

Right Propositional Neighborhood Logic over Natural Numbers with Integer Constraints for Interval Lengths

Davide Bresolin¹, Valentin Goranko², Angelo Montanari³, Guido Sciavicco⁴

¹University of Verona, Verona, Italy, ²Technical University of Denmark, Lyngby, Denmark,

³University of Udine, Udine, Italy, ⁴University of Murcia, Murcia, Spain

¹davide.bresolin@univr.it, ²vfgo@imm.dtu.dk, ³angelo.montanari@dimi.uniud.it, ⁴guido@um.es *

Abstract

Interval temporal logics are based on interval structures over linearly (or partially) ordered domains, where time intervals, rather than time instants, are the primitive ontological entities. In this paper we introduce and study Right Propositional Neighborhood Logic over natural numbers with integer constraints for interval lengths, which is a propositional interval temporal logic featuring a modality for the ‘right neighborhood’ relation between intervals and explicit integer constraints for interval lengths. We prove that it has the bounded model property with respect to ultimately periodic models and is therefore decidable. In addition, we provide an EXPSpace procedure for satisfiability checking and we prove EXPSpace-hardness by a reduction from the exponential corridor tiling problem.

1 Introduction

Interval temporal logics are based on temporal structures over linearly (or partially) ordered domains, where time intervals, rather than time instants, are the primitive ontological entities. A systematic analysis of the variety of relations between intervals on linear orders was first accomplished by Allen [1], who explored the use of interval reasoning in systems for time management and planning. The problem of representing and reasoning about time intervals arises naturally in various other fields of computer science, artificial intelligence, and temporal databases, such as theories of action and change, natural language processing, and constraint satisfaction problems. In particular, temporal logics with interval-based semantics have been proposed as a use-

ful formalism for the specification and verification of hardware [24] and of real-time systems [15].

Interval temporal logics feature modal operators that correspond to various relations between intervals. A special role is played by the thirteen different binary relations on linear orders, known as Allen’s relations [1]. In [17], Halpern and Shoham introduce a modal logic for reasoning about interval structures, nowadays known as HS, with modal operators corresponding to Allen’s interval relations. This logic turns out to be undecidable under very weak assumptions on the class of interval structures: undecidability holds for any class of interval structures over linear orders that contains at least one linear order with an infinite ascending (or descending) chain, thus including the natural time flows \mathbb{N} , \mathbb{Z} , \mathbb{Q} , and \mathbb{R} [17].

The complex and generally bad computational behavior of interval temporal logics is essentially due to the fact that formulas are evaluated over pairs of points and translate into binary relations. In a few cases, decidability has been recovered by imposing severe restrictions on the set of modalities and/or on the interval-based semantics, which essentially reduce the logic to a point-based one. For a long time, such sweeping undecidability results have discouraged attempts for practical applications and further research on interval logics. A renewed interest in the area has been recently stimulated by the discovery of some interesting decidable fragments of HS [7, 8, 9, 12, 13]. As an effect, the identification of expressive enough, yet decidable, fragments of HS has become one of the major topics of the current research agenda in interval temporal logics. In [5], positive and negative known results have been summarized, and the main proof techniques so far exploited in both directions have been presented; a very recent contribution to this classification, showing that most HS fragments involving the modal operators for the relations of left or right overlap of intervals are undecidable, has been presented in [6].

One of the most important results concerning fragments of HS is the decidability of the satisfiability problem for the

*Davide Bresolin was partly supported by the EU project FP7-ICT-223844 CON4COORD. Valentin Goranko and Guido Sciavicco were partly supported by the Spanish / South African project HS2008-0006; Guido Sciavicco was also co-financed by the Spanish project TIN2006-15460-C04-01.

so-called Propositional Neighborhood Logic (PNL) and, hence, of its single-modality fragment Right Propositional Neighborhood Logic (RPNL) [8, 9, 10]. Here we focus our attention on RPNL.

RPNL can be defined as the minimal fragment of HS containing the modal operator $\langle A \rangle$, which is the modal counterpart of the Allen’s relation *meets*. It has been shown to be decidable when interpreted over various interesting classes of linear and partial orders, including the class of all linearly ordered sets and the set of natural numbers. Moreover, sound, complete, and terminating tableau-based deduction methods have been developed for RPNL over all of them [11, 12, 13].

In search for more expressive, yet decidable, extensions of RPNL, we propose adding a simple *metric* dimension to the language of RPNL¹.

There are several previous contributions related to this proposal, which we would like to discuss briefly.

In [20], Kautz and Ladkin explore the extension of Allen’s Interval Algebra with a suitable notion of distance, but there is no interval logic involved there yet. The most important reference to a metric interval-based logical system is Duration Calculus (DC) [18, 28] – an interval logic for real-time systems, originally developed by Chaochen, Ravn and Hoare [15]. Some fragments of DC are decidable [3, 14, 16, 28], but this is often due to some semantic restrictions, such as the *locality* principle, which essentially reduce the interval system to a point-based one. In particular, the Hybrid Duration Calculus (HDC) of Bolander et al. [3], which is interpreted over natural numbers, can express all Allen’s interval relations as well as length constraints, and it is decidable. The major difference with the interval logics considered by us is that the atomic propositions, and therefore all formulae, in HDC are not interpreted in arbitrary sets of intervals, i.e., its semantics is eventually reducible to a point-based one.

In [22, 23], Montanari et al. deal with the problem of enriching the language of modal logic with a notion of distance between worlds. They take into consideration various extensions with increasing expressive power, and devise sound and complete axiomatic systems for the resulting languages. In [25], Owakine and Worrell study the so-called Metric Temporal Logic (MTL), which is a real-time extension of Linear Temporal Logic. MTL is a widely-studied metric variant of Linear Time Logic (LTL) which augments the modalities of LTL with timing constraints (see [21]). Owakine and Worrell survey results about the complexity of the satisfiability and model checking problems for fragments of MTL with respect to different semantic models. Other metric extensions of temporal logics have been

¹The use of interval logics extending RPNL with interval length constraints has been advocated in [26, 27] in the context of quality checking of medical guidelines.

proposed in the literature, including Alur and Henzinger’s Timed Propositional Temporal Logic (TPTL) [2] and Hirshfeld and Rabinovich’s Quantitative Monadic Logic of Order (QMLO) [19].

In this paper, we consider a metric extension of RPNL over natural numbers, called RPNL+INT. The language of RPNL+INT is obtained by adding to the language of RPNL an infinite set of special atomic propositions for length constraints (ℓCk), where ℓ refers to the length of the current interval, $C \in \{<, \leq, =, \geq, >\}$, and k is a natural number. We prove that the resulting logic is decidable and we analyze its complexity.

The paper is structured as follows. In Section 2, we recall the basics of RPNL, and we introduce syntax and semantics of RPNL+INT. In Section 3, we show how the classical gas-burner example can be dealt with in RPNL+INT. In Section 4, we compare the expressive power of RPNL and RPNL+INT, proving that the latter is strictly more expressive than the former. In Section 5, we prove the decidability of the satisfiability problem for RPNL+INT. In Section 6, we systematically analyze the complexity of the logic. In the conclusions, we provide an assessment of the work and we briefly outline future research directions.

2 RPNL and RPNL+INT

In this section, we provide syntax and semantics of the *Right Propositional Neighborhood Logic* (RPNL, for short) interpreted over natural numbers, and of its *metric* extension.

2.1 RPNL over Natural Numbers

The language of RPNL consists of a set \mathcal{AP} of atomic propositions, the propositional connectives \neg, \vee , and the modal operator $\langle A \rangle$. One can read $\langle A \rangle$ as ‘adjacent’; in terms of Allen’s relations between intervals, it corresponds to the relation *meets*. The other propositional connectives, as well as the logical constants \top (*true*) and \perp (*false*), and the dual modal operator $[A]$, are defined as usual. The *formulas* of RPNL denoted by φ, ψ, \dots , are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle A \rangle\varphi.$$

Hereafter, we denote by D the set of natural numbers or any prefix of it.

Given a linearly ordered domain $\mathbb{D} = \langle D, < \rangle$, an *interval* over \mathbb{D} is an ordered pair $[i, j]$ such that $i < j$. Note that logics of temporal neighborhood have also been studied in the *non-strict* version including point-intervals, that is, intervals of the form $[i, i]$, and a modal constant to capture

them. In this paper, we assume the so-called *strict* semantics, where point intervals are not admitted; however, the achieved results can be adapted to the non-strict case. An *interval structure* is a pair $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$, where $\mathbb{I}(\mathbb{D})$ is the set of all intervals over \mathbb{D} .

The semantics of RPNL is given in terms of *models* of the form $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), V \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$ is an interval structure and $V : \mathcal{AP} \rightarrow 2^{\mathbb{I}(\mathbb{D})}$ is a valuation function assigning a set of intervals to every atomic proposition. We recursively define the satisfiability relation \Vdash as follows:

- $M, [i, j] \Vdash p$ iff $p \in V([i, j])$ for any $p \in \mathcal{AP}$;
- $M, [i, j] \Vdash \neg\psi$ iff it is not the case that $M, [i, j] \Vdash \psi$;
- $M, [i, j] \Vdash \psi \vee \tau$ iff $M, [i, j] \Vdash \psi$ or $M, [i, j] \Vdash \tau$;
- $M, [i, j] \Vdash \langle A \rangle \psi$ iff there exists $r > j$ such that $M, [j, r] \Vdash \psi$.

The logic RPNL over natural numbers has been studied in [13], where the decidability of its satisfiability problem, as well as the existence of a sound, complete, and terminating tableau-based decision method for it have been shown. Thus, the results presented in this work extend and strengthen those in [13].

2.2 RPNL+INT over Natural Numbers

The logic RPNL+INT is an extension of RPNL with an infinite set of atomic propositions that express integer constraints on the length of the interval over which they are interpreted. First, we define the function $\delta : \mathbb{N}^2 \rightarrow \mathbb{N}$ as the standard distance on \mathbb{N} , that is, $\delta(i, j) = |j - i|$. Then, we add to the language of RPNL the infinite set of special atomic propositions $(\ell C k)$, where ℓ refers to the length of the current interval, $C \in \{<, \leq, =, \geq, >\}$, and k is a natural number. We call such atomic propositions *length constraints*. Clearly, we have that $k > 1$ when C is ' $<$ ', and that $k > 0$ when C is ' \leq '. The set of formulas of the resulting language is defined by the grammar:

$$\varphi ::= (\ell C k) \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A \rangle \varphi.$$

The other propositional connectives and the dual operator $[A]$ are defined as usual.

The semantics of RPNL+INT is defined as follows. Given a model $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), V \rangle$ and an interval $[i, j]$:

- $M, [i, j] \Vdash (\ell C k)$ iff $\delta(i, j) C k$.

It is immediate to observe that if we allow the negation to appear freely in the formulas, then only one of the above constraints is sufficient to define the rest of them. In general, this definition may generate less succinct formulas, as in the

case of $\ell \leq k$, which is defined by $\ell = 1 \vee \ell = 2 \vee \dots \vee \ell = k$. From now on, we assume to have only the length constraints $\ell > k$ in the language, and consider the others as definable.

Notice that RPNL+INT cannot 'look' to the left of the current interval and that it can only look inside of it as much as referring to its length. This means that we can reduce the satisfiability problem for any formula of RPNL+INT to its satisfiability over some initial interval $[0, e]$. From now on, we shall say that φ is *satisfiable* if and only if it is satisfiable over $[0, e]$, for some $e > 0$.

3 RPNL+INT at work: the gas-burner example

In the area of interval temporal logics, one of the main problem is balancing between expressive power and computational complexity. As recalled in the introduction, the previous attempts of adding metric features to temporal and modal logics have focused on point-based systems, with the notable exception of Duration Calculus (DC). A classical illustration of the use of DC to specify embedded real-time systems is the so-called *gas-burner example*. To demonstrate the expressive power of RPNL+INT from the practical point of view, we use that example to show that the logic RPNL+INT can express some meaningful properties of the gas-burner.

Consider a model of a gas-burner, where the following situations are of interest. The propositional letter *Gas* is used to indicate that the gas is flowing; thus, in our model, $M, [i, j] \Vdash Gas$ means that the gas is flowing in the interval $[i, j]$. Similarly, *Flame* means that the gas is burning, and, consequently,

$$[G](Leak \leftrightarrow Gas \wedge \neg Flame),$$

where the *universal modality* $[G]$ (capturing all intervals in the model except the initial interval and the intervals which begin anywhere in between the initial interval) is defined as:

$$[G]\varphi = \varphi \wedge [A]\varphi \wedge [A][A]\varphi,$$

expresses the fact that *Leak* holds over an interval if and only if the gas is flowing and not burning over that interval.

The safety requirement '*It is never true that the gas is leaking for more than k time units*' can be expressed as:

$$[G](\neg((\ell > k) \wedge Leak)).$$

Various other requirements of the gas-burner example can be captured in a similar way.

Besides its ability of encoding the requirements of the classical gas-burner example, RPNL+INT can be successfully used to deal with many other scenarios. As an example, it is not difficult to show that the logic RPNL (and, thus, RPNL+INT) is expressive enough to embed Linear Time Logic with the ‘‘sometimes in the future’’ operator F (LTL[F]). The presence of length constraints allow one to encode some metric forms of *until*. As an example, the condition: ‘*If the property p is true over some interval of length k starting at the right endpoint of the current one, then q is true on every shorter-than- k interval (starting at the right endpoint of the current one)*’ can be expressed as follows:

$$\langle A \rangle((\ell = k) \wedge p) \rightarrow [A]((\ell < k) \rightarrow q).$$

4 RPNL+INT is more expressive than RPNL

Here we show that, as expected, RPNL+INT is strictly more expressive than RPNL, in the sense that while there exists an obvious translation τ from RPNL-formulas to RPNL+INT-formulas (since the latter language is a conservative extension of the former) such that for every model M , every interval $[i, j]$ in M , and every formula φ of RPNL, $M, [i, j] \models \varphi$ if and only if $M, [i, d] \models \tau(\varphi)$, there is no effective translation from RPNL+INT-formulas to RPNL-formulas such that for every model M , every interval $[i, j]$ in M , and every formula φ of RPNL+INT, $M, [i, j] \models \varphi$ if and only if $M, [i, d] \models \tau(\varphi)$.

To prove our claim, we use a standard argument, based on the notion of bisimulation. Consider $\mathbb{D} = \mathbb{D}' = \mathbb{N}$, and two models $M = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), V \rangle$, $M' = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}'), V' \rangle$ (for the sake of readability, we denote the elements of the domain of M by $1, 2, \dots, i, \dots$ and those of the domain of M' by $1', 2', \dots, i', \dots$, but the domain of the two models is in fact the same). Suppose that p is false everywhere except on the intervals $[1, 3]$, $[1, 4]$, and $[1', 3']$, and consider the relation $R \subseteq \mathbb{I}(\mathbb{D}) \times \mathbb{I}(\mathbb{D}')$ defined by the union of the following sets: $\{([i, j], [i', j']) \mid [i, j] \neq [1, 4]\}$, $\{([1, 4], [1', 3']), ([1, 2], [1', 4'])\}$, $\{([i, j], [(i-1)', (j-1)']) \mid i \geq 4\}$, and $\{([i, j], [(i+2)', (j+2)']) \mid i \geq 2\}$. We show that R is a bisimulation (with respect to the language of RPNL) between M and M' . First, R -related intervals must satisfy the same propositional letters. The only p -intervals in M are $[1, 3]$ and $[1, 4]$, which are both R -related only to the p -interval $[1', 3']$, and, in turn, the latter is R -related only to the p -intervals $[1, 3]$ and $[1, 4]$. Second, we must show that the *back* and *forth* conditions hold for the relation induced by $\langle A \rangle$. Let $R_{\langle A \rangle}$ be such a relation. We show that for all x, y , if $[x, y]$ is R -related to some $[z', t']$ and $R_{\langle A \rangle}$ -related to some $[y, r]$, then there exists some s' , such that $[y, r]$ is R -related to $[t', s']$ and $[z', t']$ is $R_{\langle A \rangle}$ -related to $[t', s']$ (*forth* condition) and vice versa (*back* condition). As for the forth condition, by definition of R , we

distinguish the following cases:

- if $t = y$, then there are two possibilities. If $[y, r] \neq [1, 4]$, then we have that $[y, r]$ is R -related to $[y', r']$ and $[z', t']$ is $R_{\langle A \rangle}$ -related to $[y', r']$. Otherwise, if $[y, r] = [1, 4]$, then we have that $[1, 4]$ is R -related to $[1', 3']$ and $[z', t']$ is $R_{\langle A \rangle}$ -related to $[1', 3']$;
- if $t = y - 1$, then, by definition of R , we have that $y \geq 4$. Thus, we have that $[y, r]$ and $[t', (r-1)']$ are R -related, and $[z', t']$ and $[t', (r-1)']$ are $R_{\langle A \rangle}$ -related;
- if $t = y + 2$, then, by definition of R , we have that $y \geq 2$. Thus, we have that $[y, r]$ and $[t', (r+2)']$ are R -related, and $[z', t']$ and $[t', (r+2)']$ are $R_{\langle A \rangle}$ -related.

As for the back condition, we must show that, for all x, y , if $[x, y]$ is R -related to some $[z', t']$ and $[z', t']$ is $R_{\langle A \rangle}$ -related to some $[t', s']$, then there exists some r such that $[y, r]$ is R -related to $[t', s']$ and that $[x, y]$ is $R_{\langle A \rangle}$ -related to $[y, r]$. Again, we must distinguish the following cases:

- if $t = y$, then there are two possibilities. If $[t', s'] \neq [1', 4']$, then we have that $[t, s]$ is R -related to $[t', s']$ and $[x, y]$ is $R_{\langle A \rangle}$ -related to $[t, s]$. Otherwise, if $[t', s'] = [1', 4']$, then we have that $[1', 4']$ is R -related to $[1, 2]$ and $[x, y]$ is $R_{\langle A \rangle}$ -related to $[1, 2]$;
- if $t = y - 1$, then, by definition of R , we have that $y \geq 4$. Thus, we have that $[y, s+1]$ and $[t', s']$ are R -related, and $[x, y]$ and $[y, s+1]$ are $R_{\langle A \rangle}$ -related;
- if $t = y + 2$, then, by definition of R , we have that $y \geq 2$. Thus, we have that $[y, s-2]$ and $[t', s']$ are R -related, and $[x, y]$ and $[y, s-2]$ are $R_{\langle A \rangle}$ -related.

This proves that R is a bisimulation between M and M' with respect to RPNL. Consider now the intervals $[1, 4]$ and $[1', 3']$. They are bisimilar, but $M, [1, 4] \models (\ell > 2)$, and $M', [1, 3] \not\models (\ell > 2)$. It immediately follows that $(\ell > 2)$ cannot be expressed in the language of RPNL, and thus the claim. Since RPNL+INT is a conservative extension of RPNL, we have the following result.

Theorem 4.1. *The logic RPNL+INT is strictly more expressive than RPNL.*

5 Decidability of RPNL+INT

In this section we use a model-theoretic argument to show that the satisfiability problem for RPNL+INT has the bounded model property with respect to finitely presentable ultimately periodic models and it is therefore decidable.

Let φ be any RPNL+INT-formula and let \mathcal{AP} be the set propositional letters of the language.

Definition 5.1. The *closure* of φ is the set $CL(\varphi)$ of all subformulae of φ and their negations, after omitting all $\neg\neg\psi$. The set of *temporal requests* from $CL(\varphi)$ is the set $TF(\varphi)$ of all temporal formulae in $CL(\varphi)$, that is, $TF(\varphi) = \{\langle A \rangle \psi, [A] \psi \in CL(\varphi)\}$.

One can easily prove that $|CL(\varphi)| \leq 2|\varphi|$ and $|TF(\varphi)| \leq 2(|\varphi| - 1)$.

Definition 5.2. A φ -atom is a set $A \subseteq CL(\varphi)$ such that for every $\psi \in CL(\varphi)$, $\psi \in A$ iff $\neg\psi \notin A$, and for every $\psi_1 \vee \psi_2 \in CL(\varphi)$, $\psi_1 \vee \psi_2 \in A$ iff $\psi_1 \in A$ or $\psi_2 \in A$.

We denote the set of all φ -atoms by A_φ . Clearly, $|A_\varphi| \leq 2^{|\varphi|}$. We now introduce a suitable labeling of interval structures based on φ -atoms.

Definition 5.3. A (φ -labelled) *interval structure* (LIS for short) is a pair $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$, where $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$ is an interval structure and $\mathcal{L} : \mathbb{I}(\mathbb{D}) \rightarrow A_\varphi$ is a *labeling function*, such that for every pair of neighboring intervals $[i, j], [j, r] \in \mathbb{I}(\mathbb{D})$, if $[A] \psi \in \mathcal{L}([i, j])$, then $\psi \in \mathcal{L}([j, r])$.

Note that every interval model M is a LIS, where the labeling function is the valuation function: $\psi \in \mathcal{L}([i, j])$ if and only if $M, [i, j] \models \psi$. Thus, labelled interval structures can be thought of as *quasi-models* for φ , in which the truth of formulae that contain neither $\langle A \rangle$ nor length constraints is determined by the labeling (due to the definitions of φ -atom and LIS). To obtain a model we must also guarantee that the truth of the other formulae is in accordance with the labeling. To this end, we introduce the following notion.

Definition 5.4. A φ -labelled interval structure $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ is *fulfilling* if and only if:

- for every length constraint $(\ell > k) \in CL(\varphi)$ and every interval $[i, j] \in \mathbb{I}(\mathbb{D})$, $(\ell > k) \in \mathcal{L}([i, j])$ iff $\delta(i, j) > k$;
- for every every temporal formula $\langle A \rangle \psi \in TF(\varphi)$ and every interval $[i, j] \in \mathbb{I}(\mathbb{D})$, if $\langle A \rangle \psi \in \mathcal{L}([i, j])$, then there exists $r > j$ such that $\psi \in \mathcal{L}([j, r])$.

The notion of fulfilling LIS is analogous to a Hintikka structure. Clearly, every interval model is a fulfilling LIS. Conversely, every fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ can be transformed into a model $M(\mathbf{L})$ by defining the valuation in accordance with the labeling. Then, one can prove that for every $\psi \in CL(\varphi)$ and interval $[i, j] \in \mathbb{I}(\mathbb{D})$, $\psi \in \mathcal{L}([i, j])$ iff $M(\mathbf{L}), [i, j] \models \psi$ by a routine induction on ψ . Thus, we obtain the following.

Lemma 5.5. A formula φ of RPNL+INT is satisfiable if and only if there exists a fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ such that $\varphi \in \mathcal{L}([0, e])$ for some $e \in D$, with $e > 0$.

A fulfilling LIS with the property above is said to *satisfy* φ . Since fulfilling LISs satisfying φ may be arbitrarily large, or even infinite, if we want to prove the decidability of RPNL+INT, we must be able to restrict effectively the search for a fulfilling LIS satisfying φ . To that aim, we shall prove that each satisfiable formula is satisfied in a finitely presentable, ‘ultimately periodic’ fulfilling LIS.

Definition 5.6. Given a φ -labelled LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ and $j \in D$, the set of *temporal requests* at j , denoted by $REQ(j)$, is the set of temporal formulae from $TF(\varphi)$ belonging to the labeling of any interval ending in j .

The set $REQ(j)$ is well-defined since the temporal requests of all intervals ending in j must be the same. We denote by $REQ(\varphi)$ the set of all possible sets of temporal requests over $CL(\varphi)$. It is easy to show that $|REQ_\varphi| = 2^{\frac{|TF(\varphi)|}{2}}$. From now on, we shall use the symbol m for $\frac{|TF(\varphi)|}{2}$ and k for the maximum among all natural numbers occurring in the length constraints in φ . For example, if $\varphi = \langle A \rangle ((\ell > 3) \wedge p \rightarrow \langle A \rangle ((\ell > 5) \wedge q))$, then $m = 2$ and $k = 5$.

Observation. Given any set of temporal requests $REQ(j)$, all formulae in $REQ(j)$ can be satisfied using at most m different points r such that $r > j$.

Definition 5.7. A LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ is said to be *ultimately periodic*, with prefix L , period P , and threshold k if:

- for interval $[i, j]$ such that $j \geq L$ and $\delta(j, i) > k$, $\mathcal{L}([i, j]) = \mathcal{L}([i, j + P])$;
- for each interval $[i, j]$ such that $i \geq L$, $\mathcal{L}([i, j]) = \mathcal{L}([i + P, j + P])$.

Note that in every ultimately periodic LIS we have that for each $i \geq L$, $REQ(i) = REQ(i + P)$, and that every ultimately periodic LIS is finitely presentable: it suffices to define the labeling in it only on the intervals $[i, j]$ such that $j \leq L + P + \max(k, P)$; thereafter it can be uniquely extended by periodicity. Furthermore, we can identify a finite LIS with an ultimately periodic one with a period $P = 0$.

Definition 5.8. Given any LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$, a (k -)sequence in \mathbf{L} is a sequence of (k) consecutive points in D . Given a sequence σ in \mathbf{L} , its sequence of requests $REQ(\sigma)$ is defined as the sequence of temporal requests at the points in σ . We say that $i \in \mathbf{L}$ starts a k -sequence σ if the temporal requests at $i, \dots, i + k - 1$ form an occurrence of $REQ(\sigma)$.

Lemma 5.9. Let $\mathbf{L} = \langle \mathbb{N}, \mathbb{I}(\mathbb{N}), \mathcal{L} \rangle$ be an infinite fulfilling LIS that satisfies a formula φ on $[0, e]$ for some $e \in \mathbb{N}$. Then, there exists an infinite ultimately periodic fulfilling LIS $\bar{\mathbf{L}} = \langle \mathbb{N}, \mathbb{I}(\mathbb{N}), \bar{\mathcal{L}} \rangle$ that satisfies φ on $[0, e]$.

Proof. First, note that the interval $[0, e]$ satisfying φ in \mathbf{L} can always be chosen so that $e \leq k + 1$. We define the set REQ_{inf} as the subset of $REQ(\varphi)$ containing all and only the sets of requests that occurs infinitely often in \mathbf{L} . Let $L \in \mathbb{N}$ be the first point in \mathbf{L} such that $L > e$ and for every $i \geq L$, $REQ(i) \in REQ_{inf}$. Then, let M be the first point such that $L + k < M$ and the following conditions are satisfied:

1. every set of requests $\mathcal{R} \in REQ_{inf}$ occurs at least m times between L and M ;
2. for each point $i < L$, and any formula $\langle A \rangle \tau \in REQ(i)$, τ is satisfied on some interval $[i, j]$ where $j \leq M$;
3. the k -sequences of requests starting at L and at M are the same.

Let $P = M - L$. We will build an infinite ultimately periodic structure $\bar{\mathbf{L}}$ over the domain \mathbb{N} with prefix L , period P and threshold k . First, we shall define (*prescribed*) sets of temporal requests $\overline{REQ}(d)$ for every $d \leq M + P$ and labeling $\bar{\mathcal{L}}$ on every subinterval of $[0, M + P]$ as follows. First, for all points $d < M$ we put $\overline{REQ}(d) = REQ(d)$. Then, for all points $M + n$, where $0 \leq n \leq P$, we put $\overline{REQ}(M + n) = REQ(L + n)$. (Note that this has not changed the k -sequence of requests starting at M , because of the last condition for the choice of M .) Now, we shall define the labeling. First, for all intervals $[i, j]$ such that $i < j \leq M$ we put $\bar{\mathcal{L}}([i, j]) = \mathcal{L}([i, j])$. Then, consider any interval $[i, j]$ such that $M < j \leq M + P$:

- if $M \leq i$ we put $\bar{\mathcal{L}}([i, j]) = \mathcal{L}([i - P, j - P])$;
- if $i < M$ we put $\bar{\mathcal{L}}([i, j]) = \mathcal{L}([i, j])$.

The above construction labels all subintervals of $[0, M + P]$ in a way that is consistent with the definition of LIS, but that is not necessarily fulfilling. It could be the case that for some point $L \leq i \leq M$ and some formula $\langle A \rangle \psi \in \overline{REQ}(i)$ there are no intervals satisfying ψ . We fix such “defects” as follows. Since $\overline{REQ}(i) = REQ(i)$, there exists a point $j > i$ such that $\psi \in \mathcal{L}([i, j])$ in the original model. By condition 1 in the choice of M , there exists at least m points between M and $M + P$ with the same set of requests of j . We pick one such point j' that is “available”, in the sense that the interval $[i, j']$ either fulfills no $\langle A \rangle$ -formulas from $\overline{REQ}(i)$ or it fulfills only $\langle A \rangle$ -formulas that are fulfilled by other intervals starting at i , and we put $\bar{\mathcal{L}}([i, j']) = \mathcal{L}([i, j])$, thus fixing the defect for $\langle A \rangle \psi$ in i . By repeating such a procedure sufficiently many times going from left to right, we build a LIS where every request of every point $i \leq M$ is fulfilled before $M + P$.

To conclude the construction we extend the so defined $\bar{\mathcal{L}}$ over $\mathbb{I}(\mathbb{N})$ in the unique way satisfying the conditions in

Definition 5.7 for an ultimate periodic LIS with prefix L , period P , and threshold k , that is:

- for every $i > M + P$ we put $\overline{REQ}(i) := \overline{REQ}(i - nP)$ where n is the least non-negative integer such that $i - nP \leq M + P$;
- for every interval $[i, j]$ such that $j > M + P$, we put $\bar{\mathcal{L}}([i, j]) = \bar{\mathcal{L}}([i - nP, j - qP])$, where n and q are the least non-negative integers such that $i - nP \leq M$ and $j - qP \leq M + P$.

It is straightforward to check that the labeling $\bar{\mathcal{L}}$ defined above respects all length constraints $\ell > k'$ and their negations for all intervals, and that the resulting structure $\bar{\mathbf{L}} = \langle \mathbb{N}, \mathbb{I}(\mathbb{N}), \bar{\mathcal{L}} \rangle$ is an ultimately periodic fulfilling LIS satisfying φ on $[0, e]$. \square

Thus, we have shown that every satisfiable formula φ has an ultimately periodic model. We have to prove that we are able to bound both the length of the prefix and the period, in order to establish a (non-standard) small model property for RPNL+INT.

Definition 5.10. Given a LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$, a sequence of requests $REQ(\sigma)$ is said to *occur disjointly n times in D* if there are $2n$ distinct points $i_1 < j_1 < \dots < i_n < j_n$, such that, for each $1 \leq q \leq n$, the sequence of requests on the consecutive points i_q, \dots, j_q is $REQ(\sigma)$. In particular, this definition applies also for a single request, i.e., when $|\sigma| = 1$.

Definition 5.11. Given any $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$, the sequence of requests $REQ(\sigma)$ is said to be *abundant* in \mathbf{L} (on an interval $[i, j]$) if and only if has at least $m|REQ(\varphi)| + 1$ disjoint occurrences in D (in the interval $[i, j]$).

We observe the following combinatorial fact about abundant sequences of requests.

Lemma 5.12. *Given a LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ and any abundant sequence of requests $REQ(\sigma)$ in it, there exist an index q such that for each element $\mathcal{R} \in \{REQ(d) \mid i_q \leq d < i_{q+1}\}$, where i_q and i_{q+1} begin, respectively, the q -th and the $q + 1$ -th occurrence of σ , \mathcal{R} occurs at least m times after j_{q+1} .*

Proof. To prove this property, we shall reason by contradiction. Suppose that $REQ(\sigma)$ is abundant, that is, it occurs $n > m|REQ(\varphi)| + 1$ times in D and, for each q with $1 \leq q \leq n$, there exists a point $d(q)$ with $i_q \leq d(q) < i_{q+1}$, such that $REQ(d(q))$ occurs less than m times after j_{q+1} . Let $\Delta = \{d(q) \mid 1 \leq q \leq n\}$ the set of all such points. By hypothesis, there cannot be any $\mathcal{R} \in REQ(\varphi)$ such that \mathcal{R} occurs more than m times in Δ . Then $|\Delta| \leq m|REQ(\varphi)|$, which is a contradiction. \square

Lemma 5.13. *Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ be a fulfilling LIS that satisfies φ . Suppose that there exist an abundant k -sequence of requests $REQ(\sigma)$ and let q be the index whose existence is guaranteed by the Lemma 5.12. Then, there exists a fulfilling LIS $\bar{\mathbf{L}} = \langle \bar{\mathbb{D}}, \bar{\mathbb{I}}(\bar{\mathbb{D}}), \bar{\mathcal{L}} \rangle$ that satisfies φ such that $D' = D \setminus \{i_q, \dots, i_{q+1} - 1\}$.*

Proof. Let us fix a fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ satisfying φ at some $[0, e]$, an abundant k -sequence $REQ(\sigma)$ in \mathbf{L} , and the index q identified by Lemma 5.12. Now, let $D^- = \{i_q, \dots, i_{q+1} - 1\}$ and $D' = D \setminus D^-$. For sake of readability, the points in D' will be denoted by the same numbers as in D . Now, we have the problem of suitably re-defining the evaluation of all intervals on D' in a way preserving the temporal requests at all points in D' and still satisfying φ . First, we consider all points $d < i_q$ and, for all $0 \leq p \leq k - 1$, we put $\mathcal{L}'([d, i_{q+1} + p]) = \mathcal{L}([d, i_q + p])$. In this way, we have guaranteed that the intervals whose length has changed as an effect of the elimination of the points in D^- has now a correct labeling in terms of all length constraints of the type $\ell > k'$ and $\neg(\ell > k')$ where $k' \leq k$. Moreover, since the requests at $i_{q+1} + p$ in \mathbf{L} are equal to the requests at $i_q + p$ in \mathbf{L} , this operation is safe w.r.t. to the universal requirements. Notice also that the lengths of intervals beginning before i_q and ending after j_{q+1} are greater than k in both in \mathbf{L} and in \mathbf{L}' , and, thus, there is no need to change their labeling. The structure $\mathbf{L}' = \langle \mathbb{D}', \mathbb{I}(\mathbb{D}'), \mathcal{L}' \rangle$ defined so far is obviously a LIS, but it is not necessarily a fulfilling one. While the intervals beginning at $d \in D^-$ are not critical in terms of fulfilling, the removal of such points may generate *defects*, that is, situations in which there is a point $d < i_q$ and a formula of the type $\langle A \rangle \psi$ belonging to $REQ(d)$, which was satisfied on $[d, d']$, where $d' \in D^-$, but it is not satisfied anymore. We have to show how repair such defects. Suppose that there exists $d < i_q$ such that for some formula $\langle A \rangle \psi \in REQ(d)$, $\psi \in \mathcal{L}([d, d'])$, where $d' \in D^-$, and it is not satisfied anymore in \mathbf{L}' . First of all, notice that $\delta(d', d) > k$ in \mathbf{L} . By Lemma 5.12, there are at least m points $\{\bar{d}_1, \bar{d}_2, \dots, \bar{d}_m\}$ after j_{q+1} such that $REQ(\bar{d}_i) = REQ(d')$ for $i = 1, \dots, m$. Since $REQ(d)$ may contain at most m different requests, there exists at least one “unused” \bar{d}_h , in the sense that, in \mathbf{L} the interval $[d, \bar{d}_h]$ satisfies only those $\langle A \rangle$ -formulas of $REQ(d)$, if any, that are satisfied at other intervals starting at d . Thus we can put $\mathbf{L}'([d, \bar{d}_h]) = \mathbf{L}([d, d'])$, and correct this defect without creating a new one. Since $\delta(\bar{d}_h, d) > k$ in \mathbf{L}' , this operation does not introduce inconsistencies with the length constraints in the labeling, either.

Now, if we repeat the above procedure sufficiently many times, we obtain a finite sequence of LISs, the last one of which is the required $\bar{\mathbf{L}}$. \square

The lemma above guarantees that we can eliminate sequences of requests that occur ‘too many’ times in a LIS,

without ‘spoiling’ the LIS. Using that result we can now reduce within pre-computed bounds the lengths of the prefix and the period in ultimately periodic models of a given formula.

Theorem 5.14 (Small model property). *If φ is any satisfiable formula of RPNL+INT, then there exists a fulfilling, ultimately periodic LIS that satisfies φ , and such that each of the lengths L of the prefix and P of the period is less or equal to $|REQ(\varphi)|^k m |REQ(\varphi)|k + k - 1$.*

Proof. Let $\mathbf{L} = \langle \mathbb{D}, \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ be the ultimately periodic LIS whose existence is guaranteed by Lemma 5.9. Consider the point L , that is, the last point of the prefix. If $L \leq |REQ(\varphi)|^k m |REQ(\varphi)|k + k - 1$, then the prefix is already bounded as claimed. Otherwise, by an application of the pigeonhole principle, for at least one sequence $REQ(\sigma)$ of length k , we have that $REQ(\sigma)$ is abundant on the interval $[0, L]$. In this case, we apply Lemma 5.13 sufficiently many times to get the requested maximum length for the prefix. Notice that this elimination process do not change the conditions for ultimately periodic LISs in the prefix. Now, suppose that $P > |REQ(\varphi)|^k m |REQ(\varphi)|k + k - 1$. Again, by an application of the pigeonhole principle, for at least one sequence $REQ(\sigma)$ of length k , we have that $REQ(\sigma)$ is abundant on the interval $[L, L+P]$ and, for each n , on the interval $[L+nP, L+(n+1)P]$. By construction, the abundant sequence is always in the same position with respect to the beginning point of the respective occurrence of the period. Thus, we can multiply apply Lemma 5.13 to reduce the length of the period without changing the conditions for an ultimately periodic LIS, until we reach the desired length for the period. \square

Since ultimately periodic LIS are finitely presentable structures, decidability of RPNL+INT directly follows from Theorem 5.14.

Corollary 5.15. *The satisfiability problem for RPNL+INT, interpreted over natural numbers, is decidable.*

6 The complexity of RPNL+INT

By exploiting the results of the previous section, we can give a nondeterministic exponential time procedure for checking the satisfiability of any RPNL+INT-formula φ .

Let $n = |\varphi|$. As a first step, the algorithm guesses the length $P, L \leq 2^{nk} k(m2^n) + k - 1$ of the prefix and the period of the ultimately periodic LIS satisfying φ . Then, for every set of requests $\mathcal{R} \in REQ(\varphi)$, it guesses the number of occurrences of \mathcal{R} in the model. Such a number $n(\mathcal{R})$ can be either 0 (no occurrences), a positive integer less or equal to $2^{nk} k(m2^n) + k - 1$ (finitely many occurrences), or $+\infty$ (infinitely many occurrences).

After this first guessing phase, the procedure checks for the existence of a fulfilling ultimately periodic LIS satisfying φ with prefix P and period L . As an initial step, it guesses a labeling for each interval $[0, i]$ with $1 \leq i \leq k$. Every labeling $\mathcal{L}([0, i])$ defines a corresponding set of requests $REQ(i)$: $n(REQ(i))$ is thus decreased by 1 for every $1 \leq i \leq k$. During this phase, the algorithm also checks that $\varphi \in \mathcal{L}([0, i])$ for some i . Then, it initializes a counter d to 1 and it proceeds as follows.

1. Guess the labeling of the intervals $[d, d + i]$ with $1 \leq i \leq k - 1$, and checks if they are consistent with the sets of requests $REQ(d)$ and $REQ(d + i)$ defined in the previous iteration;
2. Guess the labeling of the interval $[d, d + k]$. This operation defines the set $REQ(d + k)$ for the first time: decrease $n(REQ(d + k))$ by 1;
3. For every formula $\langle A \rangle \psi \in REQ(d)$ that is not fulfilled by the intervals $[d, d + i]$, with $1 \leq i \leq k$, guess a set of requests $\mathcal{R} \in REQ(\varphi)$ and an atom A_ψ such that:
 - (a) $n(\mathcal{R}) > 0$,
 - (b) $\psi \in A_\psi$,
 - (c) for every $[A]\xi \in REQ(d)$, $\xi \in A_\psi$,
 - (d) $A_\psi \cap TF(\varphi) \subseteq \mathcal{R}$;
4. For every $\mathcal{R} \in REQ(\varphi)$ such that $n(\mathcal{R}) > 0$, guess an atom $A_{\mathcal{R}}$ such that, for every $[A]\xi \in REQ(d)$, $\xi \in A_{\mathcal{R}}$ and $A_{\mathcal{R}} \cap TF(\varphi) \subseteq \mathcal{R}$;
5. If $d = P$, check whether $n(\mathcal{R}) > 0$ implies $n(\mathcal{R}) = +\infty$, for all $\mathcal{R} \in REQ(\varphi)$;
6. If $d < P + L$, increase d by 1 and go to step 1. Otherwise, stop and return *true*.

The procedure ends with failure whenever an atom with the required properties cannot be guessed.

As for the complexity, we have that the amount of time needed to complete the procedure is bounded by the values of P and L , and thus the time complexity class is $\text{NTIME}(2^{nk})$. The space complexity of the procedure can be bounded by observing that the procedure needs to store only the following information:

- the values of P , L , and d , that are bounded by $2^{nk}k(m2^n) + k - 1$ and thus, can be stored with $O(nk)$ bits;
- for every set of requests, the number $n(\mathcal{R})$. Since $|REQ(\varphi)| \in O(2^n)$, all the above information needs $O(k2^n)$ bits to be stored in memory;

- at every step of the procedure, the labeling of the intervals $[d, d + i]$ and the corresponding sets of requests $REQ(d + i)$, that needs $O(nk)$ bits;
- a constant amount of space for storing temporary counters and checking for the consistency of the guessed atom.

Hence, the total amount of space needed by the procedure is bounded by $O(k2^n)$ and the space complexity class is thus $\text{SPACE}(k2^n)$.

In [13] it has been shown that non-metric RPNL is complete for NEXPTIME when interpreted over natural numbers (and also when interpreted in the class of all linearly ordered sets [12]). This lower bound already gives us that the computational complexity of any language of the class RPNL+INT is at least NEXPTIME -hard. On the other hand, the above non-deterministic procedure shows that the complexity of the most expressive language among the considered class is between $\text{SPACE}(k2^n)$ and $\text{NTIME}(2^{kn})$. To correctly locate the complexity, we need to distinguish three cases.

Case 1: k is a constant. In this case, k does not influence the complexity class. So, since we have a $\text{NTIME}(2^n)$ procedure for satisfiability, and a NEXPTIME -hardness result, we can conclude that RPNL+INT is NEXPTIME -complete when k is a constant.

Case 2: k is represented in unary. Unary encoding means that the value of k increases linearly with the length of the formula. So, $\text{NTIME}(2^{kn}) = \text{NTIME}(2^{n^2})$. Therefore, as in the previous case, RPNL+INT is NEXPTIME -complete.

Case 3: k is represented in binary. A binary codification means that the value of k increases exponentially with the length of the formula. This implies that the time complexity is non-deterministically double exponential, since $k = O(2^{|\varphi|})$ (with n binary digits we can encode numbers up to $2^n - 1$). On the other hand, the satisfiability problem for RPNL+INT is located in $\text{SPACE}(2^n)$. The following result shows that, with binary encoding, this complexity class is optimal.

Theorem 6.1. *The satisfiability problem for RPNL+INT , when the length constraints are represented in binary, is EXPSpace -complete.*

Proof. The EXPSpace upper-bound for the complexity of RPNL+INT directly follows from the above non-deterministic procedure for satisfiability checking. To show that the logic is EXPSpace -hard, we consider the *Exponential Corridor Tiling Problem*. This is the problem, given any parameter $n \in \mathbb{N}$, to establish whether a given finite set of tile types $\mathcal{T} = \{t_1, \dots, t_h\}$ can tile the 2^n -corridor $C = \{(i, j) \mid 1 \leq i \leq 2^n, j \in \mathbb{N}\}$. For every tile type

$t_i \in \mathcal{T}$, let $right(t_i)$, $left(t_i)$, $up(t_i)$, and $down(t_i)$ be the colors of the corresponding sides of t_i . To solve the problem, one must find a function $f : \mathcal{C} \rightarrow \mathcal{T}$ such that

$$\begin{aligned} right(f(i, j)) &= left(f(i + 1, j)) \\ up(f(i, j)) &= down(f(i, j + 1)). \end{aligned}$$

It is known (see e.g., [4]) that this problem is EXPSPACE-complete. So, given any set $\mathcal{T} = \{t_1, \dots, t_h\}$, if we show that there exists a RPNL+INT-formula $\varphi(\mathcal{T})$ of length polynomial in n , with numbers encoded in binary, and that $\varphi(\mathcal{T})$ is satisfiable if and only if \mathcal{T} tiles the 2^n -corridor, then RPNL+INT is EXPSPACE-hard.

First of all, consider n propositional letters $\bar{p} = p_0, \dots, p_{n-1}$. If we interpret each propositional letter as a binary value p_i , then we can define $n(\bar{p})$ as the natural number corresponding to any value of \bar{p} , that is, $n(\bar{p}) = p_{n-1}2^{n-1} + p_{n-2}2^{n-2} + \dots + p_0$. This means that $0 \leq n(\bar{p}) \leq 2^n - 1$. In our encoding, for each level of the corridor we place tiles over ‘unit’ intervals, that is, intervals of length 1. Thus, we give a unique coordinate to each tile, encoded by a unique $n(\bar{p})$. It is not difficult to show that only $2n$ formulas are necessary to express the binary successor. In the following, $[G]$ is the universal modality defined in Section 3.

$$\langle A \rangle (\neg p_{n-1} \wedge \neg p_{n-2} \wedge \dots \wedge \neg p_0 \wedge \ell = 1) \quad (1)$$

$$\bigwedge_{i=0}^{n-1} [G](p_{i-1} \wedge p_{i-2} \wedge \dots \wedge p_0 \rightarrow (p_i \leftrightarrow \langle A \rangle (\neg p_i \wedge \ell = 1))) \quad (2)$$

$$\bigwedge_{i=0}^{n-1} [G](\neg(p_{i-1} \wedge p_{i-2} \wedge \dots \wedge p_0) \rightarrow (p_i \leftrightarrow \langle A \rangle (p_i \wedge \ell = 1))) \quad (3)$$

Now, consider the following set of formulas, where we use the special propositional letters $tile, t_1, \dots, t_h$.

$$\langle A \rangle (tile \wedge \ell = 1) \quad (4)$$

$$[G](tile \rightarrow \langle A \rangle (tile \wedge \ell = 1)) \quad (5)$$

$$[G](tile \leftrightarrow \bigvee_{i=1}^h (t_i)) \quad (6)$$

$$[G] \bigwedge_{i,j=1, i \neq j}^h \neg(t_i \wedge t_j) \quad (7)$$

$$[G] \bigvee_{right(t_i)=left(t_j)} (t_i \wedge \neg(p_{n-1} \wedge p_{n-2} \wedge \dots \wedge p_0) \rightarrow \langle A \rangle (t_j \wedge \ell = 1)) \quad (8)$$

$$[G] \bigvee_{up(t_i)=down(t_j)} (t_i \wedge \rightarrow \langle A \rangle (\ell = 2^n - 1 \wedge \langle A \rangle t_j)) \quad (9)$$

Notice that, with the assumption of encoding numbers with binary representation, the operator $\ell = 2^{k-1}$ needs only n binary digits. Now, it is not difficult to show that the conjunction of the above formulas is satisfiable if and only if the set \mathcal{T} can tile the 2^n -corridor. Moreover, the encoding only requires a linear number of propositional letters, which means that the length of the formula is linear in n , and so the result. \square

7 Conclusions

In this paper we have considered an extension of Right Propositional Neighborhood Logic (RPNL) over natural numbers with integers constraints over the interval lengths. We have shown that the resulting interval logic RPNL+INT is decidable, and have determined its computational complexity depending on the representation and bounds of the constraints. Notably, when the length constraints are bounded, RPNL+INT preserves the computational complexity of the non-metric version RPNL, despite the fact that it is strictly more expressive than the latter, while without these assumptions RPNL+INT becomes EXPSPACE-complete.

In the search for even more expressive, yet decidable, metric interval logics, we intend to extend RPNL+INT over both directions (PNL+INT) as well as to consider the decision problem for interval logics that extend RPNL and PNL with rational length constraints.

Acknowledgments

We thank the anonymous referees for useful remarks and references.

References

- [1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- [2] R. Alur and T. Henzinger. A really temporal logic. *Journal of the ACM*, 41:181–204, 1994.
- [3] T. Bolander, J. Hansen, and M. R. Hansen. Decidability of a hybrid duration calculus. *Electronic Notes in Theoretical Computer Science*, 174(6):113–133, 2007.
- [4] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives of Mathematical Logic. Springer, 1997.
- [5] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. Decidable and Undecidable Fragments of Halpern and Shoham’s Interval Temporal Logic: Towards a Complete Classification. In I. Cervesato, H. Veith, and A. Voronkov, editors, *Proc. of 15th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR2008)*, volume 5330 of *LNCS*, pages 590–604. Springer, 2008.

- [6] D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. Undecidability of interval temporal logics with the *Overlap* modality. In *Proc. of 16th International Symposium on Temporal Representation and Reasoning (TIME 2009)*, pages 88–95. IEEE Computer Society Press, 2009.
- [7] D. Bresolin, V. Goranko, A. Montanari, and P. Sala. Tableau-based decision procedures for the logics of subinterval structures over dense orderings. *Journal of Logic and Computation*, doi: 10.1093/logcom/exn063, 2008.
- [8] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. On Decidability and Expressiveness of Propositional Interval Neighborhood Logics. In *Proc. of the International Symposium on Logical Foundations of Computer Science*, volume 4514 of *LNCS*, pages 84–99. Springer, 2007.
- [9] D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic*, doi: 10.1016/j.apal.2009.07.003, 2009.
- [10] D. Bresolin, A. Montanari, and P. Sala. An optimal tableau-based decision algorithm for Propositional Neighborhood Logic. In *Proceedings of STACS 2007*, volume 4393 of *LNCS*, pages 549–560. Springer, 2007.
- [11] D. Bresolin, A. Montanari, and P. Sala. An optimal tableau for Right Propositional Neighborhood Logic over trees. In *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning (TIME 2008)*, pages 110–117, Montreal, Quebec, Canada, June 2008. IEEE Computer Society Press.
- [12] D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco. Optimal Tableaux for Right Propositional Neighborhood Logic over Linear Orders. In *Proc. of the 11th European Conference on Logics in AI*, volume 5293 of *LNAI*, pages 62–75. Springer, 2008.
- [13] D. Bresolin, A. Montanari, and G. Sciavicco. An optimal decision procedure for Right Propositional Neighborhood Logic. *Journal of Automated Reasoning*, 38(1-3):173–199, 2007.
- [14] Z. Chaochen, M. R. Hansen, and P. Sestoft. Decidability and undecidability results for duration calculus. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, *STACS'93*, volume 665 of *LNCS*, pages 58–68. Springer-Verlag, 1993.
- [15] Z. Chaochen, C. A. R. Hoare, and A. P. Ravn. A calculus of durations. *Information Processing Letters*, 40(5):269–276, 1991.
- [16] N. Chetcuti-Serandio and L. F. D. Cerro. A mixed decision method for duration calculus. *Journal of Logic and Computation*, 10:877–895, 2000.
- [17] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.
- [18] M. R. Hansen and Z. Chaochen. Duration calculus: Logical foundations. *Formal Aspects of Computing*, 9:283–330, 1997.
- [19] Y. Hirshfeld and A. Rabinovich. Logics for real time: Decidability and complexity. *Fundamenta Informaticae*, 1(62):1–28, 2004.
- [20] H. Kautz and P. Ladkin. Integrating metric and qualitative temporal reasoning. In *National Conference on Artificial Intelligence*, pages 241–246, 1991.
- [21] R. Koymans. Specifying real-time properties with metric temporal logic. *Real Time Systems*, 2(4):255–299, 1990.
- [22] A. Montanari and M. de Rijke. Two-sorted metric temporal logic. *Theoretical Computer Science*, 183(2):187–214, 1997.
- [23] A. Montanari and A. Policriti. Executing metric temporal logic. In *Proceedings of the IJCAI'97 Workshop on Programming in Temporal and Non Classical Logics*, Nagoya, Japan, 1997.
- [24] B. Moszkowski. *Reasoning about digital circuits*. Tech. rep. stan-cs-83-970, Dept. of Computer Science, Stanford University, Stanford, CA, 1983.
- [25] J. Ouaknine and J. Worrell. Some recent results in metric temporal logic. In *FORMATS*, pages 1–13, 2008.
- [26] G. Sciavicco. Quality checking of medical guidelines using interval temporal logics (preliminary report). In *Proceedings of the 12th Conferencia Asociación Española para la Inteligencia Artificial (CAEPIA 2007)*, volume 2, pages 249–258, 2007.
- [27] G. Sciavicco, J. Juarez, and M. Campos. Quality checking of medical guidelines using interval temporal logics: A case-study. In *Proceedings of the 3rdth International Work-conference on the Interplay between Natural and Artificial Computation (IWINAC 2009)*, volume 5602 of *LNCS*, pages 158–167, 2009.
- [28] C. Zhou and M. R. Hansen. *Duration Calculus: A Formal Approach to Real-Time Systems*. EATCS: Monographs in Theoretical Computer Science. Springer, 2004.