Why-Provenance for Datalog with Negation

Bart Bogaerts [1,2], Marco Calautti [3], Andreas Pieris [4,5], Samuele Pollaci [2,1], and Robbe Van den Eede [1,2]

[1] Department of Computer Science, KU Leuven, Belgium
{bart.bogaerts,robbe.vandeneede}@kuleuven.be
[2] Department of Computer Science, Vrije Universiteit Brussel, Belgium
samuele.pollaci@vub.be
[3] Department of Computer Science, University of Milan, Italy
marco.calautti@unimi.it
[4] School of Informatics, University of Edinburgh, United Kingdom
apieris@exseed.ed.ac.uk
[5] Department of Computer Science, University of Cyprus, Cyprus

Datalog is a logic programming language of inductive definitions that is commonly used to answer queries over databases [1]. A standard way of explaining answers to database queries is through the notion of *why-provenance* [2]. For Datalog queries, why-provenance can be defined using the general framework of provenance semiring [6]. Given a database and a Datalog query, the why-provenance consists of all subsets of the database that, as a whole, can be used to derive the query. More specifically, the why-provenance for a Datalog query with respect to a database is defined as the set of *supports* of all *proof trees* of the query with respect to the database. A *proof tree* is a tree that represents a derivation of the query from the database facts, following the rules of the Datalog program. The *support* of a proof tree is the set of labels of all the leafs of the proof tree, which are (extensional) database facts. One can associate a decision problem with the notion of why-provenance, asking whether a given subset of the database belongs to the why-provenance. Calautti et al. have shown that the problem of why-provenance for Datalog is NP-complete [3].

In our work, we extend the notion of why-provenance to Datalog with negation, Datalog$^\neg$, both under the well-founded and the stable semantics. To accomplish this, we use the machinery of *justification theory*. Justification theory was originally introduced by Denecker and De Schreye to define a formal semantics for non-monotone inductive definitions [5, 4]. Subsequently, it grew out to a uniform framework for defining formal semantics for several monotone logics [7].

The notion of *justification graph* generalizes the notion of proof tree to Datalog$^\neg$. An important difference from proof trees is that justification graphs may have infinite branches. Different semantics are completely characterised by choosing how to treat such branches. This is formalized by means of a so-called *branch evaluation*. We generalize the notion of support, and thereby the notion of why-provenance, to Datalog$^\neg$. Starting from this, we show that the problem of why-provenance for Datalog$^\neg$ remains NP-complete, both for the well-founded branch evaluation wf and the stable branch evaluation st.

Finally, since the original stable branch evaluation st has explanatory shortcomings, we introduce an alternative stable branch evaluation st' that is semantically equivalent to st. Nevertheless, compared to st, the new stable branch evaluation st' yields explanations that are in a certain sense ‘closer to the database’, and that point to certain choices made under the stable semantics. The gain in explanatory value of st' comes with a cost in computation complexity: we show that the problem of why-provenance with respect to st' is PSPACE-complete.

References

[1] Abiteboul, S., Hull, R., and Vianu, V. *Foundations of Databases.* Addison-Wesley, 1995.

[2] Buneman, P., Khanna, S., and Wang-Chiew, T. Why and where: A characterization of data provenance. In *Database Theory — ICDT 2001* (Berlin, Heidelberg, 2001), J. Van den Bussche and V. Vianu, Eds., Springer Berlin Heidelberg, pp. 316–330.

[3] Calautti, M., Livshits, E., Pieris, A., and Schneider, M. The complexity of why-provenance for datalog queries. *Proc. ACM Manag. Data 2*, 2 (2024).

[4] Denecker, M. *Knowledge representation and reasoning in incomplete logic programming.* PhD thesis, Katholieke Universiteit Leuven, Belgium, 1993.

[5] Denecker, M., and De Schreye, D. Justification semantics: A unifiying framework for the semantics of logic programs. In *Logic Programming and Non-monotonic Reasoning, Proceedings of the Second International Workshop, Lisbon, Portugal, June 1993* (1993), L. M. Pereira and A. Nerode, Eds., MIT Press, pp. 365–379.

[6] Green, T. J., Karvounarakis, G., and Tannen, V. Provenance semirings. In *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (New York, NY, USA, 2007), PODS '07, Association for Computing Machinery, p. 31–40.

[7] Marynissen, S. *Advances in Justification Theory.* PhD thesis, Department of Computer Science, KU Leuven, Jan. 2022. Marc Denecker and Bart Bogaerts (supervisors).