Materialisation-Based Reasoning in DatalogMTL with Bounded Intervals

Przemysław A. Wałęga, Michał Zawidzki, Dingmin Wang, Bernardo Cuenca Grau

Department of Computer Science, University of Oxford, UK
{przemyslaw.walega, michal.zawidzki, dingmin.wang, bernardo.cuenca.grau}@cs.ox.ac.uk

Abstract

DatalogMTL is a powerful extension of Datalog with operators from metric temporal logic (MTL) which has received significant attention in recent years. In this paper, we investigate materialisation-based reasoning (a.k.a. forward chaining) in the context of DatalogMTL programs and datasets with bounded intervals, where partial representations of the canonical model are obtained through successive rounds of rule applications. Although materialisation does not naturally terminate in this setting, it is known that the structure of canonical models is ultimately periodic. Our first contribution in this paper is a detailed analysis of the periodic structure of canonical models; in particular, we formulate saturation conditions whose satisfaction by a partial materialisation implies the ability to recover the full canonical model via unfolding; this allows us to compute the actual periods describing the repeating parts of the canonical model as well as to establish concrete bounds on the number of rounds of rule applications required to achieve saturation. Based on these theoretical results, we propose a practical reasoning algorithm where saturation can be efficiently detected as materialisation progresses, and where the relevant periods used to evaluate entailment of queries via unfolding are efficiently computed. We have implemented our algorithm and our experiments suggest that our approach is both scalable and robust.

Introduction

DatalogMTL is a powerful rule-based language for representing temporal knowledge that has found applications in ontology-based data access (Brandt et al. 2018; Kikot et al. 2018; Kalaycı et al. 2018; Koopmann 2019) and stream reasoning (Wałęga, Kaminski, and Cuenca Grau 2019). It extends Datalog (Ceri, Gottlob, and Tanca 1989) with operators from metric temporal logic (Koymans 1990) interpreted over the rational timeline. For example, the following DatalogMTL rule states that a person can travel if they had a negative COVID test at some point in the last 2 days ($\diamondminus_{[0,2]}$) and have been double-vaccinated for 14 days ($\boxminus_{[0,14]}$):

$$CanTravel(x) \leftarrow \diamondminus_{[0,2]} NegTest(x) \land \boxminus_{[0,14]} DblVacc(x).$$

Reasoning in DatalogMTL is of high complexity, which makes its adoption in applications problematic. In particular, satisfiability and fact entailment are ExpSpace-complete (Brandt et al. 2018) and PSpace-complete in data complexity (Wałęga et al. 2019). Decision procedures are based on the observation that, although the timeline is densely ordered, the canonical model of a program and a dataset can be partitioned according to discrete sequences of intervals, inside which all time points satisfy the same facts. This enables the reduction (with exponential blow-up) of fact entailment in DatalogMTL to satisfiability of linear temporal logic (LTL) formulas (Brandt et al. 2018) or to non-emptiness of Büchi automata (Wałęga et al. 2019). As a result, canonical models have a periodic structure, with periods of bounded length (Artale et al. 2021a). Although these techniques are useful for providing complexity upper bounds for reasoning, they are based on exponential reductions and were not designed for efficient implementation.

The most common technique of choice in scalable Datalog reasoners is materialisation (a.k.a. forward chaining), where facts entailed by a program and dataset are derived in successive rounds of rule applications until a fixpoint is reached (Bry et al. 2007; Motik et al. 2014; Carral et al. 2019; Bellomarini, Sallinger, and Gottlob 2018).[1] Once the materialisation has been computed, queries can be answered directly and rules are not further considered. In contrast to Datalog where materialisation naturally terminates, in DatalogMTL a fixpoint may only be reachable after infinitely many rounds of rule applications. To circumvent this challenge, the MeTeoR reasoner (Wang et al. 2022) combines materialisation and automata-based reasoning, where automata construction ensures completeness where materialisation does not terminate; experiments, however, show a significant performance reduction in such cases (Wang et al. 2022). An alternative approach is to focus on DatalogMTL fragments for which materialisation is guaranteed to terminate (Wałęga, Zawidzki, and Cuenca Grau 2021); such fragments, however, impose significant restrictions effectively disallowing ‘recursion through time’.

In this paper, we investigate the design of reasoning algorithms for DatalogMTL that rely solely on materialisation, dispense with automata construction altogether, and do not limit recursion. We focus on bounded DatalogMTL programs and datasets, where ∞ is not mentioned as an end-

[1] In this setting, both the reasoning process and its output are often referred to as materialisation.

point of any interval; this is a natural and expressive fragment, in which reasoning is as hard as in the unrestricted language (Wałęga, Zawidzki, and Cuenca Grau 2021).

Our first contribution is a detailed analysis of the periodic structure of canonical models. We formulate *saturation conditions* that a partial materialisation needs to satisfy so that the canonical model can be recovered via unfolding; this allows us to compute the actual periods describing the repeating parts of the model based only on the form of the partial materialisation constructed so far, as well as to establish concrete bounds on the number of rounds of rule applications required to achieve saturation. This is a challenging problem since DatalogMTL rules may recursively propagate facts towards both future and past time points, and hence regularities observed in a partial materialisation may not correspond to the periodic structure of the full canonical model.

Based on these theoretical results, we propose a practical reasoning algorithm where saturation can be efficiently detected as materialisation progresses, and where the relevant periods used to evaluate entailment queries via unfolding are efficiently computed. We have implemented our algorithm and evaluated its performance on a temporal extension of Lehigh University Benchmark (Wang et al. 2022) as well as using the iTemporal benchmark generator (Bellomarini, Nissl, and Sallinger 2022). Our results show that the overhead introduced by saturation checks during materialisation is negligible, saturation can be reached after a reasonable number of rounds of rule applications, and entailment checks performed after reaching saturation can be conducted very efficiently.

Preliminaries

We recapitulate the syntax and semantics of DatalogMTL, focusing on the standard continuous semantics over the rational timeline (Brandt et al. 2018; Wałęga et al. 2019).

Time and Intervals. The *(rational) timeline* is the ordered set $\mathbb{Q}$ of rational numbers; each element of the timeline is called a *time point*. We consider binary representations of integers, and represent each rational number as a fraction with an integer numerator and a positive integer denominator.

An *interval* ϱ is a non-empty subset of $\mathbb{Q}$ satisfying two properties: (i) for all $t_1, t_2, t_3 \in \mathbb{Q}$ with $t_1 < t_2 < t_3$ and $t_1, t_3 \in \varrho$, it is the case that $t_2 \in \varrho$, and (ii) both the greatest lower bound ϱ^- and the least upper bound ϱ^+ of ϱ are in $\mathbb{Q} \cup \{-\infty, \infty\}$. The bounds ϱ^- and ϱ^+ are the *left* and *right endpoints* of ϱ, respectively, and $|\varrho| = \varrho^+ - \varrho^-$ is the *length* of ϱ. Interval ϱ is *punctual* if it has a single time point, it is *positive* if it does not contain negative time points, and *bounded* if both of its endpoints are rationals numbers. For intervals ϱ and ϱ' we define the operations $\varrho + \varrho' = \{t + t' \mid t \in \varrho \text{ and } t' \in \varrho'\}$ and $-\varrho = \{-t \mid t \in \varrho\}$. We use the standard representation $\langle \varrho^-, \varrho^+ \rangle$ for an interval ϱ, where the *left bracket* $\langle$ is either [or (, the *right bracket* $\rangle$ is either] or), and ϱ^- and ϱ^+ are representations of the left and right endpoints of ϱ, respectively. Brackets [and] indicate that the corresponding endpoints are included in the interval, whereas (and) indicate that they are not included. We often abbreviate a punctual interval $[t, t]$ as t. Since different intervals cannot have the same representation, we will often abuse notation and identify an interval representation with the interval it represents.

Syntax. Assume a function-free first-order signature. A *relational atom* is a first-order atom of the form $P(\mathbf{s})$, with P a predicate and $\mathbf{s}$ a tuple of terms whose length matches the arity of P. A *metric atom* is an expression given by the following grammar, where $P(\mathbf{s})$ is a relational atom and ϱ a positive interval:

$$M ::= \top \mid \bot \mid P(\mathbf{s}) \mid \diamondminus_\varrho M \mid \diamondplus_\varrho M \mid \boxminus_\varrho M \mid \boxplus_\varrho M \mid M \mathcal{S}_\varrho M \mid M \mathcal{U}_\varrho M.$$

A *rule* is an expression of the form

$$M' \leftarrow M_1 \wedge \cdots \wedge M_n, \qquad \text{for } n \geq 1, \tag{1}$$

where each M_i is a metric atom, whereas M' is a metric atom not mentioning $\diamondminus$, $\diamondplus$, $\mathcal{S}$, and $\mathcal{U}$, and hence generated by the following grammar:[2]

$$M' ::= \top \mid P(\mathbf{s}) \mid \boxminus_\varrho M' \mid \boxplus_\varrho M'.$$

The conjunction $M_1 \wedge \cdots \wedge M_n$ in Expression (1) is the rule's *body*; each M_i is a *body atom* and M' is the rule's *head*. A rule is *safe* if all its variables occur in the body.

A *program* is a finite set of safe rules. A program is *bounded* if all intervals it mentions are bounded. An expression (metric atom, rule, or program) is *ground* if it has no variables. The *grounding* $\mathsf{ground}(\Pi)$ of Π is the set of all ground rules obtained by assigning constants to variables in Π. A *metric fact* over an interval ϱ is an expression $M@\varrho$, with M a ground metric atom; it is relational if so is M and bounded if so is ϱ. A *dataset* is a finite set of relational facts; it is bounded if so is each of its facts. For a dataset $\mathcal{D}$ we let $t^-_\mathcal{D}$ and $t^+_\mathcal{D}$ be, respectively, the minimal and maximal rational numbers mentioned as endpoints of intervals in $\mathcal{D}$ (or 0 if $\mathcal{D}$ mentions no numbers).

Semantics. An *interpretation* $\mathfrak{I}$ specifies, for each ground relational atom $P(\mathbf{s})$ and each time point t, whether $P(\mathbf{s})$ is *satisfied* at t, in which case we write $\mathfrak{I}, t \models P(\mathbf{s})$. This notion extends to other ground metric atoms as specified in Table 1. Interpretation $\mathfrak{I}$ satisfies a metric fact $M@\varrho$, written $\mathfrak{I} \models M@\varrho$, if $\mathfrak{I}, t \models M$ for all $t \in \varrho$. Interpretation $\mathfrak{I}$ satisfies a ground rule r if, for any time point t, whenever $\mathfrak{I}$ satisfies each body atom of r at t, then $\mathfrak{I}$ also satisfies the head of r at t. Interpretation $\mathfrak{I}$ satisfies a (possibly non-ground) rule r if $\mathfrak{I}$ satisfies each rule in $\mathsf{ground}(\{r\})$; it satisfies a program Π if $\mathfrak{I}$ satisfies each rule in Π; moreover, $\mathfrak{I}$ satisfies Π in an interval ϱ if, for each $r \in \mathsf{ground}(\Pi)$ and each $t \in \varrho$, whenever $\mathfrak{I}$ satisfies each body atom of r at t, then $\mathfrak{I}$ satisfies the head of r at t. Interpretation $\mathfrak{I}$ is a *model* of a program Π if it satisfies Π, and it is a *model* of a dataset $\mathcal{D}$ if it satisfies each fact in $\mathcal{D}$. A program Π and a dataset $\mathcal{D}$ *entail* a metric fact $M@\varrho$, written as $(\Pi, \mathcal{D}) \models M@\varrho$, if each model of both Π and $\mathcal{D}$ is also a model of $M@\varrho$. We may write $\mathcal{D} \models M@\varrho$ instead of $(\emptyset, \mathcal{D}) \models M@\varrho$.

[2]For convenience, we additionally disallow $\bot$ in rule heads, which ensures consistency and allows us to focus on fact entailment and the computation of canonical interpretations.

$\mathfrak{I}, t \models \top$	for each t	
$\mathfrak{I}, t \models \bot$	for no t	
$\mathfrak{I}, t \models \diamondminus_\varrho M$	iff $\mathfrak{I}, t' \models M$ for some t' with $t - t' \in \varrho$	
$\mathfrak{I}, t \models \diamondplus_\varrho M$	iff $\mathfrak{I}, t' \models M$ for some t' with $t' - t \in \varrho$	
$\mathfrak{I}, t \models \boxminus_\varrho M$	iff $\mathfrak{I}, t' \models M$ for all t' with $t - t' \in \varrho$	
$\mathfrak{I}, t \models \boxplus_\varrho M$	iff $\mathfrak{I}, t' \models M$ for all t' with $t' - t \in \varrho$	
$\mathfrak{I}, t \models M_1 \mathcal{S}_\varrho M_2$	iff $\mathfrak{I}, t' \models M_2$ for some t' with $t - t' \in \varrho$ and $\mathfrak{I}, t'' \models M_1$ for all $t'' \in (t', t)$	
$\mathfrak{I}, t \models M_1 \mathcal{U}_\varrho M_2$	iff $\mathfrak{I}, t' \models M_2$ for some t' with $t' - t \in \varrho$ and $\mathfrak{I}, t'' \models M_1$ for all $t'' \in (t, t')$	

Table 1: Semantics of ground metric atoms

An interpretation $\mathfrak{I}$ *contains* an interpretation $\mathfrak{I}'$, written $\mathfrak{I}' \subseteq \mathfrak{I}$, if $\mathfrak{I}', t \models P(\mathbf{s})$ implies $\mathfrak{I}, t \models P(\mathbf{s})$, for each ground relational atom $P(\mathbf{s})$ and time point t. Then, $\mathfrak{I}$ is the *least* interpretation in a set X of interpretations if $\mathfrak{I} \subseteq \mathfrak{I}'$ for every $\mathfrak{I}' \in X$. Each dataset $\mathcal{D}$ admits the least interpretation $\mathfrak{I}_\mathcal{D}$ among all models of $\mathcal{D}$ and we say that a dataset $\mathcal{D}$ *represents* an interpretation $\mathfrak{I}$ if $\mathfrak{I} = \mathfrak{I}_\mathcal{D}$.

Canonical Model We let the *immediate consequence operator* T_Π for a program Π be the function mapping an interpretation $\mathfrak{I}$ to the least interpretation $T_\Pi(\mathfrak{I})$ containing $\mathfrak{I}$ and satisfying the following property for each $r \in \mathsf{ground}(\Pi)$: whenever $\mathfrak{I}$ satisfies each body atom of r at a time point t, then $T_\Pi(\mathfrak{I})$ satisfies the head of r at t. Successive applications of T_Π to $\mathfrak{I}_\mathcal{D}$ define a transfinite sequence of interpretations $T_\Pi^\alpha(\mathfrak{I}_\mathcal{D})$ for ordinals α as follows: (i) $T_\Pi^0(\mathfrak{I}_\mathcal{D}) = \mathfrak{I}_\mathcal{D}$, (ii) $T_\Pi^{\alpha+1}(\mathfrak{I}_\mathcal{D}) = T_\Pi(T_\Pi^\alpha(\mathfrak{I}_\mathcal{D}))$ for α an ordinal, and (iii) $T_\Pi^\alpha(\mathfrak{I}_\mathcal{D}) = \bigcup_{\beta<\alpha} T_\Pi^\beta(\mathfrak{I}_\mathcal{D})$ for α a limit ordinal. The *canonical model* $\mathfrak{C}_{\Pi,\mathcal{D}}$ of Π and $\mathcal{D}$ is the interpretation $T_\Pi^{\omega_1}(\mathfrak{I}_\mathcal{D})$, where ω_1 is the first uncountable ordinal. Since we do not allow $\bot$ in rule heads, $\mathfrak{C}_{\Pi,\mathcal{D}}$ is always a model of Π and $\mathcal{D}$, and it is the least such model (Brandt et al. 2017). Interpretation $\mathfrak{C}_{\Pi,\mathcal{D}}$ can be divided into regularly distributed $(\Pi, \mathcal{D})$-*intervals* whose time points satisfy the same ground atoms (Wałęga et al. 2019). In particular, the $(\Pi, \mathcal{D})$-*ruler* is the set of time points of the form $t + i \cdot \mathsf{div}(\Pi)$, for $t \in \mathbb{Q}$ mentioned in $\mathcal{D}$ and $i \in \mathbb{Z}$, and where $\mathsf{div}(\Pi) = \frac{1}{k}$, with k being the product of all denominators in the rational endpoints of the intervals mentioned in Π (for definiteness, if Π has no intervals with rational endpoints, we let $k = 1$ and hence $\mathsf{div}(\Pi) = 1$).

A $(\Pi, \mathcal{D})$-*interval* is either a punctual interval over a time point on the $(\Pi, \mathcal{D})$-ruler or an interval (t_1, t_2) with t_1 and t_2 consecutive time points on the $(\Pi, \mathcal{D})$-ruler. We say than an interpretation $\mathfrak{I}$ is a $(\Pi, \mathcal{D})$-interpretation if, for every relational fact $M@t$, it holds that $\mathfrak{I} \models M@t$ implies $\mathfrak{I} \models M@\varrho$, where ϱ is the $(\Pi, \mathcal{D})$-interval containing t. It turns out that $\mathfrak{C}_{\Pi,\mathcal{D}}$ as well as $T_\Pi^\alpha(\mathfrak{I}_\mathcal{D})$, for any ordinal α, is a $(\Pi, \mathcal{D})$-interpretation (Wałęga et al. 2019).

Reasoning. We consider *fact entailment*, that is, checking if a relational fact is entailed by a program and a dataset. This problem is PSpace-complete in data complexity, that is, when the size of a program is considered as fixed (Wałęga et al. 2019), and ExpSpace-complete in combined complexity, where complexity is measured also with respect to the program (Brandt et al. 2017). The same complexity bounds hold already in the case of bounded programs and datasets (Wałęga, Zawidzki, and Cuenca Grau 2021).

Periodic Structure of the Canonical Model

In this section, we establish the theoretical basis for our novel reasoning approach. Given a bounded program Π and a dataset $\mathcal{D}$, we present saturation conditions stating the properties that a partial materialisation $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ needs to satisfy so that the full canonical model can be recovered. We call an interpretation satisfying these conditions *saturated* and show how to compute relevant periods and exploit them to *unfold* the interpretation into the canonical model of Π and $\mathcal{D}$. This allows us to check entailment of any fact.

For convenience of presentation, we fix for the remainder of this section an arbitrary bounded program Π, bounded dataset $\mathcal{D}$, and natural number k. Moreover, for an interpretation $\mathfrak{I}$ and an interval ϱ, we let the *projection* $\mathfrak{I}\mid_\varrho$ of $\mathfrak{I}$ over ϱ be the interpretation that coincides with $\mathfrak{I}$ on ϱ and makes all relational atoms false outside ϱ. We say that an interpretation $\mathfrak{I}'$ is a *shift* of $\mathfrak{I}$ if there is a rational number q such that $\mathfrak{I} \models M@\varrho$ if and only if $\mathfrak{I}' \models M@(\varrho + q)$, for each (relational) fact $M@\varrho$. Furthermore, for a rule r, we define its *depth*, written as $\mathsf{depth}(r)$, as the sum of right endpoints of all intervals occurring in the operators of r (or 0 if r mentions no intervals), and we let $\mathsf{depth}(\Pi)$ be the maximum depth of its rules. As shown in the following proposition, the depth of a bounded rule determines the time points that can be 'affected' by an application of this rule.

Proposition 1. *For every interpretation $\mathfrak{I}$, time point t, and bounded rule r, it holds that $\mathfrak{I}$ satisfies r at t if and only if so does $\mathfrak{I}\mid_{[t-\mathsf{depth}(r), t+\mathsf{depth}(r)]}$.*

Proof. By definition, $\mathfrak{I}$ satisfies r at t if and only if the body of r does not hold at t or the head of r holds at t. By the definition of $\mathsf{depth}(r)$, all the facts corresponding to the satisfaction of the body and the head of r at t are over intervals contained in $[t - \mathsf{depth}(r), t + \mathsf{depth}(r)]$. This directly implies the claim in the proposition. □

Having introduced these basic concepts, we are now ready to define our notion of a saturated interpretation.

Definition 2. Interpretation $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ is *saturated* if there exist closed intervals ϱ_1, ϱ_2, ϱ_3, and ϱ_4 of length $2\mathsf{depth}(\Pi)$, whose endpoints are located on the $(\Pi, \mathcal{D})$-ruler and satisfy $\varrho_1^+ < \varrho_2^+ < t_\mathcal{D}^-$ and $t_\mathcal{D}^+ < \varrho_3^- < \varrho_4^-$, and such that the following properties hold:

- $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ satisfies Π in $[\varrho_1^-, \varrho_4^+]$;
- $T_\Pi^k(\mathfrak{I}_\mathcal{D})\mid_{\varrho_1}$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D})\mid_{\varrho_3}$ are shifts of $T_\Pi^k(\mathfrak{I}_\mathcal{D})\mid_{\varrho_2}$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D})\mid_{\varrho_4}$, respectively.

Any pair of intervals $[\varrho_1^-, \varrho_2^-)$ and $(\varrho_3^+, \varrho_4^+]$, for ϱ_1, ϱ_2, ϱ_3, ϱ_4 as above, will be referred to as *periods* of $T_\Pi^k(\mathfrak{I}_\mathcal{D})$.

Intuitively, a saturated interpretation contains a 'central fragment' $[\varrho_1^-, \varrho_4^+]$ which satisfies $\mathcal{D}$ and such that a single application of T_Π does not derive any new facts within this fragment (i.e., the interpretation satisfies Π within $[\varrho_1^-, \varrho_4^+]$). In the left segment of this 'central fragment' there are two intervals ϱ_1 and ϱ_2—each of length $2\mathsf{depth}(\Pi)$—in which $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ satisfies the same relational facts modulo a shift. Analogously, in the right segment of the 'central fragment' there are intervals ϱ_3 and ϱ_4 also with repeating contents.

It is worth emphasising that in the definition of a saturated interpretation (first item), we consider only a single materialisation step, which can be effectively checked. As we will show in Theorem 5, rather surprisingly, this condition is enough to guarantee that a saturated interpretation can be unfolded into the canonical model.

We next define the unfolding of a saturated interpretation relatively to a pair $(\varrho_{\mathsf{left}}, \varrho_{\mathsf{right}})$ of its periods. Although there can be many such pairs of intervals, the key properties of the unfolding are independent of the choice of periods.

Definition 3. The $(\varrho_{\mathsf{left}}, \varrho_{\mathsf{right}})$-unfolding, of a saturated interpretation $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ with periods $(\varrho_{\mathsf{left}}, \varrho_{\mathsf{right}})$, is the interpretation $\mathfrak{C}$ such that:

- $\mathfrak{C}\mid_{[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]} = T_\Pi^k(\mathfrak{I}_\mathcal{D})\mid_{[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]}$,
- $\mathfrak{C}\mid_{\varrho_{\mathsf{left}} - n\cdot|\varrho_{\mathsf{left}}|}$ is a shift of $T_\Pi^k(\mathfrak{I}_\mathcal{D})\mid_{\varrho_{\mathsf{left}}}$, for any $n \in \mathbb{N}$,
- $\mathfrak{C}\mid_{\varrho_{\mathsf{right}} + n\cdot|\varrho_{\mathsf{right}}|}$ is a shift of $T_\Pi^k(\mathfrak{I}_\mathcal{D})\mid_{\varrho_{\mathsf{right}}}$, for any $n \in \mathbb{N}$.

We observe that the unfolding is unique and can be obtained from the least interpretation coinciding with $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ on $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$ by subsequently copying the fragment of this interpretation that spans ϱ_{left} infinitely many times to the left and the fragment of this interpretation that spans ϱ_{right} infinitely many times towards the right on the timeline.

In Theorem 5 we will show that the unfolding of a saturated interpretation coincides with the canonical model. The proof of this theorem exploits a generalisation of a technical lemma from the literature (Wałęga, Zawidzki, and Cuenca Grau 2021, Lemma 11), which we provide next.

Lemma 4. *Let ϱ be a closed interval of length $\mathsf{depth}(\Pi)$ and $\mathcal{D}'$ a dataset representing $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_\varrho$. Then both of the following hold:*

1. *If $\varrho^+ \geq t_\mathcal{D}^+$, then $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_{(\varrho^+,\infty)} = \mathfrak{C}_{\Pi,\mathcal{D}'}\mid_{(\varrho^+,\infty)}$.*
2. *If $\varrho^- \leq t_\mathcal{D}^-$, then $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_{(-\infty,\varrho^-)} = \mathfrak{C}_{\Pi,\mathcal{D}'}\mid_{(-\infty,\varrho^-)}$.*

Proof sketch. Both items have similar proofs, so we focus on the first one. The inclusion $\mathfrak{C}_{\Pi,\mathcal{D}'}\mid_{(\varrho^+,\infty)} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}\mid_{(\varrho^+,\infty)}$ follows from the fact that $\mathfrak{C}_{\Pi,\mathcal{D}} \models \mathcal{D}'$, so $\mathfrak{C}_{\Pi,\mathcal{D}'} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}$. To show that $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_{(\varrho^+,\infty)} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}'}\mid_{(\varrho^+,\infty)}$ it suffices to prove inductively that, for every ordinal α and relational fact $M@t$ with $t > \varrho^+$, if $T_\Pi^\alpha(\mathfrak{I}_\mathcal{D}) \models M@t$, then $T_\Pi^\alpha(\mathfrak{I}_{\mathcal{D}'}) \models M@t$. We conduct the induction similarly to the proof due to Wałęga, Zawidzki, and Cuenca Grau (2021, Lemma 11), but using $\mathsf{depth}(\Pi)$ in the inductive step. □

Theorem 5. *The $(\varrho_{\mathsf{left}}, \varrho_{\mathsf{right}})$-unfolding $\mathfrak{C}$ of a saturated interpretation $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ with periods $(\varrho_{\mathsf{left}}, \varrho_{\mathsf{right}})$ coincides with the canonical model $\mathfrak{C}_{\Pi,\mathcal{D}}$.*

Proof sketch. We first show that $\mathfrak{C}_{\Pi,\mathcal{D}} \subseteq \mathfrak{C}$. Since $\mathfrak{C}_{\Pi,\mathcal{D}}$ is the least model of Π and $\mathcal{D}$, it suffices to show that $\mathfrak{C}$ is a model of Π and $\mathcal{D}$. As $[t_\mathcal{D}^-, t_\mathcal{D}^+] \subseteq [\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$, we obtain that the projection of $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ over $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$ is a model of $\mathcal{D}$. However, by Definition 3, $\mathfrak{C}$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ coincide on $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$, so $\mathfrak{C}$ is also a model of $\mathcal{D}$. To show that $\mathfrak{C}$ satisfies Π in $\varrho = [\varrho_{\mathsf{left}}^- + \mathsf{depth}(\Pi), \varrho_{\mathsf{right}}^+ - \mathsf{depth}(\Pi)]$, it suffices, by Proposition 1, to show that the projection of $\mathfrak{C}$ over $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$ satisfies Π in ϱ; this holds since $\mathfrak{C}$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ coincide on $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ satisfies Π in $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$, as it is saturated. It remains to argue that $\mathfrak{C}$ satisfies Π at each t outside ϱ. Assume that $t < \varrho^-$, so $t < \varrho_{\mathsf{left}}^- + \mathsf{depth}(\Pi)$. Then, there exists a unique pair of $n \in \mathbb{N}$ and $t' \in [\varrho_{\mathsf{left}}^- + \mathsf{depth}(\Pi), \varrho_{\mathsf{left}}^+ + \mathsf{depth}(\Pi))$ such that $t + n\cdot|\varrho_{\mathsf{left}}| = t'$. This, by the definition of $\mathfrak{C}$, implies that $\mathfrak{C}\mid_{[t-\mathsf{depth}(\Pi), t+\mathsf{depth}(\Pi)]}$ is a shift of $\mathfrak{C}\mid_{[t'-\mathsf{depth}(\Pi), t'+\mathsf{depth}(\Pi)]}$. Thus, by Proposition 1, $\mathfrak{C}$ satisfies Π at t if and only if $\mathfrak{C}$ satisfies Π at t'. The latter holds since $t' \in \varrho$ and, as we already showed, $\mathfrak{C}$ satisfies Π in ϱ. The proof for $t > \varrho_{\mathsf{right}}^- - \mathsf{depth}(\Pi)$ is similar, so $\mathfrak{C}_{\Pi,\mathcal{D}} \subseteq \mathfrak{C}$.

We show that $\mathfrak{C} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}$. The projection of $\mathfrak{C}$ over $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$ is in $\mathfrak{C}_{\Pi,\mathcal{D}}$ since $\mathfrak{C}$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ coincide on this interval and $T_\Pi^k(\mathfrak{I}_\mathcal{D}) \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}$. To show that the projection of $\mathfrak{C}$ over $(-\infty, \varrho_{\mathsf{left}}^-]$ is in $\mathfrak{C}_{\Pi,\mathcal{D}}$, we argue that $\mathfrak{C}\mid_{[t_i,\varrho_{\mathsf{right}}^+]} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}\mid_{[t_i,\varrho_{\mathsf{right}}^+]}$ by induction on consecutive timepoints t_i on the $(\Pi,\mathcal{D})$-ruler, with $t_0 = \varrho_{\mathsf{left}}^-$. The base holds since $\mathfrak{C}$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ coincide on $[\varrho_{\mathsf{left}}^-, \varrho_{\mathsf{right}}^+]$. For the inductive step, let $\varrho_A = [t_i, t_{i-1})$, $\varrho_B = [t_{i-1}, t_{i-1} + \mathsf{depth}(\Pi)]$, $\varrho'_A = \varrho_A + |\varrho_{\mathsf{left}}|$, and $\varrho'_B = \varrho_B + |\varrho_{\mathsf{left}}|$. Hence, it suffices to show that $\mathfrak{C}\mid_{\varrho_A} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}\mid_{\varrho_A}$. We note that $\mathfrak{C}\mid_{\varrho_B}$ is a shift of $\mathfrak{C}\mid_{\varrho'_B}$ by the construction of $\mathfrak{C}$ and the fact that $(\varrho'_B)^+ \leq \varrho_{\mathsf{left}}^+ + \mathsf{depth}(\Pi)$. Thus, by the inductive assumption and $\mathfrak{C}_{\Pi,\mathcal{D}} \subseteq \mathfrak{C}$, we obtain that $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_{\varrho_B}$ is a shift of $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_{\varrho'_B}$. By the construction of $\mathfrak{C}$ we know that $\mathfrak{C}\mid_{\varrho_A}$ is a shift of $\mathfrak{C}\mid_{\varrho'_A}$ and since the lengths of ϱ_B and ϱ'_B equal $\mathsf{depth}(\Pi)$, we can use Lemma 4 to show that $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_{\varrho_A}$ is a shift of $\mathfrak{C}_{\Pi,\mathcal{D}}\mid_{\varrho'_A}$. By the inductive assumption, $\mathfrak{C}\mid_{\varrho_{A'}} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}\mid_{\varrho_{A'}}$ and so, $\mathfrak{C}\mid_{\varrho_A} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}\mid_{\varrho_A}$. The proof of the inclusion $\mathfrak{C}\mid_{[\varrho_{\mathsf{right}}^+,\infty)} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}\mid_{[\varrho_{\mathsf{right}}^+,\infty)}$ is symmetric; thus, $\mathfrak{C} \subseteq \mathfrak{C}_{\Pi,\mathcal{D}}$. □

Theorem 5 states that the unfolding of a saturated interpretation around any pair of its periods coincides with the canonical model. Thus, we can refer to an unfolding of a saturated interpretation without explicitly referring to its periods.

We conclude this section by establishing a bound k_{max}, depending on Π and $\mathcal{D}$, which ensures that $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ is saturated for some $k \leq k_{\mathsf{max}}$. Intuitively, k_{max} is the number of facts that we can construct by combining atoms relevant to Π and $\mathcal{D}$ with $(\Pi,\mathcal{D})$-intervals contained in a sufficiently large interval ϱ. The interval ϱ is chosen so that $[t_\mathcal{D}^-, t_\mathcal{D}^+] \subseteq \varrho$, and such that its fragments $[\varrho^-, t_\mathcal{D}^-)$ and $(t_\mathcal{D}^+, \varrho^+]$ to the left and right of $\mathcal{D}$, respectively, contain so many intervals of length $2\mathsf{depth}(\Pi)$ with endpoints on the $(\Pi, \mathcal{D})$-ruler, that their contents need to repeat. These repetitions guarantee the existence of intervals $\varrho_1, \dots, \varrho_4$ from Definition 2.

Theorem 6. *There exists $k \leq k_{\mathsf{max}}$ such that $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ is saturated, where the bound k_{max} is defined as follows. Let A be the number of ground relational atoms in the grounding of Π with constants from Π and $\mathcal{D}$, let B be the number of $(\Pi, \mathcal{D})$-intervals within $[t_\mathcal{D}^-, t_\mathcal{D}^- + 2\mathsf{depth}(\Pi)]$, and let $\varrho = [t_{-w} - 2\mathsf{depth}(\Pi), t_w + 2\mathsf{depth}(\Pi)]$, where $\dots < t_{-2} < t_{-1} < t_\mathcal{D}^-$ and $t_\mathcal{D}^+ < t_1 < t_2 < \dots$ are sequences of consecutive time points on the $(\Pi, \mathcal{D})$-ruler, whil $w = 1 + B \cdot (2^A)^B$. Then k_{max} is the product of A and the number of $(\Pi, \mathcal{D})$-intervals contained in ϱ.*

Proof sketch. The canonical model $\mathfrak{C}_{\Pi,\mathcal{D}}$ satisfies at most k_{max} relational facts over $(\Pi, \mathcal{D})$-intervals contained in ϱ. Since each application of T_Π (before reaching a fixpoint) introduces at least one new relational fact over a $(\Pi, \mathcal{D})$-interval, there needs to exist $k \leq k_{\mathsf{max}}$ such that $T_\Pi^k(\mathfrak{I}_\mathcal{D}) \mid_\varrho = T_\Pi^{k+1}(\mathfrak{I}_\mathcal{D}) \mid_\varrho$. We argue that $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ is saturated for such k by showing that ϱ contains intervals $\varrho_1, \dots, \varrho_4$ satisfying the conditions from Definition 2. To show the existence of ϱ_1 and ϱ_2, we observe that ϱ contains $w = 1 + B \cdot (2^A)^B$ intervals of length $2\mathsf{depth}(\Pi)$ whose endpoints are located on the $(\Pi, \mathcal{D})$-ruler to the left of $t_\mathcal{D}^-$. We can show that there must exist amongst them two distinct intervals ϱ_1 and ϱ_2 such that $T_\Pi^k(\mathfrak{I}_\mathcal{D}) \mid_{\varrho_1}$ is a shift of $T_\Pi^k(\mathfrak{I}_\mathcal{D}) \mid_{\varrho_2}$. Indeed, each interval of the form $[t, t + 2\mathsf{depth}(\Pi)]$, for t on the $(\Pi, \mathcal{D})$-ruler, contains the same number of $(\Pi, \mathcal{D})$-intervals as $[t_\mathcal{D}^-, t_\mathcal{D}^- + 2\mathsf{depth}(\Pi)]$ does, which equals B. In each of these $(\Pi, \mathcal{D})$-intervals there can hold at most 2^A combinations of relational atoms, which gives rise to $(2^A)^B$ different contents of an interval of the form $[t, t + 2\mathsf{depth}(\Pi)]$. Additionally, these intervals can differ depending on the location of $(\Pi, \mathcal{D})$-intervals they contain. By the definition of the $(\Pi, \mathcal{D})$-ruler, there are at most B different layouts of $(\Pi, \mathcal{D})$-intervals contained in an interval of the form $[t, t + 2\mathsf{depth}(\Pi)]$. Hence, in total, there are at most $B \cdot (2^A)^B$ intervals of the form $[t, t + 2\mathsf{depth}(\Pi)]$ with different contents. Thus, in a set of $w = 1 + B \cdot (2^A)^B$ such intervals, there needs to be a pair of distinct intervals ϱ_1 and ϱ_2 such that $T_\Pi^k(\mathfrak{I}_\mathcal{D}) \mid_{\varrho_1}$ is a shift of $T_\Pi^k(\mathfrak{I}_\mathcal{D}) \mid_{\varrho_2}$. Analogously, we can show the existence of required ϱ_3 and ϱ_4 to the right of $t_\mathcal{D}^+$, so the second item from Definition 2 holds. Moreover, since $[\varrho_1^-, \varrho_4^+] \subseteq \varrho$ and $T_\Pi^k(\mathfrak{I}_\mathcal{D}) \mid_\varrho = T_\Pi^{k+1}(\mathfrak{I}_\mathcal{D}) \mid_\varrho$, we obtain that $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ satisfies Π in $[\varrho_1^-, \varrho_4^+]$. Thus, the first item from Definition 2 holds as well, and so, $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ is saturated. □

The bound k_{max} from Theorem 6 is doubly exponential in the size of Π and $\mathcal{D}$, but only exponential in the size of $\mathcal{D}$. Essentially, this bound shows us how much time is needed to perform reasoning, which is consistent with the computational complexity of reasoning in DatalogMTL, namely ExpSpace in combined complexity (Brandt et al. 2017) and PSpace in data complexity (Wałęga et al. 2019).

Algorithm 1: Reasoning via Periods Detection

Input: a bounded program Π, a bounded dataset $\mathcal{D}$, and a relational fact $M@\varrho$
Output: a Boolean value True or False

```
1 D_now := D;
2 loop
3     if D_now ⊨ M@ϱ then return True;
4     (ϱ_left, ϱ_right) := Periods(Π, D, D_now);
5     if ϱ_left ≠ ∅ and ϱ_right ≠ ∅ then
6         if Entails(Π, D, D_now, ϱ_left, ϱ_right, M@ϱ)
              then return True;
7         else return False;
8     D_now := ApplyRules(Π, D_now);
```

Algorithm

The results from the previous section suggest the reasoning procedure in Algorithm 1, which checks if a bounded program Π and a dataset $\mathcal{D}$ entail a relational fact $M@\varrho$.

The algorithm successively applies the rules of Π to the dataset (Line 8) until one of two stopping conditions hold. The first condition (Line 3) detects if the materialisation $\mathcal{D}_{\mathsf{now}}$ constructed thus far entails the input fact $M@\varrho$, in which case the algorithm reports that the input fact is entailed. The second condition checks if $\mathcal{D}_{\mathsf{now}}$ is saturated (i.e., represents a saturated interpretation). To this end, the algorithm computes in Line 4 two non-empty intervals which are periods of $\mathcal{D}_{\mathsf{now}}$ (if $\mathcal{D}_{\mathsf{now}}$ is saturated), or introduces two empty intervals (if $\mathcal{D}_{\mathsf{now}}$ is not saturated). If $\mathcal{D}_{\mathsf{now}}$ is saturated, the algorithm exploits $\mathcal{D}_{\mathsf{now}}$ and its periods to check whether $M@\varrho$ is entailed (Lines 6–7).

We next describe in detail the computations of the algorithm and establish their correctness. In what follows, we fix an arbitrary input Π, $\mathcal{D}$, and $M@\varrho$ to Algorithm 1 and assume that the notions of a saturated interpretation and its periods are relative to Π and $\mathcal{D}$. We also let the projection $\mathcal{D}' \mid_{\varrho'}$ of a dataset $\mathcal{D}'$ over an interval ϱ' be the dataset obtained by intersecting all intervals in facts from $\mathcal{D}'$ with ϱ'.

Rule Application In each iteration of the loop, Algorithm 1 applies the procedure ApplyRules (Line 8), which implements a single round of rule applications to $\mathcal{D}_{\mathsf{now}}$, and thus, mimics an application of the immediate consequence operator T_Π to $\mathfrak{I}_{\mathcal{D}_{\mathsf{now}}}$. Hence, in the beginning of a $(k+1)$st iteration of the loop, the dataset $\mathcal{D}_{\mathsf{now}}$ in Algorithm 1 represents $T_\Pi^k(\mathfrak{I}_\mathcal{D})$. This is a basic component of materialisation in DatalogMTL, already implemented in the literature (Wang et al. 2022, Algorithm 1).

First Stopping Condition In Line 3, Algorithm 1 checks whether the materialisation $\mathcal{D}_{\mathsf{now}}$ constructed so far entails the input fact $M@\varrho$. Since facts in $\mathcal{D}_{\mathsf{now}}$ are stored in a coalesced form, to check if $\mathcal{D}_{\mathsf{now}} \models M@\varrho$, it suffices to scan $\mathcal{D}_{\mathsf{now}}$ and verify whether there is $M@\varrho' \in \mathcal{D}_{\mathsf{now}}$with $\varrho \subseteq \varrho'$.

Second Stopping Condition Algorithm 1 calls the Periods procedure in Line 4 to check, in an iteration $k+1$ of the loop, if $T_\Pi^k(\mathfrak{I}_\mathcal{D})$ is saturated. The procedure searches for intervals $\varrho_1, \ldots, \varrho_4$ satisfying the conditions in Definition 2 and returns the periods of the interpretation $T^k_\Pi(\mathfrak{I}_\mathcal{D})$ if it is saturated or a pair of empty intervals otherwise. To this end, the procedure performs one round of rule applications to $\mathcal{D}_\mathsf{now}$, computing $\mathcal{D}_\mathsf{next}$. Then, it computes an interval ϱ_max as either the empty interval (if $\mathcal{D}_\mathsf{now}$ and $\mathcal{D}_\mathsf{next}$ do not coincide on $[t^-_\mathcal{D}, t^+_\mathcal{D}]$), or otherwise as the maximal interval containing $[t^-_\mathcal{D}, t^+_\mathcal{D}]$ and such that $\mathcal{D}_\mathsf{now}$ and $\mathcal{D}_\mathsf{next}$ coincide on ϱ_max. Interval ϱ_max is next used to search for ϱ_1 and ϱ_2, namely all pairs of intervals contained in ϱ_max, located to the left of $t^-_\mathcal{D}$, with endpoints on the $(\Pi, \mathcal{D})$-ruler, and of lengths $2\mathsf{depth}(\Pi)$, are compared. The first pair of such intervals with the same contents in $\mathcal{D}_\mathsf{now}$ is set as ϱ_1 and ϱ_2; otherwise ϱ_1 and ϱ_2 are empty intervals. In a similar way intervals ϱ_3 and ϱ_4 are computed. Finally, the procedure outputs a pair of intervals $([\varrho_1^-, \varrho_2^-), (\varrho_3^+, \varrho_4^+])$, or $(\emptyset, \emptyset)$ if any of the intervals $\varrho_1, \ldots, \varrho_4$ is empty.

Fact Entailment Checking after Saturation After constructing a saturated dataset $\mathcal{D}_\mathsf{now}$ (representing $T^k_\Pi(\mathfrak{I}_\mathcal{D})$, for some $k \in \mathbb{N}$) with periods $(\varrho_\mathsf{left}, \varrho_\mathsf{right})$, Algorithm 1 calls the procedure Entails (Line 6) to check whether the input fact $M@\varrho$ holds in the unfolding of $T^k_\Pi(\mathfrak{I}_\mathcal{D})$. This can be easily checked by exploiting Definition 3 of unfolding.

From our theoretical results in the previous section it follows that the algorithm is sound and complete:

Theorem 7. *Algorithm 1 outputs* True *if* $(\Pi, \mathcal{D}) \models M@\varrho$*, otherwise it outputs* False*. Moreover, the algorithm terminates after at most* $k_\mathsf{max} + 1$ *(c.f. Theorem 6) iterations of its main loop.*

We conclude this section with an example illustrating the execution of Algorithm 1.

Example 8. *Consider* $\Pi = \{\boxplus_{[0,1]} P \leftarrow P, \boxminus_{[1,1]} Q \leftarrow Q\}$, $\mathcal{D} = \{P@0, Q@1.5\}$*, and a query fact* $M@t = Q@{-4.5}$*. After 5 iterations of the loop in Algorithm 1, the dataset* $\mathcal{D}_\mathsf{now}$ *consists of facts* $P@[0,5]$ *and* $Q@t$*, for all* $t \in \{-3.5, -2.5, -1.5, -0.5, 0.5, 1.5\}$*. The stopping condition from Line 3 does not hold, but the condition in Line 5 does. Indeed,* $\mathcal{D}_\mathsf{now}$ *is saturated and its periods computed in Line 4 are* $\varrho_\mathsf{left} = [-3.5, -2.5)$ *and* $\varrho_\mathsf{right} = (4.0, 4.5]$*. The unfolding of* $\mathcal{D}_\mathsf{now}$ *is the canonical model of* Π *and* $\mathcal{D}$ *depicted in Figure 1. Then, in Line 6, Algorithm 1 can exploit* $\mathcal{D}_\mathsf{now}$*,* ϱ_left*, and* ϱ_right *to detect entailment of any fact. In particular, the input fact* $Q@{-4.5}$ *is entailed, and so, Algorithm 1 returns* True *in Line 6.*

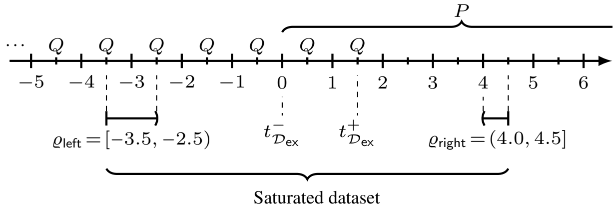

Figure 1: Canonical model $\mathfrak{C}_{\Pi,\mathcal{D}}$ of Π and $\mathcal{D}$

Experimental Evaluation

We have implemented Algorithm 1 and conducted two experiments.[3] The first experiment compares our implementation with MeTeoR (Wang et al. 2022), which we take as the baseline; we did not consider other reasoners, such as Ontop (Kalaycı et al. 2018), as they support only non-recursive programs, where canonical interpretations are always finite. The second experiment tests the scalability of our implementation on datasets of increasing size. All experiments were conducted on a Dell PowerEdge R730 server with 512 GB of RAM and two Intel Xeon E5-2640 2.6 GHz processors running Fedora 33, kernel version 5.8.17. We used as benchmarks a temporal extension of the Lehigh University Benchmark (LUBM) (Wang et al. 2022) and the iTemporal bechmark generator (Bellomarini, Nissl, and Sallinger 2022). The LUBM extension provides a fixed DatalogMTL program which extends the 56 Datalog rules in LUBM with 29 temporal rules involving recursion and mentioning all metric operators in DatalogMTL. We used the iTemporal benchmark generator to construct another recursive DatalogMTL program with 19 rules. Both of the benchmarks allow for the generation of datasets of increasing size.

Comparison with the Baseline. We compared our implementation with MeTeoR, which combines materialisation with automata-based reasoning techniques. For the LUBM benchmark, Figure 2 (left), we considered a dataset $\mathcal{D}$ with 5 million facts and query facts $F@t$, where F is a fixed relational atom (about the predicate $\mathit{FullProfessor}$) and $t \in \{-500, -400, \ldots, 500\}$. Reasoning with $\mathcal{D}$ and the fragment Π of the benchmark's program that is relevant for F is non-trivial and materialisation does not terminate for them.

Facts $F@t$ with $t > 0$ are entailed by Π and $\mathcal{D}$, so such facts are decided by MeTeoR using materialisation only. Thus, the larger t, the larger the number of materialisation steps the baseline performs; as shown in Figure 2, materialisation times increase linearly with t. In our new approach, we observe a similar linear growth up until $t = 300$, where saturation is achieved. This indicates that saturation checks during materialisation come with negligible overhead. Once saturation is reached, our approach does not require further materialisation steps and fact entailment can be decided based on the saturated dataset. Thus, fact entailment running times are roughly identical for all facts with $t \geq 300$, while for MeTeoR they continue to grow with t.

Facts $F@t$, with $t < 0$ are not entailed, hence MeTeoR must resort to automata-based techniques. Due to the optimisations implemented in MeTeoR, automata-based reasoning works well for facts with time points that lie close to the dataset, but performance degrades as t becomes located further away. In contrast, our approach proceeds as in the previous case; it will materialise in linear running time until a saturated dataset is obtained, at which point running times stabilise and become independent of t.

In the case of the iTemporal benchmark, we considered a dataset with 1000 facts and facts $G@t$ with a specific atom

[3]Our implementation can be accessed through the following link: https://github.com/wdimmy/DatalogMTLPeriodicity.

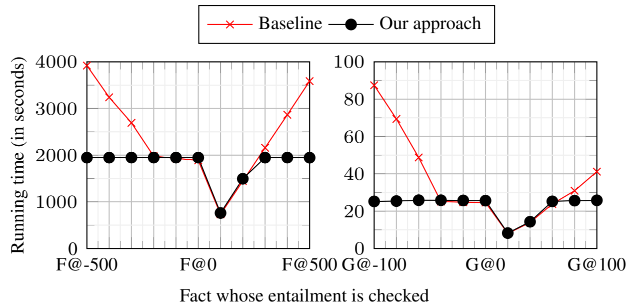

Figure 2: Comparison with the baseline on the LUBM (on the left) and iTemporal (on the right) benchmarks

G and $t \in \{-100, -80, \dots, 100\}$; we observed the same type of behaviour as for LUBM (see Figure 2, right).

Scalability in Data Size. In the second experiment, we analysed how saturation times increase with the size of the input dataset. We considered the same programs as in the first experiment, together with sequences of datasets of increasing size. For the LUBM benchmark, we generated two sequences, each containing 6 datasets. In the first sequence all the intervals of temporal facts are contained within the range $[0, 50]$ and we increase the size of datasets by introducing atoms with new constants. In the second sequence, the number of constants is the same but the number of facts increases; facts occupy increasing ranges of time, namely, $[0, 5 \cdot 10^i]$, for $i \in \{1, \dots, 6\}$. In both sequences, the datasets have 10^3, 10^4, 10^5, 10^6, 10^7, and 10^8 facts, respectively.

As depicted in Figure 3 (left), in both cases running times grow proportionally with data size; this suggests that our approach is scalable with respect to both the number of constants and the temporal range of the dataset. We observe that the running times for the second sequence of facts are around twice smaller than for the first sequence (except the first dataset which is the same in both sequences), so increasing the number of constants seems to have a larger impact on our approach than increasing the temporal domain.

We performed a similar experiment for the iTemporal benchmark (see Figure 3, right), which confirmed our observations from the LUBM experiment.

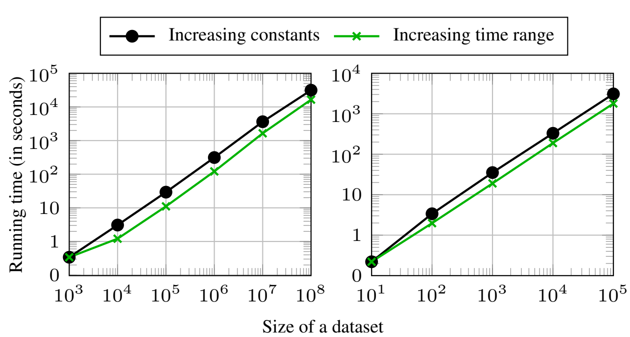

Figure 3: Scalability of our approach tested on the LUBM (on the left) and iTemporal (on the right) benchmarks

Related Work

The complexity of reasoning in DatalogMTL and its fragments has been studied in recent years (Brandt et al. 2018; Wałęga et al. 2019, 2020b), as well as their alternative semantics with favourable computational properties have been proposed (Wałęga et al. 2020a; Ryzhikov, Wałęga, and Zakharyaschev 2019). DatalogMTL has also been extended with stratified negation (Tena Cucala et al. 2021) and with unrestricted negation under stable model semantics (Wałęga et al. 2021), which is related to recent research on metric temporal ASP (Cabalar et al. 2020).

Several techniques have been proposed for reasoning in DatalogMTL. Brzoska (1998) introduced proof systems based on solving linear inequalities, MeTeoR (Wang et al. 2022) is based on an algorithm combining materialisation and automata-based techniques, whereas Ontop implements a reasoning algorithm based on query rewriting that is applicable to non-recursive programs (Kalaycı et al. 2018).

The use of blocking conditions to stop further application of inference rules while retaining completeness is routinely exploited in automated reasoning, especially in the context of modal (Bolander and Blackburn 2007; Schmidt and Waldmann 2015; Areces and Orbe 2015), temporal (Reynolds 2016; Chafik et al. 2021), and description logics (Horrocks and Sattler 1999; Schmidt and Tishkovsky 2011; Glimm, Horrocks, and Motik 2010).

Periodicity of temporal models has been extensively investigated for LTL (Manna and Wolper 1982; Sistla and Clarke 1985) and its fragments (Artale et al. 2013), the temporal logic TPTL for the specification of real-time systems (Alur and Henzinger 1994), the temporalised description logic $\mathcal{EL}$ (Gutiérrez-Basulto, Jung, and Kontchakov 2016), and the metric temporal extension $\text{LTL}^{\text{bin}}_{\mathcal{ALC}}$ of the description logic $\mathcal{ALC}$ (Baader et al. 2020). It has been shown that canonical models in LTL-based rule-languages are periodic and the lengths of offsets and periods in such interpretations are bounded by an exponential function of the number of predicates involved (Artale et al. 2021b). In contrast to our approach, however, no effective method of computing such periods was established.

There have been various other proposals for extending Datalog with temporal constructs (Baudinet, Chomicki, and Wolper 1993; Beck et al. 2015) such as $\text{Datalog}_{1\text{S}}$ and $\text{Datalog}_{n\text{S}}$ (Chomicki and Imieliński 1988; Chomicki 1990, 1995); in particular problems corresponding to finite materialisability have been considered in this setting (Chomicki and Imieliński 1988).

Conclusions and Future Work

We have proposed a novel materialisation-based approach for reasoning in DatalogMTL—a highly expressive extension of Datalog with operators from metric temporal logic. Our algorithm achieves termination by exploiting a specific saturation condition, which ensures that materialisation can be stopped after a bounded number of rounds of rule applications, without compromising completeness of reasoning. We have implemented our approach and conducted experiments supporting its feasibility in practice.

Acknowledgments

This research was supported in whole or in part by the EPSRC projects OASIS (EP/S032347/1), ConCuR (EP/V050869/1) and UK FIRES (EP/S019111/1), the SIRIUS Centre for Scalable Data Access, and Samsung Research UK. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

References

Alur, R.; and Henzinger, T. A. 1994. A Really Temporal Logic. *J. ACM*, 41(1): 181–203.

Areces, C.; and Orbe, E. 2015. Symmetric blocking. *Theor. Comput. Sci.*, 606: 25–41.

Artale, A.; Kontchakov, R.; Kovtunova, A.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2021a. First-order rewritability of ontology-mediated queries in linear temporal logic. *Artif. Intell.*, 299: 103536.

Artale, A.; Kontchakov, R.; Kovtunova, A.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2021b. First-order rewritability of ontology-mediated queries in linear temporal logic. *Artif. Intell.*, 299: 103536.

Artale, A.; Kontchakov, R.; Ryzhikov, V.; and Zakharyaschev, M. 2013. The Complexity of Clausal Fragments of LTL. In *Proc. of LPAR*, 35–52.

Baader, F.; Borgwardt, S.; Koopmann, P.; Ozaki, A.; and Thost, V. 2020. Metric Temporal Description Logics with Interval-Rigid Names. *ACM Trans. Comput. Logic*, 21(4).

Baudinet, M.; Chomicki, J.; and Wolper, P. 1993. Temporal Deductive Databases. In Tansel, A. U.; Clifford, J.; Gadia, S. K.; Segev, A.; and Snodgrass, R. T., eds., *Temporal Databases: Theory, Design, and Implementation*, 294–320. Benjamin/Cummings.

Beck, H.; Dao-Tran, M.; Eiter, T.; and Fink, M. 2015. LARS: A Logic-Based Framework for Analyzing Reasoning over Streams. In *Proc. of AAAI*, 1431–1438.

Bellomarini, L.; Nissl, M.; and Sallinger, E. 2022. iTemporal: An extensible Temporal Benchmark Generator. In *Proc. of ICDE*, 2021–2033.

Bellomarini, L.; Sallinger, E.; and Gottlob, G. 2018. The Vadalog system: Datalog-based reasoning for knowledge graphs. *Proc. VLDB Endow.*, 11(9): 975–987.

Bolander, T.; and Blackburn, P. 2007. Termination for Hybrid Tableaus. *J. Log. Comput.*, 17(3): 517–554.

Brandt, S.; Kalaycı, E. G.; Ryzhikov, V.; Xiao, G.; and Zakharyaschev, M. 2018. Querying Log Data with Metric Temporal Logic. *J. Artif. Intell. Res.*, 62: 829–877.

Brandt, S.; Kontchakov, R.; Ryzhikov, V.; Xiao, G.; and Zakharyaschev, M. 2017. Ontology-based data access with a Horn fragment of metric temporal logic. In *Proc. of AAAI*, 1070–1076.

Bry, F.; Eisinger, N.; Eiter, T.; Furche, T.; Gottlob, G.; Ley, C.; Linse, B.; Pichler, R.; and Wei, F. 2007. Foundations of rule-based query answering. In *Reasoning Web*, 1–153.

Brzoska, C. 1998. Programming in metric temporal logic. *Theor. Comput. Sci.*, 202(1-2): 55–125.

Cabalar, P.; Dieguez, M.; Schaub, T.; and Schuhmann, A. 2020. Towards metric temporal answer set programming. *Theory Pract. Log. Program.*, 20(5): 783–798.

Carral, D.; Dragoste, I.; González, L.; Jacobs, C. J. H.; Krötzsch, M.; and Urbani, J. 2019. VLog: A rule engine for knowledge graphs. In *Proc. of ISWC*, 19–35.

Ceri, S.; Gottlob, G.; and Tanca, L. 1989. What you always wanted to know about Datalog (and never dared to ask). *IEEE Trans. Knowl. Data. Eng.*, 1(1): 146–166.

Chafik, A.; Cheikh-Alili, F.; Condotta, J.-F.; and Varzinczak, I. 2021. A One-Pass Tree-Shaped Tableau for Defeasible LTL. In *Proc. of TIME*, 16:1–16:18.

Chomicki, J. 1990. Polynomial time query processing in temporal deductive databases. In *Proc. of PODS*, 379–391.

Chomicki, J. 1995. Depth-bounded bottom-up evaluation of logic programs. *J. Log. Program.*, 25(1): 1–31.

Chomicki, J.; and Imieliński, T. 1988. Temporal deductive databases and infinite objects. In *Proc. of PODS*, 61–73.

Glimm, B.; Horrocks, I.; and Motik, B. 2010. Optimized Description Logic Reasoning via Core Blocking. In *Proc. of IJCAR*, 457–471.

Gutiérrez-Basulto, V.; Jung, J. C.; and Kontchakov, R. 2016. Temporalized EL Ontologies for Accessing Temporal Data: Complexity of Atomic Queries. In *Proc. of IJCAI*, 1102–1108.

Horrocks, I.; and Sattler, U. 1999. A description logic with transitive and inverse roles and role hierarchies. *J. Log. Comput.*, 9(3): 385–410.

Kalaycı, E. G.; Xiao, G.; Ryzhikov, V.; Kalayci, T. E.; and Calvanese, D. 2018. Ontop-temporal: a tool for ontology-based query answering over temporal data. In *Proc. of CIKM*, 1927–1930.

Kikot, S.; Ryzhikov, V.; Wałęga, P. A.; and Zakharyaschev, M. 2018. On the data complexity of ontology-mediated queries with MTL operators over timed words. In *Proc. of DL*.

Koopmann, P. 2019. Ontology-based query answering for probabilistic temporal data. In *Proc. of AAAI*, 2903–2910.

Koymans, R. 1990. Specifying real-time properties with metric temporal logic. *Real-Time Syst.*, 2(4): 255–299.

Manna, Z.; and Wolper, P. 1982. Synthesis of communicating processes from Temporal Logic specifications. In *Proc. of Workshop on Logics of Programs*, 253–281.

Motik, B.; Nenov, Y.; Piro, R.; Horrocks, I.; and Olteanu, D. 2014. Parallel materialisation of Datalog programs in centralised, main-memory RDF systems. In *Proc. of AAAI*, 129–137.

Reynolds, M. 2016. A New Rule for LTL Tableaux. In *Proc. of GandALF*, 287–301.

Ryzhikov, V.; Wałęga, P. A.; and Zakharyaschev, M. 2019. Data Complexity and Rewritability of Ontology-Mediated Queries in Metric Temporal Logic under the Event-Based Semantics. In *Proc. of IJCAI*, 1851–1857.

Schmidt, R. A.; and Tishkovsky, D. 2011. Automated Synthesis of Tableau Calculi. *Log. Methods Comput. Sci.*, 7(2): 1–32.

Schmidt, R. A.; and Waldmann, U. 2015. Modal Tableau Systems with Blocking and Congruence Closure. In *Proc. of TABLEAUX*, 38–53.

Sistla, A. P.; and Clarke, E. M. 1985. The Complexity of Propositional Linear Temporal Logics. *J. ACM*, 32(3): 733–749.

Tena Cucala, D. J.; Wałęga, P. A.; Cuenca Grau, B.; and Kostylev, E. V. 2021. Stratified Negation in Datalog with Metric Temporal Operators. In *Proc. of AAAI*, 6488–6495.

Wałęga, P. A.; Cuenca Grau, B.; Kaminski, M.; and Kostylev, E. V. 2019. DatalogMTL: Computational complexity and expressive power. In *Proc. of IJCAI*, 1886–1892.

Wałęga, P. A.; Cuenca Grau, B.; Kaminski, M.; and Kostylev, E. V. 2020a. DatalogMTL over the Integer Timeline. In *Proc. of KR*, 768–777.

Wałęga, P. A.; Cuenca Grau, B.; Kaminski, M.; and Kostylev, E. V. 2020b. Tractable Fragments of Datalog with Metric Temporal Operators. In *Proc. of IJCAI*, 1919–1925.

Wałęga, P. A.; Kaminski, M.; and Cuenca Grau, B. 2019. Reasoning over streaming data in metric temporal Datalog. In *Proc. of AAAI*, 3092–3099.

Wałęga, P. A.; Tena Cucala, D. J.; Kostylev, E. V.; and Cuenca Grau, B. 2021. DatalogMTL with Negation Under Stable Models Semantics. In *Proc. of KR*, 609–618.

Wałęga, P. A.; Zawidzki, M.; and Cuenca Grau, B. 2021. Finitely materialisable Datalog programs with metric temporal operators. In *Proc. of KR*, 619–628.

Wang, D.; Hu, P.; Wałęga, P. A.; and Grau, B. C. 2022. MeTeoR: Practical Reasoning in Datalog with Metric Temporal Operators. In *Proc. of AAAI*, 5906–5913.