On Deciding the Data Complexity of Answering Linear Monadic Datalog Queries with LTL Operators (Extended Abstract)

Alessandro Artale[1], Anton Gnatenko[1], Vladislav Ryzhikov[2] and Michael Zakharyaschev[2]

[1]*Faculty of Engineering, Free University of Bozen-Bolzano, piazza Domenicani 3, Bozen-Bolzano 39100, Italy*
[2]*Birkbeck, University of London, Malet Street, London WC1E 7HX, UK*

Abstract

We study the data complexity of connected linear monadic datalog queries equipped with operators of linear temporal logic *LTL* in the rule bodies. We establish that answering a query with the operators $\bigcirc/\bigcirc^{-1}$ (at the next/previous moment) is either in AC^0, or in ACC$^0 \setminus$ AC^0, or NC^1-complete, or L-hard and in NL for data complexity, and show that checking L-hardness of answering a given query is in ExpSpace, while checking whether it lies in AC^0, ACC^0, or is NC^1-complete is in 2ExpSpace, and that these problems become, respectively, PSpace-complete and in ExpSpace for queries with just one operator $\bigcirc$ (or $\bigcirc^{-1}$). We prove further that these problems are undecidable for queries with the operators $\Diamond/\Diamond^{-1}$ (sometime in the future/past).

Keywords

Linear monadic datalog, linear temporal logic, ontology mediated query, data complexity.

We consider linear monadic datalog queries, in which predicates in the rule bodies can be prefixed by the temporal operators $\bigcirc/\bigcirc^{-1}$ (at the next/previous moment) and $\Diamond/\Diamond^{-1}$ (sometime in the future/past) of linear temporal logic *LTL* [1]. Data instances for such queries are finite sets of ground atoms that are timestamped by the moments of time $\ell \in \mathbb{Z}$ they happen at. We interpret $\Diamond$ under the strict semantics: $\Diamond P(\boldsymbol{a})$ is regarded to be true at ℓ if $P(\boldsymbol{a})$ is true at some later moment $\ell' > \ell$. The interpretation of $\bigcirc$ is as usual: $\bigcirc P(\boldsymbol{a})$ is true at ℓ if $P(\boldsymbol{a})$ holds at $\ell + 1$. The semantics of $\bigcirc^{-1}$ and $\Diamond^{-1}$ is defined as the mirror image in the past.

Example 1. Consider a temporal database that represents a schedule of flights between cities $c_1, \dots, c_n$ in a period of time. It contains the fact *Flight*(c_i, c_j) at time ℓ if there is a flight from city c_i to city c_j on day ℓ. The following temporal conjunctive query finds a city now—at time 0—and and return later:

$$\mathit{Goal}(X) \leftarrow \mathit{Flight}(X, Y) \wedge \Diamond \mathit{Flight}(Y, X). \tag{1}$$

The following query is recursive. It looks for cities, from which we can travel to a place that is sunny for at least two consecutive days with the requirement that, on our way, we pass through sunny cities only:

$$\begin{aligned} &\mathit{Goal}(X) \leftarrow \mathit{ReachSun}(X), \\ &\mathit{ReachSun}(X) \leftarrow \mathit{Sunny}(X) \wedge \bigcirc \mathit{Sunny}(X), \\ &\mathit{ReachSun}(X) \leftarrow \mathit{Flight}(X, Y) \wedge \mathit{Sunny}(Y) \wedge \bigcirc \mathit{ReachSun}(Y). \end{aligned} \tag{2}$$

The temporal datalog queries above have only monadic predicates on the left-hand side of each rule, which are known as IDB predicates, and at most one IDB predicate on the right-hand side (body) together with any number of other (EDB) predicates of arbitrary arity. We refer to such queries as *temporal monadic linear datalog queries*. Here, we only consider *connected* queries, in which each rule body gives rise to a connected variable graph.

DL 2024: 37th International Workshop on Description Logics, June 18–21, 2024, Bergen, Norway

artale@inf.unibz.it (A. Artale); anton.gnatenko@student.unibz.it (A. Gnatenko); v.ryzhikov@bbk.ac.uk (V. Ryzhikov); m.zakharyaschev@bbk.ac.uk (M. Zakharyaschev)
0000-0002-3852-9351 (A. Artale); 0000-0003-1499-2090 (A. Gnatenko); 0000-0002-6847-6465 (V. Ryzhikov); 0000-0002-2210-5183 (M. Zakharyaschev)

We are interested in the data complexity of answering temporal linear monadic datalog queries, that is, in the complexity of the following decision problem (instance checking), for any fixed query $\boldsymbol{q}$: given a temporal data instance $\mathcal{D}$, an object a in its domain and a timestamp ℓ, decide whether $\mathit{Goal}(a)$ is true at time ℓ in every model of $\boldsymbol{q}$ and $\mathcal{D}$. It is not hard to see that the upper bound on the complexity of this problem is NL, but for different queries it may vary significantly. For example, answering query (1) lies in AC^0, while answering query (2) is NL-complete. Note that, if we drop $\bigcirc$ from (2), it can also be answered in AC^0.

In fact, it is well-known that a pure linear monadic datalog query $\boldsymbol{q}$ (without temporal operators) can either lie in the complexity class AC^0, or in L, or in NL, being at least L-hard in the last two cases. On the other hand, any pure *LTL* query $\boldsymbol{q}$ (with 0-ary predicates, i.e., propositional variables) lies either in AC^0 or in $\mathrm{ACC}^0 \setminus \mathrm{AC}^0$, or is NC^1-complete [2].

Temporal datalog queries are two-dimensional, being interpreted over the object domain for the individual variables and the implicit temporal domain $(\mathbb{Z}, <)$ for the *LTL* operators. Our first observation is that the temporal operators $\bigcirc/\bigcirc^{-1}$ and $\Diamond/\Diamond^{-1}$ provide different levels of interaction between the two dimensions. Consider, for example, the following program:

$$\mathit{Goal}(X) \leftarrow A(X) \land \bigcirc R(X,Y) \land \bigcirc C(Y), \tag{3}$$

$$C(X) \leftarrow \bigcirc C(X), \tag{4}$$

$$C(X) \leftarrow B(X). \tag{5}$$

Suppose a data instance contains $A(a)$ at time 0, $R(a,b)$ at time 1, and $B(b)$ at time 5. We can infer $\mathit{Goal}(a)$ at 0 by first applying rule (5) to infer $C(b)$ at 5, then rule (4) to infer $C(b)$ at 4, 3, 2, and 1, and finally rule (3) to infer $\mathit{Goal}(a)$ at 0. Rules (4) and (5) work along the timeline of a single object, b, while the final application passes from one object, b, to another, a. To do so, it checks whether a certain condition holds for the joint timeline of a and b, namely, that they are connected by R at the next moment of time. The operators $\bigcirc/\bigcirc^{-1}$ can query the joint timeline of several objects by a number of steps bounded by the size of the query. Therefore, there is a limited interaction between the two 'phases' of inference—exploring the object and the temporal domain. On the other hand, a rule that features the operators $\Diamond/\Diamond^{-1}$ can navigate the whole temporal line in addition to the object domain:

$$C(X) \leftarrow \Diamond R(X,Y) \land D(Y). \tag{6}$$

In this case, inferring $C(a)$ at time 0 requires checking the existence of an object b that is connected to a by R at some (arbitrarily distant) moment in the future, and then passing to that object and trying to infer $D(b)$.

This distinction between $\bigcirc/\bigcirc^{-1}$ and $\Diamond/\Diamond^{-1}$ is crucial for the analysis of the behaviour of the respective queries. A query $\boldsymbol{q}$ that only features $\bigcirc/\bigcirc^{-1}$ is decomposable into a pure datalog part $\boldsymbol{q}_d$ and a pure *LTL* part $\boldsymbol{q}_t$, which are, however, exponentially larger than $\boldsymbol{q}$. These two 'one-dimensional' queries can be analysed separately using the methods of Cosmadakis et al. [3] and Kurucz et al. [4], respectively. Recall that evaluation of $\boldsymbol{q}_d$ is either in AC^0 or L-hard for data complexity. In the latter case, answering the original $\boldsymbol{q}$ is also L-hard. Otherwise, it is in NC^1, and its data complexity coincides with that of $\boldsymbol{q}_t$. This gives us the following classification:

Theorem 1. *Answering any temporal linear monadic datalog query is in NL for data complexity. Answering a connected linear query with the operators $\bigcirc/\bigcirc^{-1}$ only is either in AC^0 for data complexity, or in $ACC^0 \setminus AC^0$, or NC^1-complete, or L-hard.*

In terms of descriptive complexity, this theorem means that every temporal linear monadic datalog query can be rewritten to five types of first-order queries over data instances given as two-sorted structures (one sort for the object domain and the other for the temporal one): $\mathrm{FO}(<, \equiv)$-queries with unary predicates $t \equiv 0 \pmod{n}$, $\mathrm{FO}(<, \mathsf{MOD})$-queries with quantifiers $\exists^n t\, \psi(t)$ checking whether the number of moments of time satisfying ψ is divisible by n, $\mathrm{FO}(\mathrm{RPR})$-queries with relational primitive recursion, $\mathrm{FO}(\mathrm{DTC})$-queries with deterministic transitive closure, and $\mathrm{FO}(\mathrm{TC})$-queries with non-deterministic transitive closure. These extra predicates, quantifiers and operators complement FO($<$) with various types of recursion.

The ultimate question we would like to understand is how hard it is to extablish the *exact* data complexity of—and so the type of recursion required for—answering a given temporal linear (or arbitrary) monadic datalog query. In this paper, we provide partial answers to this question for connected linear queries. The decomposition of $\boldsymbol{q}$ into $\boldsymbol{q}_d$ and $\boldsymbol{q}_t$ mentioned above suggests a straightforward approach of utilising the respective algorithms for pure monadic datalog and pure *LTL*. For $\boldsymbol{q}_t$ we use the method of Kurucz et al. [4] as a black-box. For $\boldsymbol{q}_d$, we modify the method of Cosmadakis et al. [3] that employs an automata-theoretic criterion for boundedness. We use automata that work over a two-sorted alphabet distinguishing between the object domain and the temporal domain. The following theorem summarises our results.

Theorem 2. (i) *The problem of recognising whether answering a given connected linear temporal monadic datalog query* $\boldsymbol{q}$ *can be done in NC*^{1} *or is L-hard is*

- *PSpace-complete, if* $\boldsymbol{q}$ *contains occurrences of* $\bigcirc$ *(or* $\bigcirc^{-1}$ *) only*;
- *in ExpSpace, if* $\boldsymbol{q}$ *contains occurrences of both* $\bigcirc$ *and* $\bigcirc^{-1}$.

Recognising whether answering $\boldsymbol{q}$ *can be done in AC*^{0} *or in ACC*$^0 \setminus$ *AC*^{0} *or is NC*^{1}*-complete is in ExpSpace for* $\boldsymbol{q}$ *containing* $\bigcirc$ *(or* $\bigcirc^{-1}$ *) only, and in 2ExpSpace for* $\boldsymbol{q}$ *containing both* $\bigcirc$ *and* $\bigcirc^{-1}$.
(ii) *For* $\boldsymbol{q}$ *with* $\Diamond/\Diamond^{-1}$, *the problems in* (i) *are all undecidable.*

Our lower bound, PSpace-hardness, is obtained using the properties of *LTL* and contrasts with the only known lower bound for the case of connected queries in pure linear monadic datalog, which is coNP-hardness [5, 3].

Related Work

The first wave of research related to deciding the data complexity of datalog queries started in the mid 1980s, when the database community was working on optimisation and parallelisation of datalog programs; see [6, 7] and references therein. One of the fundamental problems considered was to decide FO-rewritability aka boundedness of any given datalog query. Boundedness was shown to be NP-complete for linear monadic single rule programs [8], PSpace-complete for linear monadic programs [9, 10], and 2ExpTime-complete for arbitrary monadic (single rule) programs [11, 12]. Boundedness of linear datalog queries with binary predicates and of ternary linear datalog queries with a single recursive rule was proved to be undecidable [13, 14].

The second wave in the 2000s was caused by the Web Ontology Language OWL and the idea of ontology-based data access, which brought large families of DLs that guarantee FO-rewritability [15, 16] (the DL-Lite family) and datalog-rewritability [17, 18] (the $\mathcal{EL}$ family) and [19] (the Horn-DL family). Other types of rule-based languages with FO-rewritability have also been identified, e.g., [20, 21, 22]. The problem of deciding the data complexity and rewritability type of large classes of ontology-mediated queries (OMQs) has been investigated since the 2010s. For instance, a complete characterisation of OMQs with an $\mathcal{EL}$-ontology was obtained in [23], establishing an AC^0/NL/P data complexity trichotomy, deciding which is ExpTime-complete. A crucial step in understanding this problem for non-Horn DLs was the discovery in [24, 25] of a connection between OMQs and non-uniform constraint satisfaction problems (CSPs) with a fixed template via MMSNP of [26]. It was used to show that deciding FO- and datalog-rewritability of OMQs with an ontology in any DL between $\mathcal{ALC}$ and $\mathcal{SHIU}$ and an atomic query is NExpTime-complete. The Feder-Vardi dichotomy of CSPs [27, 28] implies a P/coNP dichotomy of such OMQs, which is decidable in NExpTime. For monadic disjunctive datalog and OMQs with an $\mathcal{ALCI}$ ontology (that is, $\mathcal{ALC}$ with inverse roles) and a CQ, deciding FO-rewritability rises and becomes 2NExpTime-complete; deciding whether such an OMQ is rewritable to monadic datalog is between 2NExpTime and 3NExpTime [29, 25].

The survey [30] provides an overview of OMQ answering for various temporal logics and their combinations with DLs. It considers various fragments obtained by restricting Boolean and temporal operators used in ontologies and (uniform) complexity of OMQ answering for them. Any *LTL*-OMQ (without an object domain) is in NC^1 in data complexity and can be rewritten into $\mathrm{FO}(\mathrm{RPR})$ or $\mathrm{MSO}(<)$, monadic second-order logic with $<$. Such queries can also be rewritable into $\mathrm{FO}(<)$, $\mathrm{FO}(<, \equiv)$, or $\mathrm{FO}(<, \mathrm{MOD})$; see [4]. Checking for each of the latter classes if a given query is rewritable into it is ExpSpace-complete. The complexity remains the same even if the class of *LTL* ontologies is restricted to temporal Horn formulas and the queries are atomic. However, for a further restricted class with linear Horn ontologies checking rewritability into $\mathrm{FO}(<)$ and $\mathrm{FO}(<, \equiv)$ becomes PSpace-complete for negation-free queries (for atomic queries also $\mathrm{FO}(<, \mathrm{MOD})$-rewritability checking is PSpace-complete).

Finally, [31] studied the data complexity of queries mediated by a non-Horn temporal DL ontology with temporal operators on both concept and roles. For this logic, checking if the evaluation of an atomic ontology-mediated query is below coNP-hard is undecidable [32]. As shown in this paper, monadicity restores decidability when we only have the operators $\bigcirc/\bigcirc^{-1}$. However, the possibility to use arbitrary conjunctive queries as rule bodies still keeps the problem undecidable in the presence of $\Diamond/\Diamond^{-}$.

References

[1] S. Demri, V. Goranko, M. Lange, Temporal Logics in Computer Science, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2016.

[2] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, M. Zakharyaschev, First-order rewritability of ontology-mediated queries in linear temporal logic, Artificial Intelligence 299 (2021) 103536. doi:https://doi.org/10.1016/j.artint.2021.103536.

[3] S. Cosmadakis, H. Gaifman, P. Kanellakis, M. Vardi, Decidable optimization problems for database logic programs, in: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC '88, Association for Computing Machinery, New York, NY, USA, 1988, p. 477–490. doi:10.1145/62212.62259.

[4] A. Kurucz, V. Ryzhikov, Y. Savateev, M. Zakharyaschev, Deciding fo-rewritability of regular languages and ontology-mediated queries in linear temporal logic, J. Artif. Int. Res. 76 (2023). doi:10.1613/jair.1.14061.

[5] M. Y. Vardi, Decidability and undecidability results for boundedness of linear recursive queries, in: Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '88, Association for Computing Machinery, New York, NY, USA, 1988, p. 341–351. doi:10.1145/308386.308470.

[6] P. C. Kanellakis, Elements of relational database theory, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, Elsevier and MIT Press, 1990, pp. 1073–1156. URL: https://doi.org/10.1016/b978-0-444-88074-1.50022-6. doi:10.1016/b978-0-444-88074-1.50022-6.

[7] E. Dantsin, T. Eiter, G. Gottlob, A. Voronkov, Complexity and expressive power of logic programming, ACM Computing Surveys 33 (2001) 374–425.

[8] M. Y. Vardi, Decidability and undecidability results for boundedness of linear recursive queries, in: C. Edmondson-Yurkanan, M. Yannakakis (Eds.), Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 21-23, 1988, Austin, Texas, USA, ACM, 1988, pp. 341–351. URL: http://doi.acm.org/10.1145/308386.308470. doi:10.1145/308386.308470.

[9] S. S. Cosmadakis, H. Gaifman, P. C. Kanellakis, M. Y. Vardi, Decidable optimization problems for database logic programs (preliminary report), in: STOC, 1988, pp. 477–490.

[10] R. van der Meyden, Predicate boundedness of linear monadic datalog is in PSPACE, Int. J. Found. Comput. Sci. 11 (2000) 591–612. URL: https://doi.org/10.1142/S0129054100000351. doi:10.1142/S0129054100000351.

[11] M. Benedikt, B. ten Cate, T. Colcombet, M. Vanden Boom, The complexity of boundedness for guarded logics, in: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015, IEEE Computer Society, 2015, pp. 293–304. URL: https://doi.org/10.1109/LICS.2015.36. doi:10.1109/LICS.2015.36.

[12] S. Kikot, A. Kurucz, V. V. Podolskii, M. Zakharyaschev, Deciding boundedness of monadic sirups, in: L. Libkin, R. Pichler, P. Guagliardo (Eds.), PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Virtual Event, China, June 20-25, 2021, ACM, 2021, pp. 370–387. URL: https://doi.org/10.1145/3452021.3458332. doi:10.1145/3452021.3458332.

[13] G. G. Hillebrand, P. C. Kanellakis, H. G. Mairson, M. Y. Vardi, Undecidable boundedness problems for datalog programs, J. Log. Program. 25 (1995) 163–190. URL: https://doi.org/10.1016/0743-1066(95)00051-K. doi:10.1016/0743-1066(95)00051-K.

[14] J. Marcinkowski, Achilles, turtle, and undecidable boundedness problems for small DATALOG programs, SIAM J. Comput. 29 (1999) 231–257. URL: https://doi.org/10.1137/S0097539797322140. doi:10.1137/S0097539797322140.

[15] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family, J. Autom. Reasoning 39 (2007) 385–429.

[16] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-Lite family and relations, J. Artif. Intell. Res. (JAIR) 36 (2009) 1–69.

[17] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: L. P. Kaelbling, A. Saffiotti (Eds.), Proceedings of the 19th Int. Joint Conf. on Artificial Intelligence, IJCAI-05, Professional Book Center, 2005, pp. 364–369.

[18] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope further, in: K. Clark, P. F. Patel-Schneider (Eds.), Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions, 2008.

[19] U. Hustadt, B. Motik, U. Sattler, Data complexity of reasoning in very expressive description logics, in: L. P. Kaelbling, A. Saffiotti (Eds.), IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005, Professional Book Center, 2005, pp. 466–471. URL: http://ijcai.org/Proceedings/05/Papers/0326.pdf.

[20] A. Calì, G. Gottlob, A. Pieris, Towards more expressive ontology languages: The query answering problem, Artificial Intelligence 193 (2012) 87–128.

[21] J.-F. Baget, M. Leclère, M.-L. Mugnier, E. Salvat, On rules with existential variables: Walking the decidability line, Artificial Intelligence 175 (2011) 1620–1654.

[22] M. König, M. Leclère, M.-L. Mugnier, M. Thomazo, Sound, complete and minimal UCQ-rewriting for existential rules, Semantic Web 6 (2015) 451–475.

[23] C. Lutz, L. Sabellek, A complete classification of the complexity and rewritability of ontology-mediated queries based on the description logic EL, Artif. Intell. 308 (2022) 103709. URL: https://doi.org/10.1016/j.artint.2022.103709. doi:10.1016/J.ARTINT.2022.103709.

[24] M. Bienvenu, B. ten Cate, C. Lutz, F. Wolter, Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP, ACM Transactions on Database Systems 39 (2014) 33:1–44.

[25] C. Feier, A. Kuusisto, C. Lutz, Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics, Logical Methods in Computer Science 15 (2019). URL: https://doi.org/10.23638/LMCS-15(2:15)2019. doi:10.23638/LMCS-15(2:15)2019.

[26] T. Feder, M. Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory, SIAM J. Comput. 28 (1998) 57–104. URL: https://doi.org/10.1137/S0097539794266766. doi:10.1137/S0097539794266766.

[27] A. A. Bulatov, A dichotomy theorem for nonuniform CSPs, in: C. Umans (Ed.), 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, IEEE Computer Society, 2017, pp. 319–330. URL: https://doi.org/10.1109/FOCS.2017.37. doi:10.1109/FOCS.2017.37.

[28] D. Zhuk, A proof of CSP dichotomy conjecture, in: C. Umans (Ed.), 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, IEEE Computer Society, 2017, pp. 331–342. URL: https://doi.org/10.1109/FOCS.2017.38. doi:10.1109/

FOCS.2017.38.

[29] P. Bourhis, C. Lutz, Containment in monadic disjunctive datalog, MMSNP, and expressive description logics, in: C. Baral, J. P. Delgrande, F. Wolter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016, AAAI Press, 2016, pp. 207–216. URL: http://www.aaai.org/ocs/index.php/KR/KR16/paper/view/12847.

[30] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, M. Zakharyaschev, Ontology-mediated query answering over temporal data: A survey (invited talk), in: S. Schewe, T. Schneider, J. Wijsen (Eds.), 24th International Symposium on Temporal Representation and Reasoning, TIME 2017, October 16-18, 2017, Mons, Belgium, volume 90 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 1:1–1:37. URL: https://doi.org/10.4230/LIPIcs.TIME.2017.1. doi:10.4230/LIPIcs.TIME.2017.1.

[31] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, M. Zakharyaschev, First-order rewritability and complexity of two-dimensional temporal ontology-mediated queries, Journal of Artificial Intelligence Research 75 (2022) 1223–1291. doi:https://doi.org/10.1613/jair.1.13511.

[32] A. Artale, A. R. Gnatenko, V. Ryzhikov, M. Zakharyaschev, A decidable temporal DL-Lite logic with undecidable first-order and datalog-rewritability of ontology-mediated atomic queries (extended abstract), in: O. Kutz, C. Lutz, A. Ozaki (Eds.), Proceedings of the 36th International Workshop on Description Logics (DL 2023) co-located with the 20th International Conference on Principles of Knowledge Representation and Reasoning and the 21st International Workshop on Non-Monotonic Reasoning (KR 2023 and NMR 2023)., Rhodes, Greece, September 2-4, 2023, volume 3515 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3515/abstract-3.pdf.