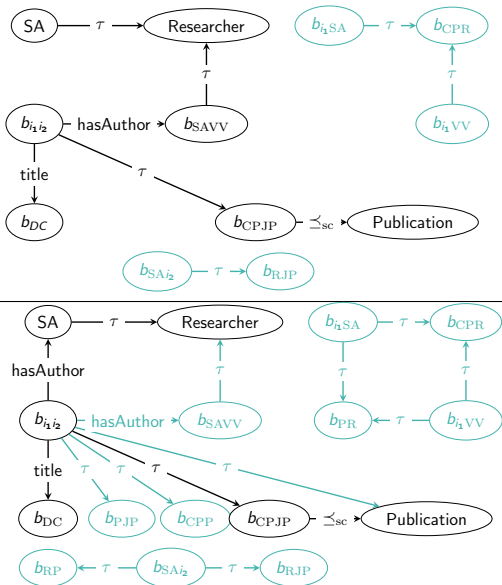
- To revisit the problem of computing a least general generalization in the entire setting of RDF.
- Algorithms to compute lggs of small-to-huge RDF graphs.
 - Memory
 - Data management system
 - MapReduce
- Perspective : Heuristics in order to compute lgg without redundant triples.

Thank you !



Questions ?

Cover graph with/without \mathcal{R} I



References I

- [1] **RDF 1.1 semantics.**
<https://www.w3.org/TR/rdf11-mt/>.