Inconsistency Handling in DatalogMTL (Extended Abstract)

Meghyn Bienvenu[1], Camille Bourgaux[2] and Atefe Khodadaditaghanaki[3]

[1]Université de Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, Talence, France
[2]DI ENS, ENS, CNRS, PSL University & Inria, Paris, France
[3]University of Paderborn, Paderborn, Germany

Abstract

This extended abstract summarizes our IJCAI'25 paper [1] on inconsistency handling in DatalogMTL, an extension of Datalog with metric temporal operators. Our work extends existing notions of conflicts and repairs to DatalogMTL and studies their properties. We also study the data complexity of the tasks of generating a single conflict / repair and query entailment under repair-based semantics.

Keywords

DatalogMTL, inconsistency handling, repair-based semantics, query answering, complexity

1. Introduction

There has been significant recent interest in formalisms for reasoning over temporal data [2]. Since its introduction in [3], the DatalogMTL language, which extends Datalog [4] with operators from metric temporal logic (MTL) [5], has risen to prominence. In DatalogMTL, facts are annotated by *time intervals* on which they are valid (e.g., $R(a, b)@[1, 5]$), and rules express dependencies between such facts (e.g., $\boxplus_{[0,2]} Q \leftarrow \diamondminus_{\{3\}} P$ states that if P holds at time $t-3$, Q holds from t to $t+2$). The complexity of reasoning in DatalogMTL has been investigated for various fragments and extensions and for different semantics (continuous vs pointwise, rational vs integer timeline) [6, 7, 8, 9, 10, 11]. Moreover, there are also several implemented reasoning systems for (fragments of) DatalogMTL [12, 13, 14, 15, 16, 17].

One important issue that has yet to be addressed is how to handle the case where the temporal dataset is inconsistent with the DatalogMTL program. Indeed, it is widely acknowledged that real-world data typically contains many erroneous or inaccurate facts, and this is true in particular for temporal sensor data, due to faulty sensors. In such cases, classical logical semantics is rendered useless, as every query is entailed from a contradiction. A prominent approach to obtain meaningful information from an atemporal dataset that is inconsistent w.r.t. a logical theory (e.g., an ontology or a set of database integrity constraints) is to use *inconsistency-tolerant semantics* based on *subset repairs*, which are maximal subsets of the dataset consistent with the theory [18, 19]. The *consistent query answering (CQA)* approach considers that a (Boolean) query is true if it holds w.r.t. every repair [20, 21]. Other natural semantics have also been proposed, such as the *brave* semantics, under which a query is true if it holds w.r.t. at least one repair [22], and the *intersection* semantics which evaluates queries w.r.t. the intersection of all repairs [21]. It is also useful to consider the minimal subsets of the dataset that are inconsistent with the theory, which are commonly referred to as *conflicts*, in order to explain the inconsistency to a user or help with debugging.

It is natural to extend these notions to the temporal setting. First work in this direction was undertaken in [23], which considered queries with linear temporal logic (LTL) operators, an atemporal DL-Lite ontology, and a sequence of datasets stating what holds at different timepoints. In that work, however, it was clear how to transfer definitions from the atemporal setting, and the main concerns were complexity

DL 2025: 38th International Workshop on Description Logics, September 3-6, 2025, Opole, Poland
meghyn.bienvenu@u-bordeaux.fr (M. Bienvenu); camille.bourgaux@ens.fr (C. Bourgaux); atefe@mail.uni-paderborn.de (A. Khodadaditaghanaki)
https://www.labri.fr/perso/meghyn/ (M. Bienvenu); https://www.di.ens.fr/camille.bourgaux/ (C. Bourgaux)

and algorithms. By contrast, in DatalogMTL, facts are annotated with time intervals, which may contain exponentially or even infinitely many timepoints (if the timeline is dense or $\infty/-\infty$ can be used as interval endpoints). One can therefore imagine multiple different ways of minimally repairing an inconsistent dataset. For example, if a dataset states that P is true from 0 to 4 and Q from 2 to 6 ($P@[0,4]$, $Q@[2,6]$), and a rule states that P and Q cannot hold at the same time ($\bot \leftarrow P \wedge Q$), one can regain consistency by removing one of the two facts, adjusting their intervals, or treating intervals as sets of points and conserving as much information as possible. Similarly, there can be multiple ways of defining conflicts to identify minimal parts of the dataset responsible for inconsistency.

2. Repairs and Conflicts in DatalogMTL

Our first contribution is to explore how the basic notions of repair and conflict, which are well studied in the atemporal setting, can be suitably adapted to DatalogMTL. We first define three different notions of repair of a dataset $\mathcal{D}$ w.r.t. a DatalogMTL program Π. We omit the formal definitions, which are somewhat tedious, as we must put datasets into a normal form and consider different ways of manipulating and comparing sets of DatalogMTL facts. Instead we give the main intuitions. The timepoints t, t'_1, t'_2 considered below depend on the chosen timeline, typically $(\mathbb{Q}, \leq)$ or $(\mathbb{Z}, \leq)$.

Strong or s-repairs we view $\mathcal{D}$ as a set of (indivisible) facts and delete a minimal subset to regain consistency with Π, straightforwardly adapting the notion of subset repairs

Pointwise or p-repairs we view $\mathcal{D}$ as the possibly infinite set of punctual facts it represents,

$$\{R(\vec{a})@[t,t] \mid t \in [t_1, t_2], R(\vec{a})@[t_1, t_2] \in \mathcal{D}\},$$

then minimally remove punctual facts until consistency with Π is achieved

Interval-based or i-repairs we consider the datasets obtained from $\mathcal{D}$ by replacing each $R(\vec{a})@[t_1, t_2] \in \mathcal{D}$ by a fact $R(\vec{a})@[t'_1, t'_2]$ whose interval $[t'_1, t'_2]$ is included in $[t_1, t_2]$, or by nothing (we retain the option to delete a fact entirely), then compare such datasets w.r.t. how much information they retain, selecting the maximal ones consistent with Π

While p-repairs achieve the maximum preservation of information, an oft-desired property, they can lead to a single original fact being replaced by (possibly infinitely) many component facts, so repairs might be much larger in size than the original dataset. Both s- and i-repairs guarantee, by definition, that the number of facts does not increase, with i-repairs striking a nice balance between preserving information and respecting the structure of the original dataset. In the same way, we can define s-conflicts, p-conflicts, and i-conflicts of an inconsistent DatalogMTL knowledge base. Furthermore, we can use the new notions of repair to transfer existing definitions of repair-based semantics to DatalogMTL, yielding x-brave, x-CQA, and x-intersection semantics for $x \in \{s, p, i\}$.

We study the formal properties of these notions. While s-repairs and s-conflicts possess similar properties to their atemporal analogs, p- and i-conflicts and repairs behave rather differently. In particular, we show that p- and i-conflicts and repairs are not guaranteed to exist. Even when they do, p-repairs and p-conflicts might contain infinitely many facts, and some datasets might only give rise to p-repairs and p-conflicts of infinite size. Moreover, for both $x = i$ and $x = p$, there can be infinitely many x-repairs / x-conflicts. One way to circumvent these negative results is to adopt the $\mathbb{Z}$ timeline and restrict datasets to only use *bounded intervals* (i.e., finite integers as endpoints).

3. Data Complexity Analysis

Our second contribution is a data complexity analysis of the main computational tasks: recognizing x-conflicts and x-repairs, generating a single x-conflict or x-repair, and testing query entailment under the x-brave, x-CQA, and x-intersection semantics. For this initial study, we focus on the two cases where x-repairs are guaranteed to exist: (i) $x = s$, and (ii) bounded datasets over $\mathbb{Z}$.

We recall that in DatalogMTL, consistency checking and query entailment are PSpace-complete w.r.t. data complexity [7], and PSpace-completeness holds for many fragments (such as core and linear) [9] as well as for DatalogMTL over $\mathbb{Z}$ [24]. We also consider *tractable fragments* for which these tasks can be done in PTime in data complexity: $\text{Datalog}_{\text{nr}}\text{MTL}$, $\text{DatalogMTL}^{\diamondsuit}_{\text{core}}$, and $\text{DatalogMTL}^{\diamondsuit}_{\text{lin}}$ (over $\mathbb{Q}$ or $\mathbb{Z}$) and propositional DatalogMTL over $\mathbb{Z}$ [6, 9, 24].

We briefly summarize our results concerning s-repairs and s-conflicts. For arbitrary DatalogMTL programs, we obtain PSpace upper bounds for all tasks (and PSpace-completeness for the decision problems) by adapting known procedures for reasoning with subset repairs and conflicts in the atemporal setting. If we consider tractable DatalogMTL fragments, then we can show that the s-repair and s-conflict recognition are in PTime, and it is also in PTime to generate a single s-repair or s-conflict. We can use the PTime upper bounds on recognizing s-repairs to obtain (co)NP upper bounds for query entailment under s-brave, s-CQA, and s-intersection semantics for tractable DatalogMTL fragments. Moreover, we provide matching lower bounds for $\text{Datalog}_{\text{nr}}\text{MTL}$ and $\text{DatalogMTL}^{\diamondsuit}_{\text{lin}}$ (as well as for $\text{DatalogMTL}^{\diamondsuit}_{\text{core}}$ in the case of the s-CQA semantics), which hold even for bounded datasets and $\mathbb{T} = \mathbb{Z}$.

The hardness results for $\text{Datalog}_{\text{nr}}\text{MTL}$ are somewhat surprising in view of the AC^0 data complexity and FO$<$-rewritability of query entailment in $\text{Datalog}_{\text{nr}}\text{MTL}$, as a result from [22] shows how to transfer FO-rewritability results from classical to brave and intersection semantics. However, the latter result relies upon the fact that in the setting of atemporal ontologies, the existence of a rewriting guarantees a data-independent bound on the size of minimal inconsistent subsets and minimal query-entailing subsets. This property fails to hold in $\text{Datalog}_{\text{nr}}\text{MTL}$.

In $\text{DatalogMTL}^{\diamondsuit}_{\text{core}}$, by contrast, [9, 24] have shown that every minimal Π-inconsistent subset contains at most two facts, and query entailment can be traced back to a single fact. This is the key to showing that query entailment under s-brave and s-intersection semantics are in PTime for $\text{DatalogMTL}^{\diamondsuit}_{\text{core}}$. For propositional DatalogMTL, we even get tractability for s-CQA semantics. This is notable in view of the notorious intractability of CQA semantics even in restricted atemporal settings.

Let us also briefly summarize the preliminary results we obtained for bounded-interval datasets over $\mathbb{Z}$. For general DatalogMTL programs, we obtain PSpace upper bounds for all tasks concerning i-repairs and i-conflicts. We further show that when we consider tractable fragments, one can tractably recognize or generate an i-conflict, using binary search to identify optimal endpoints. The situation for pointwise notions is starkly different as even in this restricted setting, a single p-conflict or p-repair may be exponentially large.

4. Future Work

We see many relevant avenues for future work. First, there remain several open questions regarding the complexity of reasoning with i- and p-repairs and conflicts in the bounded-interval $\mathbb{Z}$ setting. We are most interested in trying to extend our tractability results for s-repair-based semantics to i-repairs and are reasonably optimistic that this can be done (but at the cost of significantly more technical constructions). A nice theoretical question is to consider the decidability of i- and p-repair / conflict existence in unrestricted settings. It would also be interesting to consider DatalogMTL with negation or spatio-temporal predicates. A more practical direction is to try to devise practical SAT- or SMT-based algorithms for the identified (co)NP cases, as has been done in some atemporal settings, cf. [25]. There are also further variants of our notions that are worth exploring, such as quantitative notions of x-repairs, e.g. to take into account how much the endpoints have been adjusted in an i-repair. Another natural direction would be to extend our study beyond DatalogMTL and explore how our proposed notions of repair and conflict can be used for reasoning with inconsistent temporal description logic knowledge bases whose assertions are annotated by time intervals.

Acknowledgments

This work was supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014) and the ANR PRAIRIE 3IA Institute (ANR-19-P3IA-0001).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Bienvenu, C. Bourgaux, A. Khodadaditaghanaki, Inconsistency handling in DatalogMTL, in: Proceedings of IJCAI (to appear), 2025.
[2] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, M. Zakharyaschev, Ontology-mediated query answering over temporal data: A survey (invited talk), in: Proceedings of TIME, 2017.
[3] S. Brandt, E. G. Kalayci, R. Kontchakov, V. Ryzhikov, G. Xiao, M. Zakharyaschev, Ontology-based data access with a horn fragment of metric temporal logic, in: Proceedings of AAAI, 2017.
[4] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
[5] R. Koymans, Specifying real-time properties with metric temporal logic, Real Time Syst. 2 (1990) 255–299.
[6] S. Brandt, E. G. Kalayci, V. Ryzhikov, G. Xiao, M. Zakharyaschev, Querying log data with metric temporal logic, J. Artif. Intell. Res. 62 (2018) 829–877.
[7] P. A. Walega, B. Cuenca Grau, M. Kaminski, E. V. Kostylev, DatalogMTL: Computational complexity and expressive power, in: Proceedings of IJCAI, 2019.
[8] V. Ryzhikov, P. A. Walega, M. Zakharyaschev, Data complexity and rewritability of ontology-mediated queries in metric temporal logic under the event-based semantics, in: Proceedings of IJCAI, 2019.
[9] P. A. Walega, B. Cuenca Grau, M. Kaminski, E. V. Kostylev, Tractable fragments of datalog with metric temporal operators, in: Proceedings of IJCAI, 2020.
[10] P. A. Walega, M. Kaminski, D. Wang, B. Cuenca Grau, Stream reasoning with DatalogMTL, J. Web Semant. 76 (2023) 100776.
[11] P. A. Walega, D. J. Tena Cucala, B. Cuenca Grau, E. V. Kostylev, The stable model semantics of datalog with metric temporal operators, Theory Pract. Log. Program. 24 (2024) 22–56.
[12] E. G. Kalayci, S. Brandt, D. Calvanese, V. Ryzhikov, G. Xiao, M. Zakharyaschev, Ontology-based access to temporal data with Ontop: A framework proposal, Int. J. Appl. Math. Comput. Sci. 29 (2019) 17–30.
[13] D. Wang, P. Hu, P. A. Walega, B. Cuenca Grau, MeTeoR: Practical reasoning in datalog with metric temporal operators, in: Proceedings of AAAI, 2022.
[14] D. Wang, P. A. Walega, P. Hu, B. Cuenca Grau, Practical reasoning in DatalogMTL, CoRR abs/2401.02869 (2024).
[15] L. Bellomarini, L. Blasi, M. Nissl, E. Sallinger, The temporal Vadalog system, in: Proceedings of RuleML+RR, 2022.
[16] P. A. Walega, M. Zawidzki, D. Wang, B. Cuenca Grau, Materialisation-based reasoning in DatalogMTL with bounded intervals, in: Proceedings of AAAI, 2023.
[17] A. Ivliev, L. Gerlach, S. Meusel, J. Steinberg, M. Krötzsch, Nemo: Your friendly and versatile rule reasoning toolkit, in: Proceedings of KR, 2024.
[18] L. E. Bertossi, Database repairs and consistent query answering: Origins and further developments, in: Proceedings of PODS, 2019.
[19] M. Bienvenu, A short survey on inconsistency handling in ontology-mediated query answering, Künstliche Intelligenz 34 (2020) 443–451.

[20] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proceedings of PODS, 1999.
[21] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant semantics for description logics, in: Proceedings of RR, 2010.
[22] M. Bienvenu, R. Rosati, Tractable approximations of consistent query answering for robust ontology-based data access, in: Proceedings of IJCAI, 2013.
[23] C. Bourgaux, P. Koopmann, A. Turhan, Ontology-mediated query answering over temporal and inconsistent data, Semantic Web 10 (2019) 475–521.
[24] P. A. Walega, B. Cuenca Grau, M. Kaminski, E. V. Kostylev, DatalogMTL over the integer timeline, in: Proceedings of KR, 2020.
[25] M. Bienvenu, C. Bourgaux, Querying inconsistent prioritized data with ORBITS: algorithms, implementation, and experiments, in: Proceedings of KR, 2022.