EPiC Series in Computing
Volume 65, 2019, Pages 28–41
GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence

Ultimately-periodic Interval Model Checking for Temporal Dataset Evaluation

Dario Della Monica[1 2], Angelo Montanari[1], Aniello Murano[3], and Guido Sciavicco[4]

[1] Università di Udine, Udine, Italy
[2] Istituto Nazionale di Alta Matematica "F. Severi" (INdAM), Rome, Italy
[3] Università "Federico II" di Napoli, Napoli, Italy
[4] Università di Ferrara, Ferrara, Italy

Abstract

Temporal dataset evaluation is the problem of establishing to what extent a set of temporal data (histories) complies with a given temporal condition. Checking interval temporal logic formulas against a finite model has been recently proposed, and proved successful, as a tool to solve such a problem. In this paper, we address the problem of checking interval temporal logic specifications, supporting interval length constraints, against infinite, finitely representable models, and we show the applicability of the resulting procedure to the evaluation of incomplete temporal datasets viewed as finite prefixes of ultimately-periodic histories.

1 Introduction

One of the most notable techniques for system verification is *model checking* (MC for short), which allows one to verify the desired properties of a system against a model of its behavior [13].

MC algorithms perform, in a fully automatic way, an (implicit or explicit) exhaustive enumeration of all the states reachable by the system, and either terminate positively, proving that all properties are met, or produce a counterexample, witnessing that some behavior falsifies a property. Model checking is among the foremost applications of logic to computer science, and many breakthroughs have been achieved, effectively bridging the gap between theoretical computer science and computer engineering. It is extensively used in hardware and software verification, and it has been adopted as a standard in static analysis of systems in most university curricula worldwide. Last but not least, it has been successfully applied in new challenging areas such as, for instance, system biology and hybrid systems.

In its usual formulation, point-based temporal logics, such as LTL, CTL, and the like [28, 12, 13], are used to express properties to be verified against systems represented as (finite) Kripke structures. In recent years, an alternative *interval way* to MC has emerged. The most representative interval temporal logic is Halpern and Shoham modal logic of Allen's interval relations [1], aka HS [18]. HS is a multi-modal propositional logic, interpreted over linearly ordered sets, where each accessibility relation is a binary relation between intervals. In [16], the authors introduce and study the problem of checking an HS formula against a single finite model (*finite interval* MC), and its application to temporal dataset evaluation. The key ingredient is the concept of *temporal history*. Let us consider the case of clinical medicine. It is common practice to associate with every patient the collection of all relevant pieces of information about their health, including symptoms, tests, results, and hospitalizations that occurred during the entire observation period. Such a patient's history can be suitably represented by an interval temporal model, where proposition letters express the significant events, not necessarily in a homogeneous way (a symptom can be detected over an interval and not over any of its sub-intervals), [1], and a single temporal history may be checked against desiderata, encoded by means of an HS formula, e.g., *during every application of the therapy, the symptom disappears*. While checking a single model returns a *yes/no* answer, checking a *temporal dataset*, consisting of a statistically significant number of (temporal histories of) patients, allows one to evaluate the set of patients as a whole to rate the conformity degree of the entire dataset, e.g., to determine *how many patients comply with the specification*.

A limitation of such an approach is that it assumes histories (represented by finite interval models) to be complete. In this paper, we propose a way out by defining and solving the *ultimately-periodic interval* MC (UP-MC). UP-MC is the problem of establishing whether a finite interval model that *does not* (yet) satisfy a given requirement, expressed by a formula of interval temporal logic, can be suitably extended to an infinite, ultimately-periodic one that does. As for the specification language, HS fragments which are complete with respect to the class of ultimately-periodic models are the natural candidates. This is the case with MRPNL, the future fragment of *Metric Propositional Neighborhood Logic* (MPNL) [9]. A solution to UP-MC can be decomposed into two main steps: the first one checks whether the finite model in input *is* (the prefix of) an ultimately-periodic one; the second one checks whether its infinite extension *does* satisfy the formula. In general, there can be more than one infinite extension of a finite model fulfilling some periodicity condition. Hence, the possibly different behaviors of different extensions must be evaluated against the formula. To this end, we introduce the notion of *model rating*, which refines that of model checking, and we use it (instead of classic MC) to evaluate single models in the context of temporal dataset evaluation. Resuming the medical scenario, a finite, but incomplete, temporal history may represent a patient with an incurable, but manageable, condition. These patients are not hospitalized forever; on the contrary, they are encouraged to live a normal life once their condition is stabilized. Incomplete histories may be used to check whether patients' conditions stabilized, as it is expected that, once stabilized, their behavior will be periodic, i.e., described by conditions like: *after every application of the therapy, the symptom disappears*.

In the following, we define and solve the UP-MC problem for MRPNL. The solution is achieved in 3 steps: *(i)* we define a sequence of satisfaction relations $\langle \Vdash_i \rangle_{i \in \mathbb{N}}$, which simulate the standard one $\Vdash$ with respect to the incremental unfoldings of the model; *(ii)* we prove such a sequence to converge to a fixpoint i, i.e., $\Vdash_i$ equals $\Vdash_j$, for all $j > i$; and *(iii)* we show that the satisfaction relation at the fixpoint corresponds to the satisfaction relation with respect to the infinite unfolding of the model, thus reducing UP-MC to the finite case. The whole procedure is shown to run in polynomial time.

Structure of the paper. The paper is organized as follows. We start with a short account of related work (Section 2). Then, we provide some background knowledge (Section 3). Next, we give a motivating example and precisely state the addressed problem (Section 4). Finally, we formally define and solve the UP-MC problem (Section 5), before concluding (Section 6).

2 Related work

MC for HS specifications against finite Kripke structures, under the homogeneity assumption (a proposition letter holds on an interval if and only if it holds on all of its sub-intervals), was first investigated in [26]. Such an approach has been further elaborated in [7, 8, 27], where various fragments of HS have been studied. The MC problem for some fragments of HS extended with epistemic modalities was addressed in [22, 23, 24]. The semantic assumptions of the latter differ from those of [26] and follow-up

[1] This is especially the case when dealing with telic statements and temporal aggregations and durations [4, 19].

papers, making it difficult to compare the two approaches.

While the above main stream approaches to interval temporal logic MC are rooted in the verification setting, the problem of checking a single (finite) interval model emerged from different application contexts, and results in a much better computational behavior. Nevertheless, it has received limited attention in the literature. A common feature of the application of (interval) MC to the verification of temporal properties of a reactive system is the encoding of all the runs of the system by a finite-state transition system (a finite Kripke structure), which provides an abstract representation of infinitely many (interval) models. On the contrary, available temporal datasets, e.g., temporal databases [30], usually refer to a given temporal structure. Hence, when used for the evaluation of temporal datasets, interval MC is applied to finite, concrete interval models, and it requires no restrictive assumptions on the labeling of intervals like the homogeneity assumption.

The problem of checking a single model (path) has been investigated in the LTL setting in [25], while that of checking an HS formula against a finite interval model has been dealt with in [16]. As already noticed, interval MC for temporal dataset evaluation behaves computationally much better than interval MC for system verification: a polynomial MC algorithm for full HS has been devised in [16], while the complexity of MC for HS fragments against Kripke structures goes from coNP-complete to non-elementary, depending on the HS fragment under consideration (a summary of known results can be found in [6]).

The expansion of a finite temporal structure into a set of infinite ones, on the basis of the periodicities that are present in it, conforms with, and lifts to the interval frame, the classic approach to MC against finite Kripke structures interpreted as (infinite) sets of infinite LTL or CTL models [13, 17]. Similar approaches that address LTL bounded satisfiability checking exploiting periodicities include [2].

Runtime verification and monitoring techniques [20] are based on ideas somehow related to the ones explored here. When monitoring a property against a system execution, the monitor tries to establish the fulfillment of a full behavior from the analysis of a finite computation prefix. Such techniques also led to the development of various semantics for LTL to deal with finite prefixes, most notably LTL on finite traces [15, 14] and the 4-valued RV-LTL approach [3]. Here, we investigate the possibility of establishing the fulfillment of a periodic behavior from the analysis of a finite prefix of the model. In monitoring, people identify classes of properties for which monitoring can be effective: safety (negatively monitorable) and co-safety (positively monitorable) ones. Analogously, UP-MC is effective for sets of properties that enjoy a *generalized small model property*, that is, if a formula admits an interval model, then it admits an ultimately-periodic one.

As for the application of MC to temporal dataset evaluation, it is well known that query evaluation and constraint checking in relational databases can be naturally expressed as MC problems [31]. The applicability of MC techniques for the retrieval and the verification of temporal data, with a particular focus on medical guidelines, has also been explored in the literature [5, 29]. In general, the analysis of temporal data is focused on the search for *frequent patterns* [21], and when the pattern to be searched for is explicitly given by the user, searching for a pattern can be viewed as a form of MC. Commonly used patterns are purely existential, point-based formulas, possibly including a metric component, and thus UP-MC can be viewed, to some extent, as a generalization of such an approach: the idea of applying interval MC to temporal dataset evaluation was proposed for the first time in [16]; here, we show how to generalize it in order to deal with infinite, ultimately-periodic temporal structures.

3 Preliminaries

Let $\mathbb{D} = \langle D, < \rangle$ be a linearly ordered set. An *interval* over $\mathbb{D}$ is an ordered pair $[x, y]$, where $x, y \in D$ and $x \leq y$. Intervals of the form $[x, x]$ are called *point intervals*; the others are called *strict intervals*. Let $\mathbb{I}(\mathbb{D})$ be the set of all intervals over $\mathbb{D}$. Excluding equality, there are 12 possible relations between two

$\langle A\rangle$	$[x,y]R_A[z,t] \Leftrightarrow y = z$
$\langle L\rangle$	$[x,y]R_L[z,t] \Leftrightarrow y < z$
$\langle B\rangle$	$[x,y]R_B[z,t] \Leftrightarrow x = z, t < y$
$\langle E\rangle$	$[x,y]R_E[z,t] \Leftrightarrow y = t, x < z$
$\langle D\rangle$	$[x,y]R_D[z,t] \Leftrightarrow x < z, t < y$
$\langle O\rangle$	$[x,y]R_O[z,t] \Leftrightarrow x < z < y < t$

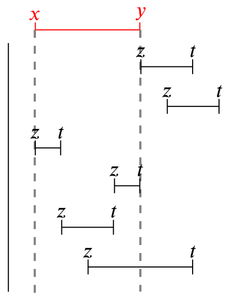

Table 1: Allen's interval relations and corresponding HS modalities.

intervals in a linear order, often called *Allen's relations* [1]: the 6 relations depicted in Table 1 and their inverses.[2] By treating sets of intervals as Kripke structures, with Allen's relations as their accessibility relations, we can associate a modality $\langle X\rangle$ with each Allen relation R_X. For each modality $\langle X\rangle$, its *transpose* $\langle\overline{X}\rangle$ corresponds to the inverse $R_{\overline{X}}$ of R_X, i.e., $R_{\overline{X}} = (R_X)^{-1}$. HS is a multi-modal logic with formulas built over a set $\mathcal{AP}$ of proposition letters, the Boolean connectives $\wedge$ and $\neg$, and the set of modalities for Allen's relations. With each subset of Allen's relations $\{R_{X_1}, \ldots, R_{X_k}\}$, we associate the HS fragment $\mathsf{X}_1\mathsf{X}_2\ldots\mathsf{X}_k$, whose formulas are defined by the grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \langle X_1\rangle\varphi \mid \ldots \mid \langle X_k\rangle\varphi.$$

where $p \in \mathcal{AP}$. The other Boolean connectives, e.g., $\vee$ and $\rightarrow$, and the dual modalities $[X]$ are defined as usual, e.g., $[X]\varphi \equiv \neg\langle X\rangle\neg\varphi$.

An *interval model* is a pair $M = \langle \mathbb{I}(\mathbb{D}), V\rangle$, where $\mathbb{D}$ is a linearly ordered set (*domain* of M) and $V : \mathcal{AP} \rightarrow 2^{\mathbb{I}(\mathbb{D})}$ is a *valuation function*, that assigns to each proposition letter $p \in \mathcal{AP}$ the set of intervals $V(p)$ on which it holds.[3] Let M be an interval model. We denote by $\mathbb{D}_M$ (resp., V_M) its domain (resp., valuation function). We define the *size*, or *cardinality,* of M as the cardinality of $\mathbb{D}_M$, and denote it by N_M (subscripts are omitted when clear from the context). Even if interval temporal logics have been studied in several classes of linearly ordered sets, we assume here every domain $\mathbb{D}$ to be either $\mathbb{N}$ (infinite case) or a prefix of it (finite case, that is, $\mathbb{D} = \{0, 1, \ldots, |\mathbb{D}| - 1\}$). With a little abuse of notation, we sometimes use M to refer to the set of its intervals, i.e., we write $[x, y] \in M$ for $[x, y] \in \mathbb{I}(\mathbb{D})$. The semantics of HS formulas is given in terms of interval models, that is, the *truth* of a formula φ on an interval $[x, y]$ of an interval model M is defined by structural induction on formulas:

$M,[x,y] \Vdash p$ iff $[x,y] \in V(p)$, for $p \in \mathcal{AP}$;
$M,[x,y] \Vdash \neg\psi$ iff $M,[x,y] \nVdash \psi$, i.e., it is not the case that $M,[x,y] \Vdash \psi$;
$M,[x,y] \Vdash \psi\wedge\gamma$ iff $M,[x,y] \Vdash \psi$ and $M,[x,y] \Vdash \gamma$;
$M,[x,y] \Vdash \langle X\rangle\psi$ iff $\exists[z,t]$ s.t $[x,y]R_X[z,t]$ and $M,[z,t] \Vdash \psi$.

We write $M \Vdash \varphi$ for $M, [0, 1] \Vdash \varphi$.

Among the many fragments of HS, a particularly significant one is *Propositional Neighborhood Logic* (PNL, for short). PNL has only two modalities $\langle A\rangle$ and $\langle\overline{A}\rangle$ corresponding to Allen's relations *meets* and *met by*, respectively. Unlike HS and most of its fragments, the satisfiability problem for PNL is decidable [10]. Moreover, when interpreted over discrete linear orders, PNL can be easily extended with metric capabilities. The resulting logic, known as *Metric Propositional Neighborhood Logic* (MPNL), features the modalities $\langle A\rangle$ and $\langle\overline{A}\rangle$, and, for each natural number k, a pre-interpreted modal constant

[2] In the recent literature on interval temporal logic, point intervals are excluded to ensure that Allen's relations are mutually exclusive; however, within our investigation, including point intervals allows us to work with a more expressive logic.

[3] With an abuse of notation we will sometimes use $V : \mathbb{I}(\mathbb{D}) \rightarrow 2^{\mathcal{AP}}$ as a function from intervals to sets of proposition letters, that is, $p \in V([x, y])$ if and only if $[x, y] \in V(p)$, for all $[x, y] \in \mathbb{I}(\mathbb{D})$ and $p \in \mathcal{AP}$.

$\mathsf{len}_{<k}$, called *length constraint* [9]. In [11], it has been shown that MPNL itself is a fragment of HS, as length constraints can be defined, in polynomial space, by using modalities $\langle B\rangle$ and $\langle E\rangle$.

In this paper, we focus on the future fragment of MPNL, called *Metric Right Propositional Neighborhood Logic* (MRPNL), obtained from MPNL by removing modality $\langle\overline{A}\rangle$, whose formulas are generated by the following grammar:

$$\varphi ::= \mathsf{len}_{<k} \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle A\rangle\varphi,$$

where $p \in \mathcal{AP}$ and $k \in \mathbb{N}$. The other Boolean connectives and logical constants, as well as the universal modality $[A]$, are defined in the standard way. Hereafter, given an MRPNL formula φ, we denote by ξ_φ (or, simply, ξ) the greatest constant k that occurs in a length constraint in φ. MRPNL is interpreted on discrete interval models by enriching HS truth relation with the following clause:

$$M, [x,y] \Vdash \mathsf{len}_{<k} \text{ iff } y - x < k.$$

Despite its simplicity, MRPNL turns out to be a very expressive HS fragment. First of all, being Boolean-complete, it allows one to define all other length constraints, i.e., $\mathsf{len}_{\sim k}$ with $\sim \in \{\leq, =, \neq, >, \geq\}$. Building on them, it is possible to mimic the (strict) *sometimes in the future* LTL modality:

$$\langle A\rangle(\mathsf{len}_{>0} \wedge \langle A\rangle(\mathsf{len}_{=0} \wedge p)).$$

Length constraints also allow one to define a metric version of the *until* (resp., *since*) LTL modality. As an example, the condition: '*p is true at a point in the future at distance k from the current interval and, until that point, q is true (pointwise)*' can be expressed as follows:

$$\langle A\rangle(\mathsf{len}_{=k} \wedge \langle A\rangle(\mathsf{len}_{=0} \wedge p)) \wedge [A](\mathsf{len}_{<k} \rightarrow \langle A\rangle(\mathsf{len}_{=0} \wedge q)).$$

Finally, MRPNL can be used to constrain interval length and to introduce metric versions of basic interval relations. More precisely, we can constrain the length of the intervals over which a given property holds to be at least (resp., at most, exactly) k. As an example, the following formula constrains p to hold only over intervals of length l, with $k \leq l \leq k'$:

$$[G](p \rightarrow \mathsf{len}_{\geq k} \wedge \mathsf{len}_{\leq k'}),$$

where the *globally-in-the-future* modality $[G]$ is defined as:

$$[G]p \equiv p \wedge [A]p \wedge [A][A]p.$$

By exploiting this capability, metric versions of most Allen's relations can be simulated (to some extent).

The finite interval MC problem, as defined in [16], consists in deciding, given a finite interval model M and an HS formula φ, whether or not $M \Vdash \varphi$. A number of meaningful problems involving temporal data, ranging from query answering to constraint checking and rule evaluation, can be expressed as finite interval MC problems. However, other significant ones require the ability of dealing with infinite streams. The MC machinery turns out to be useful to cope with them as well, provided that the relevant infinite interval models are finitely representable. In general, it is not possible to restrict the attention to finitely presentable infinite models of HS formulas. Luckily, MRPNL (in fact, MPNL as well) satisfies a natural generalization of the (finite) small model property: a formula has an infinite model if and only if it has an ultimately-periodic (and, thus, finitely representable) one [9]. This makes MPRNL an ideal candidate for the UP-MC problem.

4 Problem definition and applications

Decidability of the satisfiability problem for MPNL (and thus MRPNL) over (infinite) discrete linear orders has been proved by exploiting a finite pseudo-model property based on ultimately-periodic infinite interval models [9]. The very same class of infinite interval models is at the basis of the notion of UP-MC.

4.1 Problem definition

Let $\mathcal{AP}$ be a set of proposition letters. In what follows, we define the notion of ultimately-periodic model, which is central to this work. Intuitively, an ultimately-periodic model presents recurrent behaviors, i.e., it is characterized by the presence of periods (after a finite, possibly empty, prefix) whose corresponding intervals satisfy the same proposition letters. Notice that, in order to recognize a periodic behavior, 3 matching periods are needed to detect 2 repetitions of intervals in between consecutive periods.

Definition 1 (Ultimately-periodic model)**.** *For $z \in \mathbb{N}$, a z*-ultimately-periodic model *(z*-model *for short) is a pair $\mathcal{M} = \langle M, P \rangle$, where M is a finite interval model of cardinality N and $P \in \{0\} \cup \{n \in \mathbb{N} \mid z \leq n \leq \lfloor \frac{N}{3} \rfloor\}$, such that, for $L_{\mathcal{M}} = (N-1) - 3 \cdot P$ (we omit the subscript $_{\mathcal{M}}$ when it is clear from the context), it holds that: (i) $V([x, y]) = V([x+P, y+P])$ for all x, y such that $L < x < y \leq N-1-P$, and (ii) $V([x, y]) = V([x, y+P])$ for all x, y such that $L < y \leq N-1-P$.*

P is called the *period* of $\mathcal{M}$. Clearly, for every finite interval model M and $z \in \mathbb{N}$, $\langle M, 0 \rangle$ is a z-model. Therefore, the notion of z-model generalizes that of finite interval model.

Definition 2. *Let $\mathcal{M} = \langle M, P \rangle$ be a z-model. For $k \in \mathbb{N}$, the* k-unfolding *of $\mathcal{M}$, denoted by $\mathcal{M}^k$, is inductively defined as:*

- $\mathcal{M}^0 = M = \langle \mathbb{I}(\mathbb{D}), V \rangle$
- $\mathcal{M}^1 = \langle \mathbb{I}(\mathbb{D}^1), V^1 \rangle$, *where*
  - $\mathbb{D}^1 = \mathbb{D} \cup \{|\mathbb{D}|, |\mathbb{D}|+1, \ldots, |\mathbb{D}|+P-1\} = \{0, \ldots, |\mathbb{D}|+P-1\}$;
  - $V^1([x,y]) = \begin{cases} V([x,y]) & \textit{if } [x,y] \in \mathbb{I}(\mathbb{D}), \\ V([x-P, y-P]) & \textit{if } x > L+P, y \geq |\mathbb{D}|, \\ V([x, y-P]) & \textit{if } x \leq L+P, y \geq |\mathbb{D}|. \end{cases}$
- $\mathcal{M}^k = \langle \mathcal{M}^{k-1}, P \rangle^1$ $\quad (k > 1)$

Let $\mathcal{M}$ be a z-model. We define $\mathcal{M}^\omega = \lim_{k \mapsto \infty} \mathcal{M}^k$ (∞-*unfolding*). $\mathcal{M}^\omega$ is obtained by applying infinitely many times the 1-unfolding to $\mathcal{M}$. The next result can be easily checked.

Proposition 1. *For every z-model $\mathcal{M} = \langle M, P \rangle$, $p \in \mathcal{AP}$, $i \in \mathbb{N}$, and $[x, y] \in M$, with $y > L$, we have that $\mathcal{M}^\omega, [x, y] \Vdash p$ if and only if $\mathcal{M}^\omega, [x, y + i \cdot P] \Vdash p$. Moreover, if $x > L$, then, for all $j \in \mathbb{N}$ with $x + i \cdot P \leq y + j \cdot P$, then $\mathcal{M}^\omega, [x, y] \Vdash p$ if and only if $\mathcal{M}^\omega, [x + i \cdot P, y + j \cdot P] \Vdash p$.*

Recall that we denote by ξ_φ (or, simply, ξ) the greatest constant k that occurs in a length constraint in a formula φ. We are now ready to formalize the UP-MC problem.

Definition 3 (UP-MC)**.** *Let M be a finite interval model and φ be a formula of MRPNL. The* UP-MC *problem for MRPNL is the problem of enumerating every P such that $\mathcal{M} = \langle M, P \rangle$ is a ξ-model and $\mathcal{M}^\omega \Vdash \varphi$.*

Solving UP-MC thus means establishing to what extent a finite model that does not (yet) satisfy a formula can be extended to an infinite, ultimately-periodic one that satisfies it.

4.2 An example of application

Representing temporal information with interval temporal logic may be useful in many contexts. Let us consider the case of temporal histories of patients in a *medical* domain. Temporal data about patients, which is typically presented in the form of time series, can be abstracted in several ways. As an example, running a fever can be represented by a proposition letter lo (lower than 40 degrees) or hi (higher than or equal to 40 degrees), holding over specific time intervals. Now, consider the case in which a patient experiences low fever in an interval $[x, y]$, say a day, and during just one hour of that day, that is, over the interval $[w, z]$ strictly contained in $[x, y]$, he/she has an episode of high fever. A natural modelling choice is to represent such a situation by labelling the interval $[x, y]$ with lo and its sub-interval $[w, z]$ with hi. Notice that the choice of representing such a scenario by means of three consecutive intervals $[x, w], [w, z]$, and $[z, y]$ respectively labeled with lo, hi, and lo (this would be the case with a point-based formalism or an interval-based one under the homogeneity assumption) is rather artificial, and it somehow hides potentially relevant pieces of information such as: "*the patient presented low fever during the entire day, except for a brief episode of high fever*". In addition, a non-homogeneous labelling of intervals allows one to make temporal aggregations. In our example, lo (resp., hi) can be used to state that the patient is experiencing a fever that, on average, is low (resp., high) over the considered interval, disregarding, as before, possible short periods with no fever.

In [16], the authors introduce and study the problem of checking an HS formula against a single finite model, and its application to temporal dataset evaluation. A limitation of such an approach is that it assumes histories to be complete. Consider, for instance, a database containing the temporal histories of a group of patients that are undergoing a treatment to manage an incurable disease. Such a treatment consists of administrating a certain therapy, and it is supposed to eliminate a certain symptom for a period of time. By making use of the interval representation, one can preprocess the temporal data of the patients to let the relevant pieces of information to emerge. Suppose now that the prescribed treatment takes the form of the administration of a therapy th, which consists of some procedure that, in order to be effective, must last at least 15 minutes (time units) and at most 30 ones. Moreover, let us assume that a minimum of 24 hours (1440 time units) and a maximum of 36 ones (2160 time units) must separate two consecutive executions of the therapy. Finally, it is expected that for at least 12 hours (720 time units) after an administration a certain symptom sy (which may be non-homogeneous, as the low fever of the above examples) disappears. These requirements can be formalized as follows:

$$
\begin{aligned}
&[G](th \to (\mathsf{len}_{\geq 15} \wedge \mathsf{len}_{\leq 30}))\wedge\\
&[G](th \to (\langle A\rangle(\mathsf{len}_{\leq 2160} \wedge \langle A\rangle th) \wedge [A](\mathsf{len}_{<1440} \to \neg\langle A\rangle th)))\wedge\\
&[G](th \to [A](\mathsf{len}_{\leq 720} \to \neg\langle A\rangle sy))
\end{aligned}
$$

Now, patients with recurring, but manageable, conditions are not hospitalized forever. To improve their quality of life, they are encouraged to conduct a normal life style once their conditions are under control. We may thus expect the database not to contain complete temporal histories of patients, but only parts of them. In such a situation, finite interval MC is not applicable; on the contrary, UP-MC allows one to check that the recorded history of a patient shows a clear enough trend to conclude that his/her condition is under control.

Model rating. We conclude this section by formalizing the *model rating* problem and discussing its relationships with model checking.

First, we observe that while MC is a decision problem, UP-MC is an enumerating one. However, UP-MC generalizes MC as we can decide if $M \Vdash \varphi$ by verifying whether the z-model $\mathcal{M} = \langle M, 0\rangle$ (for any z) is such that $\mathcal{M}^\omega \Vdash \varphi$. For a z-model $\mathcal{M} = \langle M, P\rangle$, we can easily compute the greatest $n \geq 3$, say it $n_\mathcal{M}$, for which a generalized formulation of Definition 1, where $L' = N - 1 - n \cdot P$ replaces L, holds, that is: *(i')* $V([x, y]) = V(x + P, y + P)$ for all x, y such that $L' < x < y \leq N - 1 - P$, and *(ii')* $V([x, y]) = V(x, y + P)$ for all x, y such that $L' < y \leq N - 1 - P$. Definition 1 recognizes as ultimately-periodic a model that features at least 3 consecutive occurrences of an ending period. However, a measure of reliability of the prediction (the ultimately-periodic interval model), obtained by unfolding it *ad infinitum*, is given by how many consecutive occurrences of such a period are contained in the original finite model: the higher is $n_\mathcal{M}$, the more dependable is $\mathcal{M}^\omega$. We call $n_\mathcal{M}$ (we omit the subscript when it is clear from the context) the *strength* of $\mathcal{M}$, and we call the value $N - 1 - n \cdot P$ its *prefix*.

As another metric to rate a finite interval model M with respect to an MRPNL formula φ, one

$i = 0$	$\mathcal{M}, [x, y] \Vdash_i \varphi$	iff	$M, [x, y] \Vdash \varphi$	
$i > 0$	$\mathcal{M}, [x, y] \Vdash_i p$	iff	$[x, y] \in V(p)$	$(p \in \mathcal{AP})$
	$\mathcal{M}, [x, y] \Vdash_i \mathsf{len}_{<h}$	iff	$y - x < h$	$(h \in \mathbb{N})$
	$\mathcal{M}, [x, y] \Vdash_i \neg\psi$	iff	$\mathcal{M}, [x, y] \not\Vdash_i \psi$	
	$\mathcal{M}, [x, y] \Vdash_i \psi \wedge \gamma$	iff	$\mathcal{M}, [x, y] \Vdash_i \psi$ and $\mathcal{M}, [x, y] \Vdash_i \gamma$	
	$\mathcal{M}, [x, y] \Vdash_i \langle A\rangle\psi$	iff	either $y \leq L+P$ and $\exists z$ s.t. $y \leq z < N$ and $\mathcal{M}, [y, z] \Vdash_i \psi$ or $y > L+P$ and $\exists z$ s.t. $y-P \leq z < N$ and $\mathcal{M}, [y-P, z] \Vdash_{i-1} \psi$	

Table 2: Definition of the new semantic relations $\Vdash_i$ $(i \in \mathbb{N})$.

may consider the rate of ξ-models (with different periods) whose ∞-unfolding satisfies φ over the total number of legitimate ξ-models; we call such a metric the *variability* of M with respect to φ, denoted by $v_{M,\varphi}$. Formally, let $k_{M,\varphi}$ be the cardinality of the set $\{P \in \mathbb{N} \mid \langle M, P\rangle$ is a ξ-model$\}$ and let $k'_{M,\varphi}$ be the cardinality of the set $\{P \in \mathbb{N} \mid \langle M, P\rangle$ is a ξ-model such that $\langle M, P\rangle \Vdash \varphi\}$: the *variability* of M with respect to φ is defined as $v_{M,\varphi} = \frac{k'_{M,\varphi}}{k_{M,\varphi}}$.

Combining strength and variability, which can be easily computed while solving UP-MC, allows one to rate a finite interval model with respect to a formula (*model rating* problem). The usefulness of such a rate depends entirely on the intended application domain and scenario.

5 Solving UP-MC

In this section, we solve the UP-MC problem. Let φ be an MRPNL formula. We denote the set of its sub-formulas by $Sub(\varphi)$. In order to solve the UP-MC problem for MRPNL, given an MRPNL formula φ and a ξ-model $\mathcal{M} = \langle M, P\rangle$, we define an infinite sequence of alternative semantics $\Vdash_i$ (with $i \in \mathbb{N}$) such that the truth of a formula in $Sub(\varphi)$ over ($[x, y]$ in) $\mathcal{M}$, according to the $\Vdash_i$-semantics, amounts to its truth over ($[x, y]$ in) $\mathcal{M}^i$, according to the standard $\Vdash$-semantics.

Definition 4 ($\Vdash_i$-semantics)**.** *Let φ be a formula of MRPNL, $\mathcal{M} = \langle M, P\rangle$ be a ξ-model, and $[x, y] \in M$. We define the infinite sequence of truth relations $\langle \Vdash_i \rangle_{i\in\mathbb{N}}$, where, for all $i \in \mathbb{N}$, $\mathcal{M}, [x, y] \Vdash_i \varphi$ is inductively defined, on the structure of φ, according to the clauses in Table 2.*

As in the case of the standard semantics $\Vdash$ (see Section 3), other Boolean constants, connectives, and length constraints, as well as universal modalities, can be treated as defined.

The next result, which follows from Proposition 1, is used to prove the correctness of the new semantics sequence $\langle \Vdash_i \rangle_{i\in\mathbb{N}}$ (Lemma 1).

Corollary 1. *Let φ be a formula of MRPNL, $\mathcal{M} = \langle M, P\rangle$ be a ξ-model, $i \in \mathbb{N}$, and $[x, y] \in M$. Then: (i) if $y > L$ and $y - x > \xi$, then $\mathcal{M}^\omega, [x, y] \Vdash \varphi$ if and only if $\mathcal{M}^\omega, [x, y + i \cdot P] \Vdash \varphi$, and (ii) if $x > L$, then for all $j \in \mathbb{N}$ with $x + i \cdot P \leq y + j \cdot P$ $\mathcal{M}^\omega, [x, y] \Vdash \varphi$ if and only if $\mathcal{M}^\omega, [x + i \cdot P, y + j \cdot P] \Vdash \varphi$.*

Proof. If $\varphi \in \mathcal{AP}$, then the claim directly follows from Proposition 1. The other cases are easily dealt with by induction on the structure of φ. □

Let $\mathcal{F}$ be a set of formulas in the language of MRPNL, with $\xi = \max\{\xi_\phi \mid \phi \in \mathcal{F}\}$, and $\mathcal{M} = \langle M, P\rangle$ be a ξ-model. For $i, j \in \mathbb{N}$, we write $i \equiv_{\mathcal{M},\mathcal{F}} j$ as an abbreviation for $\mathcal{M}, [x, y] \Vdash_i \phi$ if and only if $\mathcal{M}, [x, y] \Vdash_j \phi$, for all $\phi \in \mathcal{F}$ and $[x, y] \in M$. We omit the model when it is clear from the context, thus writing, e.g., $i \equiv_{\mathcal{F}} j$ for $i \equiv_{\mathcal{M},\mathcal{F}} j$. The next lemma states the correctness of the new semantics sequence $\langle \Vdash_i \rangle_{i\in\mathbb{N}}$. It can be proved by induction on the structure of ϕ, the only non-trivial case being the one for the modality $\langle A\rangle$.

Lemma 1 (soundness of $\Vdash_i$-semantics)**.** *Let φ be a formula in the language of MRPNL, $\mathcal{M} = \langle M, P\rangle$ be a ξ-model, and i be such that $i \equiv_{Sub(\varphi)} i{+}1$. Then: $\mathcal{M}, [x, y] \Vdash_i \phi$ if and only if $\mathcal{M}^\omega, [x, y] \Vdash \phi$, for all $[x, y] \in M$ and $\phi \in Sub(\varphi)$.*

Proof. We proceed by induction on the structure of ϕ; the non-trivial case concerns the modality $\langle A\rangle$.

- If $\phi \in \mathcal{AP}$, then we have:

$$\mathcal{M}, [x, y] \Vdash_i \phi \overset{(\Vdash_i\text{-sem.})}{\Leftrightarrow} [x, y] \in V(\phi) \overset{(\Vdash\text{-sem.})}{\Leftrightarrow} \mathcal{M}^\omega, [x, y] \Vdash \phi.$$

- If $\phi = \neg\phi_1$, then we have:

$$\mathcal{M}, [x, y] \Vdash_i \phi \overset{(\Vdash_i\text{-sem.})}{\Leftrightarrow} \mathcal{M}, [x, y] \nVdash_i \phi_1 \overset{(\text{ind. hp.})}{\Leftrightarrow} \mathcal{M}^\omega, [x, y] \nVdash \phi_1 \overset{(\Vdash\text{-sem.})}{\Leftrightarrow} \mathcal{M}^\omega, [x, y] \Vdash \phi.$$

- If $\phi = \phi_1 \wedge \phi_2$, then we have:

$$\begin{aligned}\mathcal{M}, [x, y] \Vdash_i \phi &\overset{(\Vdash_i\text{-sem.})}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_i \phi_1 \text{ and } \mathcal{M}, [x, y] \Vdash_i \phi_2\\ &\overset{(\text{ind. hp.})}{\Leftrightarrow} \mathcal{M}^\omega, [x, y] \Vdash \phi_1 \text{ and } \mathcal{M}^\omega, [x, y] \Vdash \phi_2\\ &\overset{(\Vdash\text{-sem.})}{\Leftrightarrow} \mathcal{M}^\omega, [x, y] \Vdash \phi.\end{aligned}$$

- If $\phi = \langle A\rangle\phi_1$, then we first prove the two following preliminary results:

(†) if $y \leq L + P$, then

$$\mathcal{M}^\omega, [x, y] \Vdash \phi \Leftrightarrow \exists z. y \leq z < N \text{ and } \mathcal{M}^\omega, [y, z] \Vdash \phi_1;$$

(‡) if $L + P < y \leq N - 1 = L + 3 \cdot P$, then

$$\mathcal{M}^\omega, [x, y] \Vdash \phi \Leftrightarrow \exists z. y - P \leq z < N \text{ and } \mathcal{M}^\omega, [y - P, z] \Vdash \phi_1.$$

To prove (†), we proceed as follows:

$$\begin{aligned}\mathcal{M}^\omega, [x, y] \Vdash \phi &\overset{(\Vdash\text{-sem.})}{\Leftrightarrow} \exists z. z \geq y \text{ and } \mathcal{M}^\omega, [y, z] \Vdash \phi_1\\ &\overset{(\text{Cor. 1}(i))}{\Leftrightarrow} \exists z'. y \leq z' < N \text{ and } \mathcal{M}^\omega, [y, z'] \Vdash \phi_1.\end{aligned}$$

Conversely, to prove (‡), we first observe that $L+P < y \leq L+3\cdot P$ implies $L < y-P \leq L+2\cdot P$), and, then:

$$\begin{aligned}\mathcal{M}^\omega, [x, y] \Vdash \phi &\overset{(\Vdash\text{-sem.})}{\Leftrightarrow} \exists z. z \geq y \text{ and } \mathcal{M}^\omega, [y, z] \Vdash \phi_1\\ &\overset{(\text{Cor. 1}(ii))}{\Leftrightarrow} \exists z'. y - P \leq z' < N \text{ and } \mathcal{M}^\omega, [y - P, z'] \Vdash \phi_1.\end{aligned}$$

Now, we use an inner induction on i. If $i = 0$, then we distinguish two cases.

  - if $y \leq L + P$, then we have:

$$\begin{aligned}\mathcal{M}^\omega, [x, y] \Vdash \phi &\overset{(\dagger)}{\Leftrightarrow} \exists z. y \leq z \leq N - 1 \text{ and } \mathcal{M}^\omega, [y, z] \Vdash \phi_1\\ &\overset{(\text{ind. hp.})}{\Leftrightarrow} \exists z. y \leq z \leq N - 1 \text{ and } \mathcal{M}, [y, z] \Vdash_i \phi_1\\ &\overset{(\Vdash_0\text{-sem.})}{\Leftrightarrow} \exists z. y \leq z \leq N - 1 \text{ and } M, [y, z] \Vdash \phi_1\\ &\overset{(\Vdash\text{-sem.})}{\Leftrightarrow} M, [x, y] \Vdash \phi\\ &\overset{(\Vdash_0\text{-sem.})}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_i \phi;\end{aligned}$$

– If $L + P < y \le N - 1 = L + 3 \cdot P$, then $L < y - P \le L + 2 \cdot P$ and we have:

$$\begin{aligned}
\mathcal{M}^{\omega}, [x, y] \Vdash \phi &\overset{(\ddagger)}{\Leftrightarrow} \exists z. y - P \le z < N \text{ and } \mathcal{M}^{\omega}, [y - P, z] \Vdash \phi_1 \\
&\overset{\text{(ind. hp.)}}{\Leftrightarrow} \exists z. y - P \le z < N \text{ and } \mathcal{M}, [y - P, z] \Vdash_i \phi_1 \\
&\overset{(\Vdash_{i+1}\text{-sem.})}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_{i+1} \phi \\
&\overset{(i \equiv_{Sub(\varphi)} i+1)}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_i \phi.
\end{aligned}$$

Assume now $i > 0$. Again, we distinguish two cases.

– if $y \le L + P$, then we have:

$$\begin{aligned}
\mathcal{M}, [x, y] \Vdash_i \phi &\overset{(\Vdash_i\text{-sem.})}{\Leftrightarrow} \exists z. y \le z \le N - 1 \text{ and } \mathcal{M}, [y, z] \Vdash_i \phi_1 \\
&\overset{\text{(ind. hp.)}}{\Leftrightarrow} \exists z. y \le z \le N - 1 \text{ and } \mathcal{M}^{\omega}, [y, z] \Vdash \phi_1 \\
&\overset{(\dagger)}{\Leftrightarrow} \mathcal{M}^{\omega}, [x, y] \Vdash \phi;
\end{aligned}$$

– If $L + P < y \le N - 1 = L + 3 \cdot P$, then we have:

$$\begin{aligned}
\mathcal{M}, [x, y] \Vdash_i \phi &\overset{(\Vdash_i\text{-sem.})}{\Leftrightarrow} \exists z. y - P \le z < N \text{ and } \mathcal{M}, [y - P, z] \Vdash_{i-1} \phi_1 \\
&\overset{\text{(ind. hp.)}}{\Leftrightarrow} \exists z. y - P \le z < N \text{ and } \mathcal{M}^{\omega}, [y - P, z] \Vdash \phi_1 \\
&\overset{(\ddagger)}{\Leftrightarrow} \mathcal{M}^{\omega}, [x, y] \Vdash \phi.
\end{aligned}$$

□

For a formula φ in the language of MRPNL, we denote by $md(\varphi)$ its *modal depth*, defined inductively as usual:

$$md(\varphi) = \begin{cases} 0 & \text{if } \varphi \in \mathcal{AP} \cup \{\mathsf{len}_{<h}\}_{h \in \mathbb{N}} \\ md(\phi) & \text{if } \varphi = \neg\phi \\ \max\{md(\phi_1), md(\phi_2)\} & \text{if } \varphi = \phi_1 \wedge \phi_2 \\ 1 + md(\phi_1) & \text{if } \varphi = \langle A \rangle \phi_1. \end{cases}$$

The next lemma is used to prove termination and complexity of the UP-MC algorithm (Algorithm 1).

Lemma 2 (fixpoint of $\Vdash_i$-semantics)**.** *Let φ be a formula of MRPNL and $\mathcal{M} = \langle M, P \rangle$ be a ξ-model. For all $\phi \in Sub(\varphi)$, with modal depth d, and $[x, y] \in M$, we have: $\mathcal{M}, [x, y] \Vdash_{d+1} \phi$ if and only if $\mathcal{M}, [x, y] \Vdash_i \phi$, for all $i > d$.*

Proof. Let $i > d$, where $d = md(\phi)$. The proof proceeds by induction on the structure of ϕ.

- If $\phi \in \mathcal{AP}$, then we have:

$$\mathcal{M}, [x, y] \Vdash_{d+1} \phi \overset{(\Vdash_{d+1}\text{-sem.})}{\Leftrightarrow} [x, y] \in V(\phi) \overset{(\Vdash_i\text{-sem.})}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_i \phi$$

- If $\phi = \neg\phi_1$, then we have:

$$\mathcal{M}, [x, y] \Vdash_{d+1} \phi \overset{(\Vdash_{d+1}\text{-sem.})}{\Leftrightarrow} \mathcal{M}, [x, y] \nVdash_{d+1} \phi_1 \overset{\text{(ind. hp.)}}{\Leftrightarrow} \mathcal{M}, [x, y] \nVdash_i \phi_1 \overset{\Vdash_i\text{-sem.}}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_i \phi$$

- If $\phi = \phi_1 \wedge \phi_2$, then we have:

$$\begin{aligned}
\mathcal{M}, [x, y] \Vdash_{d+1} \phi &\overset{(\Vdash_{d+1}\text{-sem.})}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_{d+1} \phi_1 \text{ and } \mathcal{M}, [x, y] \Vdash_{d+1} \phi_2 \\
&\overset{\text{(ind. hp.)}}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_i \phi_1 \text{ and } \mathcal{M}, [x, y] \Vdash_i \phi_2 \\
&\overset{\Vdash_i\text{-sem.}}{\Leftrightarrow} \mathcal{M}, [x, y] \Vdash_i \phi
\end{aligned}$$

Algorithm 1 UP-MC.

```
function UP-MC(M, φ)                                              ▷ M = ⟨𝕀(𝔻), V⟩
    let ξ be the greatest constant that occurs in a length constraint in φ
    for each P ∈ {0} ∪ {n ∈ ℕ | ξ ≤ n ≤ ⌊N/3⌋} do
        if 𝓜 = ⟨M, P⟩ is a ξ-model then
            if FIXPOINT-CHECK(M, P, φ) then out ← out ∪ {P}
    return out
function FIXPOINT-CHECK(M, P, φ)                                  ▷ ℓ_i : Sub(φ) → 2^𝕀(𝔻)
    ℓ_0 ← ⊩_0-CHECK(M, P, φ)
    ℓ_1 ← ⊩_1-CHECK(M, P, φ)
    i ← 1
    while NOT EQUAL(ℓ_{i−1}, ℓ_i) do
        i ← i + 1
        ℓ_i ← ⊩_i-CHECK(M, P, φ)
    return [0, 1] ∈ ℓ_i(φ)
function EQUAL(ℓ_{i−1}, ℓ_i)
    for ψ ∈ Sub(φ) do
        if ℓ_{i−1}(ψ) ≠ ℓ_i(ψ) then return False
    return True
function ⊩_i-CHECK(M, P, φ)
    if i = 0 then
        return FINITE-INTERVAL-MODEL-CHECKING(M, φ)               ▷ see Algorithm 2 from [16]
    else
        for each ψ ∈ Sub(φ) do                                    ▷ ordered by size
            if ψ ∈ 𝒜𝒫 then ℓ_i(ψ) ← V(ψ)
            else if ψ = len_{<h} then ℓ_i(ψ) ← {[x, y] ∈ 𝕀(𝔻) | y − x < h}
            else if ψ = ¬τ then ℓ_i(ψ) ← 𝕀(𝔻) \ ℓ_i(τ)
            else if ψ = τ ∧ γ then ℓ_i(ψ) ← ℓ_i(τ) ∩ ℓ_i(γ)
            else if ψ = ⟨A⟩τ then
                for [y, z] ∈ 𝕀(𝔻) with y ≤ L + P do
                    if [y, z] ∈ ℓ_i(τ) then
                        for x ≤ y do ℓ_i(ψ) ← ℓ_i(ψ) ∪ {[x, y]}
                    if [y, z] ∈ ℓ_{i−1}(τ) and y + P > L + P then
                        for x ≤ y + P do ℓ_i(ψ) ← ℓ_i(ψ) ∪ {[x, y + P]}
                for [y, z] ∈ 𝕀(𝔻) with L + P < y ≤ L + 2 · P do
                    if [y, z] ∈ ℓ_{i−1}(τ) then
                        for x ≤ y + P do ℓ_i(ψ) ← ℓ_i(ψ) ∪ {[x, y + P]}
    return ℓ_i
```

- If $\phi = \langle A \rangle \phi_1$, then we have:

$$\begin{aligned}
\mathcal{M}, [x, y] \Vdash_{d+1} \phi \overset{(\Vdash_{d+1}\text{-sem.})}{\Leftrightarrow} & \text{ either } y \le L + P, \exists z. y \le z < N, \text{ and } \mathcal{M}, [y, z] \Vdash_{d+1} \phi_1 \\
& \quad \text{or } y > L + P, \exists z. y - P \le z < N, \text{ and } \mathcal{M}, [y - P, z] \Vdash_{d} \phi_1 \\
\overset{(\text{ind. hp.})}{\Leftrightarrow} & \text{ either } y \le L + P, \exists z. y \le z < N, \text{ and } \mathcal{M}, [y, z] \Vdash_{i} \phi_1 \\
& \quad \text{or } y > L + P, \exists z. y - P \le z < N, \text{ and } \mathcal{M}, [y - P, z] \Vdash_{i\text{-}1} \phi_1 \\
\overset{\Vdash_i\text{-sem.}}{\Leftrightarrow} & \ \mathcal{M}, [x, y] \Vdash_i \phi.
\end{aligned}$$ □

A solution to the UP-MC problem is given in Algorithm 1. The next theorem states correctness and complexity of the UP-MC procedure. Some considerations are in order as far as complexity is concerned. As a matter of fact, the complexity is polynomial in, inter alia, the size of the model in input. However, as observed in [16] in the context of finite interval MC, there is a class of instances $\mathcal{I} = \langle M, \varphi \rangle$ of the UP-MC problem, called *sparse* instances (see [16] for a formal definition of sparse instances of MC), which are represented in logarithmic space in the size of the model N. Therefore, for these instances we have that N is exponential in the size of the input, and thus the proposed algorithm runs in exponential time (since it is polynomial in N). A way out to this complexity blowup is given in [16], where it is shown that every sparse instance can be turned into a non-sparse, equivalent (with respect to the finite interval MC problem) one. We conjecture an analogous result can be obtained for the infinite case as well, but proving it goes beyond the purpose of this paper. Therefore, the following complexity result assumes instances to be non-sparse.

Theorem 1. *Algorithm 1 correctly solves the* UP-MC *problem. Moreover, if the input instance* $\langle M, \varphi \rangle$ *is represented in (at least) linear space in the size* N *of* M *(i.e., the instance is non-sparse), then the algorithm runs in polynomial time.*

Proof. It is easy to see that function $\Vdash_i$-CHECK correctly computes function $\ell_i : Sub(\varphi) \rightarrow 2^{\mathbb{I}(\mathbb{D})}$, which labels intervals in $\mathbb{I}(\mathbb{D})$ with formulas in $Sub(\varphi)$ that are true over it according to $\Vdash_i$. Notice that for $i = 0$ the function calls function FINITE-INTERVAL-MODEL-CHECKING, i.e., the MC algorithm for finite interval models from [16]. This suits the definition of $\Vdash_0$ in Table 2. Then, the main function UP-MC checks for periods P for which $\mathcal{M} = \langle M, P \rangle$ is a ξ-model (according to Definition 1), and for each of them, function FIXPOINT-CHECK is called, to check whether $\mathcal{M}^\omega \Vdash \varphi$.

The function FIXPOINT-CHECK computes the labelings $\ell_i(\cdot)$ for all i until the fixpoint, whose existence is guaranteed by Lemma 2, is reached. When the fixpoint i is reached, the function returns the truth value of φ over $[0, 1]$ in $\mathcal{M}$, according to $\Vdash_i$. By Lemma 1, this amounts to deciding whether $\mathcal{M}^\omega, [0, 1] \Vdash \varphi$, and thus the algorithm is correct. Termination follows from Lemma 2, which guarantees that the *while* loop in function FIXPOINT-CHECK terminates after at most $md(\varphi)$ steps.

Finally, it is easy to see that the algorithm is polynomial in N and in the size of φ: the loop at line 3 is executed at most $\lfloor \frac{N}{3} \rfloor$ times; checking (line 4) that a candidate ξ-model is indeed a ξ-model (i.e., verifying that the properties from Definition 1 hold) can be done in polynomial time in N; as already observed, the *while* loop in function FIXPOINT-CHECK is executed at most $md(\varphi)$ times, by Lemma 2; finally, functions EQUAL, FINITE-INTERVAL-MODEL-CHECKING (see [16]), and $\Vdash_i$-CHECK ($i \in \mathbb{N}$) run in polynomial time in N and in the size of φ. □

6 Conclusions

Finite interval MC has been recently proposed as a tool for temporal dataset evaluation [16]. However, checking only finite histories turns out to be rather restrictive. To overcome such a limitation, in this paper we explored the possibility of interpreting finite interval models as partial models, e.g., incomplete temporal histories. Partial models are then completed by exploiting the regularities of the models themselves in order to produce infinite ultimately-periodic ones, which can be checked against properties expressed in HS or in a fragment of it.

We formalized the UP-MC problem for the HS fragment MRPNL, and we showed how to solve it in polynomial time (for non-sparse instances).

Moreover, we argued that the proposed technique suggests an approach more general than model checking (we called it *model rating*), which makes it possible to evaluate, rather than simply check, a model against a formula, according to suitable measures of goodness related to model periodicity. Model rating can thus be used to evaluate single models in the context of temporal dataset evaluation.

A natural development of the work is the integration of the UP-MC procedure into a model checking tool. We are also thinking of lifting the proposed technique to MPNL, which can be obtained from MRPNL by adding a past modality [9].

Acknowledgments

The authors acknowledge the partial support from the Italian INdAM-GNCS project *Metodi formali per tecniche di verifica combinata*. In addition, Dario Della Monica also acknowledges the support from a Marie Curie INdAM-COFUND-2012 Outgoing Fellowship.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Comm. of the ACM*, 26(11):832–843, 1983.

[2] L. Baresi, M. M. Pourhashem, and M. Rossi. Efficient scalable verification of LTL specifications. In *Proc. of the 37th ICSE*, pages 711–721, 2015.

[3] A. Bauer, M. Leucker, and C. Schallhart. Comparing LTL semantics for runtime verification. *J. Log. Comput.*, 20(3):651–674, 2010.

[4] M. H. Böhlen, J. Gamper, and C. S. Jensen. How would you like to aggregate your temporal data? In *Proc. of the 13th TIME*, pages 121–136. IEEE Computer Society, 2006.

[5] A. Bottrighi, L. Giordano, G. Molino, S. Montani, P. Terenziani, and M. Torchio. Adopting model checking techniques for clinical guidelines verification. *Art. Int. Med.*, 48(1):1–19, 2010.

[6] L. Bozzelli, A. Molinari, A. Montanari, and A. Peron. On the complexity of model checking for syntactically maximal fragments of the interval temporal logic HS with regular expressions. In *Proc. of the 8th GandALF*, volume 256 of *EPTCS*, pages 31–45, 2017.

[7] L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Model checking for fragments of the interval temporal logic HS at the low levels of the polynomial time hierarchy. *Inf. Comp.*, 262(2):241–264, 2018.

[8] L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Which fragments of the interval temporal logic HS are tractable in model checking? *Theor. Comp. Sci.*, 764C:125–144, 2019.

[9] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. Metric propositional neighborhood logics on natural numbers. *Soft. and Sys. Mod.*, 12(2):245–264, 2013.

[10] D. Bresolin, D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco. Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. *Theor. Comp. Sci.*, 560:269–291, 2014.

[11] D. Bresolin, P. Sala, and G. Sciavicco. On begins, meets, and before. *Int. J. Found. Comp. Sci.*, 23(3):559–583, 2012.

[12] E. M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proc. of the 1981 Workshop on Logics of Programs*, volume 131 of *LNCS*, pages 52–71, 1982.

[13] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 2002.

[14] G. De Giacomo, R. De Masellis, M. Grasso, F. M. Maggi, and M. Montali. Monitoring business metaconstraints based on LTL and LDL for finite traces. In *Proc. of the 12th BPM*, pages 1–17, 2014.

[15] G. De Giacomo and M. Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *Proc. of the 23rd IJCAI*, pages 854–860, 2013.

[16] D. Della Monica, D. de Frutos-Escrig, A. Montanari, A. Murano, and G. Sciavicco. Evaluation of temporal datasets via interval temporal logic model checking. In *Proc. of the 24th TIME*, pages 11:1–11:18, 2017.

[17] E.A. Emerson. Temporal and modal logic. In *Handbook of theoretical computer science*, volume B: formal models and semantics, pages 995–1072. Elsevier MIT Press, 1990.

[18] J.Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *J. ACM*, 38(4):935–962, 1991.

[19] V. Khatri, S. Ram, R.T. Snodgrass, and P. Terenziani. Capturing telic/atelic temporal data semantics: Generalizing conventional conceptual models. *IEEE Trans. Knowl. Data Eng*, 26(3):528–548, 2014.

[20] M. Leucker and C. Schallhart. A brief account of runtime verification. *J. Log. Algebr. Program.*, 78(5):293–303, 2009.

[21] W. Lin and M.A. Orgun. Temporal data mining using hidden periodicity analysis. In *Proc. of the 12th ISMIS*, pages 49–58, 2000.

[22] A. Lomuscio and J. Michaliszyn. An epistemic Halpern-Shoham logic. In *Proc. of the 23rd IJCAI*, pages 1010–1016, 2013.

[23] A. Lomuscio and J. Michaliszyn. Decidability of model checking multi-agent systems against a class of EHS specifications. In *Proc. of the 21st ECAI*, pages 543–548, 2014.

[24] A. Lomuscio and J. Michaliszyn. Model checking multi-agent systems against epistemic HS specifications with regular expressions. In *Proc. of the 15th KR*, pages 298–308, 2016.

[25] N. Markey and P. Schnoebelen. Model checking a path. In *Proc. of the 14th CONCUR*, pages 251–265. Springer, 2003.

[26] A. Molinari, A. Montanari, A. Murano, G. Perelli, and A. Peron. Checking interval properties of computations. *Acta Inf.*, 53(6-8):587–619, 2016.

[27] A. Molinari, A. Montanari, and A. Peron. Model checking for fragments of Halpern and Shoham's interval temporal logic based on track representatives. *Inf. and Comp.*, 259(3):412–443, 2018.

[28] A. Pnueli. The temporal logic of programs. In *Proc. of the 18th FOCS*, pages 46–57, 1977.

[29] E. Quintarelli. *Model-Checking Based Data Retrieval: an Application to Semistructured and Temporal Data*. Springer, 2004.

[30] A. U. Tansel, J. Clifford, S. K. Gadia, S. Jajodia, A. Segev, and R. T. Snodgrass, editors. *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings, 1993.

[31] M. Y. Vardi. Model checking for database theoreticians. In *Proc. of the 10th ICDT*, pages 1–16, 2005.