Inconsistency Handling in DatalogMTL

Meghyn Bienvenu[1] , **Camille Bourgaux**[2] , **Atefe Khodadaditaghanaki**[3]
[1]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, Talence, France
[2]DI ENS, ENS, CNRS, PSL University & Inria, Paris, France
[3]Paderborn University, Paderborn, Germany
meghyn.bienvenu@labri.fr, camille.bourgaux@ens.fr, atefe@mail.uni-paderborn.de

Abstract

In this paper, we explore the issue of inconsistency handling in DatalogMTL, an extension of Datalog with metric temporal operators. Since facts are associated with time intervals, there are different manners to restore consistency when they contradict the rules, such as removing facts or modifying their time intervals. Our first contribution is the definition of relevant notions of conflicts (minimal explanations for inconsistency) and repairs (possible ways of restoring consistency) for this setting and the study of the properties of these notions and the associated inconsistency-tolerant semantics. Our second contribution is a data complexity analysis of the tasks of generating a single conflict / repair and query entailment under repair-based semantics.

1 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 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 extensively 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 ($\bot \leftarrow P \wedge Q$), one can regain consistency by removing one of the two facts, adjusting their intervals, or treating intervals as their sets of points and conserving as much information as possible.

In this paper, we initiate the study of inconsistency handling in DatalogMTL. After some preliminaries, we formally introduce our framework in Section 3. We define three different notions of repair based upon deleting whole facts (s-repairs), punctual facts (p-repairs), or minimally shrinking the time intervals of facts (i-repairs), which give rise to the x-brave, x-CQA, and x-intersection semantics ($x \in \{s,p,i\}$).

Likewise, we define notions of s-, p-, and i-conflict, which capture different ways to characterize minimal reasons for inconsistency. In Section 4, we study the properties of these notions. In particular, we show that p- and i-conflicts and repairs are not guaranteed to exist or be finite. In Section 5, we explore the computational properties of our framework. We provide a fairly comprehensive account of the data complexity of recognizing s-conflicts and s-repairs, generating a single s-conflict or s-repair, and testing query entailment under the s-brave, s-CQA, and s-intersection semantics. We obtain tight complexity results for several DatalogMTL fragments and identify tractable cases. We further provide some first complexity results for the i-and p-based notions.

Proofs of all claims are given in [Bienvenu *et al.*, 2025].

2 Preliminaries: DatalogMTL

Intervals We consider a timeline $(\mathbb{T}, \leq)$, (we will consider $(\mathbb{Q}, \leq)$, which is dense, and $(\mathbb{Z}, \leq)$, which is not), and call the elements of $\mathbb{T}$ *timepoints*. An *interval* takes the form $\langle t_1, t_2 \rangle$, with $t_1, t_2 \in \mathbb{T} \cup \{-\infty, \infty\}$, bracket $\langle$ being $[$ or $($, and bracket $\rangle$ either $]$ or $)$, and denotes the set of timepoints

$$\{t \mid t \in \mathbb{T}, t_1 < t < t_2\} \cup \{t_1 \mid \text{if } \langle = [\} \cup \{t_2 \mid \text{if } \rangle =]\}.$$

A *punctual* interval has the form $[t, t]$ and will also be written $\{t\}$. A *range* ϱ is an interval with non-negative endpoints.

Syntax Let $\mathbf{P}$, $\mathbf{C}$ and $\mathbf{V}$ be three mutually disjoint countable sets of predicates, constants, and variables respectively. An *atom* is of the form $P(\vec{\tau})$ where $P \in \mathbf{P}$ and $\vec{\tau}$ is a tuple of *terms* from $\mathbf{C} \cup \mathbf{V}$ of matching arity. A *literal* A is an expression built according to the following grammar:

$$A ::= P(\vec{\tau}) \mid \top \mid \boxplus_\varrho A \mid \boxminus_\varrho A \mid \diamondplus_\varrho A \mid \diamondminus_\varrho A \mid A \,\mathcal{U}_\varrho A \mid A \,\mathcal{S}_\varrho A$$

where $P(\vec{\tau})$ is an atom and ϱ is a range. Intuitively, $\mathcal{S}$ stands for 'since', $\mathcal{U}$ for 'until', $\Diamond$ for 'eventually', and $\Box$ for 'always', with $+$ indicating the future and $-$ the past. A DatalogMTL *program* Π is a finite set of rules of the form

$$B \leftarrow A_1 \wedge \ldots \wedge A_k \quad \text{or} \quad \bot \leftarrow A_1 \wedge \ldots \wedge A_k \quad \text{with } k \geq 1$$

where each A_i is a literal and B is a literal not mentioning any 'non-deterministic' operators $\diamondplus_\varrho$, $\diamondminus_\varrho$, $\mathcal{U}_\varrho$, and $\mathcal{S}_\varrho$. We call $A_1 \wedge \ldots \wedge A_k$ the *body* of the rule, and B or $\bot$ its *head*. We assume that each rule is *safe*: each variable in its head occurs in its body, and this occurrence is not in a left operand of $\mathcal{S}$ or $\mathcal{U}$. A *(temporal) dataset* $\mathcal{D}$ is a finite set of *(temporal) facts* of the form $\alpha@\iota$, with α a ground atom (i.e., α does not contain any variable) and ι a non-empty interval.

Fragments A program is *propositional* if all its predicates are nullary. It is *core* if each of its rules is either of the form $\bot \leftarrow A_1 \wedge A_2$ or of the form $B \leftarrow A$. It is *linear* if each of its rules is either of the form $\bot \leftarrow A_1 \wedge A_2$ or of the form $B \leftarrow A_1 \wedge \ldots \wedge A_k$ where at most one A_i mentions some predicate that occurs in the head of some rule (intensional predicate). We denote by $\text{DatalogMTL}^{\diamondminus}_{\mathsf{core}}$ (resp. $\text{DatalogMTL}^{\diamondminus}_{\mathsf{lin}}$) the fragment where programs are core (resp. linear) and $\diamondminus$ is the only temporal operator allowed in literals. The relation $\prec$ of dependence between predicates is defined by $P \prec Q$ iff there is a rule with P in the head and Q in the body. A program is *non-recursive* if there is no predicate P such that $P \prec^+ P$, where $\prec^+$ is the transitive closure of $\prec$. We denote by $\text{Datalog}_{\text{nr}}\text{MTL}$ the fragment of non-recursive programs.

Semantics An *interpretation* $\mathfrak{M}$ specifies for each ground atom α and timepoint $t \in \mathbb{T}$ whether α is true at t. If α is true at t in $\mathfrak{M}$, we write $\mathfrak{M}, t \models \alpha$ and say that $\mathfrak{M}$ *satisfies* α at t. The satisfaction of a ground literal by $\mathfrak{M}$ at t is then defined inductively as follows.

$\mathfrak{M}, t \models \top$	for every $t \in \mathbb{T}$
$\mathfrak{M}, t \not\models \bot$	for every $t \in \mathbb{T}$
$\mathfrak{M}, t \models \boxplus_\varrho A$	if $\mathfrak{M}, s \models A$ for all s with $s - t \in \varrho$
$\mathfrak{M}, t \models \boxminus_\varrho A$	if $\mathfrak{M}, s \models A$ for all s with $t - s \in \varrho$
$\mathfrak{M}, t \models \diamondplus_\varrho A$	if $\mathfrak{M}, s \models A$ for some s with $s - t \in \varrho$
$\mathfrak{M}, t \models \diamondminus_\varrho A$	if $\mathfrak{M}, s \models A$ for some s with $t - s \in \varrho$
$\mathfrak{M}, t \models A \,\mathcal{U}_\varrho A'$	if $\mathfrak{M}, t' \models A'$ for some t' with $t' - t \in \varrho$ and $\mathfrak{M}, s \models A$ for all $s \in (t, t')$
$\mathfrak{M}, t \models A \,\mathcal{S}_\varrho A'$	if $\mathfrak{M}, t' \models A'$ for some t' with $t - t' \in \varrho$ and $\mathfrak{M}, s \models A$ for all $s \in (t', t)$

An interpretation $\mathfrak{M}$ is a *model* of a rule $H \leftarrow A_1 \wedge \ldots \wedge A_k$ if for every grounding assignment $\nu : \mathbf{V} \mapsto \mathbf{C}$, for every $t \in \mathbb{T}$, $\mathfrak{M}, t \models \nu(H)$ whenever $\mathfrak{M}, t \models \nu(A_i)$ for $1 \leq i \leq k$, where $\nu(B)$ denotes the ground literal obtained by replacing each $x \in \mathbf{V}$ by $\nu(x)$ in B. $\mathfrak{M}$ is a model of a program Π if it is a model of all rules in Π. It is a model of a fact $\alpha@\iota$ if $\mathfrak{M}, t \models \alpha$ for every $t \in \iota$, and it is a model of a (possibly infinite) set of facts $\mathcal{B}$ if it is a model of all facts in $\mathcal{B}$. A program Π is *consistent* if it has a model. A set of facts $\mathcal{B}$ is Π-*consistent* if there exists a model $\mathfrak{M}$ of both Π and $\mathcal{B}$, written $\mathfrak{M} \models (\mathcal{B}, \Pi)$. A program Π and set of facts $\mathcal{B}$ *entail* a fact $\alpha@\iota$, written $(\mathcal{B}, \Pi) \models \alpha@\iota$, if every model of both Π and $\mathcal{B}$ is also a model of $\alpha@\iota$. Finally, we write $\mathcal{B} \models \alpha@\iota$ if $(\mathcal{B}, \emptyset) \models \alpha@\iota$ and $\Pi \models \alpha@\iota$ if $(\emptyset, \Pi) \models \alpha@\iota$.

Queries A *DatalogMTL query* is a pair $(\Pi, q(\vec{v}, r))$ of a program Π and an expression $q(\vec{v}, r)$ of the form $Q(\vec{\tau})@r$, where $Q \in \mathbf{P}$, $\vec{v} = (v_1, \ldots, v_n)$ is a tuple of variables, $\vec{\tau}$ is a tuple of terms from $\mathbf{C} \cup \vec{v}$, and r is an *interval variable*. We may simply use $q(\vec{v}, r)$ as a query when the program has been specified. A *certain answer* to $(\Pi, q(\vec{v}, r))$ over a (possibly infinite) set of facts $\mathcal{B}$ is a pair $(\vec{c}, \iota)$ such that $\vec{c} = (c_1, \ldots, c_n)$ is a tuple of constants, ι is an interval and, for every $t \in \iota$ and every model $\mathfrak{M}$ of Π and $\mathcal{B}$, we have $\mathfrak{M}, t \models Q(\vec{\tau})_{[\vec{v} \leftarrow \vec{c}]}$, where $Q(\vec{\tau})_{[\vec{v} \leftarrow \vec{c}]}$ is obtained from $Q(\vec{\tau})$ by replacing each $v_i \in \vec{v}$ by the corresponding $c_i \in \vec{c}$.

We will illustrate the notions we introduce on a running example about a blood transfusion scenario.

Example 1. *In our scenario, we wish to query the medical records of blood transfusion recipients to detect patients who exhibited symptoms or risk factors of transfusion-related adverse reactions. For example, if a patient presents a fever during the transfusion or in the next four hours, while having a normal temperature for the past 24 hours, one can suspect a febrile non-haemolytic transfusion reaction (potential fnhtr). This is represented by the following rule, where, intuitively, x represents a patient and y a blood pouch:*

$$\mathsf{PotFnhtr}(x) \leftarrow \mathsf{Fever}(x) \wedge \boxminus_{(0,24]} \mathsf{NoFever}(x) \wedge \diamondminus_{[0,4]} \mathsf{GetBlood}(x, y)$$

Another rule detects more generally relevant fever episodes:

$$\mathsf{FevEp}(x) \leftarrow \mathsf{Fever}(x) \wedge \Diamond_{[0,24]}\,(\mathsf{NoFever}(x)\,\mathcal{U}_{\{5\}}\,\mathsf{GetBlood}(x,y))$$

A patient cannot have a fever and no fever at the same time:

$$\bot \leftarrow \mathsf{Fever}(x) \wedge \mathsf{NoFever}(x)$$

We may also wish to identify patients who once produced anti-D antibodies, as they are at risk for adverse reactions to some blood types. This is represented as follows.

$$\boxplus_{[0,\infty)}\,\mathsf{AntiDRisk}(x) \leftarrow \mathsf{PositiveAntiD}(x)$$

The following dataset provides information about a patient a who received transfusion from a blood pouch b, assuming that time 0 is the time they entered the hospital.

$$\mathcal{D} = \{\mathsf{PositiveAntiD}(a)@\{-90\}, \mathsf{GetBlood}(a,b)@[24,26], \mathsf{NoFever}(a)@[0,29), \mathsf{Fever}(a)@[29,34]\}$$

Let Π consist of the DatalogMTL rules above. One can check that $\mathcal{D}$ is Π-consistent, $(a, \{29\})$ is a certain answer to the query $\mathsf{PotFnhtr}(v)@r$, $(a, [29, 34])$ is a certain answer to $\mathsf{FevEp}(v)@r$, and $(a, [-90, \infty))$ to $\mathsf{AntiDRisk}(v)@r$.

3 Repairs and Conflicts on Time Intervals

In this section, we first define three kinds of repair and conflict for temporal datasets, then extend inconsistency-tolerant semantics to this context. Before delving into the formal definitions, we illustrate the impact of dealing with *time intervals*.

Example 2. *Let Π be the program from Example 1 and*

$$\mathcal{D} = \{\mathsf{PositiveAntiD}(a)@\{-90\}, \mathsf{GetBlood}(a,b)@[24,26], \mathsf{NoFever}(a)@[0,32], \mathsf{Fever}(a)@[14,18], \mathsf{Fever}(a)@[29,34]\}.$$

$\mathcal{D}$ is Π-inconsistent because in $\mathcal{D}$, the patient a has both fever and no fever at $t \in [14,18] \cup [29,32]$. To repair the data by removing facts from $\mathcal{D}$, there are only two minimal possibilities: either remove $\mathsf{NoFever}(a)@[0,32]$, or remove both $\mathsf{Fever}(a)@[14,18]$ and $\mathsf{Fever}(a)@[29,34]$. This may be considered too drastic, since, e.g., the Fever *facts do not contradict that the patient had no fever during $[0,14)$ or $(18,29)$.*

Hence, it may seem preferable to consider each time-point independently, so that a repair may contain, e.g., the two Fever *facts as well as $\mathsf{NoFever}(a)@[0,14)$ and $\mathsf{NoFever}(a)@(18,29)$. However, with this approach, if $\mathbb{T} = \mathbb{Q}$, there are infinitely many possibilities to repair the dataset, and the number of facts in a repair may be infinite. For example, an option to repair the* Fever *and* NoFever *facts is:*

$$\{\mathsf{NoFever}(a)@[0,29), \mathsf{Fever}(a)@[30,34], \mathsf{NoFever}(a)@[29+\frac{1}{2^{2k+1}}, 29+\frac{1}{2^{2k}}), \mathsf{Fever}(a)@[29+\frac{1}{2^{2k+2}}, 29+\frac{1}{2^{2k+1}}) \mid k \in \mathbb{N}\}.$$

An intermediate approach consists in only modifying the endpoints of intervals, in order to keep more information than with fact deletion without splitting one fact into many. Again we may obtain infinitely many possibilities, e.g., the Fever *and* NoFever *facts can be repaired by $\mathsf{NoFever}(a)@[0,t)$ and $\mathsf{Fever}(a)@[t,34]$ for $t \in [29,32]$.*

Manipulating sets of temporal facts To formalize conflicts and repairs of temporal datasets, we consider three ways of comparing (possibly infinite) sets of facts w.r.t. inclusion:

Definition 1 (Pointwise inclusion, subset comparison)**.** *We say that a fact $\alpha@\iota$ is* pointwise included *in a set of facts $\mathcal{B}$ if for every $t \in \iota$, there is $\alpha@\iota' \in \mathcal{B}$ with $t \in \iota'$, i.e., if $\mathcal{B} \models \alpha@\iota$. Given sets of facts $\mathcal{B}$ and $\mathcal{B}'$, we say that $\mathcal{B}'$ is*

- *a* pointwise subset *of $\mathcal{B}$, denoted $\mathcal{B}' \sqsubseteq^p \mathcal{B}$, if every $\alpha@\iota \in \mathcal{B}'$ is pointwise included in $\mathcal{B}$;*
- *an* interval-based subset *of $\mathcal{B}$, denoted $\mathcal{B}' \sqsubseteq^i \mathcal{B}$, if $\mathcal{B}' \sqsubseteq^p \mathcal{B}$ and for every $\alpha@\iota \in \mathcal{B}$, there is* at most one *$\alpha@\iota' \in \mathcal{B}'$ such that $\iota' \subseteq \iota$;*
- *a* strong subset *of $\mathcal{B}$, written $\mathcal{B}' \sqsubseteq^s \mathcal{B}$, if $\mathcal{B}' \sqsubseteq^i \mathcal{B}$ and $\mathcal{B}' \subseteq \mathcal{B}$.*

We write $\mathcal{B}' \sqsubset^p \mathcal{B}$ to indicate that $\mathcal{B}' \sqsubseteq^p \mathcal{B}$ and $\mathcal{B} \not\sqsubseteq^p \mathcal{B}'$. For $x \in \{i, s\}$, we write $\mathcal{B}' \sqsubset^x \mathcal{B}$ if $\mathcal{B}' \sqsubseteq^x \mathcal{B}$ and $\mathcal{B}' \sqsubset^p \mathcal{B}$.

We also need to intersect (possibly infinite) sets of facts:

Definition 2 (Pointwise intersection)**.** *The* pointwise intersection *of a family $(\mathcal{B}_i)_{i\in I}$ of sets of facts is $\bigsqcap_{i\in I} \mathcal{B}_i = \{\alpha@\{t\} \mid \mathcal{B}_i \models \alpha@\{t\}$ for each $i \in I\}$. The* pointwise intersection *of a fact $\alpha@\iota$ and a set of facts $\mathcal{B}$ is $\{\alpha@\iota\} \sqcap \mathcal{B}$.*

Normal form A (possibly infinite) set of facts $\mathcal{B}$ is in *normal form* if for every pair of facts $\alpha@\iota$ and $\alpha@\iota'$ over the same ground atom, if $\alpha@\iota$ and $\alpha@\iota'$ are in $\mathcal{B}$, then $\iota \cup \iota'$ is not an interval.

Lemma 1. *If $\mathcal{B}$ is in normal form, then (1) $\mathcal{B}' \sqsubseteq^s \mathcal{B}$ iff $\mathcal{B}' \subseteq \mathcal{B}$, and (2) $\mathcal{B}' \sqsubseteq^i \mathcal{B}$ implies that the cardinality of $\mathcal{B}'$ is bounded by that of $\mathcal{B}$.*

To see why normal form is necessary, consider (1) $\mathcal{B} = \{P@[0,4], P@[1,2]\}$, which is such that $\mathcal{B} \not\sqsubseteq^i \mathcal{B}$, so that $\mathcal{B} \not\sqsubseteq^s \mathcal{B}$, and (2) $\mathcal{B} = \{P@[0,4], P@[3,7]\}$, which is such that $\{P@[0,1], P@[2,5], P@[6,7]\} \sqsubseteq^i \mathcal{B}$.

For every dataset $\mathcal{D}$, there exists a dataset $\mathcal{D}'$ in normal form such that for every $t \in \mathbb{T}$, for every ground atom α, $\mathcal{D} \models \alpha@\{t\}$ iff $\mathcal{D}' \models \alpha@\{t\}$. Moreover, such $\mathcal{D}'$ can be computed in polynomial time w.r.t. the size of $\mathcal{D}$ by merging every $\alpha@\iota_1$ and $\alpha@\iota_2$ such that $\iota_1 \cup \iota_2$ is an interval into $\alpha@\iota$ with $\iota = \iota_1 \cup \iota_2$. In the rest of this paper, we assume that *all datasets are in normal form* and *all programs are consistent*.

Conflicts, repairs, and inconsistency-tolerant semantics We are now ready to formally state the definitions of conflicts and repairs of a temporal dataset w.r.t. a DatalogMTL program. We start with the notion of conflict, which is crucial to explain inconsistency.

Definition 3 (Conflicts)**.** *Let Π be a DatalogMTL program and $\mathcal{D}$ be a dataset. Given $x \in \{p, i, s\}$, a set of facts $\mathcal{C}$ is an x-*conflict *of $\mathcal{D}$ w.r.t. Π if $\mathcal{C}$ is in normal form, $\mathcal{C} \sqsubseteq^x \mathcal{D}$, $\mathcal{C}$ is Π-inconsistent, and there is no Π-inconsistent $\mathcal{C}' \sqsubset^x \mathcal{C}$. We denote by $xConf(\mathcal{D}, \Pi)$ the set of all x-conflicts of $\mathcal{D}$ w.r.t. Π.*

Example 3. *Consider Π and $\mathcal{D}$ from Example 2. The s-conflicts are $\{\mathsf{NoFever}(a)@[0,32], \mathsf{Fever}(a)@[14,18]\}$ and $\{\mathsf{NoFever}(a)@[0,32], \mathsf{Fever}(a)@[29,34]\}$, while the p-conflicts and i-conflicts are of the form $\{\mathsf{NoFever}(a)@\{t\}, \mathsf{Fever}(a)@\{t\}\}$ with $t \in [14,18] \cup [29,32]$.*

We define repairs in a similar manner.

Definition 4 (Repairs)**.** *Let Π be a DatalogMTL program and $\mathcal{D}$ be a dataset. Given $x \in \{p, i, s\}$, a set of facts $\mathcal{R}$ is an* x-repair *of $\mathcal{D}$ w.r.t. Π if $\mathcal{R}$ is in normal form, $\mathcal{R} \sqsubseteq^x \mathcal{D}$, $\mathcal{R}$ is Π-consistent, and there is no Π-consistent $\mathcal{R}'$ such that $\mathcal{R} \sqsubset^x \mathcal{R}' \sqsubseteq^x \mathcal{D}$. We denote by $xRep(\mathcal{D}, \Pi)$ the set of all x-repairs of $\mathcal{D}$ w.r.t. Π.*

The requirement that x-repairs are in normal form ensures that when $\mathcal{D}$ is Π-consistent, $xRep(\mathcal{D}, \Pi) = \{\mathcal{D}\}$.

Example 4. *Π and $\mathcal{D}$ from Example 2 have two s-repairs:*

$\mathcal{R}_1 = \mathcal{I} \cup \{\mathsf{NoFever}(a)@[0, 32]\}$ *and*

$\mathcal{R}_2 = \mathcal{I} \cup \{\mathsf{Fever}(a)@[14, 18], \mathsf{Fever}(a)@[29, 34]\}$ *with*

$\mathcal{I} = \{\mathsf{PositiveAntiD}(a)@\{-90\}, \mathsf{GetBlood}(a, b)@[24, 26]\}$.

Every p-repair $\mathcal{R}$ is such that $\mathcal{J} \sqsubseteq^p \mathcal{R}$ with

$$\mathcal{J} = \mathcal{I} \cup \{\mathsf{NoFever}(a)@[0, 14), \mathsf{NoFever}(a)@(18, 29), \mathsf{Fever}(a)@(32, 34]\}$$

and for every $t \in [14, 18] \cup [29, 32]$, either $\mathsf{Fever}(a)@\{t\}$ or $\mathsf{NoFever}(a)@\{t\}$ is pointwise included in $\mathcal{R}$.
Finally, every i-repair $\mathcal{R}$ is such that $\mathcal{I} \subseteq \mathcal{R}$ and contains:

- *either two facts $\mathsf{NoFever}(a)@[0, t\rangle$, $\mathsf{Fever}(a)@\langle t, 34]$, where $\rangle, \langle$ are either $], ($ or $), [$ and $t \in [29, 32]$;*
- *or three facts $\mathsf{NoFever}(a)@[0, t\rangle$, $\mathsf{Fever}(a)@\langle t, 18]$, and $\mathsf{Fever}(a)@[29, 34]$, where $t \in [14, 18]$,*
  - *$\rangle, \langle$ are either $], ($ or $), [$ and*
  - *if $t = 18$, then $\rangle, \langle$ are $), [$;*
- *or three facts $\mathsf{Fever}(a)@[14, t_1\rangle$, $\mathsf{NoFever}(a)@\langle t_1, t_2\rangle'$, $\mathsf{Fever}(a)@\langle' t_2, 34]$, where $t_1 \in [14, 18]$, $t_2 \in [29, 32]$,*
  - *$\rangle, \langle$ and $\rangle', \langle'$ are either $], ($ or $), [$,*
  - *if $t_1 = 14$, then $\rangle, \langle$ are $], ($.*

We can now extend the definitions of the brave, CQA and intersection semantics to use different kinds of repairs.

Definition 5. *Consider a DatalogMTL query $(\Pi, q(\vec{v}, r))$, dataset $\mathcal{D}$, tuple $\vec{c}$ of constants from $\mathcal{D}$ with $|\vec{c}| = |\vec{v}|$, and interval ι. Given $x \in \{p, i, s\}$ such that $xRep(\mathcal{D}, \Pi) \neq \emptyset$, we say that $\vec{c}$ is an* answer *to $(\Pi, q(\vec{v}, r))$ under*

- *x-*brave semantics*, written $(\mathcal{D}, \Pi) \models^x_{brave} q(\vec{c}, \iota)$, if $(\mathcal{R}, \Pi) \models q(\vec{c}, \iota)$ for some $\mathcal{R} \in xRep(\mathcal{D}, \Pi)$;*
- *x-*CQA semantics*, written $(\mathcal{D}, \Pi) \models^x_{CQA} q(\vec{c}, \iota)$, if $(\mathcal{R}, \Pi) \models q(\vec{c}, \iota)$ for every $\mathcal{R} \in xRep(\mathcal{D}, \Pi)$;*
- *x-*intersection semantics*, written $(\mathcal{D}, \Pi) \models^x_{\cap} q(\vec{c}, \iota)$, if $(\mathcal{I}, \Pi) \models q(\vec{c}, \iota)$ where $\mathcal{I} = \bigcap_{\mathcal{R} \in xRep(\mathcal{D}, \Pi)} \mathcal{R}$.*

Proposition 1. *For every query $(\Pi, q(\vec{v}, r))$, dataset $\mathcal{D}$, tuple of constants $\vec{c}$, and interval ι, $(\mathcal{D}, \Pi) \models^x_{\cap} q(\vec{c}, \iota)$ implies $(\mathcal{D}, \Pi) \models^x_{CQA} q(\vec{c}, \iota)$, which implies $(\mathcal{D}, \Pi) \models^x_{brave} q(\vec{c}, \iota)$. None of the converse implications holds.*

Example 5. *Consider Π and $\mathcal{D}$ from Example 2. By examining the s-repairs given in Example 4, we can check that:*

- $(\mathcal{D}, \Pi) \models^s_{\cap} \mathsf{AntiDRisk}(a)@[-90, \infty)$,
- $(\mathcal{D}, \Pi) \not\models^s_{brave} \mathsf{FevEp}(a)@\{t\}$ *for every* $t \in \mathbb{T}$,
- $(\mathcal{D}, \Pi) \not\models^s_{brave} \mathsf{PotFnhtr}(a)@\{t\}$ *for every* $t \in \mathbb{T}$.

With the p-repairs (Example 4), we obtain that:

- $(\mathcal{D}, \Pi) \models^p_{\cap} \mathsf{AntiDRisk}(a)@[-90, \infty)$,
- $(\mathcal{D}, \Pi) \models^p_{\cap} \mathsf{FevEp}(a)@(32, 34]$,
- $(\mathcal{D}, \Pi) \models^p_{brave} \mathsf{PotFnhtr}(a)@\{t\}$ *for all* $t \in [29, 30]$,
- $(\mathcal{D}, \Pi) \not\models^p_{CQA} \mathsf{PotFnhtr}(a)@\{t\}$ *for every* $t \in \mathbb{T}$.

From the form of the i-repairs (Example 4), we obtain that:

- $(\mathcal{D}, \Pi) \models^i_{\cap} \mathsf{AntiDRisk}(a)@[-90, \infty)$,
- $(\mathcal{D}, \Pi) \models^i_{brave} \mathsf{FevEp}(a)@[29, 34]$,
- $(\mathcal{D}, \Pi) \not\models^i_{CQA} \mathsf{FevEp}(a)@\{t\}$ *for each* $t \in \mathbb{T}$,
- $(\mathcal{D}, \Pi) \models^i_{brave} \mathsf{PotFnhtr}(a)@\{t\}$ *for all* $t \in [29, 30]$,
- $(\mathcal{D}, \Pi) \not\models^i_{CQA} \mathsf{PotFnhtr}(a)@\{t\}$ *for each* $t \in \mathbb{T}$.

4 Properties of the Framework

We study properties of x-conflicts, x-repairs, and semantics based upon them. The results hold for $\mathbb{T} = \mathbb{Q}$ and $\mathbb{T} = \mathbb{Z}$.

4.1 Properties of Repairs and Conflicts

We will consider in particular the following properties, which are well known in the case of atemporal knowledge bases.

Definition 6. *We say that P_i holds if it holds for every dataset $\mathcal{D}$ (in normal form) and (consistent) program Π.*

P_1**:** $xRep(\mathcal{D}, \Pi) \neq \emptyset$.

P_2**:** *$\mathcal{D}$ is Π-inconsistent iff $xConf(\mathcal{D}, \Pi) \neq \emptyset$.*

P_3**:** $xRep(\mathcal{D}, \Pi)$ *and* $xConf(\mathcal{D}, \Pi)$ *are finite.*

P_4**:** *Every $\mathcal{B} \in xRep(\mathcal{D}, \Pi) \cup xConf(\mathcal{D}, \Pi)$ is finite.*

P_5**:** *For every fact $\alpha@\iota$ pointwise included in $\mathcal{D}$, $\alpha@\iota$ is pointwise included in every x-repair of $\mathcal{D}$ w.r.t. Π iff $\alpha@\iota$ has an empty pointwise intersection with every x-conflict of $\mathcal{D}$ w.r.t. Π.*

The notions based on $\sqsubseteq^s$ have all these properties, while those based on $\sqsubseteq^p$ do not have any, and those based on $\sqsubseteq^i$ only one (i-repairs and i-conflicts are finite by Lemma 1).

Proposition 2. *Properties P_1-P_5 hold for $x = s$.*

Corollary 1. $\bigcap_{\mathcal{R} \in sRep(\mathcal{D}, \Pi)} \mathcal{R} = \mathcal{D} \setminus \bigcup_{\mathcal{C} \in sConf(\mathcal{D}, \Pi)} \mathcal{C}$.

Proposition 3. *None of the properties P_1-P_5 hold for $x = p$. For $x = i$, P_4 holds but properties P_1-P_3 and P_5 do not.*

In what follows, we will provide the counterexamples used to prove Proposition 3, as well as additional examples that illustrate the properties of x-repairs and x-conflicts.

Existence of p- and i-Repairs and Conflicts

A major difference between repairs and conflicts based on $\sqsubseteq^s$ and those based on $\sqsubseteq^p$ or $\sqsubseteq^i$ is that the latter need not exist.

Example 6. *Consider the following dataset and program.*

$$\mathcal{D} = \{P@(0, \infty)\} \qquad \Pi = \{\bot \leftarrow \boxplus_{(0,\infty)} P\}$$

There is no p- or i-repair and no p- or i-conflict of $\mathcal{D}$ w.r.t. Π.

For $x \in \{p, i\}$, every Π-inconsistent $\mathcal{C} \sqsubseteq^x \mathcal{D}$ in normal form is of the form $\{P@\langle t, \infty)\}$. Since $\mathcal{C}' = \{P@(t+1, \infty)\}$ is Π-inconsistent and $\mathcal{C}' \sqsubset^x \mathcal{C}$, then $\mathcal{C}$ is not an x-conflict.

Every $\mathcal{R} \sqsubset^i \mathcal{D}$ is either empty (hence not an i-repair since, e.g., $\{P@\{1\}\}$ is Π-consistent) or of the form $\{P@\langle t_1, t_2\rangle\}$ with $\langle t_1, t_2\rangle \neq \emptyset$. If $t_2 = \infty$, $\mathcal{R}$ is Π-inconsistent. Otherwise, $\mathcal{R}' = \{P@\langle t_1, t_2 + 1\rangle\}$ is Π-consistent and $\mathcal{R} \sqsubset^i \mathcal{R}' \sqsubseteq^i \mathcal{D}$. In both cases, $\mathcal{R}$ is not an i-repair.

For every $\mathcal{R} \sqsubseteq^p \mathcal{D}$ in normal form, if there is only one $t \in (0, \infty)$ such that $\mathcal{R} \not\models P@\{t\}$, then $\mathcal{R}$ contains $P@(t, \infty)$ so $\mathcal{R}$ is Π-inconsistent. Hence, for every Π-consistent $\mathcal{R} \sqsubseteq^p \mathcal{D}$, there exist $t_1, t_2 \in (0, \infty)$ such that $t_1 < t_2$ and $\mathcal{R} \not\models P@\{t_1\}$, $\mathcal{R} \not\models P@\{t_2\}$. However, $\mathcal{R}' = \mathcal{R} \cup \{P@\{t_1\}\}$ is then Π-consistent and $\mathcal{R} \sqsubset^p \mathcal{R}' \sqsubseteq^p \mathcal{D}$ so $\mathcal{R}$ is not a p-repair.

Example 7 shows that there is no relationship between the existence of x-conflicts and the existence of x-repairs.

Example 7. *Let $\mathcal{D}_c = \mathcal{D} \cup \{R@\{0\}\}$ and $\Pi_c = \Pi \cup \{\bot \leftarrow R\}$ with $\mathcal{D}$ and Π from Example 6. We can show as in Example 6 that for $x \in \{p, i\}$, there is no x-repair of $\mathcal{D}_c$ w.r.t. Π_c. However, $\{R@\{0\}\}$ is an x-conflict of $\mathcal{D}_c$ w.r.t. Π_c. Now, let*

$$\mathcal{D}_r = \{P@[0, \infty), Q@\{0\}\}$$
$$\Pi_r = \{\bot \leftarrow Q \wedge \diamondminus_{(0,\infty)} \boxplus_{(0,\infty)} P\}.$$

For $x \in \{p, i\}$, there is no x-conflict of $\mathcal{D}_r$ w.r.t. Π_r. Indeed, every Π_r-inconsistent $\mathcal{C} \sqsubseteq^p \mathcal{D}$ has to be such that $\mathcal{C} \models P@(t, \infty)$ for some $t > 0$ and none of such $\mathcal{C}$ is minimal w.r.t. $\sqsubseteq^x$. Yet, $\{P@[0, \infty)\}$ is an x-repair of $\mathcal{D}_r$ w.r.t. Π_r.

The next examples show there is no relationship between the existence of p-repairs and the existence of i-repairs, nor between existence of p-conflicts and existence of i-conflicts.

Example 8. *The following $\mathcal{D}_i$ and Π_i have no p-repair (cf. Example 6) but $\{P@(-2, 0), Q@\{0\}\}$ is an i-repair.*

$$\mathcal{D}_i = \{P@(-2, \infty), Q@\{0\}\}$$
$$\Pi_i = \{\bot \leftarrow \boxplus_{(0,\infty)} P,\ \bot \leftarrow Q \wedge P\}$$

In the other direction, let $\mathcal{D}_p = \{P@(-\infty, \infty), Q@\{0\}\}$ and

$$\Pi_p = \{\bot \leftarrow \boxplus_{[0,\infty)} P,\ \bot \leftarrow \boxminus_{[0,\infty)} P,\ \bot \leftarrow P \wedge Q,$$
$$\bot \leftarrow Q \wedge \boxminus_{(0,10)} P \wedge \diamondminus_{[10,\infty)} P,$$
$$\bot \leftarrow Q \wedge \boxplus_{(0,10)} P \wedge \diamondminus_{[10,\infty)} P\}.$$

One can check that $\{Q@\{0\}, P@(-10, 0), P@(0, 10)\}$ is a p-repair, but one can show that there is no i-repair.

Example 9. *$\mathcal{D}_i = \{P@[0, \infty), Q@\{0\}\}$ is an i-conflict of itself w.r.t. $\Pi_i = \{\bot \leftarrow P \wedge Q \wedge \diamondminus_{(0,\infty)} \boxplus_{(0,\infty)} P\}$. However, there is no p-conflict of $\mathcal{D}_i$ w.r.t. Π_i. Indeed, every Π_i-inconsistent dataset $\mathcal{C} \sqsubseteq^p \mathcal{D}_i$ in normal form has the form $\{Q@\{0\}, P@\{0\}, P@\langle t, \infty)\}$, and $\{Q@\{0\}, P@\{0\}, P@\langle t + 1, \infty)\}$ is also Π_i-inconsistent.*

In the other direction, let $\mathcal{D}_p = \{P@[0, \infty), Q@\{0\}\}$ and

$$\Pi_p = \{\bot \leftarrow \boxplus_{(0,\infty)} P,\ \bot \leftarrow Q \wedge \boxplus_{[0,\infty)} \diamondminus_{[0,1)} P\}.$$

One can easily check that $\{Q@\{0\}, P@\{k\} \mid k \in \mathbb{N}\}$ is a p-conflict, but one can show that there is no i-conflict.

Size and Number of p- and i-Repairs and Conflicts

It follows from Lemma 1 that the i-repairs and i-conflicts of a dataset $\mathcal{D}$ w.r.t. a program Π contain at most as many facts as $\mathcal{D}$, hence are finite. In contrast, we have seen in Example 2 that a p-repair may be infinite. Example 10 shows that some datasets have *only* infinite p-repairs w.r.t. some programs, and Example 11 shows a similar result for p-conflicts.

Example 10. *Consider the following dataset and program.*

$$\mathcal{D} = \{P@(0, \infty)\} \qquad \Pi = \{\bot \leftarrow \boxplus_{[0,2]} P\}$$

There exist p-repairs of $\mathcal{D}$ w.r.t. Π, such as $\{P@(2k, 2k+2) \mid k \in \mathbb{N}\}$, but one can show that they are all infinite.

Example 11. *Consider the following dataset and program.*

$$\mathcal{D} = \{P@[0, \infty), Q@\{0\}\}$$
$$\Pi = \{\bot \leftarrow Q \wedge \boxplus_{[0,\infty)} \diamondminus_{[0,2)} P\}$$

There are p-conflicts of $\mathcal{D}$ w.r.t. Π, such as $\{Q@\{0\}, P@\{2k\} \mid k \in \mathbb{N}\}$, but one can show that they are all infinite.

Moreover, for both $x = i$ and $x = p$, there can be infinitely many x-repairs / x-conflicts:

Example 12. *The following $\mathcal{D}$ and Π have infinitely many p- and i- repairs and conflicts even if the timeline is $(\mathbb{Z}, \leq)$:*

$$\mathcal{D} = \{P@[0, \infty), Q@[0, \infty)\} \qquad \Pi = \{\bot \leftarrow P \wedge Q\}.$$

Indeed, for every $t \in [0, \infty)$, $\{P@\{t\}, Q@\{t\}\}$ is a p- and an i-conflict, and $\{P@[0, t), Q@[t, \infty)\}$ is a p- and an i-repair.

Absence of Link Between p/i- Repairs and Conflicts

Example 13 shows that a fact may be pointwise included in all p-, or i-, repairs while it is also pointwise included in a p-, or i-, conflict, respectively, and, symmetrically, that a fact may have an empty pointwise intersection with all p-, or i-, conflicts but also with some p-, or i-, repair.

Example 13. *Consider $\mathcal{D}_i$ and Π_i defined in Example 8. There is only one i-repair, $\{P@(-2, 0), Q@\{0\}\}$, but $Q@\{0\}$ belongs to the i-conflict $\{P@\{0\}, Q@\{0\}\}$. Symmetrically, $P@(0, \infty)$ has an empty intersection with every i-conflict but also with every i-repair. Indeed, $\{P@(0, \infty)\}$ is Π_i-inconsistent but is not minimal w.r.t. $\sqsubseteq^i$.*

For the p- case, we first consider again $\mathcal{D}_i$ but extend Π_i with $\bot \leftarrow Q \wedge \diamondminus_{[0,\infty)} P$. Now $\{P@(-2, 0), Q@\{0\}\}$ is the only p-repair but $\{P@\{0\}, Q@\{0\}\}$ is a p-conflict so $Q@\{0\}$ is in all p-repairs and in some p-conflict. For the other direction, consider $\mathcal{D} = \{P@[0, \infty), Q@\{0\}, R@\{0\}\}$ and

$$\Pi = \{\bot \leftarrow P \wedge Q \wedge \diamondminus_{(0,\infty)} \boxplus_{(0,\infty)} P,\ \bot \leftarrow R\}.$$

The only p-conflict of $\mathcal{D}$ w.r.t. Π is $\{R@\{0\}\}$ (cf. Example 9) so $Q@\{0\}$ has an empty intersection with every p-conflict. Yet, $\{P@[0, \infty)\}$ is a p-repair that does not contain $Q@\{0\}$.

Case of Bounded-Interval Datasets over $\mathbb{Z}$

We have seen that p- and i-repairs and conflicts need not exist, and even when they do, they may be infinite in size and/or number. Moreover, this holds not only for the dense timeline $(\mathbb{Q}, \leq)$, but also for $(\mathbb{Z}, \leq)$. We observe, however, that the negative results for $\mathbb{Z}$ crucially rely upon using ∞ or $-\infty$ as endpoints. This leads us to explore what happens when we adopt $\mathbb{T} = \mathbb{Z}$ but restrict datasets to only use *bounded intervals* (i.e., finite integers as endpoints).

The following result summarizes the properties of repairs and conflicts in this setting, showing in particular that restricting to bounded-interval datasets suffices to ensure existence and finiteness of p- and i-repairs and conflicts:

Proposition 4. *When $\mathbb{T} = \mathbb{Z}$ and datasets $\mathcal{D}$ are restricted to only use bounded intervals, P_1-P_5 hold for $x = p$, P_1-P_4 hold for $x = i$, and P_5 does not hold for $x = i$.*

4.2 Comparing the Different Semantics

The remaining examples show the following proposition.

Proposition 5. *For every Sem $\in \{brave, CQA, \cap\}$ and $x \neq y \in \{p, i, s\}$, there exist $\mathcal{D}$ and Π such that $\mathcal{D}$ has x- and y-repairs w.r.t. Π, $(\mathcal{D}, \Pi) \models^{y}_{Sem} q(\vec{c}, \iota)$ and $(\mathcal{D}, \Pi) \not\models^{x}_{Sem} q(\vec{c}, \iota)$.*

Example 14 shows the case $y = p$ and $x \in \{i, s\}$.

Example 14. *Consider our running example and recall from Example 5 that $(\mathcal{D}, \Pi) \models^{p}_{\cap} \mathsf{FevEp}(a)@\{34\}$ (hence $(\mathcal{D}, \Pi) \models^{p}_{CQA} \mathsf{FevEp}(a)@\{34\}$) while $(\mathcal{D}, \Pi) \not\models^{x}_{CQA} \mathsf{FevEp}(a)@\{34\}$ (hence $(\mathcal{D}, \Pi) \not\models^{x}_{\cap} \mathsf{FevEp}(a)@\{34\}$) for $x \in \{i, s\}$. Moreover, if we consider Π' that extends Π with*

$$Q(x) \leftarrow \mathsf{Fever}(x)\, \mathcal{U}_{(0,4)}(\mathsf{NoFever}(x)\, \mathcal{U}_{(0,4)} \mathsf{Fever}(x)),$$

$(\mathcal{D}, \Pi') \models^{p}_{brave} Q(a)@\{14\}$ but $(\mathcal{D}, \Pi') \not\models^{x}_{brave} Q(a)@\{14\}$ for $x \in \{i, s\}$.

The case $y = s$ and $x \in \{p, i\}$ is shown by Example 15 for Sem $\in \{\cap, \mathrm{CQA}\}$ and Example 16 for Sem $=$ brave.

Example 15. *Consider $\mathcal{D} = \{P@[0, 10], Q@\{5\}\}$ and*

$$\Pi = \{\bot \leftarrow P \wedge Q,\ \bot \leftarrow \boxplus_{[0,10]} P\}.$$

It is easy to check that $\{Q@\{5\}\}$ is the only s-repair so that $(\mathcal{D}, \Pi) \models^{s}_{\cap} Q@\{5\}$. However, $\{P@(0, 10]\}$ is a p- and i-repair so for $x \in \{p, i\}$, $(\mathcal{D}, \Pi) \not\models^{x}_{CQA} Q@\{5\}$.

Example 16. *Consider $\mathcal{D}_r$ and Π_r from Example 7.*

$$\mathcal{D}_r = \{P@[0, \infty), Q@\{0\}\}$$
$$\Pi_r = \{\bot \leftarrow Q \wedge \Diamond_{(0,\infty)} \boxplus_{(0,\infty)} P\}$$

Since $\{Q@\{0\}\}$ is an s-repair, $(\mathcal{D}_r, \Pi_r) \models^{s}_{brave} Q@\{0\}$. However, for $x \in \{p, i\}$, one can show that the only x-repair is $\{P@[0, \infty)\}$. Hence $(\mathcal{D}_r, \Pi_r) \not\models^{x}_{brave} Q@\{0\}$.

Example 17 illustrates the case $y = i$ and $x = s$ for Sem $\in \{\cap, \mathrm{CQA}\}$ and Example 18 shows this case for Sem $=$ brave.

Example 17. *In Example 16, the only i-repair is $\{P@[0, \infty)\}$ so $(\mathcal{D}_r, \Pi_r) \models^{i}_{\cap} P@[0, \infty)$. However, $\{Q@\{0\}\}$ is an s-repair so $(\mathcal{D}_r, \Pi_r) \not\models^{s}_{CQA} P@[0, \infty)$.*

Example 18. *Consider our running example and recall from Example 5 that $(\mathcal{D}, \Pi) \models^{i}_{brave} \mathsf{FevEp}(a)@\{29\}$ while $(\mathcal{D}, \Pi) \not\models^{s}_{brave} \mathsf{FevEp}(a)@\{29\}$.*

Example 19 illustrates the case $y = i$ and $x = p$ for Sem $\in \{\cap, \mathrm{CQA}\}$ and Example 20 shows this case for Sem $=$ brave.

Example 19. *Let $\mathcal{D} = \{T@\{0\}, P@[0, 4], Q@[0, 4]\}$ and*

$$\Pi = \{\bot \leftarrow P \wedge Q,\ R \leftarrow P\, \mathcal{U}_{(0,4)} Q\, \mathcal{U}_{(0,4)} P,\ \bot \leftarrow R \wedge T\}.$$

The i-repairs are of the form $\{T@\{0\}, P@[0, t\rangle, Q@\langle t, 4]\}$ or $\{T@\{0\}, Q@[0, t\rangle, P@\langle t, 4]\}$ so $(\mathcal{D}, \Pi) \models^{i}_{\cap} T@\{0\}$. However, $\mathcal{R} = \{P@[0, 1], Q@(1, 3), P@[3, 4]\}$ is a p-repair (note that $(\mathcal{R}, \Pi) \models R@\{0\}$, so $\mathcal{R} \cup \{T@\{0\}\}$ is Π-inconsistent). Hence $(\mathcal{D}, \Pi) \not\models^{p}_{CQA} T@\{0\}$.

Example 20. *Consider $\mathcal{D} = \{P@[0, \infty), Q@\{5\}\}$ and*

$$\Pi = \{\bot \leftarrow P \wedge Q,\ \bot \leftarrow Q \wedge \Diamond_{[0,\infty)} \boxplus_{[0,\infty)} P\}.$$

Since $\{P@[0, 5), Q@\{5\}\}$ is an i-repair, $(\mathcal{D}, \Pi) \models^{i}_{brave} Q@\{5\}$. However, one can show that the only p-repair is $\{P@[0, \infty)\}$. Hence $(\mathcal{D}, \Pi) \not\models^{p}_{brave} Q@\{5\}$.

5 Data Complexity Analysis

We explore the computational properties of our inconsistency handling framework. Specifically, we analyze the data complexity of 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 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: $\mathrm{Datalog}_{\mathrm{nr}}\mathrm{MTL}$, $\mathrm{DatalogMTL}^{\Diamond}_{\mathsf{core}}$, and $\mathrm{DatalogMTL}^{\Diamond}_{\mathsf{lin}}$ (over $\mathbb{Q}$ or $\mathbb{Z}$) and propositional DatalogMTL over $\mathbb{Z}$ [Brandt *et al.*, 2018; Walega *et al.*, 2020b; Walega *et al.*, 2020a].

All results stated in this section are w.r.t. data complexity, i.e. the input size is the size of $\mathcal{D}$. We assume a binary encoding of numbers, with rationals given as pairs of integers.

5.1 Results for s-Repairs and s-Conflicts

We can obtain PSPACE upper bounds for all tasks by adapting known procedures for reasoning with subset repairs and conflicts in the atemporal setting, cf. [Bienvenu and Bourgaux, 2016]. Specifically, an s-repair or s-conflict can be generated by a greedy approach (add / delete facts one by one while preserving (in)consistency), and query entailment under the three semantics can be done via a 'guess and check' approach.

Proposition 6. *For arbitrary DatalogMTL programs Π, (i) the size of $\mathcal{B} \in sConf(\mathcal{D}, \Pi) \cup sRep(\mathcal{D}, \Pi)$ is polynomially bounded in the size of $\mathcal{D}$, (ii) it can be decided in* PSPACE *whether $\mathcal{B} \in sConf(\mathcal{D}, \Pi)$ or $\mathcal{B} \in sRep(\mathcal{D}, \Pi)$, and (iii) a single s-conflict (resp. s-repair) can be generated in* PSPACE*. Moreover, for Sem $\in \{brave, CQA, \cap\}$, query entailment under s-Sem is* PSPACE*-complete.*

If we consider tractable DatalogMTL fragments, we obtain better bounds for the recognition and generation tasks:

Proposition 7. *For tractable DatalogMTL fragments, the tasks of testing whether $\mathcal{B} \in sConf(\mathcal{D}, \Pi)$ (resp. $\mathcal{B} \in sRep(\mathcal{D}, \Pi)$) and generating a single s-conflict (resp. s-repair) can be done in* PTIME*.*

We can use the PTIME upper bounds on recognizing s-repairs to obtain (co)NP upper bounds for query entailment in tractable DatalogMTL fragments. Moreover, for specific fragments, we can show these bounds are tight.

Proposition 8. *For tractable DatalogMTL fragments: query entailment[1] under s-brave (resp. s-CQA, s-intersection) semantics is in* NP *(resp. co*NP*). Matching lower bounds hold in $\mathrm{Datalog}_{nr}\mathrm{MTL}$ and $\mathrm{DatalogMTL}^{\Diamond}_{\mathsf{lin}}$ (and in $\mathrm{DatalogMTL}^{\Diamond}_{\mathsf{core}}$ in the case of s-CQA). The lower bounds hold even for bounded datasets and $\mathbb{T} = \mathbb{Z}$.*

Proof sketch. To illustrate, we provide the reduction from SAT used to show NP-hardness of s-brave semantics in

$\text{Datalog}_{\text{nr}}$MTL. Given a CNF $\varphi = c_1 \wedge ... \wedge c_m$ over variables $v_1, ..., v_n$, consider the $\text{Datalog}_{\text{nr}}$MTL program and dataset:

$$\Pi' = \{N'(v) \leftarrow \Diamond_{[0,\infty)}\, N(v),\ N'(v) \leftarrow \boxplus_{[0,\infty)}\, N(v),$$
$$\bot \leftarrow P(v) \wedge N'(v),\ Q' \leftarrow S\, \mathcal{U}_{(0,\infty)} M,$$
$$S \leftarrow \Diamond_{[0,2)}\, P(v),\ S \leftarrow \Diamond_{[0,2)}\, N(v)\}$$
$$\mathcal{D}' = \{P(v_j)@\{2k\} \mid v_j \in c_k\} \cup \{N(v_j)@\{2k\} \mid \neg v_j \in c_k\}$$
$$\cup\ \{M@\{2m+2\}\}$$

Then φ is satisfiable iff $(\mathcal{D}', \Pi') \models^{s}_{\text{brave}} Q'@\{2\}$. □

The hardness results for $\text{Datalog}_{\text{nr}}$MTL are somewhat surprising in view of the AC^0 data complexity and FO<-rewritability of query entailment in $\text{Datalog}_{\text{nr}}$MTL [Brandt *et al.*, 2018], 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 considered setting of atemporal ontologies, the existence of a rewriting guarantees a data-independent bound on the size of minimal inconsistent subsets and minimal consistent query-entailing subsets. As the preceding reduction shows, such a property fails to hold in $\text{Datalog}_{\text{nr}}$MTL (observe that the minimal consistent query-entailing subsets in $\mathcal{D}'$ have size $m+1$).

In DatalogMTL$^{\Diamond}_{\mathsf{core}}$, 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 our next result:

Proposition 9. *DatalogMTL$^{\Diamond}_{\mathsf{core}}$ query entailment[1] under s-brave and s-intersection semantics is in* PTIME.

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. The proof relies upon rather intricate automata constructions, which build upon and significantly extend those given in [Walega *et al.*, 2020a] for consistency checking.

Proposition 10. *When $\mathbb{T} = \mathbb{Z}$, propositional DatalogMTL query entailment under s-brave, s-CQA, and s-intersection semantics is in* PTIME *(more precisely,* NC1*-complete).*

5.2 Results for Bounded-Interval Datasets over $\mathbb{Z}$

We start by considering interval-based notions and observe that even if the binary encoding of endpoint integers leads to exponentially many choices for which sub-interval to retain for a given input fact, i-conflicts and i-repairs are of polynomial size and can be effectively recognized and generated. This allows us to establish the same general upper bounds for $x = i$ as we obtained for $x = s$.

Proposition 11. *When $\mathbb{T} = \mathbb{Z}$ and only bounded-interval datasets are considered, the results stated in Proposition 6 for the case $x = s$ hold in the case $x = i$.*

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.

Proposition 12. *For tractable DatalogMTL fragments: when $\mathbb{T} = \mathbb{Z}$ and only bounded-interval datasets are considered, it can be decided in* PTIME *whether $\mathcal{B} \in iConf(\mathcal{D}, \Pi)$ and a single i-conflict can be generated in* PTIME.

The argument does not apply to i-repairs, and we leave open the precise complexity of i-repair recognition in this case (we only get a coNP upper bound). However, we can still obtain a tight complexity result for i-brave semantics since we do not need to get a complete i-repair in this case.

Proposition 13. *For tractable DatalogMTL fragments: when $\mathbb{T} = \mathbb{Z}$ and only bounded-interval datasets are considered, query entailment[1] under i-brave (resp. i-CQA, i-intersection) is in* NP *(resp. in Π^p_2). Lower* NP *(resp. co*NP*) bounds hold for Datalog$_{nr}$MTL and DatalogMTL$^{\Diamond}_{\mathsf{lin}}$ (and for DatalogMTL$^{\Diamond}_{\mathsf{core}}$ in the case of i-CQA semantics).*

The situation for pointwise notions is starkly different:

Proposition 14. *When $\mathbb{T} = \mathbb{Z}$ and only bounded-interval datasets are considered, there exist $\mathcal{D}$ and Π such that every $\mathcal{B} \in pConf(\mathcal{D}, \Pi)$ (resp. $\mathcal{B} \in pRep(\mathcal{D}, \Pi)$) is exponentially large w.r.t. the size of $\mathcal{D}$.*

We thus only obtain EXPSPACE complexity upper bounds.

Proposition 15. *When $\mathbb{T} = \mathbb{Z}$ and only bounded-interval datasets are considered, all tasks considered in Proposition 6 for $x = s$ can be done in* EXPSPACE *in the case $x = p$.*

6 Conclusion and Future Work

This paper provides a first study of inconsistency handling in DatalogMTL, a prominent formalism for reasoning on temporal data. Due to facts having associated time intervals, there are different natural ways to define conflicts and repairs. Our results show that these alternative notions can differ significantly with regards to basic properties (existence, finiteness, or size). For s-conflicts and s-repairs, we provided a detailed picture of the data complexity landscape, with tight complexity results for several DatalogMTL fragments. Notably, we proved that query entailment in propositional DatalogMTL over $\mathbb{Z}$ is tractable for all three s-repair-based semantics.

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 (with significantly more involved constructions). It would also be interesting to consider DatalogMTL with negation or spatio-temporal predicates. A nice theoretical question is to consider the decidability of i- and p-repair / conflict existence in unrestricted settings. 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. [Bienvenu and Bourgaux, 2022]. 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.

[1] Restricted to queries with punctual intervals for DatalogMTL$^{\Diamond}_{\mathsf{lin}}$ and DatalogMTL$^{\Diamond}_{\mathsf{core}}$: Walega *et al.* [2020b] give results for consistency checking, and reductions from query entailment to consistency checking for non-punctual queries use constructs not available in these two fragments, cf. discussion in [Walega *et al.*, 2020b].

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 Zakharyaschev. 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 Vadalog 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, 2016] Meghyn Bienvenu and Camille Bourgaux. Inconsistency-tolerant querying of description logic knowledge bases. In *Lecture Notes of 2016 Reasoning Web Summer School*, 2016.

[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 Zakharyaschev. 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 Zakharyaschev. 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 Zakharyaschev. 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, Przemyslaw Andrzej Walega, and Michael Zakharyaschev. 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] Przemyslaw 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] Przemyslaw 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] Przemyslaw 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] Przemyslaw Andrzej Walega, Mark Kaminski, Dingmin Wang, and Bernardo Cuenca Grau. Stream reasoning with DatalogMTL. *J. Web Semant.*, 76:100776, 2023.

[Walega *et al.*, 2023b] Przemyslaw 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] Przemyslaw 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, Przemyslaw 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, Przemyslaw Andrzej Walega, Pan Hu, and Bernardo Cuenca Grau. Practical reasoning in DatalogMTL. *CoRR*, abs/2401.02869, 2024.