Representing and Querying Negative Knowledge in RDF

Fariz Darari*

Faculty of Computer Science, Free University of Bozen-Bolzano
Dominikanerplatz 3, Bozen-Bolzano, Italy
`fadirra@gmail.com`

Abstract. Typically, only positive data can be represented in RDF. However, negative knowledge representation is required in some application domains such as food allergies, software incompatibility and school absence. We present an approach to represent and query RDF data with negative data. We provide the syntax, semantics and an example. We argue that this approach fits into the open-world semantics of RDF according to the notion of certain answers.

1 Syntax

We present the syntax for representing positive and negative RDF data using Turtle serialization. We represent positive data `(s, p, o)` and negative data `not(s, p, o)` as reified statements of types `:posStatement` and `:negStatement` respectively[1], and they have properties `:subj`, `:pred` and `:obj` with the corresponding values `s`, `p` and `o`. In our work, `s`, `p` and `o` cannot be blank nodes.

Since negative knowledge can be represented, it is possible for data to be inconsistent. Thus, inconsistency is defined when the intersection between positive and negative data is not empty.

There will be a pre-processing step to de-reify positive and negative RDF data, and put it into the graphs `:posGraph` and `:negGraph` respectively, for querying.

As for the SPARQL syntax, we focus on SELECT queries with BGP (Basic Graph Pattern), which is of the form `(SELECT W P)`, where `W` is a set of variables and `P` is a set of positive (of the form `(s, p, o)`), and negative triple patterns (of the form `not(s, p, o)`), which are the new feature.

2 Semantics

We present the semantics for positive and negative RDF data. The positive RDF data has the semantics as in W3C's RDF Specification[2]. Moreover, following the specification, we define the semantics of negative RDF data as: if `E` is a ground negative

[1] We assume the default RDF namespace is http://example.com/.

[2] http://www.w3.org/TR/rdf-mt/

* The author was supported by the European Master's Program in Computational Logic and the project Querying Incomplete Data (QUID) funded by the FU Bozen-Bolzano.

triple `not(s, p, o)`, then `I(E) = true` if `s`, `p` and `o` are in `V`, `I(p)` is in `IP` and `<I(s), I(o)>` is not in `IEXT(I(p))`, otherwise `I(E)= false`.

As for the semantics of SPARQL, during evaluation, all positive triple patterns in the body of queries will be delegated into positive graph `:posGraph`, whereas all negative triple patterns will be delegated into negative graph `:negGraph` using GRAPH graph patterns. Then, the transformed SPARQL query is evaluated using the standard SPARQL semantics [1]. We provide an example in Section 3.

Theorem For a SPARQL query `Q` with positive and negative triple patterns and a consistent RDF data `D`, the evaluation of query `Q` over `D` gives certain answers over all models of `D`.

Proof Idea. The proof is based on the semantics of BGP and GRAPH graph pattern that will give certain answers [1], with the idea that for every negative triple `not(s, p, o)`, then it must be the case that in all models of `D`, `s`, `p` and `o` are in `V`, `I(p)` is in `IP` and `<I(s), I(o)>` is not in `IEXT(I(p))`. This is reflected by the evaluation of negative query part into all the triples in `:negGraph` that gives certain answers.

3 Example

Consider an example in the food allergies domain. Let `D` be RDF data containing `{(john, eats, egg), (john, eats, nut), (tom, eats, egg), not(john, eats, fish)}`. Given a query `Q = SELECT {?x} {(?x, eats, egg), not(?x, eats, fish)}`, the evaluation of `Q` over `D` gives `{{?x/john}}`.

Regarding the implementation, for example, the negative triple `not(john, eats, fish)` will be represented as `{(_:b1 a :negStatement), (_:b1 :subj :john), (_:b1 :pred :eats), (_:b1 :obj :fish)}` and thus, after dereification, the triple `:john :eats :fish` is stored in the graph `:negGraph`. Also note that the query `Q` during evaluation has the following form:

```
SELECT ?x WHERE { GRAPH :posGraph {?x :eats :egg}
                  GRAPH :negGraph {?x :eats :fish}}
```

4 Future Work

For future work, we will investigate the roles and interaction of blank nodes, RDF Schema and more expressive SPARQL queries with respect to negative RDF data.

References

1. M. Arenas, C. Gutierrez, J. Pérez, On the Semantics of SPARQL In: *Semantic Web Information Management: A Model Based Perspective*, R. De Virgilio, F. Giunchiglia, L. Tanca, editors, Springer, 2010 (ISBN: 978-3-642-04328-4)