Provenance Management for Evolving RDF Datasets

Argyro Avgoustaki [1, 2], Giorgos Flouris [2], Irini Fundulaki [2], Dimitris Plexousakis [1, 2]

{argiro,fgeo,fundul,dp}@ics.forth.gr

[1] Computer Science Department, University of Crete

[2] Institute of Computer Science, FORTH

www.ics.forth.gr/isl/provenance

Motivation

During the last few years we have witnessed an explosion in the publication of semantic data in the Web. Recording the provenance of such data is an essential task in order to effectively support trustworthiness, accountability and repeatability. In this context, our work:

- Introduces a new provenance model for SPARQL updates
- Allows the **reconstructability** of SPARQL INSERT Updates from their provenance
- Provides algorithmic support via the **Provenance Construction** and the **Update Reconstruction** algorithms

Model Features

- Suitable for encoding the *triple* and *attribute level provenance* of RDF quadruples
- Uses complex algebraic expressions
- Based on *how* and *where* provenance models
- Supports unions of basic graph patterns

SPARQL Update Semantics

INSERT { qp_{ins} } WHERE { gp }, where:

- qp_{ins} is a *quad pattern*
- gp is a *graph pattern* of the form gp^1 **UNION** gp^2 ... **UNION** gp^k
- gp^i is of the form $qp^i_1 \,.\, qp^i_2 \,.\, \ldots \,.\, qp^i_m$
- i: the order of a graph pattern in the WHERE clause
- m: the order of a quad pattern in gp^i
- $qp^i_j.pos$, $qp_{ins}.pos$, where pos ϵ s,p,o, are *quad pattern position identifiers*

Provenance Model

cpe represents each different way for a quadruple to be generated

pe corresponds to the provenance of one operand of a UNION operator

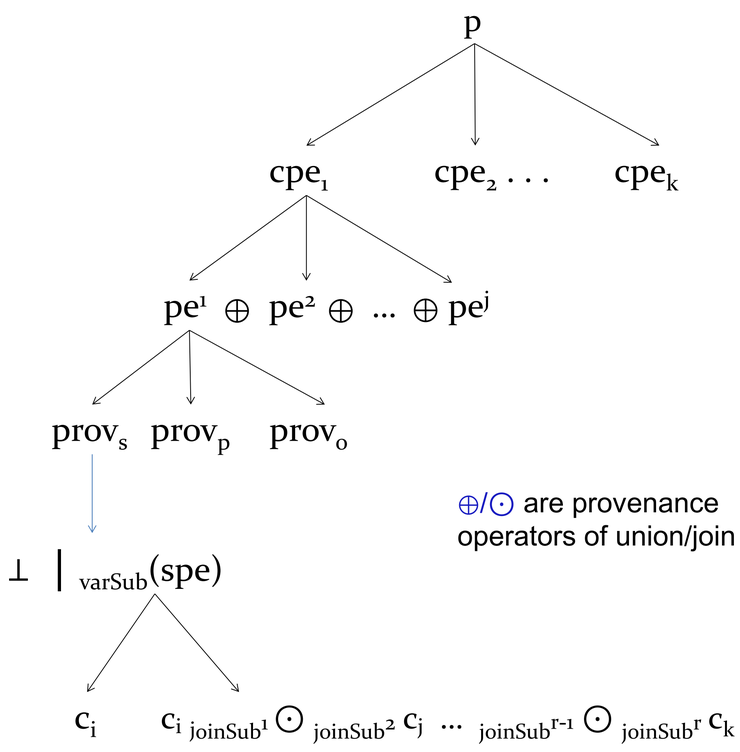
$prov_{pos}$ identifies the origin of each attribute

$\perp$ is used for constants

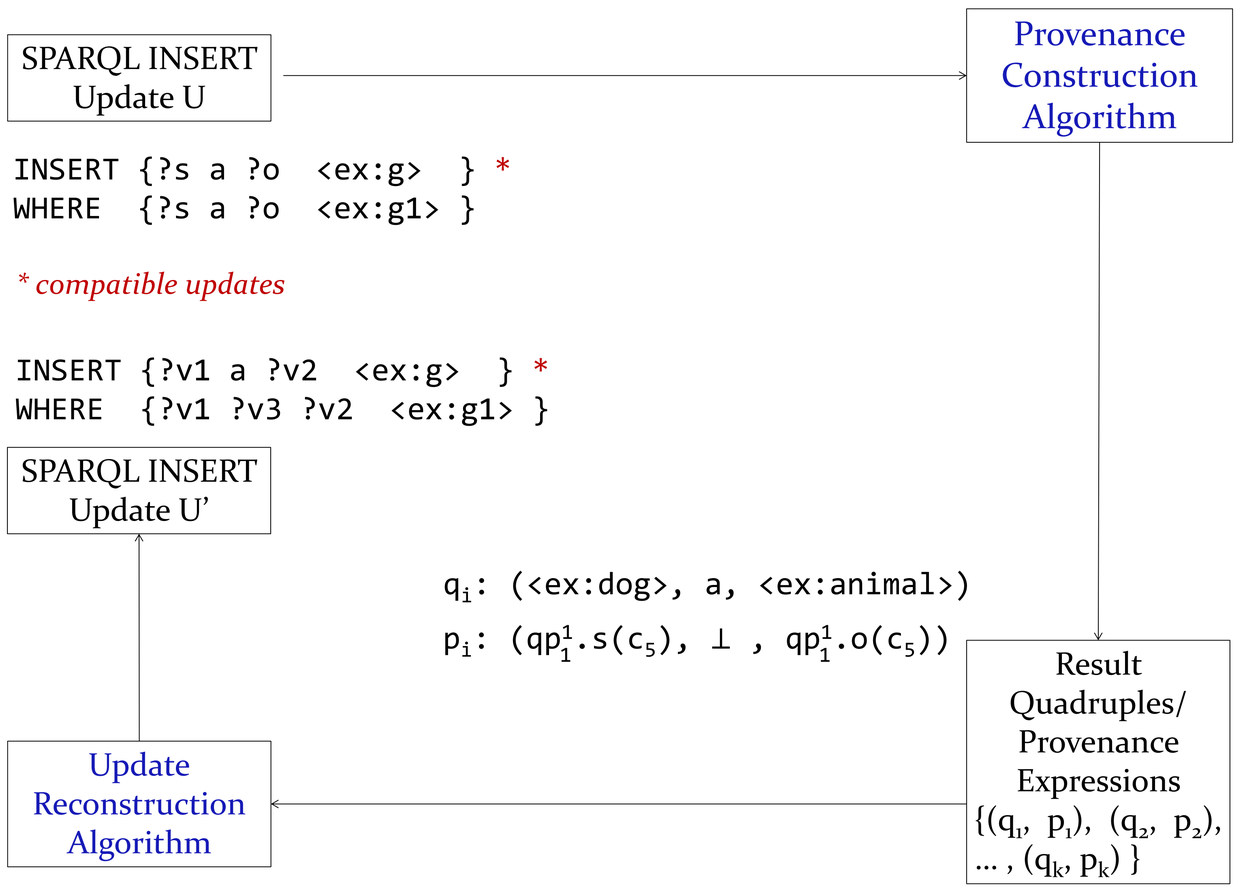
$_{varSub}$(spe) records the provenance for "copy" and join

c_i is a *quadruple identifier*

$joinSub^x$ is a set of quad pattern positions that indicates the join positions of a join operand

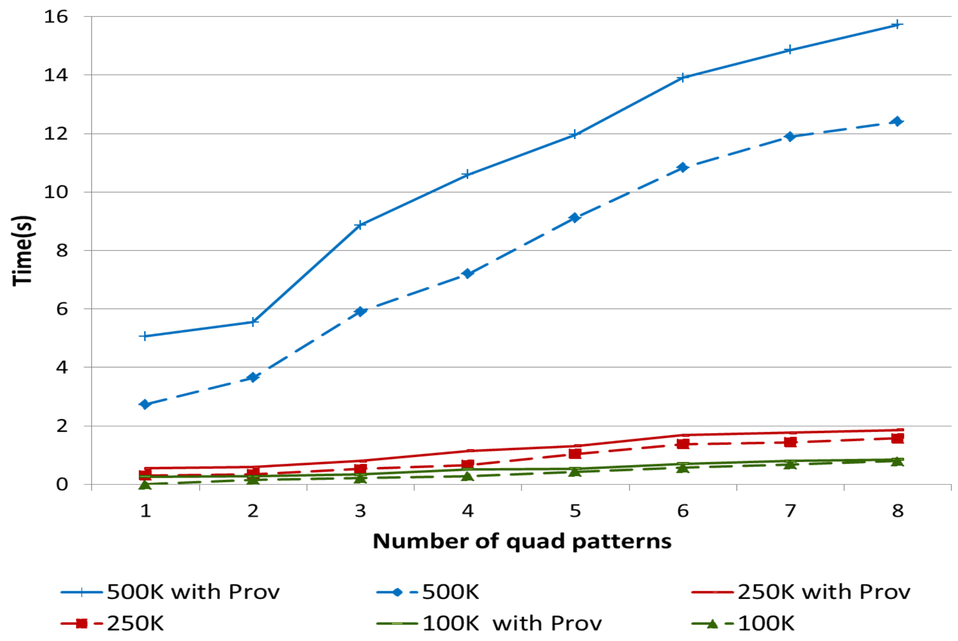


Algorithms



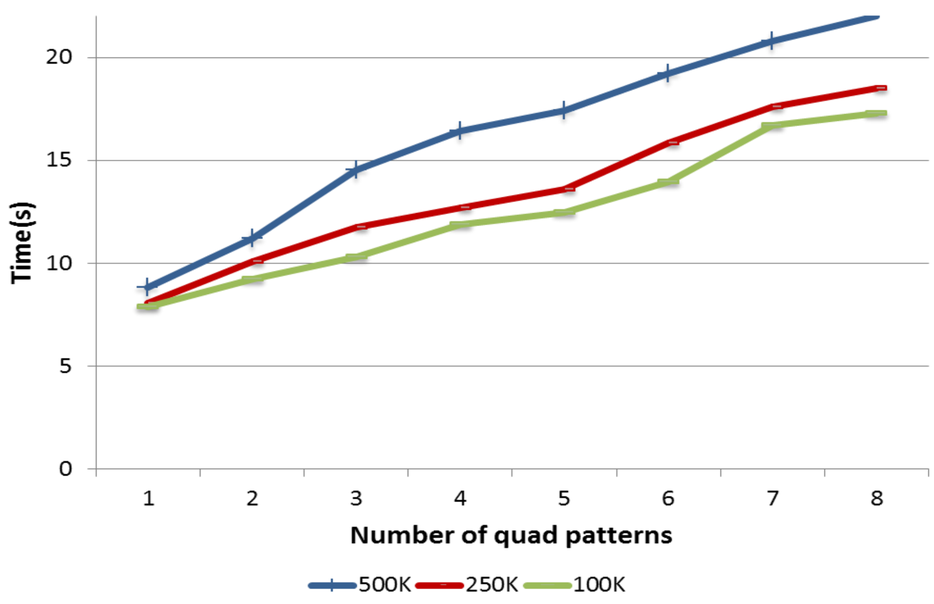
Implementation and Evaluation

- Used Virtuoso Database Engine as triple store
- Quadruples and provenance expressions are stored in a relational schema
- Excerpts of BTC dataset containing 100K, 250K and 500K unique quadruples
- **Experiment 1** measures the time required to compute the results of an INSERT update along with their provenance
- **Experiment 2** considers the time required to compute only the result quadruples
- **Experiment 3** computes the time needed for reconstructing a compatible INSERT update based on a quadruple's provenance



Experiment 1,2

The difference in computation time of Experiment 1 and Experiment 2 indicates the overhead for computing provenance



Experiment 3

Future Work

1. Support provenance management for all operations of SPARQL Update
2. Extend our model to support *FILTER* and *OPTIONAL* operators as well as SPARQL functions
3. Study the provenance of inferred quadruples using backward and forward reasoning
4. Explore the use of PROV approach
5. Consider benchmarks supporting update operations

References

1. A. Avgoustaki, G. Flouris, I. Fundulaki, and D. Plexousakis. Provenance Management for Evolving RDF datasets. In Extended Semantic Web Conference, 2016.
2. A. Avgoustaki. Provenance management for SPARQL updates. Master's thesis, University of Crete, 2014.
3. G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides. Coloring RDF triples to capture provenance. In International Semantic Web Conference, 2009.
4. P. Buneman, A. Chapman, and J. Cheney. Provenance management in curated databases. In ACM SIGMOD International Conference on Management of Data, 2006.
5. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In ACM PODS Symposium on Principles of Database Systems, 2007.