Querying Incomplete Geospatial Information in RDF*

Charalampos Nikolaou and Manolis Koubarakis

National and Kapodistrian University of Athens, Greece
{charnik,koubarak}@di.uoa.gr

1 Introduction

Incomplete information has been studied in-depth in relational databases and knowledge representation. It is also an important issue in Semantic Web frameworks such as RDF, description logics, and OWL 2. In [6], we introduced RDF^i, an extension of RDF for representing incomplete information using constraints. We defined a semantics for RDF^i and studied SPARQL query evaluation in this framework. Given the current interest in publishing geospatial datasets as linked data (e.g., by Ordnance Survey in the UK), RDF^i is an excellent framework for encoding, possibly incomplete, qualitative and quantitative geospatial information which is found in these published datasets. RDF^i is also interesting because when the constraint language used can express the topological relations of RCC-8 [8], the recent OGC standard GeoSPARQL [7] for querying geospatial information expressed in RDF, becomes a special case of RDF^i.

In this paper, we propose the problem of *implementing an efficient query processing system for incomplete temporal and geospatial information in* RDF^i as a challenge to the SSTD community. For the case of incomplete temporal information in relational databases, two such systems have been implemented in the past [1,3], but their query languages are rather limited. There is also the paper [2] which studies a closely related problem, temporal relationships in databases, but which has no related implementations. Finally, the knowledge representation language Telos [5] allows the modelling of incomplete temporal knowledge but well-known implementations such as ConceptBase have not implemented this functionality. To the best of our knowledge, no such relational database system or RDF store exists for the geospatial case, although there are some description logic reasoners that come close in terms of functionality [11]. In the rest of the paper, we present the RDF^i framework and outline the hard problems that have to be solved if such a query processing system is to become a reality. As in the theoretical foundation of [6], we concentrate on geospatial information only.

2 RDF^i by Example

RDF^i [6] is a framework that extends RDF in a general way with the ability to represent and query incomplete information. Incomplete information often

* This work was supported by the European FP7 project TELEIOS (257662) and the Greek NSRF project SWeFS (180).

M.A. Nascimento et al. (Eds.): SSTD 2013, LNCS 8098, pp. 447–450, 2013.

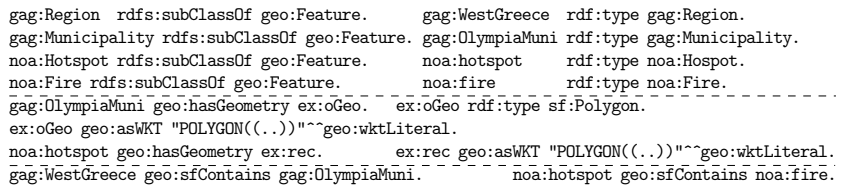


Fig. 1. RDF^{i} database using the vocabulary of GeoSPARQL

arises when sensing the real-world due to the inherent imprecision of measuring devices. For example, in wild-fire monitoring applications, satellite images are analyzed to detect hotspots (i.e., pixels of the image corresponding to geographic regions that are probably on fire). Due to the medium resolution of the satellite images, these hotspots correspond to rectangles that are 3km by 3km wide. Thus, a useful representation of the real world situation of a hotspot is to state that there is a geographic region with unknown exact coordinates where a fire is taking place, and that region is included in the known geometry for the hotspot. Such information is usually combined with other relevant sources, such as administrative boundaries, to aid decision makers in managing the fire.

This scenario is captured by the RDF^{i} database of Fig. 1 using the vocabulary offered by GeoSPARQL (namespace `geo`). This database contains data about the hotspot, the fire, the administrative region of West Greece, and the municipality of Olympia in which the hotspot has been detected. To make the example more interesting, we have assumed that the exact administrative boundary for West Greece is unknown. The second set of triples in Fig. 1 encode the geometries of Olympia and the hotspot using the Well-Known Text standard format, while the last two triples state the containment relations that hold between West Greece and the municipality of Olympia, and the hotspot and fire.

RDF^{i} uses the concept of e-literals to represent property values that exist but are unknown or partially known. Partial knowledge about property values is expressed by a quantifier-free formula of a first-order *constraint language*. For simplicity, we have not used the more powerful syntax of RDF^{i} to capture our partial knowledge about West Greece and the hotspot. Instead we have expressed it as triples as in GeoSPARQL (last two triples of Fig. 1). The following is a SPARQL query that uses the topology vocabulary extension of GeoSPARQL to query the database of Fig. 1 for fires that are inside the region of West Greece.

```
SELECT ?f WHERE {?f rdf:type noa:Fire. gag:WestGreece geo:sfContains ?f.}
```

The specification of GeoSPARQL does not propose a semantics or algorithm for computing the answer to such a query, although the answer is entailed by the triples of Fig. 1. The answer can be computed by computing the entailment of relation `geo:sfContains` between `gag:WestGreece` and `noa:hotspot` from the fact that the geometry of the hotspot is contained in the geometry of Olympia, and then include it in the computation of the transitive closure for relation `geo:sfContains` to derive the triple `gag:WestGreece geo:sfContains noa:fire`. In contrast, SPARQL query evaluation over RDF^{i} databases as studied in [6] gives an algorithm for computing such entailments.

3 Query Processing Challenges

As the example of the previous section shows, and as we have shown more generally in [6, Theorem 2], the challenge with which a system is faced for answering queries such as the above is the efficient computation of the entailment relation $\Phi \models \Theta$ where Φ, Θ are quantifier-free first-order formulae of a constraint language that is capable of expressing the topological relations of various frameworks such as RCC-8, DE-9IM, OGC Simple Features, etc. Computing such an entailment usually reduces to checking the consistency of constraint networks that involve qualitative spatial relations among regions identified by a URI and constant ones. This combination of qualitative and quantitative constraints has been studied in detail for temporal constraints but similar results do not exist for spatial constraints. Only recently there has been some work on topological relations among polygonal regions, but is limited to atomic and complete constraint networks, which are far away from real datasets.

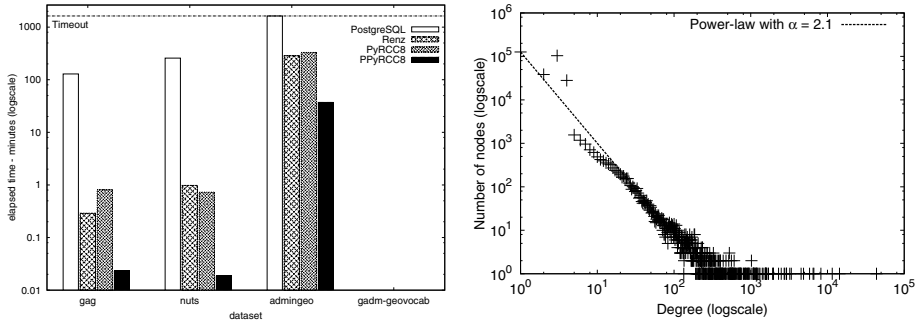
RDF stores supporting linked geospatial data are expected to scale to *billions* of triples like their non-spatial counterparts and recent work in this area is encouraging [4]. Can this level of scalability be achieved when qualitative spatial relations come into play? A good approach here might be to start with algorithms with low polynomial complexity and try to implement them as efficiently as possible. In the temporal case, this approach has been followed successfully by temporal reasoners. In addition, there might be cases where network structure can be exploited (e.g., hierarchical organization of geographical regions).

To answer the above question, we have compared the performance of checking the consistency of tractable RCC-8 constraint networks using the well-known Path Consistency (PC) algorithm as implemented in the state of the art qualitative spatial reasoners Renz Solver [9], PyRCC-8 and PyRCC-8$\bigtriangledown$ [10], and a relational counterpart of the PC algorithm of [9] as a SQL program in PostgreSQL only to reach the same conclusion. Our findings were that none of the implementations scale so as to be qualified for use in implementations of query processing algorithms for the entailment problem described above.

Fig. 2a shows the performance of these implementations using real-world linked geospatial datasets containing only qualitative spatial relations from RCC-8 (this is a much simpler problem than the one considered in Fig. 1). Each dataset is presented in increasing order of its size, while the sizes range from 1500/2000 nodes/edges to 276728/590443 nodes/edges. For the last dataset there are no measurements for any of the implementations, because they all crass either immediately (reasoners) or after some amount of time (PostgreSQL) due to memory allocation errors[1]. For the first case, this is because they allocate a two-dimensional array to represent the input constraint network. This array is of size N^2 where N is the number of nodes. Thus, even for medium-sized graphs, these implementations fail to run, a drawback that is not present in the relational-based implementation which can complete one or two iterations of PC.

The most interesting observation for such graphs is that they are sparse, hence representations of the input graph that are based on two-dimensional arrays are

[1] Setup: Intel Xeon E5620, 2.4 GHz, 12MB L3, 48GB RAM, RAID 5, Ubuntu 12.04.



(a) (b)

Fig. 2. (a) Performance of PC for various datasets (b) Degree distribution for gadm-geovocab

not appropriate. This fact is depicted in Fig. 2b where the degree distribution among the nodes of the largest dataset is shown. Fig. 2b shows that the input RCC-8 constraint network is sparse following a power-law distribution. The primary characteristic of such graphs is that most of the nodes have very low degree, while only a small number of the nodes have high degree number leading to star-shaped graphs. Another observation is that real datasets comprise constraint graphs with edges of three kinds: (non-)tangential proper part, externally connects, or equals. Such kind of edges reflect networks that are composed of components with a hierarchical structure which calls for further optimization.

References

1. Brusoni, V., Console, L., Terenziani, P., Pernici, B.: Later: Managing temporal information efficiently. IEEE Expert 12(4), 56–64 (1997)
2. Chaudhuri, S.: Temporal relationships in databases. In: VLDB, pp. 160–170 (1988)
3. Griffiths, A., Theodoulidis, B.: SQL+i: Adding temporal indeterminancy to the database language SQL. In: Morrison, R., Kennedy, J. (eds.) BNCOD 1996. LNCS, vol. 1094, pp. 204–221. Springer, Heidelberg (1996)
4. Kyzirakos, K., Karpathiotakis, M., Koubarakis, M.: Strabon: A Semantic Geospatial DBMS. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 295–311. Springer, Heidelberg (2012)
5. Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: Representing knowledge about information systems. ACM Trans. Inf. Syst. 8(4), 325–362 (1990)
6. Nikolaou, C., Koubarakis, M.: Incomplete information in RDF. In: RR (2013)
7. Open Geospatial Consortium: OGC GeoSPARQL - A geographic query language for RDF data. OGC® Implementation Standard (2012)
8. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: KR (1992)
9. Renz, J., Nebel, B.: Efficient methods for qualitative spatial reasoning. Journal of Artificial Intelligence Research (JAIR) 15, 289–318 (2001)
10. Sioutis, M., Koubarakis, M.: Consistency of Chordal RCC-8 Networks. In: ICTAI (2012)
11. Stocker, M., Sirin, E.: PelletSpatial: A hybrid RCC-8 and RDF/OWL reasoning and query engine. In: OWLED (2009)