Inconsistency Handling in DatalogMTL

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

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., $\boxplus_{[0,2]} Q \leftarrow \diamondsuit_{\{3\}} P$ 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 $(\bot \leftarrow P \land Q)$, 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 (srepairs), 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 the appendix.

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 **P**, **C** and **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:

 $A ::= P(\vec{\tau}) \mid \top \mid_{\boxplus_{\varrho}} A \mid_{\boxminus_{\varrho}} A \mid_{\diamondsuit_{\varrho}} A \mid_{\diamondsuit_{\varrho}} A \mid_{AU_{\varrho}A} \mid_{AS_{\varrho}A}$ where $P(\vec{\tau})$ is an atom and ϱ is a range. Intuitively, S stands for 'since', \mathcal{U} for 'until', \diamondsuit for 'eventually', and \square for 'always', with + indicating the future and - the past. A DatalogMTL *program* Π is a finite set of rules of the form

 $B \leftarrow A_1 \wedge ... \wedge A_k$ or $\bot \leftarrow A_1 \wedge ... \wedge A_k$ with $k \ge 1$ where each A_i is a literal and B is a literal not mentioning any 'non-deterministic' operators $\bigoplus_{\varrho}, \bigoplus_{\varrho}, \mathcal{U}_{\varrho}$, and \mathcal{S}_{ϱ} . We call $A_1 \wedge ... \wedge A_k$ 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 $\bot \leftarrow A_1 \land A_2$ or of the form $B \leftarrow A$. It is *linear* if each of its rules is either of the form $\bot \leftarrow A_1 \land A_2$ or of the form $B \leftarrow A_1 \land ... \land A_k$ where at most one A_i mentions some predicate that occurs in the head of some rule (intensional predicate). We denote by DatalogMTL $_{\text{core}}^{\diamondsuit}$ (resp. DatalogMTL $_{\text{lin}}^{\diamondsuit}$) the fragment where programs are core (resp. linear) and \diamondsuit is the only temporal operator allowed in literals. The relation \lessdot of dependence between predicates is defined by $P \lessdot Q$ 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 $P \lessdot^+ P$, where \lessdot^+ is the transitive closure of \lessdot . We denote by Datalog_{nr}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.

$$\begin{array}{ll} \mathfrak{M},t \models \top & \text{for every } t \in \mathbb{T} \\ \mathfrak{M},t \not\models \bot & \text{for every } t \in \mathbb{T} \\ \mathfrak{M},t \models \boxplus_{\varrho} A & \text{if } \mathfrak{M},s \models A \text{ for all } s \text{ with } s-t \in \varrho \\ \mathfrak{M},t \models \boxminus_{\varrho} A & \text{if } \mathfrak{M},s \models A \text{ for all } s \text{ with } t-s \in \varrho \\ \mathfrak{M},t \models \varTheta_{\varrho} A & \text{if } \mathfrak{M},s \models A \text{ for some } s \text{ with } s-t \in \varrho \\ \mathfrak{M},t \models \varTheta_{\varrho} A & \text{if } \mathfrak{M},s \models A \text{ for some } s \text{ with } t-s \in \varrho \\ \mathfrak{M},t \models \varTheta_{\varrho} A & \text{if } \mathfrak{M},s \models A \text{ for some } s \text{ with } t-s \in \varrho \\ \mathfrak{M},t \models A \, \mathcal{U}_{\varrho} A' & \text{if } \mathfrak{M},t' \models A' \text{ for some } t' \text{ with } t'-t \in \varrho \\ \text{and } \mathfrak{M},s \models A \text{ for all } s \in (t,t') \\ \mathfrak{M},t \models A \, \mathcal{S}_{\varrho} A' & \text{if } \mathfrak{M},t' \models A' \text{ for some } t' \text{ with } t-t' \in \varrho \\ \text{and } \mathfrak{M},s \models A \text{ for all } s \in (t',t) \\ \end{array}$$

An interpretation \mathfrak{M} is a *model* of a rule $H \leftarrow A_1 \wedge ... \wedge A_k$ 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 $Q \in \mathbf{P}, \ \vec{v} = (v_1, \dots, v_n)$ 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}, ι) such that $\vec{c} = (c_1, \dots, c_n)$ 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 finhtr). This is represented by the following rule, where, intuitively, x represents a patient and y a blood pouch:

$$\begin{split} \mathsf{PotFnhtr}(x) \leftarrow & \mathsf{Fever}(x) \land \boxminus_{(0,24]} \mathsf{NoFever}(x) \\ & \land \diamondsuit_{[0,4]} \mathsf{GetBlood}(x,y) \end{split}$$

Another rule detects more generally relevant fever episodes:

$$\mathsf{FevEp}(x) \leftarrow \mathsf{Fever}(x) \land$$

$$\Leftrightarrow_{[0,24]} (\mathsf{NoFever}(x) \, \mathcal{U}_{\{5\}} \mathsf{GetBlood}(x,y))$$

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

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

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.

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

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.

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 NoFever(a)@[0,32], or remove both Fever(a)@[14,18] and 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 NoFever(a)@[0,14) and 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:

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

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 NoFever(a)@[0,t) and 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 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 $\mathcal{B}' \sqsubseteq^p \mathcal{B}$ to indicate that $\mathcal{B}' \sqsubseteq^p \mathcal{B}$ and $\mathcal{B} \not\sqsubseteq^p \mathcal{B}'$. For $x \in \{i, s\}$, we write $\mathcal{B}' \sqsubseteq^x \mathcal{B}$ if $\mathcal{B}' \sqsubseteq^x \mathcal{B}$ and $\mathcal{B}' \sqsubseteq^p \mathcal{B}$.

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* $\bigcap_{i\in I} \mathcal{B}_i = \{\alpha @\{t\} \mid \mathcal{B}_i \models \alpha @\{t\} \text{ for each } i \in I\}$. *The* pointwise intersection *of a fact* $\alpha @\iota$ *and a set of facts* \mathcal{B} *is* $\{\alpha @\iota\} \cap \mathcal{B}$.

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 $\mathcal{C}' \sqsubseteq^x \mathcal{C}$. 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 $\{\text{NoFever}(a)@[0,32], \text{Fever}(a)@[14,18]\}$ and $\{\text{NoFever}(a)@[0,32], \text{Fever}(a)@[29,34]\}$, while the p-conflicts and i-conflicts are of the form $\{\text{NoFever}(a)@\{t\}, \text{Fever}(a)@\{t\}\}$ 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 $\mathcal{R} \sqsubseteq^x \mathcal{R}' \sqsubseteq^x \mathcal{D}$. 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

 $\mathcal{R}_2 = \mathcal{I} \cup \{ \text{Fever}(a)@[14,18], \text{Fever}(a)@[29,34] \}$ 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 Fever(a)@{t} or 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 NoFever(a)@[0,t \rangle , Fever(a)@ $\langle t, 34$], where \rangle , \langle are either], (or), [and $t \in [29, 32]$;
- or three facts NoFever(a)@[0,t \rangle , Fever(a)@ $\langle t, 18 \rangle$, and Fever(a)@[29,34], where $t \in [14,18]$,
 - $-\rangle,\langle$ are either],(or),[and
 - if t = 18, then $\rangle, \langle are \rangle, [$;
- or three facts $Fever(a)@[14,t_1)$, $NoFever(a)@\langle t_1,t_2\rangle'$, $Fever(a)@\langle t_2,34|$, where $t_1 \in [14,18]$, $t_2 \in [29,32]$,
 - $-\rangle$, \langle and \rangle' , \langle' are either \rangle , \langle or \rangle , \rangle ,
 - if $t_1 = 14$, then $\rangle, \langle are \rangle, \langle are \rangle$

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

Proposition 1. For every query $(\Pi, q(\vec{v}, r))$, dataset \mathcal{D} , tuple of constants \vec{c} , and interval ι , $(\mathcal{D}, \Pi) \models_{\cap}^{x} q(\vec{c}, \iota)$ implies $(\mathcal{D}, \Pi) \models_{CQA}^{x} q(\vec{c}, \iota)$, which implies $(\mathcal{D}, \Pi) \models_{brave}^{x} q(\vec{c}, \iota)$. 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:

- $(\mathcal{D}, \Pi) \models_{\Omega}^{s} \mathsf{AntiDRisk}(a)@[-90, \infty),$
- $(\mathcal{D}, \Pi) \not\models_{brave}^{s} \mathsf{FevEp}(a)@\{t\} \text{ for every } t \in \mathbb{T},$
- $(\mathcal{D}, \Pi) \not\models_{brave}^{s} \mathsf{PotFnhtr}(a)@\{t\} \text{ for every } t \in \mathbb{T}.$

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

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

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

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

P₅: 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 $\{P@(t,\infty)\}$. Since $\mathcal{C}' = \{P@(t+1,\infty)\}$ is Π -inconsistent and $\mathcal{C}' \sqsubseteq^x \mathcal{C}$, then \mathcal{C} is not an x-conflict.

Every $\mathcal{R} \sqsubseteq^i \mathcal{D}$ 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 $\mathcal{R} \sqsubseteq^i \mathcal{R}' \sqsubseteq^i \mathcal{D}$. 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 $\mathcal{R} \sqsubseteq^p \mathcal{R}' \sqsubseteq^p \mathcal{D}$ 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

$$\mathcal{D}_r = \{ P@[0, \infty), Q@\{0\} \}$$

$$\Pi_r = \{ \bot \leftarrow Q \land \bigoplus_{(0, \infty)} \boxplus_{(0, \infty)} P \}.$$

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.

$$\mathcal{D}_i = \{ P@(-2, \infty), Q@\{0\} \}$$

$$\Pi_i = \{ \bot \leftarrow \boxplus_{(0,\infty)} P, \ \bot \leftarrow Q \land P \}$$

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

$$\begin{split} \Pi_p &= \{\bot \leftarrow \boxplus_{[0,\infty)} P, \ \bot \leftarrow \boxminus_{[0,\infty)} P, \ \bot \leftarrow P \land Q, \\ & \bot \leftarrow Q \land \boxminus_{(0,10)} P \land \oplus_{[10,\infty)} P, \\ & \bot \leftarrow Q \land \boxplus_{(0,10)} P \land \oplus_{[10,\infty)} P \}. \end{split}$$

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. $\Pi_i = \{\bot \leftarrow P \land Q \land \bigoplus_{(0,\infty)} \boxplus_{(0,\infty)} P\}$. 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

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

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.

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

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.

$$\begin{split} \mathcal{D} &= \{ P@[0,\infty), Q@\{0\} \} \\ \Pi &= \{ \bot \leftarrow Q \land \boxplus_{[0,\infty)} \Phi_{[0,2)} \ P \} \end{split}$$

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) :

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

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 $\bot \leftarrow Q \land \bigoplus_{[0,\infty)} P$. 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

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

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 , $(\mathcal{D}, \Pi) \models_{Sem}^{y} q(\vec{c}, \iota)$ and $(\mathcal{D}, \Pi) \not\models_{Sem}^{x} q(\vec{c}, \iota)$.

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

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

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

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

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

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

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

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

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

$$\mathcal{D}_r = \{ P@[0,\infty), Q@\{0\} \}$$

$$\Pi_r = \{ \bot \leftarrow Q \land \bigoplus_{(0,\infty) \boxminus (0,\infty)} P \}$$

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 $\in \{ \cap, CQA \}$ 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 $(\mathcal{D}_r,\Pi_r)\not\models^s_{QA}P@[0,\infty)$.

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

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

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

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

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 $(\mathcal{D},\Pi) \models_{\cap}^{i} T@\{0\}$. 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 $(\mathcal{D},\Pi) \not\models_{COA}^{p} T@\{0\}$.

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

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

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: DatalognrMTL, DatalogMTL $_{\text{core}}^{\downarrow}$, and DatalogMTL $_{\text{lin}}^{\downarrow}$ (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 entailment¹under s-brave (resp. s-CQA, s-intersection) semantics is in NP (resp. coNP). Matching lower bounds hold in Datalog_{nr}MTL and DatalogMTL $_{lin}^{\diamondsuit}$ (and in DatalogMTL $_{core}^{\diamondsuit}$ 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

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

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

Then
$$\varphi$$
 is satisfiable iff $(\mathcal{D}', \Pi') \models_{\text{brave}}^s Q'@\{2\}.$

The hardness results for Datalog_{nr}MTL are somewhat surprising in view of the AC^0 data complexity and FO<-rewritability of query entailment in Datalog_{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 Datalog_{nr}MTL (observe that the minimal consistent query-entailing subsets in \mathcal{D}' have size m+1).

In DatalogMTL $_{\text{core}}^{\diamondsuit}$, 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 $^{\diamondsuit}_{\mathsf{core}}$ query entailment¹ 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 entailment¹ under i-brave (resp. i-CQA, i-intersection) is in NP (resp. in Π_2^p). Lower NP (resp. coNP) bounds hold for Datalog_{nr}MTL and DatalogMTL $_{\text{lin}}^{\diamondsuit}$ (and for DatalogMTL $_{\text{core}}^{\diamondsuit}$ 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 boundedinterval \mathbb{Z} setting. We are most interested in trying to extend our tractability results for s-repair-based semantics to irepairs 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.

¹Restricted to queries with punctual intervals for DatalogMTL and DatalogMTL was 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, 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.

A Proofs for Section 3

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} .

Proof. (1) By definition, $\mathcal{B}' \sqsubseteq^s \mathcal{B}$ implies $\mathcal{B}' \subseteq \mathcal{B}$ even if \mathcal{B} is not in normal form. If \mathcal{B} is in normal form and $\mathcal{B}' \subseteq \mathcal{B}$, then $\mathcal{B}' \sqsubseteq^p \mathcal{B}$ and for every $\alpha@\iota \in \mathcal{B}$, there is at most one $\alpha@\iota' \in \mathcal{B}'$ such that $\iota' \subseteq \iota$ (otherwise, since $\mathcal{B}' \subseteq \mathcal{B}$, this would contradict the fact that \mathcal{B} is in normal form). Hence $\mathcal{B}' \sqsubseteq^i \mathcal{B}$ and $\mathcal{B}' \subseteq \mathcal{B}$, i.e., $\mathcal{B}' \sqsubseteq^s \mathcal{B}$.

(2) Since $\mathcal{B}' \sqsubseteq^i \mathcal{B}$ implies $\mathcal{B}' \sqsubseteq^p \mathcal{B}$, for every $\alpha@\iota' \in \mathcal{B}'$, it holds that $\alpha@\iota'$ is pointwise included in \mathcal{B} . Since \mathcal{B} is in normal form, this implies that there exists $\alpha@\iota \in \mathcal{B}$ such that $\iota' \subseteq \iota$. Hence, for every $\alpha@\iota' \in \mathcal{B}'$, there exists $\alpha@\iota \in \mathcal{B}$ such that $\iota' \subseteq \iota$. Since $\mathcal{B}' \sqsubseteq^i \mathcal{B}$ also implies that for every $\alpha@\iota \in \mathcal{B}$, there is at most one $\alpha@\iota' \in \mathcal{B}'$ with $\iota' \subseteq \iota$, we obtain that the number of facts in \mathcal{B}' is bounded by the number of facts in \mathcal{B} .

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

Proof. Assume that $(\mathcal{D},\Pi)\models_{\mathcal{T}}^{\alpha}q(\vec{c},\iota)$. This means that $(\mathcal{I},\Pi)\models q(\vec{c},\iota)$ where $\mathcal{I}=\bigcap_{\mathcal{R}\in xRep(\mathcal{D},\Pi)}\mathcal{R}=\{\alpha@\{t\}\mid \mathcal{R}\models\alpha@\{t\} \text{ for each }\mathcal{R}\in xRep(\mathcal{D},\Pi)\}$. Let $\mathcal{R}\in xRep(\mathcal{D},\Pi)$ and $\mathfrak{M}\models(\mathcal{R},\Pi)$. For every $\alpha@\{t\}\in \mathcal{I},\mathcal{R}\models\alpha@\{t\}$ so there exists $\alpha@\iota'\in\mathcal{R}$ such that $t\in\iota'$. It follows that $\mathfrak{M},t\models\alpha$, i.e., \mathfrak{M} is a model of $\alpha@\{t\}$. Hence $\mathfrak{M}\models(\mathcal{I},\Pi)$, so for every $t\in\iota$, $\mathfrak{M},t\models Q(\vec{\tau})_{|\vec{v}\leftarrow\vec{c}|}$, where $q(\vec{v},r)=Q(\vec{r})@r$. We obtain that $(\mathcal{R},\Pi)\models q(\vec{c},\iota)$, so $(\mathcal{D},\Pi)\models_{\mathcal{L}OA}^{\mathcal{R}}q(\vec{c},\iota)$.

Now $(\mathcal{D},\Pi)\models^x_{\operatorname{CQA}}q(\vec{c},\iota)$ means that $(\mathcal{R},\Pi)\models q(\vec{c},\iota)$ for every $\mathcal{R}\in xRep(\mathcal{D},\Pi)$, and since the definition of x-CQA and $x\text{-brave semantics requires that }xRep(\mathcal{D},\Pi)\neq\emptyset$, this means that $(\mathcal{D},\Pi)\models^x_{\operatorname{brave}}q(\vec{c},\iota)$.

To see that none of the converse implications holds, consider $\Pi = \{Q \leftarrow P, Q \leftarrow R, \bot \leftarrow P \land R\}$ and $\mathcal{D} = \{P@\{0\}, R@\{0\}\}\}$. For every $x \in \{p, i, s\}, (\mathcal{D}, \Pi) \models_{\text{brave}}^x P@\{0\}$ while $(\mathcal{D}, \Pi) \not\models_{\text{CQA}}^x P@\{0\}$, and $(\mathcal{D}, \Pi) \models_{\text{CQA}}^x Q@\{0\}$ while $(\mathcal{D}, \Pi) \not\models_{\mathcal{C}}^x Q@\{0\}$.

B Proofs for Section 4.1

Lemma 2. If \mathcal{D} is a dataset (in normal form) and Π is a (consistent) DatalogMTL program, $\mathcal{R} \in sRep(\mathcal{D},\Pi)$ iff \mathcal{R} is a \subseteq -maximal Π -consistent subset of \mathcal{D} , and $\mathcal{C} \in sConf(\mathcal{D},\Pi)$ iff \mathcal{C} is a \subseteq -minimal Π -inconsistent subset of \mathcal{D} .

Proof. Let $\mathcal{R} \in sRep(\mathcal{D}, \Pi)$, i.e., \mathcal{R} is in normal form, Π -consistent and such that $\mathcal{R} \sqsubseteq^s \mathcal{D}$ and there is no Π -consistent \mathcal{R}' such that $\mathcal{R} \sqsubseteq^s \mathcal{R}' \sqsubseteq^s \mathcal{D}$. Since $\mathcal{R} \sqsubseteq^s \mathcal{D}$, then $\mathcal{R} \subseteq \mathcal{D}$. Moreover, there is no Π -consistent \mathcal{R}' such that $\mathcal{R} \subsetneq \mathcal{R}' \subseteq \mathcal{D}$ (since by Lemma 1, this would imply $\mathcal{R} \sqsubseteq^s \mathcal{R}' \sqsubseteq^s \mathcal{D}$). Hence \mathcal{R} is a \subseteq -maximal Π -consistent subset of \mathcal{D} .

In the other direction, if \mathcal{R} is a \subseteq -maximal Π -consistent subset of \mathcal{D} , then (i) \mathcal{R} is in normal form since \mathcal{D} is in normal form, (ii) by Lemma 1, $\mathcal{R} \sqsubseteq^s \mathcal{D}$, and (iii) there is no Π -consistent \mathcal{R}' such that $\mathcal{R} \sqsubseteq^s \mathcal{R}' \sqsubseteq^s \mathcal{D}$ (since this would imply that $\mathcal{R} \subsetneq \mathcal{R}' \subseteq \mathcal{D}$). Hence $\mathcal{R} \in sRep(\mathcal{D}, \Pi)$.

The case of s-conflicts is similar. \square

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

Proof. P₁: By Lemma 2 and monotonicity of DatalogMTL, one can obtain an s-repair \mathcal{R} of \mathcal{D} w.r.t. Π as follows: start from $\mathcal{R} = \emptyset$ and greedily add facts from \mathcal{D} to \mathcal{R} one by one, except if adding the fact makes \mathcal{R} Π -inconsistent.

 P_2 : If \mathcal{D} is Π -inconsistent, by Lemma 2 and monotonicity of DatalogMTL, one can obtain an s-conflict \mathcal{C} of \mathcal{D} w.r.t. Π as follows: start from $\mathcal{C}=\mathcal{D}$ and greedily removes facts one by one, except if removing the fact makes \mathcal{C} Π -consistent. In the other direction, if there exists an s-conflict $\mathcal{C}\subseteq\mathcal{D}$, since DatalogMTL is monotone, \mathcal{D} is Π -inconsistent.

 P_3 - P_4 : If \mathcal{B} is an s-repair or an s-conflict, $\mathcal{B}\subseteq\mathcal{D}$ so since \mathcal{D} is finite, so is \mathcal{B} . Moreover, $|sRep(\mathcal{D},\Pi)|$ and $|sConf(\mathcal{D},\Pi)|$ are bounded by the number of subsets of \mathcal{D} , hence are finite.

 P_5 : Let $\alpha@\iota$ be pointwise included in \mathcal{D} . Since \mathcal{D} is in normal form, there exists a unique $\alpha@\iota' \in \mathcal{D}$ such that $\iota \subseteq \iota'$. Hence $\mathcal{R} \models \alpha@\iota$ for every $\mathcal{R} \in sRep(\mathcal{D}, \Pi)$ iff $\alpha@\iota' \in \mathcal{R}$ for every $\mathcal{R} \in sRep(\mathcal{D}, \Pi)$ (since $\mathcal{R} \subseteq \mathcal{D}$). Moreover, $\alpha@\iota' \in \mathcal{R}$ for every $\mathcal{R} \in sRep(\mathcal{D},\Pi)$ iff $\alpha@\iota' \notin \mathcal{C}$ for every $\mathcal{C} \in sConf(\mathcal{D}, \Pi)$. Indeed, if $\alpha@\iota' \in \mathcal{C}$ for some sconflict C, $C \setminus \{\alpha@\iota'\}$ is Π -consistent thus can be extended to an s-repair \mathcal{R} that does not contain $\alpha@\iota'$ (as in the proof of P_1), and if $\alpha@\iota' \notin \mathcal{R}$ for some s-repair \mathcal{R} , $\mathcal{R} \cup \{\alpha@\iota'\}$ is Π -inconsistent and can be reduced to an s-conflict $\mathcal C$ that contains $\alpha@\iota'$ (as in the proof of P_2). Finally, $\alpha@\iota'\notin\mathcal{C}$ for every $C \in sConf(\mathcal{D}, \Pi)$ iff $\alpha@\iota$ has an empty pointwise intersection with every $\mathcal{C} \in sConf(\mathcal{D}, \Pi)$ (since $\mathcal{C} \subseteq \mathcal{D}$). It follows that $\mathcal{R} \models \alpha@\iota$ for every $\mathcal{R} \in sRep(\mathcal{D}, \Pi)$ iff $\alpha@\iota$ has an empty pointwise intersection with every $\mathcal{C}\in$ $sConf(\mathcal{D},\Pi).$

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

Proof. This follows from Proposition 2 (P_5) and the fact that s-repairs and s-conflicts are subsets of \mathcal{D} (so that their intersections or unions are also subsets of \mathcal{D}), and that \mathcal{D} is in normal form: for every $\alpha@\iota\in\mathcal{D}$, $\alpha@\iota\in\bigcap_{\mathcal{R}\in sRep(\mathcal{D},\Pi)}\mathcal{R}$ iff $\alpha@\iota$ is pointwise included in every s-repair iff $\alpha@\iota$ has an empty intersection with every s-conflicts iff $\alpha@\iota\notin\bigcup_{\mathcal{C}\in sConf(\mathcal{D},\Pi)}\mathcal{C}$.

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.

$$\mathcal{D}_i = \{ P@(-2, \infty), Q@\{0\} \}$$

$$\Pi_i = \{ \bot \leftarrow \boxplus_{(0, \infty)} P, \ \bot \leftarrow Q \land P \}$$

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

$$\Pi_{p} = \{ \bot \leftarrow_{\mathbb{B}[0,\infty)} P, \ \bot \leftarrow_{\mathbb{B}[0,\infty)} P, \ \bot \leftarrow P \land Q,$$

$$\bot \leftarrow_{Q} \land_{\mathbb{B}(0,10)} P \land_{\mathbb{B}[10,\infty)} P,$$

$$\bot \leftarrow_{Q} \land_{\mathbb{B}(0,10)} P \land_{\mathbb{B}[10,\infty)} P \}.$$

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.

Proof. Assume for a contradiction that \mathcal{R} is an i-repair. If $Q @ \{0\} \notin \mathcal{R}$, $\mathcal{R} = \{P @ \langle t_1, t_2 \rangle\}$ for some $t_1 \neq -\infty$ and $t_2 \neq \infty$ and $\mathcal{R}' = \{P @ \langle t_1, t_2 + 1 \rangle\}$ is Π_p -consistent, contradicting the maximality of \mathcal{R} . Hence $\mathcal{R} = \{Q @ \{0\}, P @ \langle t_1, t_2 \rangle\}$ with $\langle t_1, t_2 \rangle \subseteq (0, \infty)$ or $\langle t_1, t_2 \rangle \subseteq (-\infty, 0)$. In the first case, if $t_2 = \infty$, then \mathcal{R} is Π_p -inconsistent and otherwise, $\mathcal{R}' = \{Q @ \{0\}, P @ \langle t_1, t_2 + 1 \rangle\}$ is Π_p -consistent and $\mathcal{R} \sqsubset^i \mathcal{R}' \sqsubseteq^i \mathcal{D}_p$. The second case is similar.

Example 9. $\mathcal{D}_i = \{P@[0,\infty), Q@\{0\}\}$ is an i-conflict of itself w.r.t. $\Pi_i = \{\bot \leftarrow P \land Q \land \oplus_{(0,\infty)} \boxplus_{(0,\infty)} P\}$. 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

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

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.

Proof. Assume for a contradiction that \mathcal{C} is an i-conflict. If $Q@\{0\} \notin \mathcal{C}$, $\mathcal{C} = \{P@\langle t, \infty)\}$ is not minimal since $\{P@\langle t+1,\infty\rangle\}$ is Π_p -inconsistent. Hence $\mathcal{C} = \{Q@\{0\}, P@\langle t_1, t_2\rangle\}$. If $t_2 < \infty$, \mathcal{C} is Π_p -consistent. But if $t_2 = \infty$, $\mathcal{C}' = \mathcal{C} \setminus \{Q@\{0\}\}$ is Π_p -inconsistent and $\mathcal{C}' \sqsubset^i \mathcal{C}$.

Example 10. Consider the following dataset and program.

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

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.

Proof. Indeed, let $\mathcal{R} \sqsubseteq^p \mathcal{D}$ be a finite set of facts. If \mathcal{R} contains a fact of the form $P@\langle t, \infty \rangle$, it is Π-inconsistent. Otherwise, there exists $t \in \mathbb{T}$ such that $t > t_2$ for every $P@\langle t_1, t_2 \rangle \in \mathcal{R}$, so if \mathcal{R} is Π-consistent, then $\mathcal{R}' = \mathcal{R} \cup \{P@\{t+2\}\}$ is Π-consistent and $\mathcal{R} \sqsubseteq^p \mathcal{R}' \sqsubseteq^p \mathcal{D}$.

Example 11. Consider the following dataset and program.

$$\begin{split} \mathcal{D} &= \{P@[0,\infty), Q@\{0\}\} \\ \Pi &= \{\bot \leftarrow Q \land \boxplus_{[0,\infty)} \spadesuit_{[0,2)} \ P\} \end{split}$$

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.

Proof. Indeed, let $\mathcal{C} \sqsubseteq^p \mathcal{D}$ be a finite set of facts. If \mathcal{C} contains a fact of the form $P@\langle t, \infty \rangle$, then $\mathcal{C}' = \mathcal{C} \setminus \{P@\langle t, \infty \rangle\} \cup \{P@\langle t, t+10 \rangle, P@(t+10, \infty)\}$ is such that $\mathcal{C}' \sqsubseteq^p \mathcal{C}$ and is Π -inconsistent iff \mathcal{C} is Π -inconsistent. Otherwise, there exists $t \in \mathbb{T}$ such that $t > t_2$ for every $P@\langle t_1, t_2 \rangle \in \mathcal{C}$ and \mathcal{C} is Π -consistent.

B.1 Proof of Proposition 4

Note that because we work with finite timepoints from \mathbb{Z} , it suffices to consider intervals with square brackets [,], since e.g. the interval (2,5) can be equivalently represented as [3,4], both intervals corresponding to the same set $\{3,4\}$ of timepoints. We will therefore assume w.l.o.g. that all datasets provided as input in the following proofs use only [and].

Proposition 4. When $\mathbb{T} = \mathbb{Z}$ and datasets \mathcal{D} are restricted to only use bounded intervals, $\mathsf{P}_1\text{-}\mathsf{P}_5$ hold for x=p, $\mathsf{P}_1\text{-}\mathsf{P}_4$ hold for x=i, and P_5 does not hold for x=i.

To prove the preceding proposition, we fix some DatalogMTL program Π and dataset \mathcal{D} , over timeline (\mathbb{Z}, \leq) , such that \mathcal{D} is in normal form and its facts only use bounded intervals. We treat the cases of x=i and x=p separately.

Proof of Proposition 4 for x = i

We show the first four properties hold.

P₁: We show how we can construct an i-repair \mathcal{R} in a greedy fashion by iterating over the facts in \mathcal{D} . We start with $\mathcal{R}_0 = \emptyset$. At each stage $1 \leq k \leq n = |\mathcal{D}|$, we pick a fact $\alpha@[i,j] \in \mathcal{D}$ that has not yet been considered and set \mathcal{R}_{k+1} equal to $\mathcal{R}_k \cup \{\alpha@[i',j']\}$, with [i',j'] being some maximal subinterval of [i,j] such that $\mathcal{R}_k \cup \{\alpha@[i',j']\}$ is Π -consistent. Should all intervals lead to inconsistency with Π , then we let $\mathcal{R}_{k+1} = \mathcal{R}_k$. It follows from this construction that the final dataset \mathcal{R}_n is Π -consistent. Moreover, there cannot exist any \mathcal{R}' with $\mathcal{R} \sqsubset^i \mathcal{R}' \sqsubseteq^i \mathcal{D}$ that is Π -consistent (otherwise this would contradict the maximality of the intervals chosen during the construction). It follows that $iRep(\mathcal{D}, \Pi) \neq \emptyset$.

P₂: If \mathcal{D} is Π -inconsistent, then we can greedily construct a conflict. To do so, we start with $\mathcal{C}_0 = \mathcal{D}$, and at each stage $1 \leq k \leq n = |\mathcal{D}|$, we pick a fact $\alpha@[i,j] \in \mathcal{D}$ that has not yet been considered and set \mathcal{C}_{k+1} equal to $\mathcal{C}_k \setminus \{\alpha@[i,j]\} \cup \{\alpha@[i',j']\}$ with [i',j'] being a minimal subinterval of [i,j] such that \mathcal{C}_{k+1} is Π -inconsistent. If $\mathcal{C}_k \setminus \{\alpha@[i,j]\}$ is alrainconsistent, then $[i',j'] = \emptyset$ and $\mathcal{C}_{k+1} = \mathcal{C}_k \setminus \{\alpha@[i,j]\}$. It follows from this construction that the final dataset \mathcal{C}_n is Π -inconsistent. Moreover, there cannot exist any Π -inconsistent \mathcal{C}' with $\mathcal{C}' \sqsubset^i \mathcal{C}$ (otherwise this would contradict the minimality of the intervals chosen during the construction). It follows that $iConf(\mathcal{D}, \Pi) \neq \emptyset$. In the other direction, if there exists an *i*-conflict $\mathcal{C} \sqsubseteq^i \mathcal{D}$, since this implies that $\mathcal{D} \models \mathcal{C}$ and DatalogMTL is monotone, \mathcal{D} is Π -inconsistent.

P₃: Let $\mathcal{B} \sqsubseteq^i \mathcal{D}$. For every $\mathcal{B} \sqsubseteq^i \mathcal{D}$ and $\alpha@\iota' \in \mathcal{B}$, there exists $\alpha@\iota \in \mathcal{D}$ such that $\iota' \subseteq \iota$ (cf. proof of point 2 of Lemma 1). Since given a bounded interval [i,j] over \mathbb{Z} , there are finitely many subintervals of [i,j], the number of facts that may belong to some $\mathcal{B} \sqsubseteq^i \mathcal{D}$ is finite, and the number of \mathcal{B} such that $\mathcal{B} \sqsubseteq^i \mathcal{D}$ is finite. It follows that $iRep(\mathcal{D}, \Pi)$ and $iConf(\mathcal{D}, \Pi)$ are finite.

 P_4 : P_4 already holds without the assumption that $\mathbb{T} = \mathbb{Z}$ (Proposition 3), by point 2 of Lemma 1.

Finally we prove that Property P_5 does not hold for i-repairs and i-conflicts. Consider $\mathcal{D} = \{P@[1,5], Q@\{3\}\}$ and program $\Pi = \{\bot \leftarrow Q \land \boxplus_{[0,2]} P\}$. Then $\mathcal{R} = \{P@[4,5], Q@\{3\}\}$ is an i-repair: it is Π -consistent, in normal form, and any \mathcal{R}' with $\mathcal{R} \sqsubset^i \mathcal{R}' \sqsubseteq^i \mathcal{D}$ must be of the

form $\{P@[u,5], Q@\{3\}\}$ with $u \in \{1,2,3\}$, hence Π -inconsistent. We can further observe that there is a unique i-conflict: $\mathcal{C} = \{P@[3,5], Q@\{3\}\}$. Finally observe that P@[1,2] has an empty pointwise intersection with the unique conflict \mathcal{C} but is not pointwise included in the i-repair \mathcal{R} .

Proof of Proposition 4 for x = p

Given a set of facts \mathcal{B} , let $\operatorname{tp}(\mathcal{B}) = \{\alpha @ \{t\} \mid \mathcal{B} \models \alpha @ \{t\}\}$. If $\mathbb{T} = \mathbb{Z}$ and \mathcal{B} uses only bounded intervals, $\operatorname{tp}(\mathcal{B})$ is a finite set of facts with punctual intervals. Moreover, for every two sets of facts \mathcal{B} and \mathcal{B}' , $\mathcal{B}' \sqsubseteq^{\mathcal{P}} \mathcal{B}$ iff $\operatorname{tp}(\mathcal{B}') \subseteq \operatorname{tp}(\mathcal{B})$.

Lemma 3. If $\mathbb{T} = \mathbb{Z}$, \mathcal{D} is a dataset that uses only bounded intervals, and Π is a DatalogMTL program, $\mathcal{R} \in pRep(\mathcal{D},\Pi)$ iff \mathcal{R} is in normal form and $tp(\mathcal{R})$ is $a \subseteq$ -maximal Π -consistent subset of $tp(\mathcal{D})$; and $\mathcal{C} \in pConf(\mathcal{D},\Pi)$ iff \mathcal{C} is in normal form and $tp(\mathcal{C})$ is $a \subseteq$ -minimal Π -inconsistent subset of $tp(\mathcal{D})$.

Proof. Let $\mathcal{R} \in pRep(\mathcal{D}, \Pi)$, i.e., \mathcal{R} is in normal form, Π -consistent and such that $\mathcal{R} \sqsubseteq^p \mathcal{D}$ and there is no Π -consistent \mathcal{R}' such that $\mathcal{R} \sqsubseteq^p \mathcal{R}' \sqsubseteq^p \mathcal{D}$. It follows that $\operatorname{tp}(\mathcal{R})$ is Π -consistent, $\operatorname{tp}(\mathcal{R}) \subseteq \operatorname{tp}(\mathcal{D})$ and there is no Π -consistent \mathcal{R}' such that $\operatorname{tp}(\mathcal{R}) \subseteq \mathcal{R}' \subseteq \operatorname{tp}(\mathcal{D})$ (otherwise it would hold that $\mathcal{R} \sqsubseteq^p \mathcal{R}' \sqsubseteq^p \mathcal{D}$). Hence $\operatorname{tp}(\mathcal{R})$ is a \subseteq -maximal Π -consistent subset of $\operatorname{tp}(\mathcal{D})$.

In the other direction, if \mathcal{R} is in normal form and $\operatorname{tp}(\mathcal{R})$ is a \subseteq -maximal Π -consistent subset of $\operatorname{tp}(\mathcal{D})$, then \mathcal{R} is Π -consistent and such that $\mathcal{R} \sqsubseteq^p \mathcal{D}$, and there is no Π -consistent \mathcal{R}' such that $\mathcal{R} \sqsubseteq^p \mathcal{R}' \sqsubseteq^p \mathcal{D}$ (otherwise it would hold that $\operatorname{tp}(\mathcal{R}) \subsetneq \operatorname{tp}(\mathcal{R}') \subseteq \operatorname{tp}(\mathcal{D})$). Hence $\mathcal{R} \in pRep(\mathcal{D}, \Pi)$.

The case of p-conflicts is similar.

Properties P_1 to P_4 follows from Lemma 3 as in the proof of Proposition 2, using the fact that normalization and tp do not change set finiteness when applied in the context of $\mathbb{T} = \mathbb{Z}$ and bounded intervals.

For Property P_5 , let $\alpha@\iota$ be pointwise included in \mathcal{D} . First, $\mathcal{R} \models \alpha@\iota$ for every $\mathcal{R} \in pRep(\mathcal{D},\Pi)$ iff for each $t \in \iota$, $\alpha@\{t\} \in \mathsf{tp}(\mathcal{R})$ for every $\mathcal{R} \in pRep(\mathcal{D},\Pi)$. Moreover, $\alpha@\{t\} \in \mathsf{tp}(\mathcal{R})$ for every $\mathcal{R} \in pRep(\mathcal{D},\Pi)$ iff $\alpha@\{t\} \notin \mathsf{tp}(\mathcal{C})$ for every $\mathcal{C} \in pConf(\mathcal{D},\Pi)$. Indeed, if $\alpha@\{t\} \in \mathsf{tp}(\mathcal{C})$ for some p-conflict \mathcal{C} , $\mathsf{tp}(\mathcal{C}) \setminus \{\alpha@\{t\}\}$ is Π -consistent thus can be extended then normalized into a p-repair \mathcal{R} such that $\mathcal{R} \not\models \alpha@\{t\}$, and if $\alpha@\{t\} \notin \mathsf{tp}(\mathcal{R})$ for some p-repair \mathcal{R} , $\mathsf{tp}(\mathcal{R}) \cup \{\alpha@\{t\}\}$ is Π -inconsistent and can be reduced then normalized into a p-conflict \mathcal{C} such that $\mathcal{C} \models \alpha@\{t\}$. It follows that $\mathcal{R} \models \alpha@\iota$ for every $\mathcal{R} \in pRep(\mathcal{D},\Pi)$ iff $\alpha@\iota$ has an empty pointwise intersection with every $\mathcal{C} \in pConf(\mathcal{D},\Pi)$.

C Proofs for Section 4.2

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

$$\mathcal{D}_r = \{ P@[0, \infty), Q@\{0\} \}$$

$$\Pi_r = \{ \bot \leftarrow Q \land \Phi_{(0, \infty)} \boxplus_{(0, \infty)} P \}$$

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 $(\mathcal{D}_r,\Pi_r)\not\models^x_{brave}Q@\{0\}$.

Proof. We show that for $x \in \{p, i\}$, the only x-repair is $\{P@[0,\infty)\}$. Let $\mathcal{R} \sqsubseteq^p \mathcal{D}$ be in normal form such that $Q@\{0\} \in \mathcal{R}$. If \mathcal{R} contains a fact of the form $P@\langle t,\infty\rangle$, then \mathcal{R} is Π_r -inconsistent. Otherwise, there is $t \in \mathbb{T}$ such that for every $P@\langle t_1,t_2\rangle \in \mathcal{R}$, $t_2 < t$, and $\mathcal{R}' = \{P@[0,t],Q@\{0\}\}$ is Π_r -consistent and such that $\mathcal{R} \sqsubseteq^p \mathcal{R}' \sqsubseteq^x \mathcal{D}_r$. Thus \mathcal{R} is not an x-repair.

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

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

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 $(\mathcal{D},\Pi) \not\models^{p}_{brave} Q@\{5\}$.

Proof. We show that the only p-repair is $\{P@[0,\infty)\}$. Let $\mathcal{R} \sqsubseteq^p \mathcal{D}$ be in normal form such that $Q@\{5\} \in \mathcal{R}$. If \mathcal{R} contains a fact of the form $P@\langle t,\infty\rangle$, \mathcal{R} is Π -inconsistent. Otherwise, there is $t \in \mathbb{T}$ such that for every $P@\langle t_1,t_2\rangle \in \mathcal{R}$, $t_2 < t$, and $\mathcal{R}' = \mathcal{R} \cup \{P@\{t\}\}$ is Π -consistent iff \mathcal{R} is, and is such that $\mathcal{R} \sqsubseteq^p \mathcal{R}' \sqsubseteq^p \mathcal{D}$. Hence \mathcal{R} is not a p-repair. \square

D Proofs for Section 5.1

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.

Proof. By definition, every dataset $\mathcal{B} \in sRep(\mathcal{D}, \Pi) \cup sConf(\mathcal{D}, \Pi)$ is such that $\mathcal{B} \sqsubseteq^s \mathcal{D}$, hence $\mathcal{B} \subseteq \mathcal{D}$, so the size of \mathcal{B} cannot exceed that of \mathcal{D} .

We recall that for (unrestricted) DatalogMTL programs, consistency checking and query entailment are PSPACEcomplete w.r.t. data complexity. This holds both for $\mathbb{T} = \mathbb{Q}$ and for $\mathbb{T} = \mathbb{Z}$ [Walega et al., 2019; Walega et al., 2020a]. It follows that we can decide in PSPACE whether a given subset $\mathcal{B} \subseteq \mathcal{D}$ is an s-conflict or s-repair. Indeed, by Lemma 2, $\mathcal{B} \in sConf(\mathcal{D}, \Pi)$ iff \mathcal{B} is a \subseteq -minimal Π -inconsistent subset of \mathcal{D} . This can be tested by (i) checking that \mathcal{B} is Π inconsistent, and (ii) testing, for each $\alpha@\iota \in \mathcal{B}$, whether $\mathcal{B} \setminus \{\alpha@\iota\}$ is Π -consistent. Similarly, due to Lemma 2, $\mathcal{B} \in \mathit{sRep}(\mathcal{D},\Pi)$ iff \mathcal{B} is a \subseteq -maximal Π -consistent subset of \mathcal{D} . It therefore suffices to test (i) whether \mathcal{B} is Π consistent, and (ii) whether $\mathcal{B} \cup \{\alpha@\iota\}$ is Π -inconsistent for each $\alpha@\iota \in \mathcal{D} \setminus \mathcal{B}$. As both procedures involve only a polynomial number of consistency checks, we obtain PSPACE procedures for recognizing s-conflicts and s-repairs.

Given a Π -inconsistent dataset \mathcal{D} , a single s-conflict $\mathcal{C} \in sConf(\mathcal{D},\Pi)$ can be obtained by starting with \mathcal{D} and removing facts until any further removal yields a Π -consistent dataset. Such a procedure runs in PSPACE when consistency checking is in PSPACE. In a similar manner, we can generate a single $\mathcal{R} \in sRep(\mathcal{D},\Pi)$ by starting from the empty set and adding facts from \mathcal{D} until any further addition leads to Π -inconsistency. Again, due to the need to perform only

polynomially many consistency checks, each in PSPACE, we obtain a PSPACE generation procedure.

The lower bounds for query entailment under s-repair semantics come from the consistent case. Since by Lemma 2, s-repairs are \subseteq -maximal Π -consistent subsets of \mathcal{D} , we obtain the upper bounds using the following 'standard' procedures for brave, CQA and intersection semantics:

- To decide if $(\mathcal{D},\Pi) \models^s_{\text{brave}} q(\vec{c},\iota)$, guess $\mathcal{R} \subseteq \mathcal{D}$ and check that $\mathcal{R} \in sRep(\mathcal{D},\Pi)$ and that $(\mathcal{R},\Pi) \models q(\vec{c},\iota)$.
- To decide if $(\mathcal{D}, \Pi) \not\models_{\text{CQA}}^s q(\vec{c}, \iota)$, guess $\mathcal{R} \subseteq \mathcal{D}$ and check that $\mathcal{R} \in sRep(\mathcal{D}, \Pi)$ and that $(\mathcal{R}, \Pi) \not\models q(\vec{c}, \iota)$.
- To decide if $(\mathcal{D},\Pi) \not\models_{\cap}^{s} q(\vec{c},\iota)$, guess $\{\alpha_{1}@\iota_{1},\ldots,\alpha_{n}@\iota_{n}\}\subseteq \mathcal{D}$ together with $\mathcal{B}_{1},\ldots,\mathcal{B}_{n}$ such that $\mathcal{B}_{i}\subseteq \mathcal{D}$ and $\alpha_{i}@\iota_{i}\notin \mathcal{B}_{i}$ for $1\leq i\leq n$, and check that $\mathcal{B}_{i}\in sRep(\mathcal{D},\Pi)$ for $1\leq i\leq n$ and $(\mathcal{D}\setminus\{\alpha_{1}@\iota_{1},\ldots,\alpha_{n}@\iota_{n}\},\Pi)\not\models q(\vec{c},\iota)$.

Using NPSPACE = PSPACE, we get PSPACE procedures for the three semantics. \Box

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.

Proof. We can use the same procedures as sketched in the proof of Proposition 6 to test whether a given subset $\mathcal{B}\subseteq\mathcal{D}$ belongs to $sConf(\mathcal{D},\Pi)$, or to generate a single $\mathcal{B}\in sConf(\mathcal{D},\Pi)$, and likewise for $sRep(\mathcal{D},\Pi)$. As these procedures involve polynomially many consistency checks, and such checks can be done in PTIME for the considered tractable DatalogMTL fragments, we obtain PTIME upper bounds for the s-conflict and s-repair recognition and generation tasks.

Proposition 8. For tractable DatalogMTL fragments: query entailment¹ under s-brave (resp. s-CQA, s-intersection) semantics is in NP (resp. coNP). Matching lower bounds hold in Datalog_{nr}MTL and DatalogMTL $_{lin}^{\diamondsuit}$ (and in DatalogMTL $_{core}^{\diamondsuit}$ in the case of s-CQA). The lower bounds hold even for bounded datasets and $\mathbb{T} = \mathbb{Z}$.

Proof. We can use the same procedures as described in the proof of Proposition 6 to perform query entailment under the s-brave, s-CQA, and s-intersection semantics. As s-repairs are of polynomial size (Proposition 6) and s-repair checking is in PTIME for all tractable fragments (Proposition 7), we obtain an NP procedure for s-brave semantics and coNP procedures for s-CQA and s-intersection semantics.

We now turn to the lower bounds, starting with the case of the s-CQA semantics. The proof is by reduction from (UN)SAT. Let $\varphi = c_1 \wedge ... \wedge c_m$ be a conjunction of clauses over variables $v_1,...,v_n$. We define a program Π and dataset

 \mathcal{D} as follows.

$$\begin{split} \Pi &= \{N'(v) \leftarrow \Leftrightarrow_{[0,\infty)} N(v), \ P'(v) \leftarrow \Leftrightarrow_{[0,\infty)} P(v), \\ &\perp \leftarrow P'(v) \wedge N'(v), \ Q \leftarrow \Leftrightarrow_{[0,\infty)} U, \\ &\perp \leftarrow P(v) \wedge U, \ \bot \leftarrow N(v) \wedge U \} \\ \mathcal{D} &= \{P(v_j)@\{2k\} \mid v_j \in c_k\} \cup \{N(v_j)@\{2k\} \mid \neg v_j \in c_k\}, \\ &\cup \{U@\{2k\} \mid 1 \leq k \leq m\} \end{split}$$

Note that since the punctual facts use only even integers, \mathcal{D} is in normal form even if we consider $\mathbb{T}=\mathbb{Z}$. We show that φ is unsatisfiable iff $(\mathcal{D},\Pi)\models^s_{\text{CQA}} Q@\{2m\}$. This establishes coNP-hardness of query entailment under s-CQA semantics for Datalog_nrMTL, DatalogMTL $^{\diamondsuit}_{\text{lin}}$, and DatalogMTL $^{\diamondsuit}_{\text{core}}$, since Π belongs to all three fragments. Moreover, \mathcal{D} is bounded and uses endpoints from \mathbb{Z} .

- (\Rightarrow) Assume that φ is unsatisfiable, and let $\mathcal R$ be an s-repair of $\mathcal D$ w.r.t. Π . Let ν be the valuation of $v_1,...,v_n$ such that $\nu(v_j)=$ true iff $P(v_j)@\{2k\}\in \mathcal R$ for some $1\leq k\leq m$. Since ν does not satisfy φ , there exists a clause c_ℓ such that
- $\nu(v_j) = \text{false}$, hence $P(v_j)@\{2\ell\} \notin \mathcal{R}$, for every $v_j \in c_\ell$
- $\nu(v_j) = \text{true}$, hence $P(v_j)@\{2k\} \in \mathcal{R}$ for some $1 \leq k \leq m$, for every $\neg v_j \in c_\ell$. Due to the consistency of \mathcal{R} and the rules $N'(v) \leftarrow \diamondsuit_{[0,\infty)} \ N(v), \ P'(v) \leftarrow \diamondsuit_{[0,\infty)} \ P(v)$, and $\bot \leftarrow P'(v) \land N'(v)$, it follows that $N(v_j)@\{2\ell\} \notin \mathcal{R}$, for every $\neg v_j \in c_\ell$.

By maximality of \mathcal{R} , it follows that $U@\{2\ell\} \in \mathcal{R}$, and by $Q \leftarrow \diamondsuit_{[0,\infty)} U$, we must have $(\mathcal{R},\Pi) \models Q@\{2m\}$. It follows that $(\mathcal{D},\Pi) \models_{\mathrm{COA}}^s Q@\{2m\}$.

(\Leftarrow) In the other direction, assume that $(\mathcal{D},\Pi) \models^s_{\text{CQA}} Q@\{2m\}$ and let ν be a valuation of $v_1,...,v_n$. Let \mathcal{R} be an s-repair that contains every $P(v_j)@\{2k\}$ such that $\nu(v_j) =$ true and every $N(v_j)@\{2k\}$ such that $\nu(v_j) =$ false. Since $(\mathcal{R},\Pi) \models Q@\{2m\}$, there exists $1 \leq k \leq m$ such that $U@\{2k\} \in \mathcal{R}$. Hence, there is no v_j such that $P(v_j)@\{2k\}$ or $N(v_j)@\{2k\}$ is in \mathcal{R} , i.e., ν does not satisfy c_k . It follows that φ is unsatisfiable.

For the s-brave semantics in Datalog_{nr}MTL, we give a reduction from SAT. Given $\varphi = c_1 \wedge ... \wedge c_m$ over variables $v_1, ..., v_n$, we consider the following program and dataset:

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

We show that φ is satisfiable iff $(\mathcal{D}',\Pi')\models^s_{\text{brave}} Q'@\{2\}$. As the program Π' is formulated in Datalog_{nr}MTL, this yields the desired NP lower bound for Datalog_{nr}MTL.

(\Rightarrow) Assume that φ is satisfiable and let ν be a valuation of $v_1,...,v_n$ that satisfies φ . Let $\mathcal R$ be an s-repair that contains every $P(v_j)@\{2k\}$ such that $\nu(v_j)=$ true and every $N(v_j)@\{2k\}$ such that $\nu(v_j)=$ false. Since ν satisfies φ , for every clause c_k , either there exists $v_j\in c_k$ such that $\nu(v_j)=$

true, i.e., $P(v_j)@\{2k\} \in \mathcal{R}$, or there exists $\neg v_j \in c_k$ such that $\nu(v_j) = \text{false}$, i.e., $N(v_j)@\{2k\} \in \mathcal{R}$. By the rules $S \leftarrow \diamondsuit_{[0,2)} P(v)$ and $S \leftarrow \diamondsuit_{[0,2)} N(v)$, it follows that for every $1 \leq k \leq m$, $(\mathcal{R}, \Pi') \models S@[2k, 2k+2)$, i.e., $(\mathcal{R}, \Pi') \models S@[2, 2m+2)$. By the rule $Q' \leftarrow S \mathcal{U}_{(0,\infty)} M$, it follows that $(\mathcal{R}, \Pi') \models Q'@\{2\}$. Hence $(\mathcal{D}', \Pi') \models_{\text{hrave}}^s Q'@\{2\}$.

 $(\Leftarrow) \text{ In the other direction, assume that } (\mathcal{D}',\Pi') \models^s_{\text{brave}} Q'@\{2\} \text{ and let } \mathcal{R} \text{ be an } s\text{-repair of } \mathcal{D}' \text{ w.r.t. } \Pi' \text{ such that } (\mathcal{R},\Pi') \models Q'@\{2\}. \text{ Let } \nu \text{ be the valuation of } v_1,...,v_n \text{ such that } \nu(v_j) = \text{ true iff } P(v_j)@\{2k\} \in \mathcal{R} \text{ for some } 1 \leq k \leq m. \text{ Since } (\mathcal{R},\Pi') \models Q'@\{2\} \text{ and } M@\{2m+2\} \text{ is the only occurence of } M \text{ in } \mathcal{D}' \text{ and cannot be inferred via } \Pi', \text{ it must be the case that } (\mathcal{R},\Pi') \models S@(2,2m+2). \text{ Again, by examining } \mathcal{D}' \text{ and } \Pi', \text{ this can only be achieved if for every } 1 \leq k \leq m, \text{ there is some } P(v_j)@\{2k\} \text{ or } N(v_j)@\{2k\} \text{ in } \mathcal{R}. \text{ If } P(v_j)@\{2k\} \in \mathcal{R}, \text{ then } \nu(v_j) = \text{ true and if } N(v_j)@\{2k\} \in \mathcal{R}, \text{ by consistency of } \mathcal{R} \text{ w.r.t. } N'(v) \longleftrightarrow_{[0,\infty)} N(v), N'(v) \longleftrightarrow_{[0,\infty)} N(v), \text{ and } \bot \leftarrow P(v) \land N'(v), \text{ then } \nu(v_j) = \text{ false. It follows that } \nu \text{ satisfies each clause } c_k, \text{ hence the formula } \varphi.$

To show coNP-hardness of query entailment under the sintersection semantics in Datalog_{nr}MTL, we consider

$$\Pi'' = \Pi' \cup \{\bot \leftarrow Q' \land Q''\} \text{ and } \mathcal{D}'' = \mathcal{D}' \cup \{Q''@\{2\}\}.$$

We argue that $(\mathcal{D}'', \Pi'') \models_{\cap}^{s} Q''@\{2\}$ iff φ is unsatisfiable.

 $(\Rightarrow) \text{ Assume that } \varphi \text{ is unsatisfiable, and let } \mathcal{R} \text{ be an } s\text{-repair of } \mathcal{D} \text{ w.r.t. } \Pi. \text{ Let } \nu \text{ be the valuation of } v_1,...,v_n \text{ such that } \nu(v_j) = \text{true iff } P(v_j) @\{2k\} \in \mathcal{R} \text{ for some } 1 \leq k \leq m. \text{ As in the } s\text{-CQA case, since } \nu \text{ does not satisfy } \varphi, \text{ there exists a clause } c_k \text{ such that for every } v_j \in c_k, P(v_j) @\{2k\} \notin \mathcal{R}, \text{ and for every } \neg v_j \in c_k, N(v_j) @\{2k\} \notin \mathcal{R}. \text{ Hence } (\mathcal{R}, \Pi'') \not\models S@\{t\} \text{ for } t \in [2k, 2k+2), \text{ so } (\mathcal{R}, \Pi'') \not\models S@(2, 2m+2) \text{ and since } M \text{ occurs only at time } m+1, (\mathcal{R}, \Pi'') \not\models Q'@\{2\}. \text{ By maximality of } \mathcal{R}, \text{ it follows that } Q''@\{2\} \in \mathcal{R}. \text{ Hence } (\mathcal{D}'', \Pi'') \models_{\Omega}^{s} Q''@\{2\}.$

 $(\Leftarrow) \text{ In the other direction, assume that } (\mathcal{D}'',\Pi'') \models_{\cap}^s Q''@\{2\} \text{ and let } \nu \text{ be a valuation of } v_1,...,v_n. \text{ Let } \mathcal{R} \text{ be an } s\text{-repair that contains every } P(v_j)@\{2k\} \text{ such that } \nu(v_j) = \text{true and every } N(v_j)@\{2k\} \text{ such that } \nu(v_j) = \text{false. Since } Q''@\{2\} \in \mathcal{R}, \text{ by the rule } \bot \leftarrow Q' \land Q'', (\mathcal{R},\Pi'') \not\models Q'@\{2\}. \text{ It follows that there exists } t \in (2,2m+2) \text{ such that } (\mathcal{R},\Pi'')\not\models S@\{t\}, \text{ which implies that there is some } 1 \leq k \leq m \text{ such that there is no } P(v_j)@\{2k\} \text{ or } N(v_j)@\{2k\} \text{ in } \mathcal{R}. \text{ It follows that } \nu \text{ does not satisfy } c_k. \text{ Hence } \varphi \text{ is unsatisfiable.}$

The preceding reductions for s-brave and s-intersection semantics use programs with $\mathcal U$ and thus do not yield hardness results for DatalogMTL $_{\mathrm{lin}}^{\diamondsuit}$. We provide a different but similar reduction from SAT to show NP-hardness of query entailment under s-brave semantics in DatalogMTL $_{\mathrm{lin}}^{\diamondsuit}$. Starting again from a SAT instance $\varphi = c_1 \wedge ... \wedge c_m$ over variables

 $v_1, ..., v_n$, we consider the program and dataset:

$$\begin{split} \Pi_{\mathsf{lin}}' &= \{ N'(v) \leftarrow \Diamond_{[0,\infty)} \ N(v), \ P'(v) \leftarrow \Diamond_{[0,\infty)} \ P(v), \\ &\perp \leftarrow P'(v) \wedge N'(v), \ S \leftarrow F, \\ &S \leftarrow P(v) \wedge \Diamond_{\{2\}} \ S, \ S \leftarrow N(v) \wedge \Diamond_{\{2\}} \ S \} \\ \mathcal{D}_{\mathsf{lin}}' &= \{ P(v_j) @ \{2k\} \mid v_j \in c_k \} \ \cup \ \{ N(v_j) @ \{2k\} \mid \neg v_j \in c_k \} \\ &\cup \ \{ F @ \{0\} \} \end{split}$$

Using arguments similar to those given previously, it can be shown that φ is satisfiable iff $(\mathcal{D}'_{\text{lin}}, \Pi'_{\text{lin}}) \models^s_{\text{brave}} S@\{2m\}$. Indeed, each repair \mathcal{R} induces a valuation $\nu_{\mathcal{R}}$ of v_1, \ldots, v_n , and we can use $(\mathcal{R}, \Pi'_{\text{lin}}) \models S@\{2m\}$ and the rules for transmitting S to argue that each clause is satisfied by $\nu_{\mathcal{R}}$. Conversely, every satisfying valuation allows us to build such a repair \mathcal{R} with $(\mathcal{R}, \Pi'_{\text{lin}}) \models S@\{2m\}$.

Finally, we adapt the preceding reduction to show coNP-hardness in the case of s-intersection semantics in DatalogMTL $_{\rm lin}^{\diamondsuit}$. Given a CNF $\varphi=c_1\wedge\ldots\wedge c_m$ over variables v_1,\ldots,v_n , we consider the program and dataset:

$$\Pi_{\mathsf{lin}}^{\prime\prime} = \Pi_{\mathsf{lin}}^{\prime} \cup \{\bot \leftarrow S \land S^{\prime}\} \quad \mathcal{D}_{\mathsf{lin}}^{\prime} = \mathcal{D}_{\mathsf{lin}}^{\prime} \cup \{S^{\prime} @ \{2m\}\}$$

It is not hard to see that φ is unsatisfiable iff $(\mathcal{D}''_{lin}, \Pi''_{lin}) \models_{\cap}^s S' @\{2m\}$. Indeed, $S' @\{2m\}$ belongs to the intersection of s-repairs iff there is no repair that entails $S @\{2m\}$ iff there is no satisfying valuation for φ .

Proposition 9. Datalog $MTL_{core}^{\diamondsuit}$ query entailment¹ under s-brave and s-intersection semantics is in PTIME.

Proof. The following properties of DatalogMTL have been shown in [Walega *et al.*, 2020a; Walega *et al.*, 2020b], for both the integer and rational timelines:

- If D is Π-inconsistent, then there exists α₁@ι₁, α₂@ι₂ ∈ D such that {α₁@ι₁, α₂@ι₂} is Π-inconsistent, i.e. every s-conflict contains at most 2 facts.
- If $(\mathcal{D},\Pi) \models \beta@\{t\}$, then either $\Pi \models \beta@\{t\}$ (e.g. if β is a nullary predicate and there is a rule $\beta \leftarrow \top$) or there exists $\alpha@\iota \in \mathcal{D}$ such that $(\{\alpha@\iota\},\Pi) \models \beta@\{t\}$.

It follows that to decide whether $(\mathcal{D},\Pi) \models_{\text{brave}}^s \beta@\{t\}$ (note that here we focus on queries with punctual intervals) it suffices to check whether:

- $\Pi \models \beta@\{t\}$, or
- there exists $\alpha@\iota \in \mathcal{D}$ such that $\{\alpha@\iota\}$ is Π -consistent and $(\{\alpha@\iota\}, \Pi) \models \beta@\{t\}$

Indeed, a Π -consistent $\{\alpha@\iota\}\subseteq\mathcal{D}$ can always be greedily extended into a s-repair of \mathcal{D} w.r.t. Π . Thus, we need only perform a linear number of consistency and punctual query entailment checks. Such checks being possible in PTIME for DatalogMTL $_{\text{core}}^{\diamondsuit}$, we obtain a PTIME procedure for punctual query entailment under s-brave semantics.

We can proceed similarly for *s*-intersection semantics. To decide whether $(\mathcal{D}, \Pi) \models^s_{\cap} \beta@\{t\}$, we must check whether:

- $\Pi \models \beta@\{t\}$, or
- there exists $\alpha@\iota\in\mathcal{D}$ such that
 - $\{\alpha@\iota\}$ is Π -consistent,

- $-(\{\alpha@\iota\},\Pi)\models\beta@\{t\},$ and
- for every $\alpha'@\iota'\in\mathcal{D}$, if $\{\alpha'@\iota'\}$ is Π -consistent, then the set $\{\alpha@\iota,\alpha'@\iota'\}$ is also Π -consistent,

and then return yes if one of these two conditions is satisfied. Note that the described procedure runs in PTIME as it involves only quadratically many consistency and entailment checks, and each such check is possible in PTIME. The correctness of this procedure follows from the aforementioned properties of entailment in DatalogMTL $_{core}^{\diamondsuit}$, together with Corollary 1, which tells us that $\alpha@\iota\in\mathcal{D}$ belongs to every s-repair iff it appears in no s-conflict, which is checked by the third point of the second condition, since s-conflicts contain at most two facts in this case.

D.1 Proof of Proposition 10

We first restate the proposition we wish to prove.

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).

Let us start by considering a propositional DatalogMTL program Π , dataset \mathcal{D} , and query $Q@\langle t_B, t_E \rangle$. Following [Walega et~al., 2020a], we will assume w.l.o.g. that 0 is the least integer mentioned in the input dataset \mathcal{D} , and will denote by d the largest integer mentioned in \mathcal{D} (note that \mathcal{D} may also use ∞ or $-\infty$ as endpoints). Moreover, as we work over \mathbb{Z} , we may assume w.l.o.g. that any interval $\langle t_1, t_2 \rangle$ mentioned in \mathcal{D} is such that $\langle = [$ if $t_1 \in \mathbb{Z}$ and $\rangle =]$ if $t_2 \in \mathbb{Z}$ (i.e. we use '(' and ')' only for endpoints $-\infty$ and ∞). We further assume, as we have done throughout the paper, that \mathcal{D} is in normal form. Finally, we assume w.l.o.g. that all propositions in \mathcal{D} , as well as the query proposition Q, occur in Π .

To simplify the presentation, we focus on the case in which the query is $Q@[t_B,t_E]$, where the query interval has endpoints t_B,t_E such that $0 < t_B \le t_E < d$ and uses square brackets. The proof is straightforwardly adapted to handle queries using $-\infty$ or ∞ as endpoints, or integers outside the range (0,d).

The starting point for the proof will be the original and succinct NFA constructions from [Walega *et al.*, 2020a], which were used to show that, when $\mathbb{T} = \mathbb{Z}$, consistency checking in propositional DatalogMTL is in NC¹ \subseteq PTIME. We will slightly modify these consistency-checking NFAs to make them better suited to our purposes and then use the modified succinct NFA as a building block for constructing more complex NFAs for testing query entailment under *s*-brave, *s*-CQA, and *s*-intersection semantics.

As part of our procedure, we will need to test query entailment under classical semantics, which will be done via a standard reduction to consistency checking, using the program

$$\Pi^Q = \Pi \cup \{Q' \leftarrow Q \mathcal{U}_{[0,\infty)} E, \bot \leftarrow B \land Q'\}$$

where Q', B, E are fresh propositions that do not occur in Π nor \mathcal{D} . The reduction is provided in the following lemma.

Lemma 4. For every dataset \mathcal{D}^{\dagger} such that \mathcal{D}^{\dagger} is Π -consistent, the following are equivalent²:

1.
$$(\mathcal{D}^{\dagger}, \Pi) \models Q@[t_B, t_E]$$

2.
$$\mathcal{D}^{\dagger} \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}\$$
 is Π^Q -inconsistent

In what follows, we will denote by \mathcal{D}^Q the dataset $\mathcal{D} \cup \{S@\{t_B-1\}, E@\{t_E+1\}\}$. We will denote by $\operatorname{prop}(\Pi)$ and $\operatorname{prop}(\Pi^Q)$ the set of propositions appearing in Π and in Π^Q respectively. We will use $\operatorname{lit}(\Pi)$ (resp. $\operatorname{lit}(\Pi^Q)$) for the (finite!) set of literals without nested temporal operators, with all propositions drawn from $\operatorname{prop}(\Pi)$ (resp. $\operatorname{prop}(\Pi^Q)$) and all integer endpoints bounded by the maximal integer in Π (which by our assumptions is the same in Π^Q).

The following definition, taken from [Walega *et al.*, 2020a], introduces consistent sets of literals and a successor relation between them, which will be used when defining automata states and transitions.

Definition 7. A set $q \subseteq \text{lit}(\Pi)$ is Π -consistent if there is a model \mathfrak{M} of Π such that $\mathfrak{M}, 0 \models L$ iff $L \in q$, for each $L \in \text{lit}(\Pi)$. We use $\text{con}(\Pi)$ for the set of all Π -consistent subsets of $\text{lit}(\Pi)$.

For $q, q' \in con(\Pi)$, q' is a Π -successor of q if there is a model \mathfrak{M} of Π such that for all $L \in lit(\Pi)$:

- $\mathfrak{M}, 0 \models L \text{ iff } L \in q, \text{ and }$
- $\mathfrak{M}, 1 \models L \text{ iff } L \in q'.$

We can define $con(\Pi^Q)$ and Π^Q -successors in the same way, simply considering Π^Q in place of Π .

In the original construction, the input dataset is associated with a word over the alphabet $\mathcal{P}(\mathsf{prop}(\Pi))$, with each letter specifying the set of propositions that are stated to hold at the considered timepoint. For our purposes, it will be convenient to work with a larger alphabet:

$$\Sigma = \mathcal{P}(\Sigma_{\mathsf{prop}}^+) \cup \mathcal{P}(\Sigma_{-\infty}) \cup \mathcal{P}(\Sigma_{\infty})$$

where the component alphabets of annotated propositions are defined as follows:

- $\Sigma_{\text{prop}}^+ = \text{prop}(\Pi) \cup \Sigma_{\text{open}} \cup \Sigma_{\text{close}} \cup \{B, E\}$
- $\Sigma_{-\infty} = \{ P_{-\infty} \mid P \in \operatorname{prop}(\Pi) \}$
- $\Sigma_{\infty} = \{ P_{\infty} \mid P \in \operatorname{prop}(\Pi) \}$
- $\Sigma_{\text{open}} = \{ P_{\text{open}} \mid P \in \text{prop}(\Pi) \}$
- $\Sigma_{\mathsf{close}} = \{ P_{\mathsf{close}} \mid P \in \mathsf{prop}(\Pi) \}$

Intuitively, P_{open} and P_{close} are used to indicate the start and end of an interval of a P-fact, $P_{-\infty}$ indicates a P-fact with endpoint $-\infty$, and P_{∞} indicates a P-fact with endpoint ∞ . Given any $\sigma \in \Sigma$, we denote by σ^- the set $\sigma \cap \text{prop}(\Pi)$. In this manner, we can 'convert' any letter in our expanded alphabet into a symbol in the original alphabet. We will further introduce σ_O^- as the set $\sigma \cap (\text{prop}(\Pi) \cup \{B, E\})$.

The following definition shows how we will construct the word $w_{\Pi,\mathcal{D},Q}$ from the input dataset \mathcal{D} (which has minimum and maximum integer endpoints 0 and d) and query $Q@[t_B,t_E]$ over this expanded alphabet:

Definition 8. The word $w_{\Pi,\mathcal{D},Q}$ is $\sigma_{-1}\sigma_0 \dots \sigma_d \sigma_{d+1}\sigma_{d+2}$, where:

semantics of \mathcal{U} , which considers the open interval (t,t'), equivalent over \mathbb{Z} to [t+1,t'-1], in between the considered timepoints t,t'.

²We need to use t_B-1 and t_E+1 rather than t_B and t_E due to the

- $\sigma_{-1} \subseteq \Sigma_{-\infty}$ is the set of $P_{-\infty}$ such that \mathcal{D} contains some fact of the form $P@(-\infty, m]$ or $P@(-\infty, \infty)$
- $\sigma_{d+1} \subseteq \Sigma_{\mathsf{prop}}^+$ is the set of $P \in \mathsf{prop}(\Pi)$ such that \mathcal{D} contains some fact of the form $P@[m,\infty)$ or $P@(-\infty,\infty)$
- σ_{d+2} ⊆ Σ_∞ is the set of P_∞ such that D contains some fact of the form P@[m, ∞) or P@(-∞, ∞)
- for every $0 \le j \le d$, $\sigma_j \subseteq \Sigma_{\mathsf{prop}}^+$ is the set comprising all:
 - P ∈ prop(Π) such that $\mathcal{D} \models P@\{j\}$
 - P_{open} such that \mathcal{D} contains P@[j,j'] or $P@[j,\infty)$
 - P_{close} such that \mathcal{D} contains some P@[j',j] or $P@(-\infty,j]$
 - $B \text{ if } j = t_B 1$
 - $E \ if \ j = t_E + 1$

Remark 1. If $t \in [0,d]$ is not mentioned in \mathcal{D}^Q , then $\sigma_t \subseteq \operatorname{prop}(\Pi)$. In other words, the propositions from $\Sigma_{\operatorname{open}} \cup \Sigma_{\operatorname{close}} \cup \{B, E\}$ can only appear in σ_t if timepoint t is explicitly mentioned in \mathcal{D}^Q . Further note that if t < t' are mentioned in \mathcal{D}^Q , but there is no t'' mentioned with t < t'' < t', then $\sigma_j = \sigma_{j'}$ for all t < j, j' < t'.

We now define NFAs that can check Π - and Π^Q -consistency. Note that both automata are defined for words over the alphabet Σ , but when reading σ , it will ignore elements of σ from $\Sigma_{\text{open}} \cup \Sigma_{\text{close}}$. Moreover, the NFA for Π -consistency will further ignore B, E. Later constructions will make use of the full alphabet.

Definition 9. The NFA $\mathfrak{A}_{\Pi} = (S_{\Pi}, \Sigma, \delta_{\Pi}, s_0, s_f)$ is defined as follows:

- the set of states S_{Π} is $con(\Pi) \cup \{s_0, s_f\}$, with s_0 the unique initial state and s_f the unique final state
- the alphabet is Σ
- the transition function δ_{Π} allows all and only the following transitions:
 - **–** for σ ⊆ $\Sigma_{-\infty}$ and q ∈ con(Π):

$$q \in \delta_{\Pi}(s_0, \sigma)$$
 iff $\exists_{(-\infty,0]} P \in q \text{ for each } P_{-\infty} \in \sigma$

- for $\sigma \subseteq \Sigma_{\text{prop}}^+$, $q \in \text{con}(\Pi)$, and $q' \in \text{con}(\Pi)$:

$$q' \in \delta_{\Pi}(q, \sigma)$$
 iff q' is Π -successor of q and $\sigma^- \subseteq q'$

- for $\sigma \subseteq \Sigma_{\infty}$ and $g \in con(\Pi)$:

$$s_f \in \delta_{\Pi}(q,\sigma)$$
 iff $\boxplus_{[0,\infty)} P \in q \text{ for every } P_{\infty} \in \sigma$

The NFA for Π^Q -consistency is essentially the same but uses ${\rm con}(\Pi^Q)$ and σ_Q^- in place of ${\rm con}(\Pi)$ and σ^- :

Definition 10. The NFA $\mathfrak{A}_{\Pi}^Q = (S_{\Pi}^Q, \Sigma, \delta_{\Pi}^Q, s_0, s_f)$ is defined as follows:

- the set of states S_{Π}^{Q} is $con(\Pi^{Q}) \cup \{s_{0}, s_{f}\}$, with s_{0} the unique initial state and s_{f} the unique final state
- the alphabet is Σ
- the transition function δ_{Π}^Q allows all and only the following transitions:

$$\begin{split} &-\textit{for }\sigma\subseteq \Sigma_{-\infty}\textit{ and }q\in \text{con}(\Pi^Q),\\ &q\in \delta_\Pi^Q(s_0,\sigma)\quad \textit{iff}\quad \ \, \boxminus_{(-\infty,0]}P\in q\textit{ for each }P_{-\infty}\in\sigma\\ &-\textit{for }\sigma\subseteq \Sigma_{\text{prop}}^+,\,q\in \text{con}(\Pi^Q),\textit{ and }q'\in \text{con}(\Pi^Q). \end{split}$$

 $q' \in \delta_{\Pi}^Q(q,\sigma) \quad \textit{ iff } \quad q' \textit{ is } \Pi^Q \textit{-successor of } q \textit{ and } \sigma_Q^- \subseteq q'$

- for $\sigma \subseteq \Sigma_{\infty}$ and $q \in con(\Pi^Q)$:

$$s_f \in \delta_\Pi^Q(q,\sigma) \quad \textit{ iff } \quad \boxplus_{[0,\infty)} \ P \in q \textit{ for every } P_\infty \in \sigma$$

The NFAs \mathfrak{A}_Π and \mathfrak{A}_Π^Q are largely similar to the NFA $\mathcal{A}_{\Pi,\mathcal{D}}$ given in [Walega et al., 2020a]. The key difference is that we use a single initial and final state and use the first and the final letters of the word $w_{\Pi,\mathcal{D},Q}$ to ensure that the timepoints before 0 and after d are correctly handled. In the original construction, a set of initial and final states was used, and which states were taken to be initial / final depended on \mathcal{D} . Our modification ensures that \mathfrak{A}_Π and \mathfrak{A}_Π^Q can be constructed fully independently from \mathcal{D} . By adapting the analogous result (Lemma 6) in [Walega et al., 2020a], we obtain the following:

Lemma 5. \mathcal{D} is Π -consistent iff \mathfrak{A}_{Π} accepts $w_{\Pi,\mathcal{D},Q}$. Likewise, \mathcal{D}^Q is Π^Q -consistent iff \mathfrak{A}_{Π}^Q accepts $w_{\Pi,\mathcal{D},Q}$.

Proof. We briefly describe the relationship between our construction and the one in [Walega et al., 2020a]. Let $w_{\Pi,\mathcal{D},Q} =$ $\sigma_{-1}\sigma_0\ldots\sigma_d\sigma_{d+1}\sigma_{d+2}$ be the word associated with \mathcal{D} and the query according to Definition 8. In the construction of [Walega et al., 2020a], no query is considered, and one uses instead the word $\sigma_0^- \dots \sigma_d^- \sigma_{d+1}^-$, which we shall denote by $w_{\Pi,\mathcal{D}}^*.$ The NFA $\mathcal{A}_{\Pi,\mathcal{D}}$ defined in [Walega $\mathit{et~al.},\ 2020a$] has states $con(\Pi)$ and alphabet Σ_{prop} . Comparing the definitions of \mathfrak{A}_{Π} and $\mathcal{A}_{\Pi,\mathcal{D}}$, we can observe that the transitions of \mathfrak{A}_{Π} between states $q, q' \in \mathsf{con}(\Pi)$ precisely mirror the transitions of $\mathcal{A}_{\Pi,\mathcal{D}}$. Moreover, our transitions from s_0 to $q \in con(\Pi)$ simulate the set of initial states in $\mathcal{A}_{\Pi,\mathcal{D}}$, which ensures that $q_{-1} \in \delta_{\Pi}(s_0, \sigma_{-1})$ iff q_{-1} is an initial state of $A_{\Pi,\mathcal{D}}$. Likewise, the transitions to s_f simulate the final states in $\mathcal{A}_{\Pi,\mathcal{D}}$: $s_f \in \delta_{\Pi}(q_{d+1},\sigma_{d+2})$ iff q_{d+1} is a final state of $A_{\Pi,\mathcal{D}}$. From the preceding observations, one can show that every accepting run $s_0q_{-1}q_0q_1\dots q_dq_{d+1}s_f$ of \mathfrak{A}_Π on $w_{\Pi,\mathcal{D},Q}$ gives rise to the accepting run $q_{-1}q_0\dots q_dq_{d+1}$ of $\mathcal{A}_{\Pi,\mathcal{D}}$ on $w_{\Pi,\mathcal{D}}^*$, and vice-versa. It follows that \mathfrak{A}_{Π} accepts $w_{\Pi,\mathcal{D},\mathcal{Q}}$ iff $\mathcal{A}_{\Pi,\mathcal{D}}$ accepts $w_{\Pi,\mathcal{D}}$. Moreover, by Lemma 6 of [Walega et al., 2020a], the latter holds iff \mathcal{D} is Π -consistent, which establishes the first statement.

For \mathfrak{A}_{Π}^Q , we can apply the same construction but starting from the program Π^Q , in which case $\mathcal{A}_{\Pi^Q,\mathcal{D}^Q}$ will use the set of states $\operatorname{con}(\Pi^Q)$ and alphabet $\mathcal{P}(\operatorname{prop}(\Pi^Q))$. The word $w_{\Pi^Q,\mathcal{D}^Q}^*$ associated with \mathcal{D}^Q will additionally include B and E at positions t_B-1 and t_E+1 , i.e. $\sigma_{t_B-1}^-$ is replaced by $(\sigma_{t_B-1})_Q^-$ and $\sigma_{t_E+1}^-$ by $(\sigma_{t_E+1})_Q^-$. We can again notice that the transitions of \mathfrak{A}_{Π}^Q between states $q,q'\in\operatorname{con}(\Pi^Q)$ corresponds to the transitions of $\mathcal{A}_{\Pi^Q,\mathcal{D}^Q}$, and that the transitions from s_0 and to s_f simulate the initial and final states of $\mathcal{A}_{\Pi^Q,\mathcal{D}^Q}$. We can show in this manner that \mathfrak{A}_{Π}^Q accepts $w_{\Pi,\mathcal{D},Q}$ iff $\mathcal{A}_{\Pi^Q,\mathcal{D}^Q}$ accepts $w_{\Pi,\mathcal{D},Q}^*$ iff \mathcal{D}^Q is Π^Q -consistent

(the final equivalence is due to Lemma 6 of [Walega *et al.*, 2020a]).

The preceding lemma, like the corresponding Lemma 6 from [Walega *et al.*, 2020a], does not yield a PTIME procedure since the word that encodes $\mathcal D$ may have size exponential in $\mathcal D$, due to the binary encoding of endpoint integers. In [Walega *et al.*, 2020a], this issue was overcome by devising a polynomial-size representation $\overline{w}_{\Pi,\mathcal D}^*$ of $w_{\Pi,\mathcal D}^*$ (recall that we are using $w_{\Pi,\mathcal D}^*$ to refer to the word encoding from that paper) and a so-called 'succinct' NFA $\overline{\mathcal A}_{\Pi,\mathcal D}$, both over an extended alphabet, with the property that $\mathcal A_{\Pi,\mathcal D}$ accepts $w_{\Pi,\mathcal D}^*$ iff $\overline{\mathcal A}_{\Pi,\mathcal D}$ may be exponentially long, it can be split into polynomially many segments, and within each segment, the same propositions are asserted (cf. Remark 1).

As we will adopt the same technique for our constructions, we recall the main elements of the approach. First, the following lemma is proven to characterize when it is possible to move from a state q to q' when reading a word consisting of a single repeated symbol. We have slightly adapted the formulation of the original result (Lemma 7 from [Walega et al., 2020a]) so that it uses our notation and our modified base NFAs and alphabet.

Lemma 6. For all $q, q' \in \text{con}(\Pi)$ and $\sigma \subseteq \Sigma_{\text{prop}}^+$, it is possible to construct a finite set $T_{\Pi}(q, q', \sigma)$ of pairs of nonnegative integers such that the following are equivalent, for all $\ell \in \mathbb{N}$:

- 1. there is a run of \mathfrak{A}_{Π} from q to q' on word σ^{ℓ}
- 2. $\ell = a + n \cdot b$ for some $(a,b) \in T_{\Pi}(q,q',\sigma)$ and $n \in \mathbb{N}$

We can similarly construct a set $T_{\Pi}^Q(q,q',\sigma)$, for $q,q' \in \text{con}(\Pi^Q)$, which satisfies the analogous properties with \mathfrak{A}_{Π} replaced by \mathfrak{A}_{Π}^Q .

The succinct NFAs and succinct word encoding can now be defined as follows. Our definitions adapt Definition 8 in [Walega *et al.*, 2020a] to our setting:

Definition 11. Let \mathfrak{A}_{Π} and \mathfrak{A}_{Π}^{Q} be the previously defined NFAs. Define the following sets of pairs

$$\begin{split} \mathsf{paths}(\Pi) = & \{(a,b) \mid (a,b) \in T_\Pi(q,q',\sigma) \, \textit{for some} \\ & q,q' \in \mathsf{con}(\Pi) \, \textit{and} \, \sigma \subseteq \Sigma_{\mathsf{prop}}^+ \} \\ \mathsf{paths}(\Pi^Q) = & \{(a,b) \mid (a,b) \in T_\Pi^Q(q,q',\sigma) \, \textit{for some} \\ & q,q' \in \mathsf{con}(\Pi^Q) \, \textit{and} \, \sigma \subseteq \Sigma_{\mathsf{prop}}^+ \} \end{split}$$

and let $\overline{\Sigma}$ be the alphabet defined as follows³:

$$\overline{\Sigma} = \!\! \{ (\sigma, \tau_\Pi, \tau_\Pi^Q) \mid \sigma \in \Sigma, \tau_\Pi \subseteq \mathsf{paths}(\Pi), \tau_\Pi^Q \subseteq \mathsf{paths}(\Pi^Q) \}$$

Definition 12. The NFA $\overline{\mathfrak{A}}_{\Pi} = (S_{\Pi}, \overline{\Sigma}, \overline{\delta}_{\Pi}, s_0, s_f)$ has a transition function $\overline{\delta}_{\Pi}$ which allows all and only the following transitions:

• for $\sigma \subseteq \Sigma_{-\infty}$:

$$\overline{\delta}_{\Pi}(s_0,(\sigma,\emptyset,\emptyset)) = \delta_{\Pi}(s_0,\sigma)$$

• for $\sigma\subseteq\Sigma^+_{\mathrm{prop}},\ \tau_\Pi\subseteq\mathrm{paths}(\Pi),\ \tau_\Pi^Q\subseteq\mathrm{paths}(\Pi^Q),\ and\ q\in\mathrm{con}(\Pi)$:

$$\overline{\delta}_{\Pi}(q,(\sigma,\tau_{\Pi},\tau_{\Pi}^{Q})) = \{q' \in \mathsf{con}(\Pi) \mid \tau_{\Pi} \cap T_{\Pi}(q,q',\sigma) \neq \emptyset\}$$

• for $\sigma \subseteq \Sigma_{\infty}$ and $q \in con(\Pi)$:

$$\overline{\delta}_{\Pi}(q,(\sigma,\emptyset,\emptyset)) = \delta_{\Pi}(q,\sigma)$$

The transition function $\overline{\delta}_{\Pi}^Q$ of the NFA $\overline{\mathfrak{A}}_{\Pi}^Q = (S_{\Pi}^Q, \overline{\Sigma}, \overline{\delta}_{\Pi}^Q, s_0, s_f)$ is defined as follows:

• for $\sigma \subseteq \Sigma_{-\infty}$:

$$\overline{\delta}_{\Pi}^{Q}(s_{0},(\sigma,\emptyset,\emptyset)) = \delta_{\Pi}^{Q}(s_{0},\sigma)$$

• for $\sigma\subseteq \Sigma_{\mathrm{prop}}^+,\ \tau_\Pi\subseteq\mathrm{paths}(\Pi),\ \tau_\Pi^Q\subseteq\mathrm{paths}(\Pi^Q),\ and\ q\in\mathrm{con}(\Pi^Q).$

$$\overline{\delta}_{\Pi}^Q(q,(\sigma,\tau_{\Pi},\tau_{\Pi}^Q)) = \{q' \in \mathsf{con}(\Pi^Q) \mid \tau_{\Pi}^Q \cap T_{\Pi}^Q(q,q',\sigma) \neq \emptyset\}$$

• for $\sigma \subseteq \Sigma_{\infty}$ and $q \in con(\Pi^Q)$:

$$\overline{\delta}_{\Pi}^{Q}(q,(\sigma,\emptyset,\emptyset)) = \delta_{\Pi}^{Q}(q,\sigma)$$

Definition 13. Let $0 = t_1 < \ldots < t_N = d$ be all of the integers mentioned in \mathcal{D} , and let $\varrho_0, \ldots, \varrho_{m+1}$ be the subsequence of all non-empty intervals in the sequence

$$\{t_1\}, (t_1, t_2), \{t_2\}, \dots, (t_{N-1}, t_N), \{t_N\}, \{t_N + 1\}$$

Then we define the word $\overline{w}_{\Pi,\mathcal{D},Q}$ as⁴

$$(\sigma_{-1},\emptyset,\emptyset)(\sigma_0^*,\tau_0,\tau_0')\dots(\sigma_{m+1}^*,\tau_{m+1},\tau_{m+1}')(\sigma_{d+2},\emptyset,\emptyset)$$

with σ_{-1} and σ_{d+2} defined as in Definition 8 and for every $1 \le j \le m$:

- if $\varrho_j = \{t_k\}$ for some $1 \le k \le N$, then $\sigma_i^* = \sigma_{t_k}$
- if $\varrho_j = (t_k, t_{k+1})$ for some $0 \le k \le N$, then $\sigma_i^* = \{ P \in \mathsf{prop}(\Pi) \mid \mathcal{D} \models P@(t_k, t_{k+1}) \}$
- $\tau_i = \{(a,b) \in \mathsf{paths}(\Pi) \mid |\varrho_i| = a + n \cdot b \text{ for some } n \in \mathbb{N}\}$
- $\tau_j' = \{(a,b) \in \mathsf{paths}(\Pi^Q) | |\varrho_j| = a + n \cdot b \text{ for some } n \in \mathbb{N} \}$

where $|\rho_j|$ is the cardinality of the set of timepoints associated with ρ_j .

We can now state the key lemma concerning the succinct NFAs, whose proof is a straightforward adaptation of the analogous Lemma 9 in [Walega *et al.*, 2020a], using Lemma 6 and Remark 1:

 $^{^3}$ We use the Greek letter τ , with sub- and superscripts, for the elements of paths(Π) and paths(Π^Q), whereas σ was used in [Walega et al., 2020a], which we use instead for elements of Σ . While we have generally tried to keep notations consistent with the latter paper, some changes were needed to accommodate the many further symbols needed to specify our constructions.

⁴A note to readers who compare our definition with its analogue in [Walega *et al.*, 2020a]: we point out that, for purely presentational reasons, we denote the sequence by $\varrho_0, \ldots, \varrho_{m+1}$ (with starting index 0 and final index m+1) rather than $\varrho_1, \ldots, \varrho_m$.

Lemma 7.

- 1. NFA \mathfrak{A}_{Π} accepts $w_{\Pi,\mathcal{D},Q}$ iff NFA $\overline{\mathfrak{A}}_{\Pi}$ accepts $\overline{w}_{\Pi,\mathcal{D},Q}$.
- 2. NFA \mathfrak{A}_{Π}^{Q} accepts $w_{\Pi,\mathcal{D},Q}$ iff NFA $\overline{\mathfrak{A}}_{\Pi}^{Q}$ accepts $\overline{w}_{\Pi,\mathcal{D},Q}$.

Having recalled, and suitably adapted, the main constructions from [Walega *et al.*, 2020a], we are now ready to build upon these constructions to obtain NFAs that can decide query entailment under s-brave, s-CQA, and s-intersection semantics.

Construction for s**-brave semantics**

We will start with s-brave semantics, as it is the simplest case and will provide useful components for tackling the two other semantics. The following characterization, which follows from the definition of s-brave semantics and Lemma 4, shows us what we need to check:

Lemma 8. $(\mathcal{D}, \Pi) \models^s_{brave} Q@[t_B, t_E]$ iff there exists a subset $\mathcal{D}' \subseteq \mathcal{D}$ such that:

- \mathcal{D}' is Π -consistent, and
- $\mathcal{D}' \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}\)$ is Π^Q -inconsistent

Our aim is thus to construct an NFA $\mathfrak{A}_{\mathsf{brave}}$ which accepts $\overline{w}_{\Pi,\mathcal{D},Q}$ iff such a subset $\mathcal{D}' \subseteq \mathcal{D}$ exists. To do so, we will build intermediate automata which work on an expanded alphabet, which encodes not only the input dataset \mathcal{D} but also a candidate subset \mathcal{D}' . Formally, we consider the alphabet:

$$\begin{split} \overline{\Sigma^2} = & \{ (\zeta^1, \zeta^2, \tau_\Pi, \tau_\Pi^Q) \mid \zeta^1, \zeta^2 \in \Sigma, \\ & \tau_\Pi \subseteq \mathsf{paths}(\Pi), \tau_\Pi^Q \subseteq \mathsf{paths}(\Pi^Q) \} \end{split}$$

where intuitively we use ζ^1 to specify a letter in the encoding of the input dataset \mathcal{D} and ζ^2 for a letter in the encoding of a possible subset \mathcal{D}' . As we will later need words encoding more than two datasets, we can more generally define

$$\begin{split} \overline{\Sigma^k} = & \{ (\zeta^1, \zeta^2, \dots, \zeta^k, \tau_\Pi, \tau_\Pi^Q) \mid \zeta^1, \dots, \zeta^k \in \Sigma, \\ & \tau_\Pi \subseteq \mathsf{paths}(\Pi), \tau_\Pi^Q \subseteq \mathsf{paths}(\Pi^Q) \} \end{split}$$

Given a word W over $\overline{\Sigma^k}$ and $1 \leq \ell \leq k$, we will use $\operatorname{proj}(W,\ell)$ to denote the ℓ th projection of W, which is the word over alphabet Σ obtained by replacing each letter in W by its ℓ th component, i.e. replacing $(\zeta^1,\zeta^2,\ldots,\zeta^k,\tau_\Pi,\tau_\Pi^Q)$ by ζ^ℓ . We will use $\operatorname{proj}^+(W,\ell)$ for the word over $\overline{\Sigma}$ obtained by keeping the ℓ th component followed by the elements of $\operatorname{paths}(\Pi)$ and $\operatorname{paths}(\Pi^Q)$, i.e. replacing $(\zeta^1,\zeta^2,\ldots,\zeta^k,\tau_\Pi,\tau_\Pi^Q)$ by $(\zeta^\ell,\tau_\Pi,\tau_\Pi^Q)$.

The following definition specifies when a word over $\overline{\Sigma^k}$ is well formed, in the sense that the symbols in $\Sigma_{-\infty}$, Σ_{∞} , Σ_{open} , Σ_{close} are used in a coherent manner.

Definition 14. We say that a word

$$w = w_{-1}w_0 \dots w_M w_{M+1} w_{M+2}$$

over alphabet Σ is proper if $M \geq 0$ and the following conditions hold:

- $w_{-1} \subseteq \Sigma_{-\infty}$
- $w_{M+2} \subseteq \Sigma_{\infty}$

- $w_j \subseteq \Sigma_{\mathsf{prop}}^+ \text{ for } 0 \le j \le M$
- $w_{M+1} \subseteq \Sigma_{\mathsf{prop}}$
- for every $P_{-\infty} \in w_{-1}$, either:
 - there exists j such that $P_{\text{close}} \in w_j$, j^* is the least such j, and $P \in w_{j'}$ for every $0 \le j' \le j^*$
 - there is no j with $P_{\mathsf{close}} \in w_j$, in which case $P \in w_j$ for every $0 \le j \le M+1$, and further we have $P_{\infty} \in w_{M+2}$
- for every $P_{\infty} \in w_{M+2}$, either:
 - there exists j such that $P_{\mathsf{open}} \in w_j$, j^* is the largest such j, and $P \in w_{j'}$ for every $j^* \leq j' \leq M+1$
 - there is no j with $P_{\mathsf{open}} \in w_j$, in which case $P \in w_j$ for every $0 \le j \le M+1$, and further we have $P_{-\infty} \in w_{-1}$
- if $P_{\text{open}} \in w_j$ and $P_{\text{open}} \in w_{j'}$ with j < j', then there exists j'' with $j \leq j'' < j'$ and $P_{\text{close}} \in w_{j''}$
- if $P_{\text{close}} \in w_j$ and $P_{\text{close}} \in w_{j'}$ with j < j', then there exists j'' with $j < j'' \le j'$ and $P_{\text{open}} \in w_{j''}$
- if $P_{\mathsf{open}} \in w_j$ and $P_{\mathsf{close}} \in w_{j'}$ with $j \leq j'$, and there is no j'' with $j \leq j'' < j'$ such that $P_{\mathsf{close}} \in w_{j''}$, then $P \in w_{j^*}$ for every $j \leq j^* \leq j'$
- if $P_{\mathsf{open}} \in w_j$ and there is no $j \leq j'$ with $P_{\mathsf{close}} \in w_{j'}$, then $P \in w_{j'}$ for every $j \leq j' \leq M+1$ and $P_{\infty} \in w_{M+2}$
- if $P_{\mathsf{close}} \in w_j$ and there is no $j' \leq j$ with $P_{\mathsf{open}} \in w_{j'}$, then $P \in w_{j'}$ for every $0 \leq j' \leq j$ and $P_{-\infty} \in w_{-1}$
- if $P \notin w_j$ and $P \in w_{j+1}$ and $j \ge 0$, then $P_{\mathsf{open}} \in w_{j+1}$
- if $P \in w_j$ and $P \notin w_{j+1}$ and $j \leq M$, then $P_{\mathsf{close}} \in w_j$

We call a word W over $\overline{\Sigma^k}$ proper if $\operatorname{proj}(W,\ell)$ is proper for every $1 \leq \ell \leq k$, and furthermore, $Z \in w_j^\ell$ iff $Z \in w_j^{\ell'}$ for all $Z \in \{B, E\}$, $1 \leq \ell, \ell' \leq k$, and $0 \leq j \leq M$.

Remark 2. It can be easily verified, by examining Definition 13 and recalling that \mathcal{D} is in normal form, that the word $\overline{w}_{\Pi,\mathcal{D},\mathcal{Q}}$ is proper.

Next we show how to extract datasets from the projections of proper $\overline{\Sigma^k}$ words. As such words are succinct representations, where a single symbol defines the propositions that hold over a whole interval, such an extraction is not deterministic and depends on which concrete endpoints are used. To this end, we define when a set of timepoints and corresponding intervals is compatible with the constraints that the $\overline{\Sigma^k}$ -word imposes via the elements in paths(Π) and paths(Π^Q), and then show how to perform the extraction w.r.t. a compatible set of endpoint timepoints.

Definition 15. Consider a set I of integer timepoints $0 = t_1^* < \ldots < t_{N^*}^*$ and let $\Theta_I = \varrho_0^*, \ldots, \varrho_{m^*+1}^*$ be the subsequence of all non-empty intervals in the sequence

$$\{t_1^*\}, (t_1^*, t_2^*), \{t_2^*\}, \dots, (t_{N^*-1}^*, t_{N^*}^*), \{t_{N^*}^*\}, \{t_{N^*}^*+1\}$$

We say that I is compatible with a word

$$W = W_{-1}W_0 \dots W_{m^*}W_{m^*+1}W_{m^*+2} \in (\overline{\Sigma^k})^*$$

with $W_j=(\zeta_j^1,\zeta_j^2,\ldots,\zeta_j^k,\tau_j,\tau_j^Q)$ if the following conditions hold:

- if $\zeta_i^i \cap (\Sigma_{\mathsf{open}} \cup \Sigma_{\mathsf{close}}) \neq \emptyset$, then $\varrho_i^* = \{t\}$ for some $t \in I$
- if $(a,b) \in \tau_j \cup \tau_j^Q$, then $|\rho_j| = a + n \cdot b$ for some $n \in \mathbb{N}$

Remark 3. By comparing Definitions 8, 13, and 15, we can see that the set of integer endpoints $I_{\mathcal{D}}$ given by $0 = t_1 < \ldots < t_N = d$ is compatible with $\overline{w}_{\Pi,\mathcal{D},Q}$. Indeed, as previously noted in Remark 1, symbols from $\Sigma_{\mathsf{open}} \cup \Sigma_{\mathsf{close}}$ can only occur at timepoints in $I_{\mathcal{D}}$.

Now we formalize how to extract datasets from words over Σ^k and compatible sets of integer timepoints.

Definition 16. Consider a proper word $W \in (\overline{\Sigma^k})^*$. Let I be a set of integers $0 = t_1^* < \ldots < t_{N^*}^*$ that is compatible with W, and let $\Theta_I = \varrho_1^*, \ldots, \varrho_{m^*+1}^*$ be as in Definition 15. Further let $\operatorname{proj}(W, \ell) = w_{-1}w_0 \ldots w_{m^*}w_{m^*+1}w_{m^*+2}$. We define the dataset $\mathcal{D}_{W,I}^\ell$ induced by W, I, and level ℓ as the dataset which contains all and only the following facts:

- $P@(-\infty,\infty)$, if $P_{-\infty} \in w_{-1}$, $P_{\infty} \in w_{m^*+2}$, and there is no j with $P_{\text{close}} \in w_j$
- $P@(-\infty, t^{\dagger}]$, if $P_{-\infty} \in w_{-1}$, $\varrho_j^* = \{t^{\dagger}\}$, $P_{\mathsf{close}} \in w_j$, and $P_{\mathsf{close}} \notin w_{j'}$ for all j' < j
- $P@[t^{\dagger}, t^{\ddagger}]$, if $j \leq j'$, $\varrho_j^* = \{t^{\dagger}\}$, $\varrho_{j'}^* = \{t^{\ddagger}\}$, $P_{\mathsf{open}} \in w_j$, $P_{\mathsf{close}} \in w_{j'}$, and $P_{\mathsf{close}} \not\in w_{j''}$ for all $j \leq j'' < j'$
- $P@[t^{\dagger}, \infty)$, if $P_{\infty} \in w_{m^*+2}$, $\varrho_j^* = \{t^{\dagger}\}$, $P_{\mathsf{open}} \in w_j$, and $P_{\mathsf{open}} \notin w_{j'}$ for all j' > j

where $P \in \mathsf{prop}(\Pi)$.

Remark 4. If we consider the set of integer endpoints $I_{\mathcal{D}}$ given by $0=t_1<\ldots< t_N=d$ and take $W=\overline{w}_{\Pi,\mathcal{D},Q}$, then $\mathcal{D}^1_{W,I_{\mathcal{D}}}=\mathcal{D}$, i.e. we have shown how to 'decode' $\overline{w}_{\Pi,\mathcal{D},Q}$, using $I_{\mathcal{D}}$, to obtain the original dataset \mathcal{D} . Note that the decoding does not consider the special propositions B,E, which will be handled separately.

Our next step will be to give conditions on proper $\overline{\Sigma^k}$ -words W which can be used to test whether the dataset $\mathcal{D}_{W,I}^\ell$ is a subset of $\mathcal{D}_{W,I}^{\ell'}$, for $1 \leq \ell, \ell' \leq k$ and compatible I.

Definition 17. Consider a proper word $W \in (\overline{\Sigma^k})^*$ and $\ell, \ell' \in \{1, \dots, k\}$. Further let $\operatorname{proj}(W, \ell) = w_{-1}w_0 \dots w_{m^*}w_{m^*+1}w_{m^*+2}$ and $\operatorname{proj}(W, \ell') = w'_{-1}w'_0 \dots w'_{m^*}w'_{m^*+1}w'_{m^*+2}$. We say that level ℓ' of W is syntactically included in level ℓ of W, written $(W, \ell') \subseteq^{\operatorname{syn}}(W, \ell)$, if the following conditions hold:

- (i) if $P_{-\infty} \in w'_{-1}$, then $P_{-\infty} \in w_{-1}$
- (ii) if $P_{-\infty} \in w'_{-1}$ and there exists j such that $P_{\mathsf{close}} \in w_j$, with j^* the least such j, then $P_{\mathsf{close}} \in w'_{j^*}$
- (iii) if $P_{\infty} \in w'_{m^*+2}$, then $P_{\infty} \in w_{m^*+2}$
- (iv) if $P_{\infty} \in w'_{m^*+2}$ and there exists j such that $P_{\text{open}} \in w_j$, with j^* the greatest such j, then $P_{\text{open}} \in w'_{j^*}$
- (v) if $P_{\text{open}} \in w'_j$, then $P_{\text{open}} \in w_j$
- (vi) if $P_{\text{open}} \in w'_j$ and there exists $j' \geq j$ such that $P_{\text{close}} \in w_{j'}$, with j^* the least such j', then $P_{\text{close}} \in w'_{j^*}$
- (vii) if $P_{\mathsf{close}} \in w'_i$, then $P_{\mathsf{close}} \in w_i$

(viii) if $P_{\text{close}} \in w'_j$ and there exists $j' \leq j$ such that $P_{\text{open}} \in w_{j'}$, with j^* the greatest such j', then $P_{\text{open}} \in w'_{j^*}$

Lemma 9. Consider a proper word $W \in (\overline{\Sigma^k})^*$, $\ell, \ell' \in \{1, \ldots, k\}$, and a set of integers I compatible with W. Then $(W, \ell') \subseteq^{\text{syn}} (W, \ell)$ iff $\mathcal{D}_{W, I}^{\ell'} \subseteq \mathcal{D}_{W, I}^{\ell}$.

Proof. Let I be the considered set of integers that is compatible with W and whose associated sequence of intervals is $\Theta_I = \varrho_1^*, \dots, \varrho_{m^*+1}^*$. Further let $\operatorname{proj}(W,\ell) = w_{-1}w_0 \dots w_{m^*}w_{m^*+1}w_{m^*+2}$ and $\operatorname{proj}(W,\ell') = w'_{-1}w'_0 \dots w'_{m^*}w'_{m^*+1}w'_{m^*+2}$.

First suppose that $(W, \ell') \subseteq^{\text{syn}} (W, \ell)$ and let $\varphi \in \mathcal{D}_{W,I}^{\ell'}$. By Definition 16, there are four cases to consider depending on the shape of φ :

Case 1 $\varphi = P@(-\infty, \infty)$

This implies that $P_{-\infty} \in w'_{-1}$, $P_{\infty} \in w'_{m^*+2}$, and there is no j with $P_{\text{close}} \in w'_{j}$. Since $(W,\ell') \subseteq^{\text{syn}} (W,\ell)$, we know that $P_{-\infty} \in w_{-1}$ and $P_{\infty} \in w_{m^*+2}$. Furthermore, there cannot exist j such that $P_{\text{close}} \in w_{j}$, since this would imply $P_{\text{close}} \in w'_{j}$ (by point (vii) of Definition 17). Hence $\varphi = P@(-\infty,\infty) \in \mathcal{D}^{\ell}_{W,I}$.

Case 2 $\varphi = P@(-\infty, t^{\dagger}]$

In this case, $P_{-\infty} \in w'_{-1}$, $P_{\text{close}} \in w'_{j}$, and $P_{\text{close}} \notin w'_{j'}$ for all j' < j, and $\varrho_j^* = \{t^{\dagger}\}$. Since $(W, \ell') \subseteq^{\text{syn}} (W, \ell)$, it follows that $P_{-\infty} \in w_{-1}$, $P_{\text{close}} \in w_j$, and there is no j' < j with $P_{\text{close}} \in w_{j'}$ (by point (ii) of Definition 17). Hence $\varphi = P@(-\infty, t^{\dagger}] \in \mathcal{D}_{W,I}^{\ell}$.

Case 3 $\varphi = P@[t^{\dagger}, t^{\ddagger}]$

In this case, there exists some $j \leq j'$ such that $\varrho_j^* = \{t^\dagger\}$, $\varrho_{j'}^* = \{t^\dagger\}$, $P_{\text{open}} \in w_j'$, $P_{\text{close}} \in w_{j'}'$, and $P_{\text{close}} \not\in w_{j''}'$ for all $j \leq j'' < j'$. Since $(W, \ell') \subseteq^{\text{syn}} (W, \ell)$, we must have $P_{\text{open}} \in w_j$ and $P_{\text{close}} \in w_{j'}$. Moreover, since $P_{\text{close}} \not\in w_{j''}'$ for all $j \leq j'' < j'$, there cannot be any $j \leq j'' < j'$ with $P_{\text{close}} \in w_{j''}$ (by point (vi) of Definition 17). From all this, we obtain $\varphi = P@[t^\dagger, t^\dagger] \in \mathcal{D}_{W,I}^\ell$.

Case 4 $\varphi = P@[t^{\dagger}, \infty)$

In this case, $P_{\infty} \in w'_{m^*+2}$ and there exists j with $\varrho_j^* = \{t^{\dagger}\}$ and $P_{\text{open}} \in w'_j$ and such that there is no j' > j with $P_{\text{open}} \in w'_{j'}$. From $(W, \ell') \subseteq^{\text{syn}} (W, \ell)$, we get $P_{\infty} \in w_{m^*+2}$, $P_{\text{open}} \in w_j$ and there is no j' > j with $P_{\text{open}} \in w_{j'}$ (by point (iv) of Definition 17), hence $\varphi = P@[t^{\dagger}, \infty) \in \mathcal{D}_{W,I}^{\ell}$.

We can thus conclude that $\mathcal{D}_{W,I}^{\ell'} \subseteq \mathcal{D}_{W,I}^{\ell}$.

Next suppose that we have $\mathcal{D}_{W,I}^{\ell'} \subseteq \mathcal{D}_{W,I}^{\ell}$. To show that $(W,\ell') \subseteq^{\mathsf{syn}} (W,\ell)$, we must show the eight items in Definition 17. We give the proof for a few items, the other items can be proved analogously:

Item (i) Suppose $P_{-\infty} \in w'_{-1}$. As W is proper, either $P_{\infty} \in w'_{m^*+2}$ or there is some j such that $P_{\mathsf{close}} \in w'_{j}$. Thus, there must exist some fact $P@(-\infty, t^\dagger]$ or $P@(-\infty, \infty)$ in $\mathcal{D}^{\ell'}_{W,I}$, and hence also in $\mathcal{D}^{\ell}_{W,I}$. It follows that $P_{-\infty} \in w_{-1}$.

Item (ii) Suppose that $P_{-\infty} \in w'_{-1}$ and there exists j such that $P_{\text{close}} \in w_j$, with j^* the least such j. As $P_{\text{close}} \in w_{j^*}$ and

I is compatible, we have $\varrho_{j^*}^*=\{t^\dagger\}$ for some $t^\dagger\in I$. Due to item (i), we also have $P_{-\infty}\in w_{-1}$, hence $P@(-\infty,t^\dagger]$ in $\mathcal{D}_{W,I}^\ell$. Moreover, from the definition of $\mathcal{D}_{W,I}^\ell$, this is the unique P-fact with endpoint $-\infty$. From $P_{-\infty}\in w_{-1}'$ and properness, we know that $\mathcal{D}_{W,I}^{\ell'}$ must also contain a single P-fact with endpoint $-\infty$. Due to $\mathcal{D}_{W,I}^{\ell'}\subseteq \mathcal{D}_{W,I}^\ell$, this must be the same P-fact, i.e. $P@(-\infty,t^\dagger]\in \mathcal{D}_{W,I}^{\ell'}$. This implies in turn that $P_{\text{close}}\in w_{j^*}'$, as required.

Item (vii) Suppose $P_{\text{close}} \in w'_j$. As I is compatible, we have $\varrho_j^* = \{t^{\ddagger}\}$ for some $t^{\ddagger} \in I$. Due to properness, we know that either $P_{-\infty} \in w'_{-1}$ or there is some $j' \leq j$ such that $P_{\text{open}} \in w'_{j'}$, with j^* be the largest such j'. In the former case, the fact $P@(-\infty, t^{\ddagger}]$ belongs to $\mathcal{D}_{W,I}^{\ell'}$, hence to $\mathcal{D}_{W,I}^{\ell}$, which implies $P_{\text{close}} \in w_j$. In the latter case, there is some fact $P@[t^{\dagger}, t^{\ddagger}]$ in $\mathcal{D}_{W,I}^{\ell'}$, hence in $\mathcal{D}_{W,I}^{\ell}$, which also implies $P_{\text{close}} \in w_j$.

Item (viii) Suppose that $P_{\text{close}} \in w'_j$ and there exists $j' \leq j$ such that $P_{\text{open}} \in w_{j'}$, with j^* the greatest such j'. By item (vii), we have $P_{\text{close}} \in w_j$. Due to compatibility of I, there exist $t^\dagger, t^\ddagger \in I$ such that $\varrho_j^* = \{t^\ddagger\}$ and $\varrho_{j'}^* = \{t^\dagger\}$. It follows that $P@[t^\dagger, t^\ddagger]$ belongs to $\mathcal{D}_{W,I}^\ell$. Moreover, due to properness, either there is $j'' \leq j$ such that $P_{\text{close}} \in w'_{j''}$ or $P_{-\infty} \in w'_{-1}$. In either case, we know that $\mathcal{D}_{W,I}^{\ell'}$ must contain a P-fact whose second endpoint is t^\ddagger . Since $\mathcal{D}_{W,I}^{\ell'} \subseteq \mathcal{D}_{W,I}^\ell$, this fact must be $P@[t^\dagger, t^\ddagger]$. It follows then from the definition of $\mathcal{D}_{W,I}^{\ell'}$ that $P_{\text{open}} \in w'_{j^*}$.

To construct our NFA $\mathfrak{A}_{\mathsf{brave}}$ that decides query entailment under s-brave semantics, we will need the following intermediate NFAs over the alphabet $\overline{\Sigma^2}$:

- $\mathfrak{A}_{\mathsf{proper}}$ tests whether the given input word W is proper and can be obtained by intersecting many simple NFAs that test for the various conditions of properness.
- $\mathfrak{A}^{1,2}_{\subseteq}$ tests whether the input word W is proper and such that $(W,2)\subseteq^{\mathsf{syn}}(W,1)$, which can be obtained by intersecting $\mathfrak{A}_{\mathsf{proper}}$ with NFAs that check for each of the conditions in items (i)-(viii).
- $\overline{\mathfrak{A}}_{\Pi}^2$ is used to check Π -consistency of the level-2 dataset encoded in W. It can be defined like $\overline{\mathfrak{A}}_{\Pi}$ except that it runs on $\overline{\Sigma^2}$ -words and ignores the component ζ^1 in the input letters $(\zeta^1, \zeta^2, \tau_\Pi, \tau_\Pi^Q)$, i.e. it simulates running $\overline{\mathfrak{A}}_{\Pi}$ over $\operatorname{proj}^+(W, 2)$.
- $\overline{\mathfrak{A}}_{\Pi}^{\overline{Q},2}$ is used to check Π^Q -inconsistency of the level-2 dataset encoded in W. It is obtained by adapting $\overline{\mathfrak{A}}_{\Pi}^Q$ so that it runs on $\overline{\Sigma^2}$ -words and ignores the ζ^1 component, and then taking the complement.

We then let $\mathfrak{A}^2_{\text{brave}}$ be the NFA obtained by intersecting $\mathfrak{A}^{1,2}_{\subseteq}$, $\overline{\mathfrak{A}}^2_{\Pi}$, and $\overline{\mathfrak{A}}^{\bar{Q},2}_{\Pi}$. Finally, we define $\mathfrak{A}_{\text{brave}}$ as the NFA over alphabet $\overline{\Sigma}$ that accepts precisely those $\overline{\Sigma}$ -words that are equal

to $\operatorname{proj}^+(W,1)$ for some $W\in \mathfrak{A}^2_{\operatorname{brave}}$ (i.e. we project away the ζ^2 component). The following lemma shows that the NFA thus constructed can be used to decide s-brave semantics.

Lemma 10. The NFA $\mathfrak{A}_{\mathsf{brave}}$ accepts $\overline{w}_{\Pi,\mathcal{D},Q}$ iff $(\mathcal{D},\Pi) \models^s_{\mathit{brave}} Q@[t_B,t_E].$

Proof. First suppose that $\mathfrak{A}_{\mathsf{brave}}$ accepts $\overline{w}_{\Pi,\mathcal{D},Q}$. Then we know that there exists a $\overline{\Sigma^2}$ -word W that is accepted by $\mathfrak{A}^2_{\mathsf{brave}}$ and such that $\mathsf{proj}^+(W,1) = \overline{w}_{\Pi,\mathcal{D},Q}$. We know from Remark 4 that $\mathcal{D}^1_{W,I_{\mathcal{D}}} = \mathcal{D}$, where $I_{\mathcal{D}}$ is the set of integer endpoints given by $0 = t_1 < \ldots < t_N = d$. Let $\mathcal{D}' = \mathcal{D}^2_{W,I_{\mathcal{D}}}$. As $\mathfrak{A}^{1,2}_{\subseteq}$ accepts W, and $I_{\mathcal{D}}$ is compatible with $\overline{w}_{\Pi,\mathcal{D},Q}$ (Remark 3), we have $(W,2)\subseteq^{\mathsf{syn}}(W,1)$. By Lemma $9,\mathcal{D}'\subseteq\mathcal{D}$. Moreover, the acceptance of W by $\overline{\mathfrak{A}}^2_{\Pi}$, and $\overline{\mathfrak{A}}^{\bar{Q},2}_{\Pi}$ ensures that \mathcal{D}' is Π -consistent and that $(\mathcal{D}')^Q$ is Π^Q -inconsistent (recall that properness ensures that $\mathsf{proj}^+(W,1)$ and $\mathsf{proj}^+(W,2)$ have B and E at the same positions). We conclude by Lemma 8 that $(\mathcal{D},\Pi)\models^s_{\mathsf{brave}}Q@[t_B,t_E]$.

Lemma 8 that $(\mathcal{D},\Pi) \models_{\text{brave}}^s Q@[t_B,t_E]$. Now suppose that $(\mathcal{D},\Pi) \models_{\text{brave}}^s Q@[t_B,t_E]$. By Lemma 8, there exists a subset $\mathcal{D}' \subseteq \mathcal{D}$ such that:

- \mathcal{D}' is Π -consistent, and
- $\mathcal{D}' \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}\$ is Π^Q -inconsistent

We let $w^2_{\Pi,\mathcal{D},Q}$ be obtained by applying Definition 8 to \mathcal{D}' , but using d even if the largest integer in \mathcal{D}' is smaller (so that the two words have the same length). Then let $\sigma_0^{**},\ldots,\sigma_m^{**}$ be defined like $\sigma_0^*,\ldots,\sigma_m^*$ in Definition 13, using the same set of integers and sequence of intervals $\varrho_0,\ldots,\varrho_{m+1}$, but using the word $w^2_{\Pi,\mathcal{D},Q}$ in place of $w_{\Pi,\mathcal{D},Q}$. We define W^* as

$$(\sigma_{-1}, \sigma'_{-1}, \emptyset, \emptyset)(\sigma_0^*, \sigma_0^{**}, \tau_0, \tau'_0) \dots (\sigma_m^*, \sigma_m^{**}, \tau_m, \tau'_m)(\sigma_{d+2}, \sigma'_{d+2}, \emptyset, \emptyset)$$

By construction, W^* is proper and, since $\mathcal{D}'\subseteq \mathcal{D}$, such that $(W^*,2)\subseteq^{\operatorname{syn}}(W^*,1)$, so W^* is accepted by $\mathfrak{A}^{1,2}_\subseteq$. Moreover, since \mathcal{D}' is Π -consistent and $\mathcal{D}'\cup \{B@\{t_B-1\}, E@\{t_E+1\}\}$ is Π^Q -inconsistent, W^* is also accepted by the NFAs $\overline{\mathfrak{A}}^0_\Pi$, and $\overline{\mathfrak{A}}^{\bar{Q},2}_\Pi$. It follows that $\mathfrak{A}^2_{\operatorname{brave}}$ accepts W^* , and since $\operatorname{proj}^+(W^*,1)=w_{\Pi,\mathcal{D},Q}$, the NFA $\mathfrak{A}_{\operatorname{brave}}$ will accept $\overline{w}_{\Pi,\mathcal{D},Q}$.

Construction for s**-CQA semantics**

We will now describe how to obtain an NFA \mathfrak{A}_{CQA} to decide s-CQA semantics, building upon the notions and components introduced for s-brave semantics. Unlike the s-brave semantics, for which it is sufficient to find a Π -consistent subset that entails the query, for the s-CQA semantics, we will need to be able to detect s-repairs.

We thus aim to define an NFA \mathfrak{A}_{rep} over alphabet $\overline{\Sigma^2}$ that will recognize (encodings of) repairs of the level-1 dataset. For this, we will first define an NFA \mathfrak{A}_{better} over the alphabet $\overline{\Sigma^3}$ that is the intersection of the following NFAs:

• $\mathfrak{A}^3_{\text{proper}}$, which tests properness of words over $\overline{\Sigma^3}$

- $\mathfrak{A}^{2,3}_{\subseteq}$ and $\mathfrak{A}^{3,1}_{\subseteq}$, which check respectively whether the input word W is proper and such that $(W,2)\subseteq^{\mathsf{syn}}(W,3)$ (respectively, proper and $(W,3)\subseteq^{\mathsf{syn}}(W,1)$), defined analogously to the previously mentioned $\mathfrak{A}^{1,2}_{\subseteq}$
- the intersection of $\mathfrak{A}^3_{\mathsf{proper}}$ and of the complement of $\mathfrak{A}^{3,2}_{\subseteq}$, which accepts proper W such that $(W,3) \not\subseteq^{\mathsf{syn}} (W,2)$
- $\overline{\mathfrak{A}}_{\Pi}^3$ (defined analogously to $\overline{\mathfrak{A}}_{\Pi}^2$) is used to check Π -consistency of the encoded level-3 dataset

We then let $\mathfrak{A}_{\mathsf{nobetter}}$ be the NFA over $\overline{\Sigma^2}$ that accepts those W such that there does not exist any W' over $\overline{\Sigma^3}$ such that:

- $proj^+(W,1) = proj^+(W',1)$
- $proj^+(W, 2) = proj^+(W', 2)$
- W' is accepted by $\mathfrak{A}_{\mathsf{better}}$

We can obtain $\mathfrak{A}_{\mathsf{nobetter}}$ by first projecting $\mathfrak{A}_{\mathsf{better}}$ onto $\overline{\Sigma^2}$ (dropping ζ^3 component), then complementing it.

We then define the NFA $\mathfrak{A}_{\mathsf{rep}}$ as the intersection of $\mathfrak{A}_{\mathsf{proper}}$, $\mathfrak{A}_{\subseteq}^{1,2}$, $\overline{\mathfrak{A}}_{\Pi}^{2}$, and $\mathfrak{A}_{\mathsf{nobetter}}$. We further define an NFA $\mathfrak{A}_{\mathsf{rep}}^{\bar{Q}}$ as the intersection of $\mathfrak{A}_{\mathsf{rep}}$ and $\overline{\mathfrak{A}}_{\Pi}^{Q,2}$ (testing Π^{Q} -consistency of the level-2 dataset extended with the B, E facts). Finally, we let $\mathfrak{A}_{\mathsf{CQA}}$ be an NFA that accepts precisely those $\overline{\Sigma}$ -words that are *not* equal to $\mathsf{proj}^{+}(W,1)$ for some $\overline{\Sigma}^{2}$ -word W accepted by $\mathfrak{A}_{\mathsf{rep}}^{\bar{Q}}$ (obtained by $\mathsf{projecting}$ then complementing $\mathfrak{A}_{\mathsf{rep}}^{\bar{Q}}$).

Lemma 11. The NFA \mathfrak{A}_{CQA} accepts $\overline{w}_{\Pi,\mathcal{D},Q}$ iff $(\mathcal{D},\Pi) \models^s_{CQA} Q@[t_B,t_E]$.

Proof. First let us suppose that $(\mathcal{D}, \Pi) \not\models^s_{\text{CQA}} Q@[t_B, t_E]$. It follows that there is an s-repair $\mathcal{D}' \subseteq \mathcal{D}$ such that $(\mathcal{D}', \Pi) \not\models Q@[t_B, t_E]$, which means:

- \mathcal{D}' is Π -consistent,
- there is no \mathcal{D}'' with $\mathcal{D}' \subseteq \mathcal{D}'' \subseteq \mathcal{D}$ that is Π -consistent,
- $\mathcal{D}' \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}\$ is Π^Q -consistent

Let W be the unique $\overline{\Sigma^2}$ -word such that:

- $\operatorname{proj}^+(W,1) = \overline{w}_{\Pi,\mathcal{D},Q}$
- $\operatorname{proj}^+(W,2)$ is the word $\overline{w}_{\Pi,\mathcal{D}',Q,d}$, defined like $\overline{w}_{\Pi,\mathcal{D},Q}$ but using dataset \mathcal{D}' and final integer d (even if \mathcal{D}' has a smaller greatest integer endpoint)

By construction, W is proper. Moreover, if we let $I_{\mathcal{D}}$ be the set of integer endpoints given by $0=t_1<\ldots< t_N=d$, then $\mathcal{D}=\mathcal{D}^1_{W,I_{\mathcal{D}}}$ and $\mathcal{D}'=\mathcal{D}^2_{W,I_{\mathcal{D}}}$. Since $\mathcal{D}'\subseteq\mathcal{D}$, it follows from Lemma 9 that $(W,2)\subseteq^{\mathrm{syn}}(W,1)$. Thus, W is accepted by $\mathfrak{A}^{1,2}_\subseteq$. Moreover, since $\mathcal{D}'\cup\{B@\{t_B-1\},E@\{t_E+1\}\}$ is Π^Q -consistent, W is also accepted by $\overline{\mathfrak{A}}^{1}_\Pi$. We need to show W is also accepted by $\mathfrak{A}_{\mathrm{nobetter}}$, which is equivalent to there does not exist any $\overline{\Sigma^3}$ -word W' accepted by $\mathfrak{A}_{\mathrm{better}}$ and such that $\mathrm{proj}^+(W,1)=\mathrm{proj}^+(W',1)$ and $\mathrm{proj}^+(W,2)=\mathrm{proj}^+(W',2)$. For this, we note that if such a W' were to exist, it would imply existence of a subset $\mathcal{D}'\subseteq\mathcal{D}''\subseteq\mathcal{D}$ that is Π -consistent, which we know does not exist. As W is accepted by $\mathfrak{A}^{1,2}_\subseteq$, $\overline{\mathfrak{A}}^2_\Pi$, and $\mathfrak{A}_{\mathrm{nobetter}}$, it is accepted by $\mathfrak{A}_{\mathrm{rep}}$.

Moreover, since $\mathcal{D}' \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}$ is Π^Q -consistent, W is also accepted by $\overline{\mathfrak{A}}_{\Pi}^{Q,2}$, hence by $\mathfrak{A}_{\text{rep}}^{Q}$. Since $\operatorname{proj}^+(W,1)=\overline{w}_{\Pi,\mathcal{D},Q}$, we can conclude that $\overline{w}_{\Pi,\mathcal{D},Q}$ is not accepted by $\mathfrak{A}_{\text{CQA}}$.

For the other direction, suppose for a contradiction that $(\mathcal{D},\Pi)\models_{\mathsf{CQA}}^sQ@[t_B,t_E]$ but $\mathfrak{A}_{\mathsf{CQA}}$ does not accept $\overline{w}_{\Pi,\mathcal{D},Q}$. This means that there exists a $\overline{\Sigma^2}$ -word W that is accepted by $\mathfrak{A}^{\bar{Q}}_{\mathsf{rep}}$ and such that $\mathsf{proj}^+(W,1)=\overline{w}_{\Pi,\mathcal{D},Q}$. From the definition of $\mathfrak{A}^{\bar{Q}}_{\mathsf{rep}}$, we have that W is accepted by $\mathfrak{A}_{\mathsf{rep}}$ and $\overline{\mathfrak{A}}^{Q,2}_{\Pi}$. We let $I_{\mathcal{D}}$ be the set of integer endpoints given by $0=t_1<\ldots< t_N=d$, recall that $\mathcal{D}=\mathcal{D}^1_{W,I_{\mathcal{D}}}$ and define \mathcal{D}' as $\mathcal{D}^2_{W,I_{\mathcal{D}}}$. Acceptance by $\overline{\mathfrak{A}}^{Q,2}_{\Pi}$ implies that $\mathcal{D}'\cup\{B@\{t_B-1\},E@\{t_E+1\}\}$ is Π^Q -consistent, hence $(\mathcal{D}',\Pi)\not\models Q@[t_B,t_E]$. Acceptance by $\mathfrak{A}^{-2}_{\mathsf{rep}}$ means acceptance by $\mathfrak{A}^{-2}_{\mathsf{c}}$, $\overline{\mathfrak{A}}^2_{\mathsf{H}}$, and $\mathfrak{A}_{\mathsf{nobetter}}$, which yields:

- $(W, 2) \subseteq^{\text{syn}} (W, 1)$, hence $\mathcal{D}' \subseteq \mathcal{D}$ (Lemma 9)
- \mathcal{D}' is Π -consistent, due to $\overline{\mathfrak{A}}_{\Pi}^2$
- \mathcal{D}' is an *s*-repair of \mathcal{D} , due to $\mathfrak{A}_{\mathsf{nobetter}}$ (and its components), which ensure there does not exist any dataset \mathcal{D}'' with $\mathcal{D}' \subsetneq \mathcal{D}'' \subseteq \mathcal{D}$ that is Π -consistent.

It follows that \mathcal{D}' is an s-repair with $(\mathcal{D}',\Pi) \not\models Q@[t_B,t_E]$, so we may conclude that $(\mathcal{D},\Pi) \not\models^s_{COA} Q@[t_B,t_E]$.

Construction for s**-intersection semantics**

We now consider the case of *s*-intersection semantics, building upon the NFAs constructed for the *s*-brave and *s*-CQA semantics. We start with a lemma that resumes what we need to check:

Lemma 12. $(\mathcal{D}, \Pi) \not\models_{\cap}^{s} Q@[t_B, t_E]$ iff there exists a subset $\mathcal{D}' \subseteq \mathcal{D}$ such that:

- $\mathcal{D}' \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}\$ is Π^Q -consistent
- for every $\varphi \in \mathcal{D} \setminus \mathcal{D}'$, there exists some s-repair \mathcal{R} of \mathcal{D} such that $\varphi \notin \mathcal{R}$

As before, we can consider words W over $\overline{\Sigma^2}$ which are accepted by $\mathfrak{A}^{1,2}_{\subset}$, which ensures that the level-2 dataset is a subset of the level-1 dataset. We can check the level-2 dataset satisfies the first item of Lemma 12 using $\overline{\mathfrak{A}}_{\Pi}^{Q,2}$. To ensure that the level-2 dataset satisfies the second item, we will proceed as follows. First, we will construct an NFA $\mathfrak{A}_{missing}$ that tests for the complementary property, i.e. whether there exists some fact φ , present in the level-1 dataset but absent from the level-2 dataset, that is included in every s-repair. For this, we will consider an NFA \mathfrak{A}_{pick1} which accepts proper words W over $\overline{\Sigma}^3$ such that $(W,3) \subseteq^{\text{syn}} (W,1)$, $(W,3) \not\subseteq^{\text{syn}} (W,2)$, and proj(W, 3) encodes a single fact. Intuitively, we use the level-3 dataset to pick one fact from the level-1 dataset that is absent in the level-2 dataset. We can further consider an NFA $\mathfrak{A}_{\mathsf{absentrep}}$ over $\overline{\Sigma^4}$ that accepts proper words W such that $(W,3) \not\subseteq^{\mathsf{syn}} (W,4)$, that are accepted by $\mathfrak{A}^{1,4}_{\mathsf{rep}}$ (defined like \mathfrak{A}_{rep} but testing whether the level-4 dataset is an s-repair of the level-1 dataset), which can be obtained as a straightforward combination and modification of earlier NFAs. We

then let $\mathfrak{A}_{missing}$ be the NFA over $\overline{\Sigma^2}$ that accepts those proper words W such that there exists a proper word W' over $\overline{\Sigma}^3$ such that:

- $proj^+(W,1) = proj^+(W',1)$
- $proj^+(W, 2) = proj^+(W', 2)$
- W' is accepted by \mathfrak{A}_{pick1}
- there is *no* proper word W'' over $\overline{\Sigma^4}$ such that:
 - $\text{proj}^+(W', 1) = \text{proj}^+(W'', 1)$
 - $\text{proj}^+(W', 2) = \text{proj}^+(W'', 2)$
 - $\text{proj}^+(W',3) = \text{proj}^+(W'',3)$
 - W'' is accepted by $\mathfrak{A}_{absentrep}$

Such an NFA can be constructed using intersection, projection, and complementation of simpler automata. We then let $\mathfrak{A}_{\mathsf{missing}}^c$ be the complement of $\mathfrak{A}_{\mathsf{missing}}$. The final NFA \mathfrak{A}_{\cap} can thus be defined <u>as</u> the *complement* of the automaton that accepts words over $\overline{\Sigma}$ that are equal to $\operatorname{proj}^+(W,1)$ for some $\overline{\Sigma^2}$ -word W that is accepted by \mathfrak{A}_{proper} , $\mathfrak{A}^{1,2}_{\subset}$, $\overline{\mathfrak{A}}^{Q,2}_{\Pi}$,

The correctness is stated in the following lemma, which combines ideas from the proofs of Lemmas 10 and 11.

Lemma 13. The NFA \mathfrak{A}_{\cap} accepts $\overline{w}_{\Pi,\mathcal{D},\mathcal{Q}}$ iff $(\mathcal{D},\Pi) \models_{\cap}^{s}$ $Q@[t_{B},t_{E}].$

Proof. First suppose that $(\mathcal{D}, \Pi) \not\models_{\cap}^{s} Q@[t_B, t_E]$. We aim to show that \mathfrak{A}_{\cap} does not accept $\overline{w}_{\Pi,\mathcal{D},Q}$, or equivalently, that $\overline{w}_{\Pi,\mathcal{D},Q}$ is accepted by $\mathfrak{A}_{\mathsf{proper}},\mathfrak{A}^{1,2}_{\subseteq},\overline{\mathfrak{A}}^{Q,2}_{\Pi}$, and $\mathfrak{A}^c_{\mathsf{missing}}$. By Lemma 12, since $(\mathcal{D},\Pi)\not\models^s_{\cap}Q@[t_B,t_E]$, we can find

a subset $\mathcal{D}' \subseteq \mathcal{D}$ such that:

- $\mathcal{D}' \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}\$ is Π^Q -consistent
- for every $\varphi \in \mathcal{D} \setminus \mathcal{D}'$, there exists some s-repair \mathcal{R} of \mathcal{D} such that $\varphi \notin \mathcal{R}$

Similarly to the proof of Lemma 11, we can use \mathcal{D}' to construct a proper $\overline{\Sigma^2}$ -word W such that:

- $\operatorname{proj}^+(W,1) = \overline{w}_{\Pi,\mathcal{D},\mathcal{Q}}$
- $\operatorname{proj}^+(W,2)$ is the word $\overline{w}_{\Pi,\mathcal{D}',Q,d}$, defined like $\overline{w}_{\Pi,\mathcal{D},Q}$ but using dataset \mathcal{D}' and final integer d

For the compatible set of integer endpoints $I_{\mathcal{D}}$ given by 0 = $t_1 < \ldots < t_N = d$, we have $\mathcal{D} = \mathcal{D}_{W,I_{\mathcal{D}}}^1$ and $\mathcal{D}' = \mathcal{D}_{W,I_{\mathcal{D}}}^2$. Again arguing similarly to Lemma 11, we obtain that W is accepted by $\mathfrak{A}_{\mathsf{proper}}, \mathfrak{A}^{1,2}_{\subseteq}$, and $\overline{\mathfrak{A}}^{Q,2}_{\Pi}$.

It remains to show that W is accepted by $\mathfrak{A}^{c}_{\mathsf{missing}}$, or equiv-

alently, that there is no word W' over $\overline{\Sigma^3}$ such that:

- $proj^+(W,1) = proj^+(W',1) = proj^+(W'',1)$
- $\operatorname{proj}^+(W, 2) = \operatorname{proj}^+(W', 2) = \operatorname{proj}^+(W'', 2)$
- $proj^+(W',3) = proj^+(W'',3)$
- W' is accepted by \mathfrak{A}_{pick1}
- there is no proper word W'' over $\overline{\Sigma^4}$ such that:
 - $proj^+(W', 1) = proj^+(W'', 1)$
 - $\text{proj}^+(W', 2) = \text{proj}^+(W'', 2)$

- $\text{proj}^+(W',3) = \text{proj}^+(W'',3)$
- W'' is accepted by $\mathfrak{A}_{absentren}$

Suppose for a contradiction that such a word W' over $\overline{\Sigma}^3$ were to exist. Then since W' is accepted by \mathfrak{A}_{pick1} , and W' coincides with W on the first two projections (encoding $\mathcal D$ and $\mathcal D'),$ the dataset $\mathcal D^3_{W,I_{\mathcal D}}$ must contain a single fact $\varphi \in \mathcal{D} \setminus \mathcal{D}'$. From our initial assumption, we know that there exist some s-repair \mathcal{R} of \mathcal{D} such that $\varphi \notin \mathcal{R}$. Let W''be the unique word that coincides with W' on levels 1, 2, and 3 and is such that $\operatorname{proj}^+(W'',4)$ is the word $\overline{w}_{\Pi,\mathcal{R},Q,d}$. Since \mathcal{R} is an s-repair, W'' is accepted by $\mathfrak{A}^{1,4}_{\mathsf{rep}}$. Moreover, as $\varphi \in \mathcal{D}^3_{W,I_{\mathcal{D}}} = \mathcal{D}^3_{W'',I_{\mathcal{D}}}$ but $\varphi \notin \mathcal{R}$, we also have $(W'',3) \not\subseteq^{\text{syn}} (W'',4)$. It follows that W'' is accepted by $\mathfrak{A}_{\mathsf{absentrep}}.$ However, this contradicts our earlier assumption about W'. Thus, we may conclude that no W' satisfying the earlier conditions exists, and hence that W is accepted by $\mathfrak{A}_{\mathsf{missing}}^c$, as desired.

For the other direction, let us suppose that \mathfrak{A}_{\cap} does not accept $\overline{w}_{\Pi,\mathcal{D},\mathcal{Q}}$. We can therefore find a $\overline{\Sigma^2}$ -word W such that $\operatorname{proj}^+(W,1) = \overline{w}_{\Pi,\mathcal{D},Q}$ and W is accepted by $\mathfrak{A}_{\operatorname{proper}}, \mathfrak{A}^{1,2}_{\subset}$, $\overline{\mathfrak{A}}_{\Pi}^{Q,2}$, and $\mathfrak{A}_{\mathsf{missing}}^c$. We let $I_{\mathcal{D}}$ be the set of integer endpoints given by $0=t_1$ $\stackrel{\smile}{<}\dots$ $< t_N=d$, recall that $\mathcal{D}=\mathcal{D}^1_{W,I_{\mathcal{D}}}$ and define \mathcal{D}' as $\mathcal{D}^2_{W,I_{\mathcal{D}}}$. Acceptance by $\mathfrak{A}_{\mathsf{proper}}$, $\mathfrak{A}^{1,2}_{\subset}$, and $\overline{\mathfrak{A}}^{Q,2}_{\Pi}$ implies that $\mathcal{D}' \subseteq \mathcal{D}$ and $\mathcal{D}' \cup \{B@\{t_B-1\}, E@\{t_E+1\}\}\$ is Π^Q -consistent, hence $(\mathcal{D}',\Pi) \not\models Q@[t_B,t_E]$. Thus, the first condition of Lemma 11 is satisfied by \mathcal{D}' .

It remains to show that the second condition is also satisfied by \mathcal{D}' . Suppose for a contradiction that this is not the case. Then there exists $\varphi \in \mathcal{D}' \setminus \mathcal{D}$ such that every s-repair \mathcal{R} of \mathcal{D} contains φ . Let us consider the unique word W' over $\overline{\Sigma^3}$

- $proj^+(W,1) = proj^+(W',1)$
- $proj^+(W, 2) = proj^+(W', 2)$
- $\mathcal{D}_{W,I_{\mathcal{D}}}^3 = \{\varphi\}$

Further let W'' be any proper word over $\overline{\Sigma^4}$ such that:

- $proj^+(W', 1) = proj^+(W'', 1)$
- $proj^+(W', 2) = proj^+(W'', 2)$
- $proj^+(W',3) = proj^+(W'',3)$

Let $\mathcal{D}^4_{W'',I_{\mathcal{D}}}$ be the dataset associated with level 4 of W''. We consider two cases, depending on whether this dataset is a repair. If $\mathcal{D}^4_{W'',I_{\mathcal{D}}}$ is not an s-repair of \mathcal{D} , then W'' is not accepted by $\mathfrak{A}^{1,4}_{\text{rep}}$, hence not by $\mathfrak{A}_{\text{absentrep}}$. If $\mathcal{D}^4_{W'',I_{\mathcal{D}}}$ is an srepair of \mathcal{D} , then by our earlier assumption, $\mathcal{D}^4_{W'',I_{\mathcal{D}}}$ contains φ . It follows then that W'' is not accepted by $(W'',3) \not\subset^{syn}$ (W'',4), hence not accepted by $\mathfrak{A}_{absentrep}$. Thus, there is no W'' matching W' on levels 1, 2, and 3 and which is accepted by $\mathfrak{A}_{absentrep}$. This establishes that W' satisfies the required conditions, and so W will be accepted by $\mathfrak{A}_{missing}$. This yields the desired contradiction, since we know that \hat{W} is accepted by $\mathfrak{A}_{\text{missing}}^c$, the complement of $\mathfrak{A}_{\text{missing}}$.

Complexity of the NFA procedures

We note that all of the considered NFAs are defined independently from the input dataset, so the NFA construction takes constant time w.r.t. data complexity. We only need \mathcal{D} to construct the input word $\overline{w}_{\Pi,\mathcal{D},Q}$. Constructing the sets σ_j^* is straightforward, as we basically just copy the input propositions and use the annotation propositions to indicate interval endpoints. The only non-trivial step is to compute the sets τ_j and τ_j' of pairs in paths(Π) and paths(Π^Q), but it was already shown in [Walega *et al.*, 2020a] that such pairs can be computed in TC₀. As NFA membership testing is complete for NC1 \subseteq PTIME, we obtain the same complexity for query entailment under the three semantics, with the lower bound coming from consistency checking / query entailment under classical semantics.

E Proofs for Section 5.2

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.

Proof. (i) The size of $\mathcal{B} \in iConf(\mathcal{D},\Pi) \cup iRep(\mathcal{D},\Pi)$ is polynomially bounded in the size of \mathcal{D} . This is immediate from the definition of i-conflicts and i-repairs and the assumption of bounded intervals. Indeed, if $\mathcal{C} \in iConf(\mathcal{D},\Pi)$, then $\mathcal{C} \sqsubseteq^i \mathcal{D}$, hence \mathcal{C} contains at most as many facts as \mathcal{D} by point 2 of Lemma 1. Moreover, every fact $\alpha@[t_1,t_2] \in \mathcal{C}$ is such that there exists $\alpha@[t_1',t_2'] \in \mathcal{C}$ with $t_1' \leq t_1 \leq t_2 \leq t_2'$ (cf. proof of point 2 of Lemma 1). It follows that all the interval endpoints t mentioned in \mathcal{C} are such that $t_{\min} \leq t \leq t_{\max}$, with t_{\min} (resp. t_{\max}) the least (maximal) integer mentioned in \mathcal{D} , and so the binary encoding of integer endpoints in \mathcal{C} are no larger than the encodings of the integers in \mathcal{D} . The same argument applies to i-repairs.

(ii) It can be decided in PSPACE whether $\mathcal{B} \in iConf(\mathcal{D},\Pi)$ or $\mathcal{B} \in iRep(\mathcal{D},\Pi)$. We first show how to test whether $\mathcal{C} \in iConf(\mathcal{D},\Pi)$. Testing whether \mathcal{C} is in normal form and such that $\mathcal{C} \sqsubseteq^i \mathcal{D}$ is clearly in PTIME, as it simply requires a straightforward comparison of the facts in \mathcal{C} and \mathcal{D} and their endpoints. We also have to test whether \mathcal{C} is Π -inconsistent and that there is no Π -inconsistent $\mathcal{C}' \sqsubseteq^i \mathcal{C}$. For the latter, it suffices to consider the polynomially many sets \mathcal{C}' obtained by replacing some fact $\alpha@[t_1,t_2] \in \mathcal{C}$ by either $\alpha@[t_1+1,t_2]$ or $\alpha@[t_1,t_2-1]$ (if $t_1 < t_2$) or with nothing (if $t_1 = t_2$). It follows that the minimality check can be carried out by means of polynomially many consistency checks. The overall procedure will run in PSPACE, yielding the desired upper bound.

We can use a similar idea to obtain a procedure for repair checking. Indeed suppose we wish to test whether $\mathcal{R} \in iRep(\mathcal{D},\Pi)$. We first check in PSPACE that \mathcal{R} is in normal form, that $\mathcal{R} \sqsubseteq^i \mathcal{D}$, and that \mathcal{R} is Π -consistent, returning no if any of these conditions is violated. It remains to verify whether \mathcal{R} satisfies the maximality condition. This can be checked by considering each of the polynomially many $\mathcal{R} \sqsubseteq^i \mathcal{R}' \sqsubseteq^i \mathcal{D}$ that can be obtained from \mathcal{R} by replacing a fact $\alpha@[t_1,t_2] \in \mathcal{R}$ that issues from the fact $\alpha@[t_1,t_2'] \in \mathcal{D}$, with $t_1' \leq t_1 \leq t_2 \leq t_2'$, by either the fact $\alpha@[t_1-1,t_2]$ (provided $t_1-1 \geq t_1'$) or $\alpha@[t_1,t_2+1]$ (provided $t_2+1 \leq t_2$), and

verifying that every such \mathcal{R}' is Π -inconsistent. However, this is not by itself sufficient, as we also need to ensure that:

(*) If $\mathcal D$ contains a fact $\beta@[u_1,u_2]$ but $\mathcal R$ contains no fact $\beta@\iota$ with $\iota\subseteq[u_1,u_2]$, then every $\mathcal R'$ that can be obtained from $\mathcal R$ by adding a fact $\beta@\{t\}$, with $u_1\leq t\leq u_2$, is Π -inconsistent.

Iterating over all (exponentially many) such timepoints t can be done in polynomial space, so we obtain a PSPACE procedure.

- (iii) A single i-conflict (resp. i-repair) can be generated in PSPACE. We begin by showing how to construct an i-conflict for a dataset \mathcal{D} that is inconsistent w.r.t. program Π . The idea will be to start from the whole \mathcal{D} and remove facts or restrict intervals to obtain a conflict that is minimal w.r.t. \sqsubseteq^i . More precisely, we consider the following procedure (we use binary search in prevision of the proof of Proposition 12 where we consider tractable fragments and the number of consistency calls need to be more strictly bounded):
- 1. Set $C_0 = \mathcal{D}$, and let n be the number of facts in \mathcal{D} .
- 2. For j = 1 until n:
 - Let $\alpha@[t_1, t_2]$ be a fact from \mathcal{D} that has not yet been considered (and so currently appears in \mathcal{C}_{j-1}).
 - If $C_{j-1} \setminus \{\alpha@[t_1,t_2]\}$ is Π -inconsistent, set

$$\mathcal{C}_j = \mathcal{C}_{j-1} \setminus \{\alpha@[t_1, t_2]\}$$

and increment j.

• Otherwise, use binary search to identify a sub-interval $[t_1^*,t_2^*]$ of $[t_1,t_2]$ such that $\mathcal{C}_{j-1}\setminus\{\alpha@[t_1,t_2]\}\cup\{\alpha@[t_1^*,t_2^*]\}$ is Π -inconsistent, and for every proper sub-interval $[t_1',t_2']$ of $[t_1^*,t_2^*]$, the dataset $\mathcal{C}_{j-1}\setminus\{\alpha@[t_1,t_2]\}\cup\{\alpha@[t_1',t_2']\}$ is Π -consistent. Set

$$C_j = C_{j-1} \setminus \{\alpha@[t_1, t_2]\} \cup \{\alpha@[t_1^*, t_2^*]\}$$

and increment j.

By construction, the final set C_n is an *i*-conflict of (\mathcal{D}, Π) . Indeed, we check at every stage that C_i is Π -inconsistent. Moreover, if there exists $\mathcal{B} \sqsubset^i \mathcal{C}_n$ that is Π -inconsistent, this would contradict the minimality of some $\alpha@[t_1^*, t_2^*]$. Regarding the complexity of the procedure, we note that when examining fact $\alpha @ [t_1, t_2]$ at stage j, we can first use binary search and consistency checks to identify the largest $t_1^* \leq t_2$ such that $C_{j-1}\setminus\{\alpha@[t_1,t_2]\}\cup\{\alpha@[t_1^*,t_2]\}$ is Π -inconsistent. We may then do a second binary search to identify the least t_2^* such that $C_{i-1}\setminus\{\alpha@[t_1,t_2]\}\cup\{\alpha@[t_1^*,t_2^*]\}$ is Π -inconsistent. Each binary search will require at most m rounds, with m the number of bits in the binary encoding of t_2 . As m is linear w.r.t. the size of \mathcal{D} , the procedure requires only linearly many consistency checks to examine a given fact, and thus the overall procedure involves quadratically many consistency checks. This yields the desired PSPACE upper bound.

Let us now show how to build an *i*-repair of a dataset \mathcal{D} w.r.t. program Π . Again, we will do so in a greedy fashion.

- 1. Set $\mathcal{R}_0 = \emptyset$, and let n be the number of facts in \mathcal{D} .
- 2. For j = 1 until n:

- Let $\alpha@[t_1,t_2]$ be a fact from $\mathcal D$ that has not yet been considered.
- If $\mathcal{R}_{i-1} \cup \{\alpha@[t_1, t_2]\}$ is consistent with Π , set

$$\mathcal{R}_j = \mathcal{R}_{j-1} \cup \{\alpha@[t_1, t_2]\}$$

and increment j.

- · Otherwise,
 - (a) find the smallest $t_1^* \in [t_1, t_2]$ such that $\mathcal{R}_{j-1} \cup \{\alpha@\{t_1^*\}\}$ is Π -consistent,
 - (b) find the largest $t_2^* \in [t_1^*, t_2]$ such that $\mathcal{R}_{j-1} \cup \{\alpha@[t_1^*, t_2^*]\}$ is Π -consistent, set

$$\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{\alpha@[t_1^*, t_2^*]\}$$

and increment j.

By construction, the final set \mathcal{R}_n is an i-repair of (\mathcal{D},Π) , as consistency is checked for each \mathcal{R}_j , and we ensure that the interval $[t_1^*,t_2^*]$ of each added fact $\alpha@[t_1^*,t_2^*]$ cannot be further extended without losing consistency. As for complexity, step (a) (resp. step (b)) can be done by iterating over the (potentially exponentially many) $t \in [t_1,t_2]$ (resp. $t \in [t_1^*,t_2]$) in polynomial space, yielding a PSPACE upper bound.

Finally, we show the desired PSPACE completeness results for query entailment under i-brave, i-CQA and i-intersection. The lower bounds come from the consistent case, since query entailment under classical semantics is PSPACE-complete w.r.t. data complexity, even when $\mathbb{T}=\mathbb{Z}$ and bounded-interval datasets are considered. Indeed, the reduction used to show that checking consistency when $\mathbb{T}=\mathbb{Z}$ is PSPACE-hard given in the proof of Theorem 2 in [Walega et~al., 2020a] can be easily adapted to use a bounded-interval dataset if the program is not required to belong to a specific DatalogMTL fragment: for every predicate P such that some $P(\vec{c})@(-\infty,\infty)$ occurs in \mathcal{D}_w , add two rules $\boxplus P(\vec{x}) \leftarrow P(\vec{x})$ and $\boxminus P(\vec{x}) \leftarrow P(\vec{x})$ in the program and replace every $P(\vec{c})@(-\infty,\infty)$ by $P(\vec{c})@\{0\}$ in the dataset.

The upper bounds for the three semantics rely upon the facts that i-repairs are polynomial in size (point (i)), they can be recognized in PSPACE (point (ii)), and their number is exponentially bounded (since for each $\alpha@\iota\in\mathcal{D}$, there are at most exponentially many $\alpha@\iota'$ with $\iota'\subseteq\iota$). Indeed, it follows that the 'guess and check' procedures for i-brave and i-CQA semantics described in the proof of Proposition 6 still yield PSPACE upper bounds in this case.

For the *i*-intersection case, we show that for each $\alpha@\iota\in\mathcal{D}$, there exist at most two *i*-repairs \mathcal{B}_1 and \mathcal{B}_2 such that for every $t\in\iota$, $\alpha@\{t\}\not\in \bigcap_{\mathcal{B}\in iRep(\mathcal{D},\Pi)}\mathcal{B}$ implies that $\mathcal{B}_1\sqcap\mathcal{B}_2\not\models\alpha@\{t\}$. Indeed, if there exist two *i*-repairs \mathcal{B}_1 and \mathcal{B}_2 such that for every $t\in\iota$, $\mathcal{B}_1\sqcap\mathcal{B}_2\not\models\alpha@\{t\}$, then for every $t\in\iota$, $\alpha@\{t\}\not\in\bigcap_{\mathcal{B}\in iRep(\mathcal{D},\Pi)}\mathcal{B}$ and \mathcal{B}_1 and \mathcal{B}_2 are as required. Otherwise, if for every pair of *i*-repairs \mathcal{B}_1 and \mathcal{B}_2 there exists $t\in\iota$ such that $\mathcal{B}_1\sqcap\mathcal{B}_2\models\alpha@\{t\}$, it means that every *i*-repair contains some $\alpha@\iota'$ with $\iota'\subseteq\iota$ and these intervals ι' pairwise intersect. It follows that there must exist $t\in\iota$ such that $\alpha@\{t\}\in\bigcap_{\mathcal{B}\in iRep(\mathcal{D},\Pi)}\mathcal{B}$. Let t_1 and t_2 be the smallest and largest such t. If $t_1-1\in\iota$, there exists \mathcal{B}_1 such that $\mathcal{B}_1\not\models\alpha@\{t_1-1\}$ so by definition of t_1 , \mathcal{B}_1 contains a fact of the form $\alpha@[t_1,t']$ with $[t_1,t']\subseteq\iota$. Similarly, if

 $t_2+1\in\iota$, there exists \mathcal{B}_2 such that $\mathcal{B}_2\not\models\alpha@\{t_2+1\}$ so by definition of t_2 , \mathcal{B}_2 contains a fact of the form $\alpha@[t^*,t_2]$ with $[t^*,t_2]\subseteq\iota$. It follows that for every $t\in\iota$, $\alpha@\{t\}\not\in\prod_{\mathcal{B}\in iRep(\mathcal{D},\Pi)}\mathcal{B}$ implies that $t< t_1$ or $t>t_2$ and $\mathcal{B}_1\sqcap\mathcal{B}_2\not\models\alpha@\{t\}$. Hence, one can decide whether $(\mathcal{D},\Pi)\not\models^i_{\cap}q(\vec{c},\iota)$ as follows: guess $\mathcal{B}_1,\ldots,\mathcal{B}_n$ (with $n\leq 2*|\mathcal{D}|$), and check that $\mathcal{B}_k\in iRep(\mathcal{D},\Pi)$ for $1\leq k\leq n$ and that $(\prod_{k=1}^n\mathcal{B}_k,\Pi)\not\models q(\vec{c},\iota)$.

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.

Proof. For tractable DatalogMTL fragments, for which consistency checking is in PTIME w.r.t. data complexity, the procedures described in points (ii) and (iii) of the proof of Proposition 11 to verify whether $\mathcal{B} \in iConf(\mathcal{D},\Pi)$ or generate a single i-conflict run in PTIME.

Note that condition (\star) in the *i*-repair checking procedure prevents us to extend Proposition 12 to *i*-repairs in the same way as we did for *i*-conflicts.

Proposition 13. For tractable DatalogMTL fragments: when $\mathbb{T} = \mathbb{Z}$ and only bounded-interval datasets are considered, query entailment¹ under *i*-brave (resp. *i*-CQA, *i*-intersection) is in NP (resp. in Π_2^p). Lower NP (resp. coNP) bounds hold for Datalog $_{nr}$ MTL and DatalogMTL $_{lin}^{\diamondsuit}$ (and for DatalogMTL $_{core}^{\diamondsuit}$ in the case of *i*-CQA semantics).

Proof. The lower bounds follow from the reductions used in the proof of Proposition 8, since they use only punctual facts over \mathbb{Z} so that s- and i-repairs coincide.

The NP upper bound for i-brave semantics comes from the fact that by monotonicity of DatalogMTL, it is sufficient to guess $\mathcal{B} \sqsubseteq^i \mathcal{D}$ such that \mathcal{B} is Π -consistent and $(\mathcal{B}, \Pi) \models q(\vec{c}, \iota)$ to show that there exists an i-repair \mathcal{R} of \mathcal{D} w.r.t. Π such that $\mathcal{B} \sqsubseteq^i \mathcal{R}$ and $(\mathcal{R}, \Pi) \models q(\vec{c}, \iota)$. Indeed, when $\mathbb{T} = \mathbb{Z}$ and intervals are bounded, every Π -consistent $\mathcal{B} \sqsubseteq^i \mathcal{D}$ can be greedily extended to an i-repair.

The Π_2^p upper bounds for i-CQA and i-intersection use the same procedures as described in the proof of Proposition 11, using the fact that in the case of tractable DatalogMTL fragments, i-repairs can be recognized in coNP. Indeed, one can check that a Π -consistent $\mathcal{B} \sqsubseteq^i \mathcal{D}$ is not an i-repair by guessing \mathcal{B}' such that $\mathcal{B} \sqsubseteq^i \mathcal{B}' \sqsubseteq^i \mathcal{D}$ and \mathcal{B}' is Π -consistent. \square

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} .

Proof. We start with the case of *p*-repairs. Consider the following program and dataset (note that Π belongs to Datalog_{nr}MTL, DatalogMTL $_{\text{core}}^{\diamondsuit}$, DatalogMTL $_{\text{lin}}^{\diamondsuit}$ and propositional DatalogMTL):

$$\Pi = \{A^- \leftarrow \diamondsuit_{\{1\}} A, B^- \leftarrow \diamondsuit_{\{1\}} B, \\ \bot \leftarrow A \land B, \bot \leftarrow A \land A^-, \bot \leftarrow B \land B^-\} \\ \mathcal{D} = \{A@[0, 2^n], B@[0, 2^n]\}$$

Due to the binary encoding of input intervals, \mathcal{D} has size polynomial in n. However, it is not hard to see that any p-repair \mathcal{R} of (\mathcal{D},Π) must take the form

$$\{D_0@\{0\},\ldots,D_{2^n}@\{2^n\}\}$$

where for every $0 \le j \le 2^n$, $D_j \in \{A, B\}$ and for every $0 < j \le 2^n$, $D_j = A$ iff $D_{j-1} = B$. Hence, every p-repair is of size exponential in n.

Note that if we take $\Pi = \{\bot \leftarrow A \land B\}$ (available in every DatalogMTL fragment in which a dataset is not trivially consistent with any program), this is enough to show that *some* repairs can be exponentially large.

For p-conflicts, consider the following program and dataset (here Π belongs to Datalog_{nr}MTL and propositional DatalogMTL):

$$\begin{split} \Pi = &\Pi' \cup \{\bot \leftarrow S \land (C \ \mathcal{U}_{(0,\infty)}E)\} \text{ with} \\ \Pi' = &\{C \leftarrow A \land \ \varphi_{\{1\}} \ B, \ C \leftarrow B \land \ \varphi_{\{1\}} \ A\} \\ \mathcal{D} = &\{S@\{-1\}, A@[0, 2^{n+2}], B@[0, 2^{n+2}], E@\{2^{n+2}\}\} \end{split}$$

Again, due to the binary encoding of input intervals, \mathcal{D} has size polynomial in n. However, we show that every p-conflict \mathcal{C} of (\mathcal{D}, Π) has size at least 2^n .

Let $\mathcal{C} \in pConf(\mathcal{D},\Pi)$. Since $\mathcal{C} \sqsubseteq^p \mathcal{D}$ and violates $\bot \leftarrow S \land (C\ \mathcal{U}_{(0,\infty)}E)$, it must be the case that $S@\{-1\} \in \mathcal{C}$, $E@\{2^{n+2}\} \in \mathcal{C}$ and for every $t \in \mathbb{Z}$, if $-1 < t < 2^{n+2}$, $(\mathcal{C},\Pi') \models C@\{t\}$.

First note that (i) there is no $t \in [0, 2^{n+2} - 1]$ such that both $\mathcal{C} \not\models A@\{t\}$ and $\mathcal{C} \not\models A@\{t+1\}$ (otherwise $(\mathcal{C}, \Pi') \not\models C@\{t\}$).

Assume for a contradiction that there exists $t \in [0, 2^{n+2} - 3]$ such that $\mathcal{C} \models A@[t, t+3]$ (i.e., \mathcal{C} contains a fact of the form $A@\iota$ such that ι contains at least four timepoints).

- If $\mathcal{C} \not\models B@\{t+1\}$ and $\mathcal{C} \not\models B@\{t+2\}$, then $(\mathcal{C},\Pi') \not\models C@\{t+1\}$ and \mathcal{C} is Π -consistent.
- If $\mathcal{C} \models B@\{t+1\}$, then 'removing $A@\{t+1\}$ from \mathcal{C} ' maintains Π -inconsistency so \mathcal{C} is not \sqsubseteq^p minimal. More precisely, let \mathcal{C}' be the normal form of $\operatorname{tp}(\mathcal{C}) \setminus \{A@\{t+1\}\}$. It holds that $\mathcal{C}' \sqsubseteq^p \mathcal{C}$ and \mathcal{C}' is Π -inconsistent because
 - $(C', \Pi') \models C@\{t\}$ by rule $C \leftarrow A \land \bigoplus_{\{1\}} B$,
 - $(C',\Pi') \models C@\{t+1\}$ by rule $C \leftarrow B \land \bigoplus_{\{1\}} A$, and
 - for every t' such that $-1 < t' < 2^{n+2}$, $t' \neq t$, and $t' \neq t+1$, $(\mathcal{C}', \Pi') \models C@\{t'\}$ because $(\mathcal{C}, \Pi') \models C@\{t'\}$ and only facts that hold at t' and t'+1 are relevant to this entailment and t' and t'+1 are different from t+1.
- If C |= B@{t+2}, then 'removing A@{t+2} from C' maintains Π-inconsistency (as above) so C is not □^p minimal.

In every case, we obtain that $\mathcal{C} \notin pConf(\mathcal{D}, \Pi)$, so we conclude that (ii) every fact of the form $A@\iota$ in \mathcal{C} is such that ι contains at most three timepoints.

It follows from (i) and (ii) that the fact $A@[0, 2^{n+2}]$ is split in \mathcal{C} in at least 2^n facts.

If we only require that *some* p-conflicts are of exponential size, one can consider the following DatalogMTL $_{\text{lin}}^{\diamondsuit}$ program

and dataset.

$$\begin{split} \Pi = & \{A^- \leftarrow \diamondsuit_{\{1\}} A, B^- \leftarrow \diamondsuit_{\{1\}} B, \\ & \perp \leftarrow A \land B, \perp \leftarrow A \land A^-, \perp \leftarrow B \land B^-, \\ & Q' \leftarrow Q \land A, Q' \leftarrow Q \land B, Q \leftarrow \diamondsuit Q', \\ & \perp \leftarrow P \land Q'\} \\ \mathcal{D} = & \{Q@\{0\}, A@[0, 2^n], B@[0, 2^n], P@\{2^n\}\} \end{split}$$

It is not hard to check that $\{Q@\{0\}, P@\{2^n\}\} \cup \{A@\{2k\} \mid 0 \le k \le 2^{n-1}\} \cup \{B@\{2k-1\} \mid 1 \le k \le 2^{n-1}\}$ is a p-conflict of $\mathcal D$ w.r.t. Π .

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.

Proof. By Lemma 3, $\mathcal{R} \in pRep(\mathcal{D},\Pi)$ iff \mathcal{R} is in normal form and $tp(\mathcal{R})$ is a \subseteq -maximal Π -consistent subset of $tp(\mathcal{D})$; and $\mathcal{C} \in pConf(\mathcal{D},\Pi)$ iff \mathcal{C} is in normal form and $tp(\mathcal{C})$ is a \subseteq -minimal Π -inconsistent subset of $tp(\mathcal{D})$. Moreover, the number of facts in $tp(\mathcal{D})$ is exponentially bounded by the size of \mathcal{D} , since each $\alpha@[t_1,t_2] \in \mathcal{D}$ corresponds to at most exponentially many $\alpha@\{t\}$ w.r.t. the size of the binary encodings of t_1 and t_2 . Applying the procedures described in the proof of Proposition 6 using $tp(\mathcal{D})$ instead of \mathcal{D} thus yields ExpSpace upper bounds instead of PSpace ones. \square