Cost-based logical optimization for DAG plans of reformulated queries

Damian Bursztyn[1], François Goasdoué[2], Ioana Manolescu[1], Alessandro Solimando[1], and Michaël Thomazo[1]

[1] INRIA & U. Paris-Sud, France [2] U. Rennes 1 & INRIA, France

Query answering in presence of *semantic constraints* must cope with both *explicit* and *implicit* data. Two families of query answering techniques have been proposed in literature, namely *Saturation*-based, which adds all the implicit knowledge to the explicit graph, that is then queried in a classical way, and *Reformulation*-based, which does not alter the set of explicit data, but reformulates the original query under the given contraints in such a way that ensure completeness also w.r.t. implicit knowledge.

Different reformulation algorithms have been proposed in literature. Union of conjunctive queries (`UCQ`) reformulation [6, 10, 8, 11, 4, 1, 5, 3, 7] applies to various fragments of RDF, ranging from the Description Logics (DL) one up to the Database (DB) one. Another notable reformulation technique computes semi-conjunctive queries (`SCQ`) reformulations [9] and can be applied to the DL fragment of RDF. Join of unions of conjunctive queries (`JUCQ`) reformulation, a generalization of both `UCQ` and `SCQ`, has been proposed for the DB fragment of RDF [2], and has shown to improve the execution performance of reformulated queries over both `UCQ`s and `SCQ`s.

To improve the performance of reformulated query evaluation performance even further, in the context of query answering for RDF, we investigate `SPJUM`, which allows plans composed by semi-join, projection, join, union and materialization operators, and thus strictly subsumes `JUCQ`. Join and union operators can appear in the computed plans at any level, differently from all the previous proposals in literature. Semi-join operators are explored for improving performance, materialization allows to save extra work in case of worth reuse opportunities for intermediate results.

In this context, we propose *cost-based* algorithms, which compute a plans for executing the query having a contained cost.

References

1. P. Adjiman, F. Goasdoué, and M.-C. Rousset. SomeRDFS in the semantic web. *JODS*, 8, 2007.
2. D. Bursztyn, F. Goasdoué, and I. Manolescu. Optimizing reformulation-based query answering in RDF. In *International Conference on Extending Database Technology (EDBT)*, pages 265–276, 2015.
3. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *JAR*, 39(3), 2007.
4. G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, R. Rosati, M. Ruzzi, and D. F. Savo. MASTRO: A reasoner for effective ontology-based data access. In *ORE*, 2012.
5. F. Goasdoué, K. Karanasos, J. Leblay, and I. Manolescu. View selection in semantic web databases. *PVLDB*, 2011.
6. F. Goasdoué, I. Manolescu, and A. Roatiş. Efficient query answering against dynamic RDF databases. In *EDBT*, 2013.
7. G. Gottlob, G. Orsi, and A. Pieris. Ontological queries: Rewriting and optimization. In *ICDE*, 2011. Keynote.
8. Z. Kaoudi, I. Miliaraki, and M. Koubarakis. RDFS reasoning and query answering on DHTs. In *ISWC*, 2008.
9. M. Thomazo. Compact rewriting for existential rules. *IJCAI*, 2013.
10. J. Urbani, A. Margara, C. J. H. Jacobs, F. van Harmelen, and H. E. Bal. Dynamite: Parallel materialization of dynamic RDF data. In *ISWC*, 2013.
11. J. Urbani, F. van Harmelen, S. Schlobach, and H. Bal. QueryPIE: Backward reasoning for OWL Horst over very large knowledge bases. In *ISWC*, 2011.