

Inconsistency Handling in DatalogMTL (Extended Abstract)

Meghyn Bienvenu¹, Camille Bourgaux², Atefe Khodadaditaghanaki³

¹Université de Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, Talence, France

²DI ENS, ENS, CNRS, PSL University & Inria, Paris, France

³University of Paderborn, Paderborn, Germany

meghyn.bienvenu@u-bordeaux.fr, camille.bourgaux@ens.fr, atefe@mail.uni-paderborn.de

Abstract

This extended abstract summarizes our IJCAI'25 paper [Bienvenu *et al.*, 2025] on inconsistency handling in DatalogMTL, an extension of Datalog with metric temporal operators.

Introduction There has been significant recent interest in formalisms for reasoning over temporal data [Artale *et al.*, 2017]. Since its introduction by Brandt *et al.* [2017; 2018], the DatalogMTL language, which extends Datalog [Abiteboul *et al.*, 1995] with operators from metric temporal logic (MTL) [Koymans, 1990], 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 may use atoms involving metric temporal logic operators like $\boxplus_{[t_1, t_2]} \varphi$ and $\boxminus_{[t_1, t_2]} \varphi$ (which hold at time t if φ holds at all, resp. some, timepoints in the interval $[t + t_1, t + t_2]$) to express dependencies between such facts (e.g., $\boxplus_{[0, 2]} Q \leftarrow \boxminus_{[3, 3]} P$ 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) [Brandt *et al.*, 2018; Walega *et al.*, 2019; Ryzhikov *et al.*, 2019; Walega *et al.*, 2020b; Walega *et al.*, 2023a; Walega *et al.*, 2024]. Moreover, there are also several implemented reasoning systems for (fragments of) DatalogMTL [Kalayci *et al.*, 2019; Wang *et al.*, 2022; Wang *et al.*, 2024; Bellomarini *et al.*, 2022; Walega *et al.*, 2023b; Ivliev *et al.*, 2024].

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 [Bertossi, 2019; Bienvenu, 2020]. The *consistent query answering (CQA)* approach considers that a (Boolean) query is true if it holds w.r.t. every repair [Arenas *et al.*, 1999; Lembo *et al.*, 2010].

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 [Bienvenu and Rosati, 2013], and the *intersection* semantics which evaluates queries w.r.t. the intersection of all repairs [Lembo *et al.*, 2010]. It is also useful to consider the minimal subsets of the dataset that are inconsistent with the theory, called *conflicts*, 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 by Bourgaux *et al.* [2019], who 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 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 ($\perp \leftarrow P \wedge Q$), 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.

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:

- **Subset or s -repairs:** we view $\mathcal{D}$ as a set of facts and delete a minimal subset to regain consistency with Π
- **Pointwise or p -repairs:** we view $\mathcal{D}$ as the possibly infinite set of the 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 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 we show that 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).

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 [Walega *et al.*, 2019], and PSPACE-completeness holds for many fragments (such as core and linear) [Walega *et al.*, 2020b] as well as for DatalogMTL over $\mathbb{Z}$ [Walega *et al.*, 2020a]. We also consider some *tractable fragments* for which these tasks can be performed in PTIME w.r.t. data complexity: Datalog_{nr}MTL, DatalogMTL_{core} $^\diamond$, and DatalogMTL_{lin} $^\diamond$ (over $\mathbb{Q}$ or $\mathbb{Z}$) and propositional DatalogMTL over $\mathbb{Z}$ [Brandt *et al.*, 2018; Walega *et al.*, 2020b; Walega *et al.*, 2020a].

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 Datalog_{nr}MTL and DatalogMTL_{lin} $^\diamond$ (as well as for DatalogMTL_{core} $^\diamond$ in the case of the s -CQA semantics), which hold even for bounded datasets and $\mathbb{T} = \mathbb{Z}$.

The hardness results for Datalog_{nr}MTL are somewhat surprising in view of the AC⁰ data complexity and FO<-rewritability of query entailment in Datalog_{nr}MTL, as a result from [Bienvenu and Rosati, 2013] 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 Datalog_{nr}MTL.

In DatalogMTL_{core} $^\diamond$, by contrast, Walega *et al.* [2020b; 2020a] 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 DatalogMTL_{core} $^\diamond$. For propositional DatalogMTL, we even get tractability for s -CQA semantics – notable in view of the notorious intractability of CQA semantics even in restricted atemporal settings.

Let us also briefly summarize our results 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.

Discussion and Future Work We initiated an investigation into how classical atemporal notions of repairs and conflicts can be suitably defined for inconsistent DatalogMTL knowledge bases, proposing three distinct approaches which differ on how the temporal intervals of facts can be modified. Our results show that s -repairs and s -conflicts retain desirable properties from the atemporal setting, which is why we made them the focus of our complexity analysis. In future work, we plan to analyze in more detail the complexity of reasoning with i -repairs and i -conflicts in the bounded-interval $\mathbb{Z}$ setting, as we believe the i -notions provide a nice tradeoff between the simplicity of the s -notions and preservation of information of the p -notions. Our results also raise several interesting theoretical questions, such as the decidability of i - and p -repair / conflict existence in unrestricted settings. It would also be worth exploring quantitative variants of our notions, e.g. to take into account how much interval endpoints have been adjusted in an i -repair. A more practical direction is to devise practical SAT- or SMT-based algorithms for the identified (co)NP cases, as has been done in some atemporal settings, cf. [Bienvenu and Bourgaux, 2022].

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).

References

- [Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [Arenas *et al.*, 1999] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of PODS*, 1999.
- [Artale *et al.*, 2017] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyashev. Ontology-mediated query answering over temporal data: A survey (invited talk). In *Proceedings of TIME*, 2017.
- [Bellomarini *et al.*, 2022] Luigi Bellomarini, Livia Blasi, Markus Nissl, and Emanuel Sallinger. The temporal Vada-log system. In *Proceedings of RuleML+RR*, 2022.
- [Bertossi, 2019] Leopoldo E. Bertossi. Database repairs and consistent query answering: Origins and further developments. In *Proceedings of PODS*, 2019.
- [Bienvenu and Bourgaux, 2022] Meghyn Bienvenu and Camille Bourgaux. Querying inconsistent prioritized data with ORBITS: algorithms, implementation, and experiments. In *Proceedings of KR*, 2022.
- [Bienvenu and Rosati, 2013] Meghyn Bienvenu and Riccardo Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proceedings of IJCAI*, 2013.
- [Bienvenu *et al.*, 2025] Meghyn Bienvenu, Camille Bourgaux, and Atefe Khodadaditaghanaki. Inconsistency handling in DatalogMTL, 2025. arxiv.org/abs/2505.10394 [cs.LO].
- [Bienvenu, 2020] Meghyn Bienvenu. A short survey on inconsistency handling in ontology-mediated query answering. *Künstliche Intelligenz*, 34(4):443–451, 2020.
- [Bourgaux *et al.*, 2019] Camille Bourgaux, Patrick Koopmann, and Anni-Yasmin Turhan. Ontology-mediated query answering over temporal and inconsistent data. *Semantic Web*, 10(3):475–521, 2019.
- [Brandt *et al.*, 2017] Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyashev. Ontology-based data access with a horn fragment of metric temporal logic. In *Proceedings of AAAI*, 2017.
- [Brandt *et al.*, 2018] Sebastian Brandt, Elem Güzel Kalayci, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyashev. Querying log data with metric temporal logic. *J. Artif. Intell. Res.*, 62:829–877, 2018.
- [Ivliev *et al.*, 2024] Alex Ivliev, Lukas Gerlach, Simon Meusel, Jakob Steinberg, and Markus Krötzsch. Nemo: Your friendly and versatile rule reasoning toolkit. In *Proceedings of KR*, 2024.
- [Kalayci *et al.*, 2019] Elem Güzel Kalayci, Sebastian Brandt, Diego Calvanese, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyashev. Ontology-based access to temporal data with Ontop: A framework proposal. *Int. J. Appl. Math. Comput. Sci.*, 29(1):17–30, 2019.
- [Koymans, 1990] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real Time Syst.*, 2(4):255–299, 1990.
- [Lembo *et al.*, 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of RR*, 2010.
- [Ryzhikov *et al.*, 2019] Vladislav Ryzhikov, Przemysław Andrzej Walega, and Michael Zakharyashev. Data complexity and rewritability of ontology-mediated queries in metric temporal logic under the event-based semantics. In *Proceedings of IJCAI*, 2019.
- [Walega *et al.*, 2019] Przemysław Andrzej Walega, Bernardo Cuenca Grau, Mark Kaminski, and Egor V. Kostylev. DatalogMTL: Computational complexity and expressive power. In *Proceedings of IJCAI*, 2019.
- [Walega *et al.*, 2020a] Przemysław Andrzej Walega, Bernardo Cuenca Grau, Mark Kaminski, and Egor V. Kostylev. DatalogMTL over the integer timeline. In *Proceedings of KR*, 2020.
- [Walega *et al.*, 2020b] Przemysław Andrzej Walega, Bernardo Cuenca Grau, Mark Kaminski, and Egor V. Kostylev. Tractable fragments of datalog with metric temporal operators. In *Proceedings of IJCAI*, 2020.
- [Walega *et al.*, 2023a] Przemysław Andrzej Walega, Mark Kaminski, Dingmin Wang, and Bernardo Cuenca Grau. Stream reasoning with DatalogMTL. *J. Web Semant.*, 76:100776, 2023.
- [Walega *et al.*, 2023b] Przemysław Andrzej Walega, Michal Zawidzki, Dingmin Wang, and Bernardo Cuenca Grau. Materialisation-based reasoning in DatalogMTL with bounded intervals. In *Proceedings of AAAI*, 2023.
- [Walega *et al.*, 2024] Przemysław Andrzej Walega, David J. Tena Cucala, Bernardo Cuenca Grau, and Egor V. Kostylev. The stable model semantics of datalog with metric temporal operators. *Theory Pract. Log. Program.*, 24(1):22–56, 2024.
- [Wang *et al.*, 2022] Dingmin Wang, Pan Hu, Przemysław Andrzej Walega, and Bernardo Cuenca Grau. MeTeoR: Practical reasoning in datalog with metric temporal operators. In *Proceedings of AAAI*, 2022.
- [Wang *et al.*, 2024] Dingmin Wang, Przemysław Andrzej Walega, Pan Hu, and Bernardo Cuenca Grau. Practical reasoning in DatalogMTL. *CoRR*, abs/2401.02869, 2024.