Towards Learning Commonalities in SPARQL

Sara El Hassad, François Goasdoué, Hélène Jaudoin

Towards Learning Commonalities in SPARQL

Sara El Hassad, François Goasdoué, and Hélène Jaudoin

IRISA, Univ. Rennes 1, Lannion, France
{sara.el-hassad,fg,helene.jaudoin}@irisa.fr

Abstract. Finding the commonalities between descriptions of data or knowledge is a foundational reasoning problem of Machine Learning, which amounts to computing a *least general generalization (*`lgg`*)* of such descriptions. We revisit this old problem in the popular conjunctive fragment of SPARQL, a.k.a. Basic Graph Pattern Queries (BGPQs). In particular, we define this problem in all its generality by considering general BGPQs, while the literature considers unary tree-shaped BGPQs only. Further, when *ontological knowledge* is available as RDF Schema constraints, we take advantage of it to devise much more pregnant `lgg`s.

Keywords: BGP queries, RDF, RDFS, least general generalization

1 Introduction

Finding commonalities between descriptions of data and knowledge is a fundamental Machine Learning problem. It was formalized in early 70's as computing a *least general generalization* (`lgg`) of First Order Logic formulae [4].

We revisit this old reasoning problem in the setting of SPARQL, the RDF query language by W3C, which may have varied theoretical and practical applications. For instance, an `lgg` of queries is a best upper approximation thereof by a single query in *knowledge approximation*, is the largest set of commonalities that may be recommended for view materialization or shared processing in *query optimization*, or may help recommending users to each other, especially in a social context, if what they ask for is enough related in *recommendation*, etc.

Our contribution is to carefully study and define a pregnant notion of `lgg` for the well-established conjunctive fragment of SPARQL, a.k.a. Basic Graph Pattern Queries (BGPQs). Our results significantly depart from the literature by considering *general* BGPQs, instead of *unary tree-shaped* BGPQs [1, 3], and crucially by taking advantage of *ontological knowledge* formalized as RDF Schema constraints, when available. Proofs for this paper's claims are delegated to [2].

2 Preliminaries

The RDF data model allows specifying *RDF graphs*, which are sets of *well-formed triples* from $(\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$ with $\mathcal{U}$, $\mathcal{B}$, $\mathcal{L}$ pairwise disjoint sets of URIs, of blank nodes (unknown values) and of literals (constants) respectively [5]. A triple (s, p, o) states that its *subject* s has *property* p whose value is the *object* o. Importantly, the RDF standard provides built-in property URIs to state facts for classes (unary relations) and properties (binary relations), called *RDF statements*, and ontological constraints relating classes and

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

RDFS statement	Triple
Subclass	$(\mathtt{s}, \text{rdfs:subClassOf}, \mathtt{o})$
Subproperty	$(\mathtt{s}, \text{rdfs:subPropertyOf}, \mathtt{o})$
Domain typing	$(\mathtt{s}, \text{rdfs:domain}, \mathtt{o})$
Range typing	$(\mathtt{s}, \text{rdfs:range}, \mathtt{o})$

Table 1. RDF & RDFS statements.

Rule [6]	Entailment rule
rdfs2	$(\mathtt{p}, \hookleftarrow_d, \mathtt{o}), (\mathtt{s}_1, \mathtt{p}, \mathtt{o}_1) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$
rdfs3	$(\mathtt{p}, \hookrightarrow_r, \mathtt{o}), (\mathtt{s}_1, \mathtt{p}, \mathtt{o}_1) \rightarrow (\mathtt{o}_1, \tau, \mathtt{o})$
rdfs5	$(\mathtt{p}_1, \preceq_{\text{sp}}, \mathtt{p}_2), (\mathtt{p}_2, \preceq_{\text{sp}}, \mathtt{p}_3) \rightarrow (\mathtt{p}_1, \preceq_{\text{sp}}, \mathtt{p}_3)$
rdfs7	$(\mathtt{p}_1, \preceq_{\text{sp}}, \mathtt{p}_2), (\mathtt{s}, \mathtt{p}_1, \mathtt{o}) \rightarrow (\mathtt{s}, \mathtt{p}_2, \mathtt{o})$
rdfs9	$(\mathtt{s}, \preceq_{\text{sc}}, \mathtt{o}), (\mathtt{s}_1, \tau, \mathtt{s}) \rightarrow (\mathtt{s}_1, \tau, \mathtt{o})$
rdfs11	$(\mathtt{s}, \preceq_{\text{sc}}, \mathtt{o}), (\mathtt{o}, \preceq_{\text{sc}}, \mathtt{o}_1) \rightarrow (\mathtt{s}, \preceq_{\text{sc}}, \mathtt{o}_1)$
ext1	$(\mathtt{p}, \hookleftarrow_d, \mathtt{o}), (\mathtt{o}, \preceq_{\text{sc}}, \mathtt{o}_1) \rightarrow (\mathtt{p}, \hookleftarrow_d, \mathtt{o}_1)$
ext2	$(\mathtt{p}, \hookrightarrow_r, \mathtt{o}), (\mathtt{o}, \preceq_{\text{sc}}, \mathtt{o}_1) \rightarrow (\mathtt{p}, \hookrightarrow_r, \mathtt{o}_1)$
ext3	$(\mathtt{p}, \preceq_{\text{sp}}, \mathtt{p}_1), (\mathtt{p}_1, \hookleftarrow_d, \mathtt{o}) \rightarrow (\mathtt{p}, \hookleftarrow_d, \mathtt{o})$
ext4	$(\mathtt{p}, \preceq_{\text{sp}}, \mathtt{p}_1), (\mathtt{p}_1, \hookrightarrow_r, \mathtt{o}) \rightarrow (\mathtt{p}, \hookrightarrow_r, \mathtt{o})$

Table 2. Sample RDF entailment rules.

properties, called *RDF Schema (RDFS) statements*, as shown in Table 1. Hereafter, we use the shorthands τ, $\preceq_{\text{sc}}$, $\preceq_{\text{sp}}$, $\hookleftarrow_d$ and $\hookrightarrow_r$ for the built-in property URIs rdf:type, rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain and rdfs:range respectively. The semantics of an RDF graph $\mathcal{G}$ is its *saturation* (a.k.a. *closure*), denoted $\mathcal{G}^\infty$, defined as the set of $\mathcal{G}$ triples together with all the *implicit* triples that can be derived from them and entailment rules from the RDF standard. Table 2 shows some rules that use RDFS constraints to derive implicit facts and constraints.

The *Basic Graph Pattern Queries (BGPQs)* form the conjunctive (or select-project-join) fragment of SPARQL. A BGPQ is of the form $q(\bar{x}) \leftarrow t_1, \ldots, t_\alpha$, where $\{t_1, \ldots, t_\alpha\}$ is a subset of $(\mathcal{U} \cup \mathcal{B} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L} \cup \mathcal{V})$ with $\mathcal{V}$ a set of variables pairwise disjoint with $\mathcal{U}, \mathcal{B}, \mathcal{L}$, and $\bar{x}$ is a subset of the variables occurring in $t_1, \ldots, t_\alpha$ called *answer variables*; for boolean queries, $\bar{x}$ is empty. The head of q is $head(q) = q(\bar{x})$ and the body of q is $body(q) = \{t_1, \ldots, t_\alpha\}$.

Two standard reasoning tasks characterize how RDF graphs contribute to queries. *Query entailment* indicates if an RDF graph holds some answer(s) to a query. Given a BGPQ q, an RDF graph $\mathcal{G}$ and a set $\mathcal{R}$ of RDF entailment rules, $\mathcal{G}$ *entails* q, noted $\mathcal{G} \models_{\mathcal{R}} q$, iff $\mathcal{G} \models_{\mathcal{R}} body(q)$ holds, i.e., there exists a homomorphism ϕ from q's variables and blank nodes to $\mathcal{G}^\infty$'s values (URIs, literals and blank nodes) such that $[body(q)]_\phi \subseteq \mathcal{G}^\infty$. Importantly, $\mathcal{G} \models_{\mathcal{R}} q$ holds iff $\mathcal{G}^\infty \models_\emptyset q$ holds. We note $\mathcal{G} \models^\phi_{\mathcal{R}} q$ the entailment $\mathcal{G} \models_{\mathcal{R}} q$ due to the homomorphism ϕ. *Query answering* identifies *all* the answers to a query that an RDF graph holds. Given a BGPQ q with head $q(\bar{x})$, the *answer set of* q *against* $\mathcal{G}$ is $q(\mathcal{G}) = \{(\bar{x})_\phi \mid \mathcal{G} \models^\phi_{\mathcal{R}} body(q)\}$ where we denote by $(\bar{x})_\phi$ the tuple of $\mathcal{G}^\infty$ values obtained by replacing every answer variable $x_i \in \bar{x}$ by its image $\phi(x_i)$.

Finally, queries can be compared through the generalization/specialization relationship of *entailment between queries*, which is the obvious adaptation of query entailment to the presence of variables in queries. Given two BGPQs q, q' with *same* arity, whose heads are $q(\bar{x})$ and $q'(\bar{x}')$, and a set $\mathcal{R}$ of RDF entailment rules at hand, q *entails* q', noted $q \models_{\mathcal{R}} q'$, iff $body(q) \models^\phi_{\mathcal{R}} body(q')$ *with* $(\bar{x}')_\phi = \bar{x}$.

3 Least General Generalization of BGPQs

A *least general generalization* (lgg) of two[1] descriptions d_1, d_2 is a most specific description d generalizing d_1, d_2 for some generalization/specialization rela-

[1] This easily generalizes to lggs of n descriptions [2].

tion [4]. In our SPARQL setting, we use BGPQs as descriptions and entailment between BGPQs as generalization/specialization relation:

Definition 1 (lgg of BGPQs). *Let q_1, q_2 be two BGPQs with the same arity and $\mathcal{R}$ a set of RDF entailment rules.*
- *A* generalization *of q_1, q_2 is a BGPQ q_g such that $q_1 \models_{\mathcal{R}} q_g$ and $q_2 \models_{\mathcal{R}} q_g$.*
- *A* least general generalization *of q_1, q_2 is a generalization $q_{\texttt{lgg}}$ of q_1, q_2 such that for any other generalization q_g of q_1, q_2: $q_{\texttt{lgg}} \models_{\mathcal{R}} q_g$.*

Unfortunately, this natural definition is of limited practical interest as exemplified next. Consider the BGPQs q_1 and q_2 in Figure 1, which respectively ask for the conference papers having some contact author, and for the journal papers having some author. Clearly, with the RDF entailment rules shown in Table 2, an `lgg` of q_1 and q_2 is the *very* general BGPQ $q_{\texttt{lgg}}(x) \leftarrow (x, \tau, y)$ asking for *the resources having some type*. However, by considering the ontological constraints displayed in Figure 1 that hold in the scientific publication domain, i.e., the context in which the queries are asked, a more pregnant `lgg` would be $q_{\texttt{lgg}}(x) \leftarrow (x, \tau, \text{Publication}), (x, \text{hasAuthor}, y), (y, \tau, \text{Researcher})$ asking for *the publications having some researcher as author*, since (i) having a contact author is having an author, (ii) only publications have authors, (iii) only researchers are authors, and (iv) conference (resp. journal) papers are publications.

To devise such elaborate `lgg`s that rely on ontological knowledge, we revisit the notion of entailment between BGPQs in order to account for extra RDFS constraints. We first complement a BGPQ w.r.t. ontological knowledge:

Definition 2 (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 body of which, without loss of generality, does not contain blank nodes*[2]*. The* saturation *of q w.r.t. $\mathcal{O}$, denoted $q^{\infty}_{\mathcal{O}}$, is a BGPQ with the same answer variables as q and whose body, denoted $body(q^{\infty}_{\mathcal{O}})$, is the maximal subset of $(\mathcal{O} \cup body(q))^{\infty}$ 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.*

Intuitively, the saturation of a BGPQ comprises all the triples in the saturation of its body augmented with the constraints, except those triples that only follow from the ontological constraints, i.e., which are not related to what the query is asking for. This corresponds to the *non-hatched* subset of $(\mathcal{O} \cup body(q))^{\infty}$ shown in Figure 2. This Figure also displays the saturations $q_1{}^{\infty}_{\mathcal{O}}, q_2{}^{\infty}_{\mathcal{O}}$ of the two BGPQs q_1, q_2 w.r.t. the constraints $\mathcal{O}$ shown in Figure 1. Importantly, we proved that a BGPQ and its saturation w.r.t. ontological constraints are *equivalent for the central RDF reasoning tasks of query entailment and query answering* [2]:

Theorem 1. *Let $\mathcal{R}$ be a set of RDF entailment rules, $\mathcal{O}$ a set of RDFS statements, and q a BGPQ whose saturation w.r.t. $\mathcal{O}$ is $q^{\infty}_{\mathcal{O}}$. For any RDF graph $\mathcal{G}$ whose set of RDFS statements is $\mathcal{O}$, (i) $\mathcal{G} \models_{\mathcal{R}} q$ holds iff $\mathcal{G} \models_{\mathcal{R}} q^{\infty}_{\mathcal{O}}$ holds, and (ii) $q(\mathcal{G}) = q^{\infty}_{\mathcal{O}}(\mathcal{G})$ holds.*

Building on BGPQ saturation, we generalize entailment between BGPQs to:

[2] In SPARQL queries, blank nodes are equivalent to non-answer variables [7].

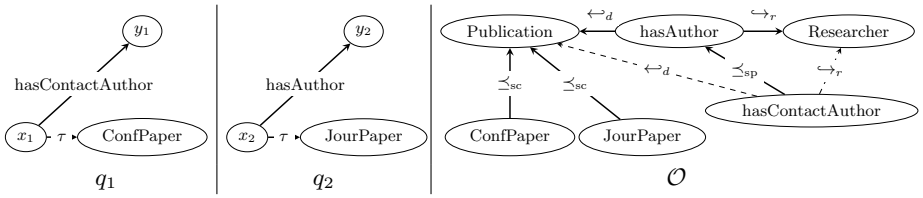

Fig. 1. Sample BGPQs q_1 and q_2; sample set of ontological constraints $\mathcal{O}$.

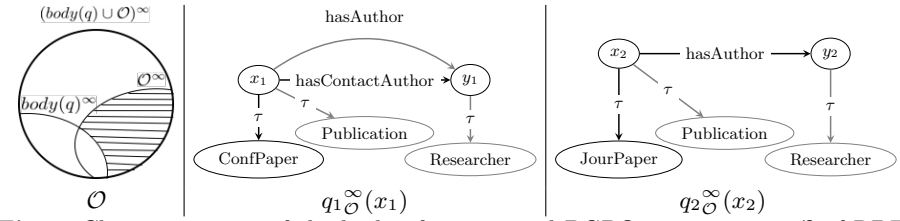

Fig. 2. Characterization of the body of a saturated BGPQ q w.r.t. a set $\mathcal{O}$ of RDFS constraints; saturations of q_1 and q_2 w.r.t. $\mathcal{O}$, triples in grey are added by saturation.

Definition 3 (Entailment between BGPQs w.r.t. RDFS constraints). *Given a set $\mathcal{R}$ of RDF entailment rules, a set $\mathcal{O}$ of RDFS statements, and two BGPQs q and q' with the same arity, q* entails *q'* w.r.t. *$\mathcal{O}$, denoted $q \models_{\mathcal{R},\mathcal{O}} q'$, iff $q^\infty_\mathcal{O} \models_\emptyset q'$ holds.*

When $\mathcal{O}$ is empty, the above definition coincides with standard entailment between BGPQs. Further, we proved fundamental properties for a BGPQ entailed by another w.r.t. ontological constraints: the former *generalizes* the latter for the central RDF reasoning tasks of query entailment and query answering [2]:

Theorem 2. *Let $\mathcal{R}$ be a set of RDF entailment rules, $\mathcal{O}$ a set of RDFS statements, and two BGPQs q and q' such that $q \models_{\mathcal{R},\mathcal{O}} q'$. For any RDF graph $\mathcal{G}$ whose set of RDFS statements is $\mathcal{O}$, (i) if $\mathcal{G} \models_\mathcal{R} q$ holds then $\mathcal{G} \models_\mathcal{R} q'$ holds, and (ii) $q(\mathcal{G}) \subseteq q'(\mathcal{G})$ holds.*

With the above notion of entailment between BGPQs endowed with ontological knowledge, we revise the definition of `lgg` (Definition 1) in order to use $\models_{\mathcal{R},\mathcal{O}}$ instead of $\models_\mathcal{R}$. We therefore propose to investigate as next challenge:

Problem 1. Given *two* BGPQs q_1, q_2 with same arity, a set $\mathcal{O}$ of RDFS statements, and a set $\mathcal{R}$ of RDF entailment rules, compute an `lgg` of q_1, q_2 w.r.t. $\mathcal{O}$.

References

1. Bühmann, L., Lehmann, J., Westphal, P.: Dl-learner - a framework for inductive learning on the semantic web. J. Web Semantics 39 (2016)
2. El Hassad, S., Goasdoué, F., Jaudoin, H.: Learning commonalities in RDF and SPARQL (research report). `https://hal.inria.fr/hal-01386237` (2016)
3. Lehmann, J., Bühmann, L.: Autosparql: Let users query your knowledge base. In: ESWC (2011)
4. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence 5 (1970)
5. Resource description framework 1.1. `https://www.w3.org/TR/rdf11-concepts`
6. RDF 1.1 semantics. `https://www.w3.org/TR/rdf11-mt/`
7. SPARQL. `http://www.w3.org/TR/rdf-sparql-query`