The Complexity of Clausal Fragments of LTL

A. Artale,[1] R. Kontchakov,[2] V. Ryzhikov,[1] and M. Zakharyaschev[2]

[1] KRDB Research Centre
Free University of Bozen-Bolzano
I-39100 Bolzano, Italy
{artale,ryzhikov}@inf.unibz.it

[2] Dept. of Computer Science and Inf. Systems
Birkbeck, University of London
London WC1E 7HX, UK
{roman,michael}@dcs.bbk.ac.uk

Abstract. We introduce and investigate a number of fragments of propositional temporal logic LTL over the flow of time $(\mathbb{Z}, <)$. The fragments are defined in terms of the available temporal operators and the structure of the clausal normal form of the temporal formulas. We determine the computational complexity of the satisfiability problem for each of the fragments, which ranges from NLogSpace to PTime, NP and PSpace.

1 Introduction

We consider the (PSpace-complete) propositional temporal logic LTL over the flow of time $(\mathbb{Z}, <)$. Our aim is to investigate how the computational complexity of the satisfiability problem for LTL-formulas depends on the form of their clausal representation and the available temporal operators.

Sistla and Clarke [26] showed that satisfiability of LTL-formulas with all standard operators ('next-time', 'always in the future', 'eventually' and 'until') is PSpace-complete; see also [18, 19]. Ono and Nakamura [22] proved that for formulas with only 'always in the future' and 'eventually' the satisfiability problem becomes NP-complete. Since then a number of fragments of LTL of different complexity have been identified. For example, Chen and Lin [10] observed that the complexity does not change if we restrict attention to temporal Horn formulas. Demri and Schnoebelen [12] determined the complexity of fragments that depend on three parameters: the available temporal operators, the number of nested temporal operators, and the number of propositional variables in formulas. Markey [21] analysed fragments defined by the allowed set of temporal operators, their nesting and the use of negation. Dixon *et al.* [13] introduced a XOR fragment of LTL and showed its tractability. Bauland *et al.* [7] systematically investigated the complexity of fragments given by both temporal operators and Boolean connectives (using Post's lattice of sets of Boolean functions).

In this paper, we classify temporal formulas according to their clausal normal form. Recall [14] that any LTL-formula over $(\mathbb{N}, <)$ can be transformed into an equisatisfiable formula in the so-called *separated normal form* that consists of initial clauses (setting conditions at moment 0), step clauses (defining transitions between consecutive states), and eventuality clauses (defining the states that must be reached infinitely often). Our clausal normal form is a slight generalisation

of the separated normal form. The main building blocks are *positive temporal literals* λ given by the following grammar:

$$\lambda \quad ::= \quad \bot \quad | \quad p \quad | \quad \bigcirc_F \lambda \quad | \quad \bigcirc_P \lambda \quad | \quad \Box_F \lambda \quad | \quad \Box_P \lambda \quad | \quad \circledast \lambda, \qquad (1)$$

where p is a propositional variable, $\bigcirc_F$ and $\bigcirc_P$ are the next- and previous-time operators, and $\Box_F$, $\Box_P$, $\circledast$ are the operators 'always in the future,' 'always in the past' and 'always.' We say that a temporal formula φ is in *clausal normal form* if

$$\varphi \quad ::= \quad \lambda \quad | \quad \neg\lambda \quad | \quad \circledast(\neg\lambda_1 \vee \dots \vee \neg\lambda_n \vee \lambda_{n+1} \vee \dots \vee \lambda_{n+m}) \quad | \quad \varphi_1 \wedge \varphi_2. \quad (2)$$

Conjunctions of positive and *negative* ($\neg\lambda$) literals can be thought of as initial clauses, while conjunctions of $\circledast$-formulas generalise both step and eventuality clauses of the separated normal form. Similarly to [15] one can show that any LTL-formula over $(\mathbb{Z}, <)$ is equisatisfiable to a formula in clausal normal form.

We consider twelve fragments of LTL that will be denoted by $\mathsf{LTL}_\alpha^{\Box,\bigcirc}$, $\mathsf{LTL}_\alpha^{\Box}$ and $\mathsf{LTL}_\alpha^{\circledast}$, for $\alpha \in \{\mathit{bool}, \mathit{horn}, \mathit{krom}, \mathit{core}\}$. The superscript in the language name indicates the temporal operators that can be used in its positive literals. Thus, $\mathsf{LTL}_\alpha^{\Box,\bigcirc}$ uses all types of positive literals, $\mathsf{LTL}_\alpha^{\Box}$ can only use the $\Box$-operators:

$$\lambda \quad ::= \quad \bot \quad | \quad p \quad | \quad \Box_F \lambda \quad | \quad \Box_P \lambda \quad | \quad \circledast \lambda,$$

and $\mathsf{LTL}_\alpha^{\circledast}$ only the $\circledast$-operator:

$$\lambda \quad ::= \quad \bot \quad | \quad p \quad | \quad \circledast \lambda.$$

The subscript α in the language name refers to the form of the clauses

$$\neg\lambda_1 \vee \dots \vee \neg\lambda_n \vee \lambda_{n+1} \vee \dots \vee \lambda_{n+m} \qquad (3)$$

$(m, n \geq 0)$ that can be used in the formulas φ:

- *bool*-clauses are arbitrary clauses of the form (3),
- *horn*-clauses have at most one positive literal (that is, $m \leq 1$),
- *krom*-clauses are binary (that is, $n + m \leq 2$),
- *core*-clauses are binary with at most one positive literal ($n + m \leq 2$, $m \leq 1$).

The tight complexity bounds in Table 1 show how the complexity of the satisfiability problem for LTL-formulas depends on the form of clauses and the available temporal operators. The PSPACE upper bound for $\mathsf{LTL}_{\mathit{bool}}^{\Box,\bigcirc}$ is well-known [18, 26, 24, 25]; the matching lower bound can be obtained already for $\mathsf{LTL}_{\mathit{horn}}^{\Box,\bigcirc}$ without $\Box_F$ and $\Box_P$ by a standard encoding of deterministic Turing machines with polynomial tape [10]. The NP upper bound for $\mathsf{LTL}_{\mathit{bool}}^{\Box}$ is also well-known [22], and the PTIME and NLOGSPACE lower bounds for $\mathsf{LTL}_{\mathit{horn}}^{\circledast}$ and $\mathsf{LTL}_{\mathit{core}}^{\circledast}$ coincide with the complexity of the respective non-temporal languages. The upper bounds for the $\mathsf{LTL}_\alpha^{\circledast}$ fragments can be obtained by embedding into the the underlying propositional fragments; see the full paper [6] for details.

The main contributions of this paper are the remaining complexity results in Table 1. The complexity of the $\mathsf{LTL}_\alpha^{\Box}$ fragments matches the complexity of

temporal operators	$\boxplus, \Box_F, \Box_P, \bigcirc_F, \bigcirc_P$	$\boxplus, \Box_F, \Box_P$	$\boxplus$
α	$\mathsf{LTL}_\alpha^{\Box,\bigcirc}$	$\mathsf{LTL}_\alpha^{\Box}$	$\mathsf{LTL}_\alpha^{\boxplus}$
bool	PSPACE ($\leq$ [26])	NP ($\leq$ [22])	NP
horn	PSPACE ($\geq$ [10])	PTIME [$\leq$ Th. 3]	PTIME
krom	NP [$\leq$ Th. 1]	NP [$\geq$ Th. 5]	NLOGSPACE
core	NP [$\geq$ Th. 2]	NLOGSPACE [$\leq$ Th. 4]	NLOGSPACE

Table 1. The complexity of clausal fragments of LTL.

the underlying non-temporal fragments except for the Krom case, where we can use the clauses $\neg p \vee \neg\Box_F q$ and $q \vee r$ to say that $p \to \Diamond_F r$ (if p then eventually r), which allows one to encode 3-colourability and results in NP-hardness. It is known that the addition of the operators $\bigcirc_F$ and $\bigcirc_P$ to the language with $\Box_F$ and $\Box_P$ usually increases the complexity (note that the proofs of the lower bounds for the $\mathsf{LTL}_\alpha^{\Box,\bigcirc}$ fragments require only $\boxplus$ and $\bigcirc_F$). It is rather surprising that this does not happen in the case of the Krom fragment, while the complexity of the corresponding core fragment jumps from NLOGSPACE to NP.

We prove the upper bounds using two different techniques. The existence of models for $\mathsf{LTL}_{krom}^{\Box,\bigcirc}$-formulas is checked in Section 3 by guessing a small number of types and exponentially large distances between them (given in binary) and then using unary automata (and the induced arithmetic progressions) to verify correctness of the guess in polynomial time. In Section 4.1, we design a calculus for $\mathsf{LTL}_{core}^{\Box}$ in which derivations can be thought of as paths in a graph over the propositions labelled by moments of time. Thus, the existence of such derivations is essentially the graph reachability problem and can be solved in NLOGSPACE.

2 The Clausal Normal Form for LTL

The *propositional linear-time temporal logic* LTL (see, e.g., [16, 17] and references therein) we consider in this paper is interpreted over the flow of time $(\mathbb{Z}, <)$. LTL-*formulas* are built from propositional variables $p_0, p_1, \ldots$, propositional constants $\top$ and $\bot$, the Boolean connectives $\wedge$, $\vee$, $\to$ and $\neg$, and two binary temporal operators $\mathcal{S}$ ('since') and $\mathcal{U}$ ('until'), which are assumed to be 'strict.' So, the other temporal operators mentioned in the introduction can be defined via $\mathcal{S}$ and $\mathcal{U}$ as follows:

$$\bigcirc_F\varphi = \bot\,\mathcal{U}\,\varphi, \quad \Diamond_F\varphi = \top\,\mathcal{U}\,\varphi, \quad \Box_F\varphi = \neg\Diamond_F\neg\varphi, \quad \boxplus\varphi = \Diamond_P\Diamond_F\varphi,$$
$$\bigcirc_P\varphi = \bot\,\mathcal{S}\,\varphi, \quad \Diamond_P\varphi = \top\,\mathcal{S}\,\varphi, \quad \Box_P\varphi = \neg\Diamond_P\neg\varphi, \quad \boxtimes\varphi = \Box_P\Box_F\varphi.$$

A *temporal interpretation*, $\mathfrak{M}$, defines a truth-relation between moments of time $n \in \mathbb{Z}$ and propositional variables p_i. We write $\mathfrak{M}, n \models p_i$ to indicate that p_i is true at the moment n in the interpretation $\mathfrak{M}$. This truth-relation is extended to

all LTL-formulas as follows (the Booleans are interpreted as expected):

$$\mathfrak{M}, n \models \varphi \,\mathcal{U}\, \psi \text{ iff there is } k > n \text{ with } \mathfrak{M}, k \models \psi \text{ and } \mathfrak{M}, m \models \varphi, \text{ for } n < m < k,$$
$$\mathfrak{M}, n \models \varphi \,\mathcal{S}\, \psi \text{ iff there is } k < n \text{ with } \mathfrak{M}, k \models \psi \text{ and } \mathfrak{M}, m \models \varphi, \text{ for } k < m < n.$$

An LTL-formula φ is *satisfiable* if there is an interpretation $\mathfrak{M}$ such that $\mathfrak{M}, 0 \models \varphi$; in this case we call $\mathfrak{M}$ a *model* of φ. We denote the length of φ by $|\varphi|$.

Recall that LTL-formulas of the form (2) were said to be in *clausal normal form*, and the class of such formulas was denoted by $\mathsf{LTL}^{\Box,\bigcirc}_{bool}$. The clauses (3) will often be represented as $\lambda_1 \wedge \dots \wedge \lambda_n \to \lambda_{n+1} \vee \dots \vee \lambda_{n+m}$ (where the empty disjunction is $\bot$ and the empty conjunction is $\top$).

Lemma 1 (clausal normal form). *For every LTL-formula, one can construct an equisatisfiable $\mathsf{LTL}^{\Box,\bigcirc}_{bool}$-formula. The construction requires logarithmic space.*

The proof of this lemma is similar to the proof of [15, Theorem 3.3.1] and uses fixed-point unfolding and renaming [15, 23]. For example, we can replace every positive occurrence (that is, an occurrence in the scope of an even number of negations) of $p \,\mathcal{U}\, q$ in a given formula φ with a fresh propositional variable r and add the conjuncts $\boxtimes(r \to \bigcirc_F q \vee \bigcirc_F p)$, $\boxtimes(r \to \bigcirc_F q \vee \bigcirc_F r)$ and $\boxtimes(r \to \Diamond_F q)$ to φ. The result contains no positive occurrences of $p\mathcal{U}q$ and is equisatisfiable with φ: the first two conjuncts are the fixed-point unfolding $(p \,\mathcal{U}\, q) \to \bigcirc_F q \vee \big(\bigcirc_F p \wedge \bigcirc_F (p \,\mathcal{U}\, q)\big)$, while the last conjunct ensures that the fixed-point is eventually reached.

The next lemma allows us to consider an even more restricted classes of formulas. In what follows, we do not distinguish between a set of formulas and the conjunction of its members, and we write $\boxtimes \Phi$ for the conjunction $\bigwedge_{\chi \in \Phi} \boxtimes \chi$.

Lemma 2. *Let $\mathcal{L}$ be one of $\mathsf{LTL}^{\Box,\bigcirc}_{\alpha}$, $\mathsf{LTL}^{\Box}_{\alpha}$, $\mathsf{LTL}^{\boxtimes}_{\alpha}$, for $\alpha \in \{bool, horn, krom, core\}$. For any $\mathcal{L}$-formula φ, one can construct, in log-space, an equisatisfiable $\mathcal{L}$-formula*

$$\Psi \;\wedge\; \boxtimes \Phi, \tag{4}$$

where Ψ is a conjunction of propositional variables from Φ, and Φ is a conjunction of clauses of the form (3) containing only $\bigcirc_F$, $\Box_P$, $\Box_F$ for $\mathsf{LTL}^{\Box,\bigcirc}_{\alpha}$, only $\Box_P$, $\Box_F$ for $\mathsf{LTL}^{\Box}_{\alpha}$, and only $\boxtimes$ for $\mathsf{LTL}^{\boxtimes}_{\alpha}$, in which the temporal operators are not nested.

Proof. First, we take a fresh variable p and replace all the conjuncts of the form λ and $\neg\lambda$ in φ by $\boxtimes(\neg p \vee \lambda)$ and $\boxtimes(\neg p \vee \neg\lambda)$, respectively; we set $\Psi = p$. For an $\mathsf{LTL}^{\Box,\bigcirc}_{\alpha}$ or $\mathsf{LTL}^{\Box}_{\alpha}$-formula, we replace the temporal literals $\boxtimes\, \lambda$ with $\Box_F \Box_P \lambda$. Then, for each $\bigcirc_P \lambda$, we take a fresh variable, denoted $\overline{\bigcirc_P \lambda}$, replace each occurrence of $\bigcirc_P \lambda$ with $\overline{\bigcirc_P \lambda}$ and add the conjuncts $\boxtimes(\bigcirc_F \overline{\bigcirc_P \lambda} \to \lambda)$ and $\boxtimes(\lambda \to \bigcirc_F \overline{\bigcirc_P \lambda})$ to the resulting formula. In a similar manner, we use fresh propositional variables as abbreviations for nested temporal operators and obtain the required equisatisfiable formula. Clearly, this can be done in logarithmic space. ❑

We now characterise the structure of interpretations satisfying formulas φ^* of the form (4) in a way similar to other known descriptions of temporal models; see, e.g., [16, 17]. This characterisation will be used in the upper bound proofs of

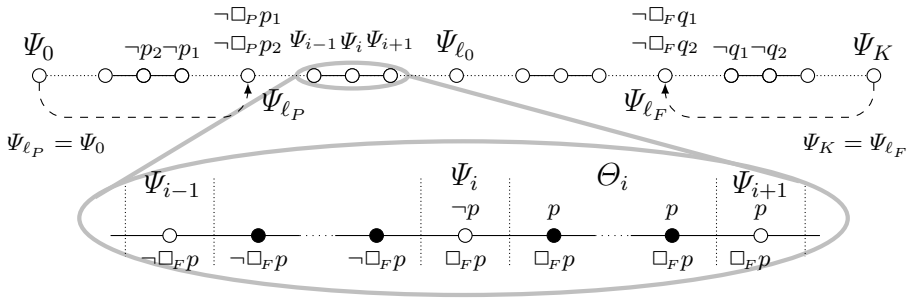

Fig. 1. The structure of a model in Lemma 3.

Theorems 1 and 3. For each $\Box_F p$ in Φ, we take a fresh propositional variable, $\overline{\Box_F p}$, and call it the *surrogate* of $\Box_F p$; likewise, for each $\Box_P p$ in Φ we take its surrogate $\overline{\Box_P p}$. Let $\overline{\Phi}$ be the result of replacing all the $\Box$-literals in Φ with their surrogates. By a *type* for $\overline{\Phi}$ we mean any set of literals that contains either p or $\neg p$ (but not both), for each variable p in $\overline{\Phi}$ (including the surrogates).

The proof of the following lemma is standard and can be found in [6]. The reader may find useful Fig. 1 illustrating the conditions of the lemma.

Lemma 3 (structure of models). *Let φ be an $\mathsf{LTL}^{\Box,\bigcirc}_{bool}$-formula of the form (4) and $K = |\varphi|+4$. Then φ is satisfiable iff there exist integers $m_0 < m_1 < \dots < m_K$ and types $\Psi_0, \Psi_1, \dots, \Psi_K$ for $\overline{\Phi}$ such that*:

($\mathbf{B}_0$) $m_{i+1} - m_i < 2^{|\overline{\Phi}|}$, *for* $0 \le i < K$;

($\mathbf{B}_1$) *there exists* ℓ_0, $0 < \ell_0 < K$, *such that* $\Psi \subseteq \Psi_{\ell_0}$;

($\mathbf{B}_2$) $\overline{\Box_F p} \in \Psi_i \Rightarrow p, \overline{\Box_F p} \in \Psi_{i+1}$ *and* $\overline{\Box_F p} \in \Psi_{i+1} \setminus \Psi_i \Rightarrow p \notin \Psi_{i+1}$ $(0 \le i < K)$,
$\overline{\Box_P p} \in \Psi_i \Rightarrow p, \overline{\Box_P p} \in \Psi_{i-1}$ *and* $\overline{\Box_P p} \in \Psi_{i-1} \setminus \Psi_i \Rightarrow p \notin \Psi_{i-1}$ $(0 < i \le K)$;

($\mathbf{B}_3$) *there exist* $\ell_F < K$ *and* $\ell_P > 0$ *such that*
- $\Psi_{\ell_F} = \Psi_K$ *and, for each* $\neg\overline{\Box_F p} \in \Psi_{\ell_F}$, *there is* $j \ge \ell_F$ *with* $\neg p \in \Psi_j$,
- $\Psi_{\ell_P} = \Psi_0$ *and, for each* $\neg\overline{\Box_P p} \in \Psi_{\ell_P}$, *there is* $j \le \ell_P$ *with* $\neg p \in \Psi_j$;

($\mathbf{B}_4$) *the following formulas are consistent, for* $0 \le i < K$:

$$\psi_i \;=\; \Psi_i \quad\wedge\quad \bigwedge_{k=1}^{m_{i+1}-m_i-1} \bigcirc_F^k \Theta_i \quad\wedge\quad \bigcirc_F^{m_{i+1}-m_i} \Psi_{i+1} \quad\wedge\quad \circledast\, \overline{\Phi},$$

where $\bigcirc_F^k \Psi$ is the result of attaching k operators $\bigcirc_F$ to each literal in Ψ and

$$\Theta_i \;=\; \big\{p,\ \overline{\Box_F p} \mid \overline{\Box_F p} \in \Psi_i\big\} \cup \big\{\neg\overline{\Box_F p} \mid \neg\overline{\Box_F p} \in \Psi_i\big\} \cup \big\{p,\ \overline{\Box_P p} \mid \overline{\Box_P p} \in \Psi_{i+1}\big\} \cup \big\{\neg\overline{\Box_P p} \mid \neg\overline{\Box_P p} \in \Psi_{i+1}\big\}.$$

The intuition behind this lemma is as follows (see Fig. 1). If φ is satisfiable, then it has a model $\mathfrak{M}$ that consists of the initial fragments of models $\mathfrak{M}_i$ of

the formulas ψ_i: namely, the types of the moments $m_i, \dots, m_{i+1}$ in $\mathfrak{M}$ coincide with the types of the moments $0, \dots, (m_{i+1} - m_i)$ in $\mathfrak{M}_i$. By $(\mathbf{B}_4)$, we have $\mathfrak{M}, 0 \models \circledast \overline{\Phi}$. Then $(\mathbf{B}_1)$ makes sure that $\mathfrak{M}, 0 \models \Psi$. Conditions $(\mathbf{B}_2)$ and $(\mathbf{B}_3)$ guarantee that if $\overline{\Box_{\scriptscriptstyle F} p} \in \Psi_i$ then $p \in \Psi_j$ for all types Ψ_j located to the right of Ψ_i in Fig. 1 and, conversely, if $\overline{\Box_{\scriptscriptstyle F} p} \notin \Psi_i$ then $\neg p \in \Psi_j$, for some Ψ_j to the right of Ψ_i; and symmetrically for the $\Box_{\scriptscriptstyle P}$-literals. It follows that $\mathfrak{M}, 0 \models \circledast \Phi$.

3 Binary-Clause LTL and Arithmetic Progressions

In this section, we prove NP-completeness of the satisfiability problem for $\mathsf{LTL}^{\Box,\bigcirc}_{krom}$ and $\mathsf{LTL}^{\Box,\bigcirc}_{core}$. The key ingredient of the proof of the upper bound is an encoding of condition $(\mathbf{B}_4)$ for *binary clauses* by means of arithmetic progressions (via unary automata). The proof of the lower bound is by reduction of the problem whether a given set of arithmetic progressions covers all the natural numbers.

Let φ be an $\mathsf{LTL}^{\Box,\bigcirc}_{krom}$-formula of the form (4). By Lemma 3, to check satisfiability of φ it suffices to guess $K + 1$ types for $\overline{\Phi}$ and K natural numbers $n_i = m_{i+1} - m_i$, for $0 \leq i < K$, whose binary representation, by $(\mathbf{B}_0)$, is polynomial in $|\overline{\Phi}|$. Evidently, $(\mathbf{B}_1)$–$(\mathbf{B}_3)$ can be checked in polynomial time. Our aim now is to show that $(\mathbf{B}_4)$ can also be verified in polynomial time, which will give a nondeterministic polynomial-time algorithm for checking satisfiability of $\mathsf{LTL}^{\Box,\bigcirc}_{krom}$-formulas.

Theorem 1. *The satisfiability problem for* $\mathsf{LTL}^{\Box,\bigcirc}_{krom}$*-formulas is in* NP.

Proof. In view of Lemma 2, we write $\bigcirc$ in place of $\bigcirc_{\scriptscriptstyle F}$. We denote propositional literals (p or $\neg p$) by L and temporal literals (p, $\neg p$, $\bigcirc p$ or $\neg \bigcirc p$) by D. We assume that $\bigcirc \neg p$ is the same as $\neg \bigcirc p$. We use $\psi_1 \models \psi_2$ as a shorthand for '$\mathfrak{M}, 0 \models \psi_2$ whenever $\mathfrak{M}, 0 \models \psi_1$, for any interpretation $\mathfrak{M}$.' Thus, the problem is as follows: given a set Φ of binary clauses of the form $D_1 \vee D_2$, types Ψ and Ψ' for Φ, a set Θ of propositional literals and a number $n > 0$ (in binary), decide whether

$$\Psi \quad \wedge \quad \bigwedge\nolimits_{k=1}^{n-1} \bigcirc^k \Theta \quad \wedge \quad \bigcirc^n \Psi' \quad \wedge \quad \circledast \Phi \tag{5}$$

has a satisfying interpretation. For $0 \leq k \leq n$, we set:

$$F^k_\Phi(\Psi) = \big\{ L' \mid L \wedge \circledast \Phi \models \bigcirc^k L', \text{ for } L \in \Psi \big\},$$
$$P^k_\Phi(\Psi') = \big\{ L \mid \bigcirc^k L' \wedge \circledast \Phi \models L, \text{ for } L' \in \Psi' \big\}.$$

Lemma 4. *Formula* (5) *is satisfiable iff the following conditions hold*:

$(\mathbf{L}_1)$ $F^0_\Phi(\Psi) \subseteq \Psi$, $F^n_\Phi(\Psi) \subseteq \Psi'$ *and* $P^0_\Phi(\Psi') \subseteq \Psi'$, $P^n_\Phi(\Psi') \subseteq \Psi$;
$(\mathbf{L}_2)$ $\neg L \notin F^k_\Phi(\Psi)$ *and* $\neg L \notin P^{n-k}_\Phi(\Psi')$, *for all* $L \in \Theta$ *and* $0 < k < n$.

Proof. Clearly, if (5) is satisfiable then the above conditions hold. For the converse direction, observe that if $L' \in F^k_\Phi(\Psi)$ then, since Φ is a set of binary clauses, there is a sequence of $\bigcirc$-prefixed literals $\bigcirc^{k_0} L_0 \rightsquigarrow \bigcirc^{k_1} L_1 \rightsquigarrow \dots \rightsquigarrow \bigcirc^{k_m} L_m$ such that

$k_0 = 0$, $L_0 \in \Psi$, $k_m = k$, $L_m = L'$, each k_i is between 0 and n and the $\rightsquigarrow$ relation is defined by taking $\bigcirc^{k_i} L_i \rightsquigarrow \bigcirc^{k_{i+1}} L_{i+1}$ just in one of the three cases: $k_{i+1} = k_i$ and $L_i \to L_{i+1} \in \Phi$ or $k_{i+1} = k_i + 1$ and $L_i \to \bigcirc L_{i+1} \in \Phi$ or $k_{i+1} = k_i - 1$ and $\bigcirc L_i \to L_{i+1} \in \Phi$ (we assume that, for example, $\neg q \to \neg p \in \Phi$ whenever Φ contains $p \to q$). So, suppose conditions $\mathbf{(L_1)}$–$\mathbf{(L_2)}$ hold. We construct an interpretation satisfying (5). By $\mathbf{(L_1)}$, both $\Psi \wedge \circledast \Phi$ and $\bigcirc^n \Psi' \wedge \circledast \Phi$ are consistent. So, let $\mathfrak{M}_\Psi$ and $\mathfrak{M}_{\Psi'}$ be such that $\mathfrak{M}_\Psi, 0 \models \Psi \wedge \circledast \Psi$ and $\mathfrak{M}_\Psi, n \models \Psi' \wedge \circledast \Psi$, respectively. Let $\mathfrak{M}$ be an interpretation that coincides with $\mathfrak{M}_\Psi$ for all moments $k \leq 0$ and with $\mathfrak{M}_{\Psi'}$ for all $k \geq n$; for the remaining k, $0 < k < n$, it is defined as follows. First, for each $p \in \Theta$, we make p true at k and, for each $\neg p \in \Theta$, we make p false at k; such an assignment exists due to $\mathbf{(L_2)}$. Second, we extend the assignment by making L true at k if $L \in F^k_\Phi(\Psi) \cup P^{n-k}_\Phi(\Psi')$. Observe that we have $\{p, \neg p\} \not\subseteq F^k_\Phi(\Psi) \cup P^{n-k}_\Phi(\Psi')$: for otherwise $L \wedge \circledast \Phi \models \bigcirc^k p$ and $\bigcirc^{n-k} L' \wedge \circledast \Phi \models \neg p$, for some $L \in \Psi$ and $L' \in \Psi'$, whence $L \wedge \circledast \Phi \models \bigcirc^n \neg L'$, contrary to $\mathbf{(L_1)}$. Also, by $\mathbf{(L_2)}$, any assignment extension at this stage does not contradict the choices made due to Θ. Finally, all propositional variables not covered in the previous two cases get their values from $\mathfrak{M}_\Psi$ (or $\mathfrak{M}_{\Psi'}$). We note that the last choice does not depend on the assignment that is fixed by taking account of the consequences of $\boxplus \Phi$ with Ψ, Ψ' and Θ (because if the value of a variable depended on those sets of literals, the respective literal would be among the logical consequences and would have been fixed before). ❑

Thus, it suffices to show that conditions $\mathbf{(L_1)}$ and $\mathbf{(L_2)}$ can be checked in polynomial time. First, we claim that there is a polynomial-time algorithm which, given a set Φ of binary clauses of the form $D_1 \vee D_2$, constructs a set Φ^* of binary clauses that is 'sound and complete' in the following sense:

$\mathbf{(S_1)}$ $\boxplus \Phi^* \models \boxplus \Phi$;

$\mathbf{(S_2)}$ if $\boxplus \Phi \models \boxplus(L \to \bigcirc^k L_k)$ then either $k = 0$ and $L \to L_0 \in \Phi^*$, or $k \geq 1$ and there are $L_0, L_1, \ldots, L_{k-1}$ with $L = L_0$ and $L_i \to \bigcirc L_{i+1} \in \Phi^*$, for $0 \leq i < k$.

Intuitively, the set Φ^* makes explicit the consequences of $\boxplus \Phi$ and can be constructed in time $(2|\Phi|)^2$ (the number of temporal literals in Φ^* is bounded by the doubled length $|\Phi|$ of Φ as each of its literal can only be prefixed by $\bigcirc$). Indeed, we start from Φ and, at each step, add $D_1 \vee D_2$ to Φ if it contains both $D_1 \vee D$ and $\neg D \vee D_2$; we also add $L_1 \vee L_2$ if Φ contains $\bigcirc L_1 \vee \bigcirc L_2$ (and *vice versa*). This procedure is sound since we only add consequences of $\boxplus \Phi$; completeness follows from the completeness proof for temporal resolution [15, Section 6.3].

Our next step is to encode Φ^* by means of unary automata. Let L, L' be literals. Consider a nondeterministic finite automaton $\mathfrak{A}_{L,L'}$ over $\{0\}$ such that the literals of Φ^* are its states, with L being the initial state and L' the only accepting state, and $\big\{(L_1, L_2) \mid L_1 \to \bigcirc L_2 \in \Phi^*\big\}$ is its transition relation. By $\mathbf{(S_1)}$ and $\mathbf{(S_2)}$, for all $k > 0$, we have

$$\mathfrak{A}_{L,L'} \text{ accepts } 0^k \quad \text{iff} \quad \boxplus \Phi \models \boxplus(L \to \bigcirc^k L').$$

Then both $F^k_\Phi(\Psi)$ and $P^k_\Phi(\Psi')$ can be defined in terms of the language of $\mathfrak{A}_{L,L'}$:

$$F^k_\Phi(\Psi) = \big\{L' \mid \mathfrak{A}_{L,L'} \text{ accepts } 0^k, \text{ for } L \in \Psi\big\},$$
$$P^k_\Phi(\Psi') = \big\{L \mid \mathfrak{A}_{\neg L,\neg L'} \text{ accepts } 0^k, \text{ for } L' \in \Psi'\big\}$$

(recall that $\bigcirc^k L' \to L$ is equivalent to $\neg L \to \bigcirc^k \neg L'$). Note that the numbers n and k in conditions $\mathbf{(L_1)}$ and $\mathbf{(L_2)}$ are in general exponential in the length of Φ and, therefore, the automata $\mathfrak{A}_{L,L'}$ do not immediately provide a polynomial-time procedure for checking these conditions: although it can be shown that if $\mathbf{(L_2)}$ does not hold then it fails for a polynomial number k, this is not the case for $\mathbf{(L_1)}$, which requires the accepting state to be reached in a fixed (exponential) number of transitions. Instead, we use the *Chrobak normal form* [11] to decompose the automata into a polynomial number of polynomial-sized arithmetic progressions (which can have an exponential common period; cf. the proof of Theorem 2). In what follows, given a and b, we denote by $a + b\mathbb{N}$ the set $\{a + bm \mid m \in \mathbb{N}\}$ (the arithmetic progression with initial term a and common difference b).

It is known that every N-state unary automaton $\mathfrak{A}$ can be converted (in polynomial time) into an equivalent automaton in Chrobak normal form (e.g., by using Martinez's algorithm [28]), which has $O(N^2)$ states and gives rise to M arithmetic progressions $a_1 + b_1\mathbb{N}, \dots, a_M + b_M\mathbb{N}$ such that

$\mathbf{(A_1)}$ $M \leq O(N^2)$ and $0 \leq a_i, b_i \leq N$, for $1 \leq i \leq M$;
$\mathbf{(A_2)}$ $\mathfrak{A}$ accepts 0^k iff $k \in a_i + b_i\mathbb{N}$, for some $1 \leq i \leq M$.

By construction, the number of arithmetic progressions is bounded by a quadratic function in the length of Φ.

We are now in a position to give a polynomial-time algorithm for checking $\mathbf{(L_1)}$ and $\mathbf{(L_2)}$, which requires solving Diophantine equations. In $\mathbf{(L_2)}$, for example, to verify that, for each $p \in \Theta$, we have $\neg p \notin F^k_\Phi(\Psi)$, for all $0 < k < n$, we take the automata $\mathfrak{A}_{L,\neg p}$, for $L \in \Psi$, and transform them into the Chrobak normal form to obtain arithmetic progressions $a_i + b_i\mathbb{N}$, for $1 \leq i \leq M$. Then there is k, $0 < k < n$, with $\neg p \in F^k_\Phi(\Psi)$ iff one of the equations $a_i + b_i m = k$ has an integer solution, for some k, $0 < k < n$. The latter can be verified by taking the integer $m = \lfloor -a_i/b_i \rfloor$ and checking whether either $a_i + b_i m$ or $a_i + b_i(m+1)$ belongs to the open interval $(0, n)$, which can clearly be done in polynomial time.

This completes the proof of Theorem 1. ❑

The matching lower bound for $\mathsf{LTL}^{\Box,\bigcirc}_{core}$-formulas, even without $\Box_F/\Box_P$, can be obtained using NP-hardness of deciding inequality of regular languages over a unary alphabet [27]. In the proof of Theorem 2, we give a more direct reduction of the NP-complete problem 3SAT and repeat the argument of [27, Theorem 6.1] to construct a small number of arithmetic progressions (each with a small initial term and common difference) that give rise to models of exponential size.

Theorem 2. *The satisfiability problem for* $\mathsf{LTL}^{\Box,\bigcirc}_{core}$*-formulas is* NP*-hard.*

Proof. The proof is by reduction of 3SAT. Let $f = \bigwedge_{i=1}^{n} C_i$ be a 3CNF with variables $p_1, \dots, p_m$ and clauses $C_1, \dots, C_n$. By a propositional assignment for f we

	1	2	3	4	5	**6**	7	8	9	**10**	11	12	13	14	**15**	**16**	17	18	19	20	**21**	22	23	24	**25**	26	27	28	29	**30**
2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
3	1		0	1		0	1		0	1		0	1		0	1			0	1		0	1		0	1			0	1	0
5	1				0	1				0	1				0	1				0	1				0	1				0

Fig. 2. Positive numbers encoding assignments for 3 variables p_1, p_2, p_3 (shaded).

understand a function $\sigma\colon \{p_1, \dots, p_m\} \to \{0, 1\}$. We represent such assignments by sets of positive natural numbers. More precisely, let $P_1, \dots, P_m$ be the first m prime numbers; it is known that P_m does not exceed $O(m^2)$ [1]. A natural number $k > 0$ is said to *represent* an assignment σ if k is equivalent to $\sigma(p_i)$ modulo P_i, for all i, $1 \le i \le m$. Clearly, not every natural number represents an assignment since each element of

$$j + P_i \cdot \mathbb{N}, \qquad \text{for } 1 \le i \le m \text{ and } 2 \le j < P_i, \tag{6}$$

is equivalent to j modulo P_i with $j \ge 2$. On the other hand, every natural number that does not represent an assignment belongs to one of those arithmetic progressions (see Fig. 2).

Let C_i be a clause in f, say, $C_i = p_{i_1} \vee \neg p_{i_2} \vee p_{i_3}$. Consider

$$P_{i_1}^1 P_{i_2}^0 P_{i_3}^1 + P_{i_1} P_{i_2} P_{i_3} \cdot \mathbb{N}. \tag{7}$$

A natural number represents an assignment that makes C_i true iff it does not belong to the progressions (6) and (7). In the same way we construct a progression of the form (7) for every clause in f. Thus, a natural number $k > 0$ *does not* belong to the constructed progressions of the form (6) and (7) iff k represents a satisfying assignment for f.

To complete the proof, we show that the defined progressions can be encoded in $\mathsf{LTL}_{core}^{\Box,\bigcirc}$. Take a propositional variable d (it will be shared by all formulas below). Given an arithmetic progression $a + b\mathbb{N}$ (with $a \ge 0$ and $b > 0$), let

$$\begin{aligned} \theta_{a,b} \;=\;& u_0 \wedge \bigwedge\nolimits_{j=1}^{a} \circledast(u_{j-1} \to \bigcirc_F u_j) \wedge \\ & \circledast(u_a \to v_0) \wedge \bigwedge\nolimits_{j=1}^{b} \circledast(v_{j-1} \to \bigcirc_F v_j) \wedge \circledast(v_b \to v_0) \wedge \circledast(v_0 \to d), \end{aligned}$$

where $u_0, \dots, u_a$ and $v_0, \dots, v_b$ are fresh propositional variables. It is not hard to see that, in every model of $\theta_{a,b}$, if k belongs to $a + b\mathbb{N}$, then d is true at moment k. Thus, we take a conjunction φ_f of the $\theta_{a,b}$ for arithmetic progressions (6) and (7) together with $p \wedge \circledast(\bigcirc_F p \to p) \wedge \circledast(p \to d) \wedge \circledast(\neg \circledast d)$, where p is a fresh variable (the last formula makes both p and d true at all moments $k \le 0$). The size of the $\mathsf{LTL}_{core}^{\Box,\bigcirc}$-formula φ_f is $O(n \cdot m^6)$. It is readily checked that φ_f is satisfiable iff f is satisfiable. ❑

4 Core and Horn Fragments without Next-Time

Let φ be an $\mathsf{LTL}_{horn}^{\Box}$-formula. By applying Lemma 2, we can transform φ to the form $\Psi \wedge \circledast \Phi^+ \wedge \circledast \Phi^-$, where Ψ is a set of propositional variables while Φ^+ and

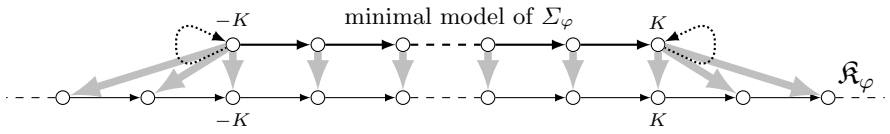

Fig. 3. The minimal model of Σ_φ and $\mathfrak{K}_\varphi$.

Φ^- are sets of *positive* and *negative clauses* of the form

$$\lambda_1 \wedge \lambda_2 \wedge \dots \wedge \lambda_{k-1} \to \lambda_k \quad \text{and} \quad \neg\lambda_1 \vee \neg\lambda_2 \vee \dots \vee \neg\lambda_k, \tag{8}$$

respectively. Trivially, $\Psi \wedge \circledast\, \Phi^+$ is satisfiable. Since all clauses in Φ^+ have at most one positive literal and are constructed from variables possibly prefixed by $\Box_{\scriptscriptstyle F}$ or $\Box_{\scriptscriptstyle P}$, the formula $\Psi \wedge \circledast\, \Phi^+$ has a *canonical model* $\mathfrak{K}_\varphi$ defined by taking

$$\mathfrak{K}_\varphi, n \models p \quad \text{iff} \quad \mathfrak{M}, n \models p, \quad \text{for every model } \mathfrak{M} \text{ of } \Psi \wedge \circledast\, \Phi^+,\ n \in \mathbb{Z}$$

(indeed, $\mathfrak{K}_\varphi, 0 \models \Psi \wedge \circledast\, \Phi^+$ follows from the observation that $\mathfrak{K}_\varphi, n \models \Box_{\scriptscriptstyle F} p$ iff $\mathfrak{M}, n \models \Box_{\scriptscriptstyle F} p$, for every model $\mathfrak{M}$ of $\Psi \wedge \circledast\, \Phi^+$; and similarly for $\Box_{\scriptscriptstyle P} p$). If we consider the canonical model $\mathfrak{K}_\varphi$ in the context of Lemma 3 then, since the language does not contain $\bigcirc_{\scriptscriptstyle F}$ or $\bigcirc_{\scriptscriptstyle P}$, we have $m_{i+1} - m_i = 1$ for all i. Thus, $\mathfrak{K}_\varphi$ can be thought of as a sequence of $(\ell_{\scriptscriptstyle F} - \ell_{\scriptscriptstyle P} + 1)$-many states, the first and last of which repeat indefinitely. Let $K = |\varphi| + 4$.

Obviously, φ is satisfiable iff there is no negative clause $\neg\lambda_1 \vee \dots \vee \neg\lambda_k$ in Φ^- such that all the λ_i are true in $\mathfrak{K}_\varphi$ at some moment n with $|n| \leq K$. This condition can be encoded by means of propositional Horn clauses in the following way. For each variable p, we take $2K + 1$ variables p^n, $|n| \leq K$, and, for each $\Box_{\scriptscriptstyle F} p$ and $\Box_{\scriptscriptstyle P} p$, we take $2K + 1$ variables, denoted $(\Box_{\scriptscriptstyle F} p)^n$ and $(\Box_{\scriptscriptstyle P} p)^n$, $|n| \leq K$, respectively. Consider the following set Σ_φ of propositional Horn clauses, $|n| \leq K$:

$(\mathbf{H}_0)$ p^0, if $p \in \Psi$,

$(\mathbf{H}_1)$ $\lambda_1^n \wedge \dots \wedge \lambda_{k-1}^n \to \lambda_k^n$, if $(\lambda_1 \wedge \dots \wedge \lambda_{k-1} \to \lambda_k) \in \Phi^+$,

$(\mathbf{H}_2)$ $(\Box_{\scriptscriptstyle F} p)^n \to (\Box_{\scriptscriptstyle F} p)^{n+1}$ if $n < K$, $\quad (\Box_{\scriptscriptstyle P} p)^n \to (\Box_{\scriptscriptstyle P} p)^{n-1}$ if $n > -K$,

$(\mathbf{H}_3)$ $(\Box_{\scriptscriptstyle F} p)^n \to p^{n+1}$, $\quad (\Box_{\scriptscriptstyle P} p)^n \to p^{n-1}$,

$(\mathbf{H}_4)$ $(\Box_{\scriptscriptstyle F} p)^n \wedge p^n \to (\Box_{\scriptscriptstyle F} p)^{n-1}$ if $n > -K$, $\quad (\Box_{\scriptscriptstyle P} p)^n \wedge p^n \to (\Box_{\scriptscriptstyle P} p)^{n+1}$ if $n < K$,

$(\mathbf{H}_5)$ $(\Box_{\scriptscriptstyle F} p)^K \leftrightarrow p^K$, $\quad (\Box_{\scriptscriptstyle P} p)^{-K} \leftrightarrow p^{-K}$,

$(\mathbf{H}_6)$ $(\Box_{\scriptscriptstyle F} p)^{-K} \leftrightarrow p^{-K}$, $\quad (\Box_{\scriptscriptstyle P} p)^K \leftrightarrow p^K$.

Clearly, $|\Sigma_\varphi| \leq O(|\varphi|^2)$. It is readily seen that the minimal model of Σ_φ corresponds to the canonical model $\mathfrak{K}_\varphi$ as shown in Fig. 3. As propositional Horn satisfiability is PTIME-complete, we obtain the following:

Theorem 3. *The satisfiability problem for* $\mathsf{LTL}^{\Box}_{horn}$*-formulas is in* PTIME.

4.1 Temporal Derivations for $\mathsf{LTL}^{\Box}_{core}$ in NLogSpace

In $\mathsf{LTL}^{\Box}_{core}$-formulas, all clauses are binary: $k = 2$ in (8). Satisfiability of propositional binary clauses is known to be NLogSpace-complete. However, in the reduction $\varphi \mapsto \Sigma_\varphi$ above, the clauses $\mathbf{(H_4)}$ are ternary. In this section we show how to modify the reduction to ensure membership in NLogSpace. More precisely, we define two types of derivation from $\Psi \wedge \circledast\, \Phi^+$: a 0-derivation of (λ, n) will mean that $\mathfrak{K}_\varphi, n \models \lambda$, while a $\forall$-derivation of λ from λ' that $\mathfrak{K}_\varphi, 0 \models \circledast\, \lambda' \to \circledast\, \lambda$. We then show that these derivations define $\mathfrak{K}_\varphi$ and that satisfiability of φ can be checked by a nondeterministic algorithm in logarithmic space.

Denote by $\to^*$ the transitive and reflexive closure of the relation $\to$ over literals given by the clauses of Φ^+. We require the following derivation rules over the pairs (λ, n), where λ is a positive temporal literal in φ and $n \in \mathbb{Z}$:

$$
\begin{array}{lll}
\mathbf{(R_1)} & (\lambda_1, n) \Rightarrow (\lambda_2, n), \quad \text{if } \lambda_1 \to^* \lambda_2, & \\
\mathbf{(R_2)} & (\Box_{\scriptscriptstyle F} p, n) \Rightarrow (\Box_{\scriptscriptstyle F} p, n+1), & (\Box_{\scriptscriptstyle P} p, n) \Rightarrow (\Box_{\scriptscriptstyle P} p, n-1), \\
\mathbf{(R_3)} & (\Box_{\scriptscriptstyle F} p, n) \Rightarrow (p, n+1), & (\Box_{\scriptscriptstyle P} p, n) \Rightarrow (p, n-1), \\
\mathbf{(R_4)} & (\Box_{\scriptscriptstyle F} p, 0) \Rightarrow (\Box_{\scriptscriptstyle F} p, -1), & (\Box_{\scriptscriptstyle P} p, 0) \Rightarrow (\Box_{\scriptscriptstyle P} p, 1), \quad \text{if } p' \to^* p \text{ for } p' \in \Psi, \\
\mathbf{(R_5)} & (p, n) \Rightarrow (\Box_{\scriptscriptstyle F} p, n-1), & (p, n) \Rightarrow (\Box_{\scriptscriptstyle P} p, n+1).
\end{array}
$$

The rules in $\mathbf{(R_1)}$–$\mathbf{(R_4)}$ mimic $\mathbf{(H_1)}$–$\mathbf{(H_4)}$ above ($\mathbf{(H_4)}$ at moment 0 only) and reflect the semantics of LTL in the sense that whenever $(\lambda, n) \Rightarrow (\lambda', n')$ and $\mathfrak{K}_\varphi, n \models \lambda$ then $\mathfrak{K}_\varphi, n' \models \lambda'$. For example, consider $\mathbf{(R_4)}$. It only applies if p follows (by $\to^*$) from the initial conditions in Ψ, in which case $\mathfrak{K}_\varphi, 0 \models p$, and so $\mathfrak{K}_\varphi, 0 \models \Box_{\scriptscriptstyle F} p$ implies $\mathfrak{K}_\varphi, -1 \models \Box_{\scriptscriptstyle F} p$. The rules in $\mathbf{(R_5)}$ are different: for instance, we can only apply $(p, n) \Rightarrow (\Box_{\scriptscriptstyle F} p, n-1)$ if we know that p holds at all $m \geq n$.

A sequence $\mathfrak{d}\colon (\lambda_0, n_0) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell)$, for $\ell \geq 0$, is called a 0-*derivation of* (λ_ℓ, n_ℓ) if $\lambda_0 \in \Psi$, $n_0 = 0$ and all applications of $\mathbf{(R_5)}$ are *safe* in the following sense: for any $(p, n_i) \Rightarrow_{\mathbf{(R_5)}} (\Box_{\scriptscriptstyle F} p, n_i - 1)$, there is $\lambda_j = \Box_{\scriptscriptstyle F} q$, for some q and $0 \leq j < i$; similarly, for any $(p, n_i) \Rightarrow_{\mathbf{(R_5)}} (\Box_{\scriptscriptstyle P} p, n_i + 1)$, there is $\lambda_j = \Box_{\scriptscriptstyle P} q$ with $0 \leq j < i$. In this case we write $\Psi \Rightarrow^0 (\lambda_\ell, n_\ell)$. For example, consider

$$\varphi \quad = \quad p \quad \wedge \quad \circledast(p \to \Box_{\scriptscriptstyle F} q) \quad \wedge \quad \circledast(q \to r) \quad \wedge \quad \circledast(p \to r).$$

Evidently, $\mathfrak{K}_\varphi, -1 \models \Box_{\scriptscriptstyle F} r$. The following sequence is a 0-derivation of $(\Box_{\scriptscriptstyle F} r, -1)$ because the application of $\mathbf{(R_5)}$ is safe due to $\Box_{\scriptscriptstyle F} q$:

$$(p, 0) \Rightarrow_{\mathbf{(R_1)}} (\Box_{\scriptscriptstyle F} q, 0) \Rightarrow_{\mathbf{(R_3)}} (q, 1) \Rightarrow_{\mathbf{(R_1)}} (r, 1) \Rightarrow_{\mathbf{(R_5)}} (\Box_{\scriptscriptstyle F} r, 0) \Rightarrow_{\mathbf{(R_4)}} (\Box_{\scriptscriptstyle F} r, -1).$$

Intuitively, if we can derive $(r, 1)$ using $(\Box_{\scriptscriptstyle F} q, 0)$, then we can also derive (r, n) for any $n \geq 1$, and so we must also have $(\Box_{\scriptscriptstyle F} r, 0)$, which justifies the application of $\mathbf{(R_5)}$. This argument is formalised in the following lemma:

Lemma 5 (monotonicity). *Let $\mathfrak{d}$ be a 0-derivation of (λ_ℓ, n_ℓ) with a suffix*

$$\mathfrak{s}\colon (\Box_{\scriptscriptstyle F} q, n_s) \Rightarrow (\lambda_{s+1}, n_{s+1}) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell), \tag{9}$$

where none of the λ_i contains $\Box_{\scriptscriptstyle F}$. Then $\Psi \Rightarrow^0 (\lambda_\ell, m)$, for all $m \geq n_\ell$. Similarly, if there is a suffix beginning with some $\Box_{\scriptscriptstyle P} q$ then $\Psi \Rightarrow^0 (\lambda_\ell, m)$, for all $m \leq n_\ell$. Moreover, these 0-derivations only contain the rules used in $\mathfrak{d}$ and $\mathbf{(R_2)}$.

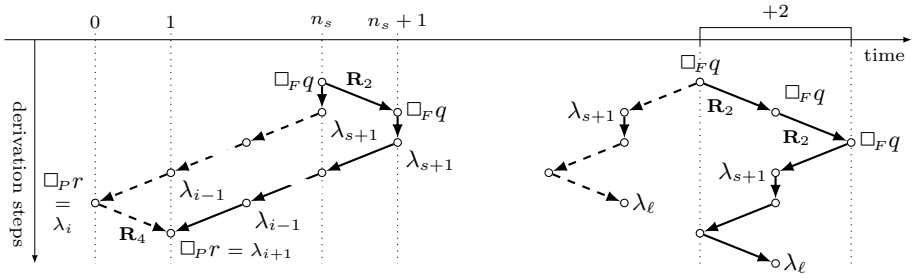

Fig. 4. Removing applications of $(\mathbf{R}_4)$ (left) and shifting a 0-derivation by 2 (right): dashed arrows show the original derivation and solid ones the resulting derivation.

Proof. We first remove all applications of $(\mathbf{R}_4)$ in $\mathfrak{s}$. Let $(\lambda_i, n_i) \Rightarrow_{(\mathbf{R}_4)} (\lambda_{i+1}, n_{i+1})$ be the first one. By definition, $n_i = 0$ and, since $\Box_F q$ is the last $\Box_F$ in $\mathfrak{d}$, we have $n_{i+1} = 1$ and $\lambda_i = \lambda_{i+1} = \Box_P r$, for some r. So we can begin $\mathfrak{s}$ with $(\Box_F q, n_s) \Rightarrow_{(\mathbf{R}_2)} (\Box_F q, n_s{+}1) \Rightarrow (\lambda_{s+1}, n_{s+1}{+}1) \Rightarrow \cdots \Rightarrow (\lambda_i, n_i{+}1) \Rightarrow (\lambda_{i+2}, n_{i+2})$; see Fig. 4 on the left-hand side. We repeatedly apply this operation to obtain a suffix $\mathfrak{s}$ of the form (9) that does not use $(\mathbf{R}_4)$. We then replace $\mathfrak{s}$ in $\mathfrak{d}$ with $(\Box_F q, n_s) \Rightarrow_{(\mathbf{R}_2)} \cdots \Rightarrow_{(\mathbf{R}_2)} (\Box_F q, n_s + k) \Rightarrow (\lambda_{s+1}, n_{s+1} + k) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell + k)$, where $k = m - n_\ell$; see Fig. 4 on the right-hand side. ❑

However, 0-derivations are not enough to obtain all literals that are true in $\mathfrak{K}_\varphi$. Indeed, consider the formula

$$\varphi \quad = \quad r \quad \wedge \quad \circledast(r \to \Box_F q) \quad \wedge \quad \circledast(\Box_F q \to q) \quad \wedge \quad \circledast(\Box_P q \to p).$$

Clearly, $\mathfrak{K}_\varphi, n \models p$ for all $n \in \mathbb{Z}$, but neither (p, n) nor $(\Box_P q, n)$ is 0-derivable. On the other hand, for each $n \in \mathbb{Z}$, there is a 0-derivation of (q, n): for example,

$$(r, 0) \Rightarrow_{(\mathbf{R}_1)} (\Box_F q, 0) \Rightarrow_{(\mathbf{R}_1)} (q, 0) \Rightarrow_{(\mathbf{R}_5)} (\Box_F q, -1) \Rightarrow_{(\mathbf{R}_1)} (q, -1).$$

These 0-derivations correspond to $\mathfrak{K}_\varphi, 0 \models \circledast\, q$, from which we can derive $\circledast\, p$ by means of the second type of derivations. A sequence $\mathfrak{d}\colon (\lambda_0, n_0) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell)$ is called a $\forall$-*derivation of* λ_ℓ *from* λ_0 if it uses only $(\mathbf{R}_1)$–$(\mathbf{R}_3)$ and $(\mathbf{R}_5)$, whose applications are not necessarily safe. So we write $\Psi \Rightarrow^\forall \lambda$ if there is a $\forall$-derivation of λ from some q such that $\Psi \Rightarrow^0 (q, n)$, for all $n \in \mathbb{Z}$. In the example above, $(q, 0) \Rightarrow_{(\mathbf{R}_5)} (\Box_P q, 1) \Rightarrow_{(\mathbf{R}_1)} (p, 1)$ is a $\forall$-derivation of p from q, whence $\Psi \Rightarrow^\forall p$.

Lemma 6 (soundness). *If* $\Psi \Rightarrow^0 (\lambda, n)$ *then* $\mathfrak{K}_\varphi, n \models \lambda$. *If* $\Psi \Rightarrow^\forall \lambda$ *then* $\mathfrak{K}_\varphi, 0 \models \circledast\, \lambda$.

Proof. By induction on the derivation length, using Lemma 5 for $(\mathbf{R}_5)$. ❑

Lemma 7 (completeness). *If* $\mathfrak{K}_\varphi, n \models \lambda$ *then either* $\Psi \Rightarrow^0 (\lambda, n)$ *or* $\Psi \Rightarrow^\forall \lambda$.

Proof. Let $\mathfrak{M}$ be an interpretation such that, for all p and $n \in \mathbb{Z}$, we have $\mathfrak{M}, n \models p$ iff $\Psi \Rightarrow^0 (p, n)$ or $\Psi \Rightarrow^\forall p$. It suffices to show that $\mathfrak{M}, 0 \models \Psi \wedge \circledast \Phi^+$. Indeed, if we assume that there are p' and n' such that $\mathfrak{K}_\varphi, n' \models p'$ but neither $\Psi \Rightarrow^0 (p', n')$ nor $\Psi \Rightarrow^\forall p'$, we will obtain $\mathfrak{M}, n' \models \neg p'$ contrary to our assumption (other types of literals are considered analogously).

Thus, we have to show that $\mathfrak{M}$ is a model of $\Psi \wedge \circledast \Phi^+$. Suppose $p \in \Psi$. Then trivially $\Psi \Rightarrow^0 (p, 0)$, and so $\mathfrak{M}, 0 \models p$. Suppose $\lambda_1 \to \lambda_2 \in \Phi^+$ and $\mathfrak{M}, n \models \lambda_1$. We consider three cases depending on the shape of λ_1 and show that $\mathfrak{M}, n \models \lambda_2$.

$\lambda_1 = p$. If $\Psi \Rightarrow^\forall p$ then, by $(\mathbf{R}_1)$, $\Psi \Rightarrow^\forall \lambda_2$. Otherwise, there is a 0-derivation of (p, n), and so $\Psi \Rightarrow^0 (\lambda_1, n) \Rightarrow_{(\mathbf{R}_1)} (\lambda_2, n)$.

$\lambda_1 = \Box_F p$. Then $\mathfrak{M}, m \models p$ for all $m > n$. Consider $\mathfrak{M}, n+1 \models p$. If $\Psi \Rightarrow^\forall p$ then, by $(\mathbf{R}_5)$, $(\mathbf{R}_1)$, $\Psi \Rightarrow^\forall \lambda_2$. Otherwise, there is a 0-derivation $\mathfrak{d}$ of $(p, n+1)$.
- (F) If $\Box_F$ occurs in $\mathfrak{d}$ then $\Psi \Rightarrow^0 (p, n+1) \Rightarrow_{(\mathbf{R}_5)} (\Box_F p, n) \Rightarrow_{(\mathbf{R}_1)} (\lambda_2, n)$.
- (P) If $\Box_P$ occurs in $\mathfrak{d}$ then, by Lemma 5, $\Psi \Rightarrow^0 (p, m)$ for each $m \leq n+1$. Thus, $\Psi \Rightarrow^0 (p, m)$ for *all* $m \in \mathbb{Z}$, and so, by $(\mathbf{R}_5)$ and $(\mathbf{R}_1)$, $\Psi \Rightarrow^\forall \lambda_2$.
- (0) If $\mathfrak{d}$ contains neither $\Box_F$ nor $\Box_P$ then $n = -1$ and $\lambda \to^* p$, for some $\lambda \in \Psi$ (by $(\mathbf{R}_1)$). As $\mathfrak{M}, 1 \models p$ and we assumed $\Psi \not\Rightarrow^\forall p$, there is a 0-derivation $\mathfrak{d}'$ of $(p, 1)$, which must contain $\Box_F$ or $\Box_P$. If $\mathfrak{d}'$ contains $\Box_F$ then $\Psi \Rightarrow^0 (p, 1) \Rightarrow_{(\mathbf{R}_5)} (\Box_F p, 0) \Rightarrow_{(\mathbf{R}_4)} (\Box_F p, -1) \Rightarrow_{(\mathbf{R}_1)} (\lambda_2, n)$. If $\Box_P$ occurs in $\mathfrak{d}'$ then, by the argument in (P), $\Psi \Rightarrow^\forall \lambda_2$.

$\lambda_1 = \Box_P p$. The proof is symmetric.

In each of these cases, we have either $\Psi \Rightarrow^0 (\lambda_2, n)$ or $\Psi \Rightarrow^\forall \lambda_2$. Observe that $\Psi \Rightarrow^0 (\lambda_2, n)$ implies $\mathfrak{M}, n \models \lambda_2$. Indeed, this clearly holds for $\lambda_2 = p$. If $\lambda_2 = \Box_F p$ then, by repetitive applications of $(\mathbf{R}_2)$ and an application of $(\mathbf{R}_3)$, we obtain $\Psi \Rightarrow^0 (p, m)$, for all $m > n$, which means $\mathfrak{M}, n \models \Box_F p$. The case $\lambda_2 = \Box_P p$ is symmetric. If $\Psi \Rightarrow^\forall \lambda_2$ then, independently of whether λ_2 is p', $\Box_F p'$ or $\Box_P p'$, we have $\Psi \Rightarrow^\forall p'$, so $\mathfrak{M}, m \models p'$ for all $m \in \mathbb{Z}$, whence, $\mathfrak{M}, n \models \lambda_2$. ❑

Next, in Lemmas 8 and 9, we provide efficient criteria for checking the conditions $\Psi \Rightarrow^0 (\lambda, n)$ and $\Psi \Rightarrow^\forall \lambda$ by restricting the range of numbers that can be used in 0-derivations (numbers in $\forall$-derivations can simply be ignored). Given a 0-derivation $\mathfrak{d}\colon (\lambda_0, n_0) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell)$, we define its *reach* as

$$r(\mathfrak{d}) = \max\{|n_i| \mid 0 \leq i \leq \ell\}.$$

We say that $\mathfrak{d}$ *right-stutters*, if there are $v < w$ such that $\lambda_v = \lambda_w$, $n_v < n_w$ and $n_i > 0$, for all i, $v \leq i \leq w$ (in particular, $(\mathbf{R}_4)$ is not applied between v and w). Symmetrically, $\mathfrak{d}$ *left-stutters* if there are $v < w$ such that $\lambda_v = \lambda_w$, $n_v > n_w$ and $n_i < 0$, for all i, $v \leq i \leq w$.

Lemma 8 (checking $\Rightarrow^0$). *$\Psi \Rightarrow^0 (\lambda, n)$ iff there exists a 0-derivation $\mathfrak{d}$ of (λ, m) such that $r(\mathfrak{d}) \leq 2|\varphi|$ and one of the following conditions holds*:

$(\mathbf{C}_1)$ $m = n$;
$(\mathbf{C}_2)$ *$\mathfrak{d}$ contains $\Box_F$ and either $m \leq n$ or $\mathfrak{d}$ left-stutters*;
$(\mathbf{C}_3)$ *$\mathfrak{d}$ contains $\Box_P$ and either $m \geq n$ or $\mathfrak{d}$ right-stutters.*

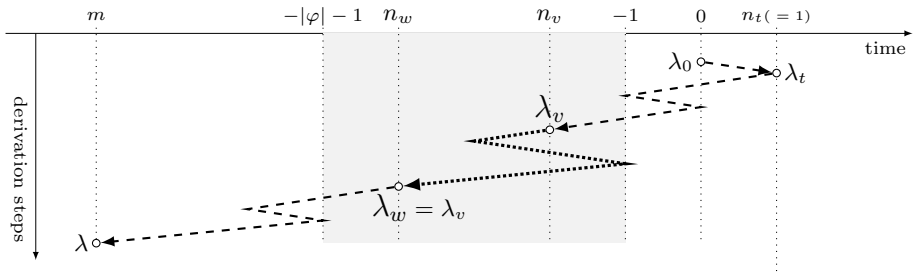

Fig. 5. Left-stuttering: n_v and n_w occur between -1 and $-|\varphi|-1$ (shaded) and the fragment of the derivation from n_v to n_w can be repeated any number of times (incl. 0).

Proof. ($\Rightarrow$) Let $\mathfrak{d}\colon (\lambda_0, n_0) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell)$ be a 0-derivation of (λ, n). If $r(\mathfrak{d}) \leq |\varphi|$ then $\mathfrak{d}$ satisfies $\mathbf{(C_1)}$. Otherwise, we take the first $\Box$-literal in $\mathfrak{d}$, say $\lambda_t = \Box_F q$ (the case of $\Box_P q$ is symmetric). Clearly, $|n_t| \leq 1$. Let $u > t$ be the smallest index with $|n_u| > |\varphi|$. Since adjacent n_i and n_{i+1} differ by at most 1, the segment between (λ_t, n_t) and (λ_u, n_u) contains a repeating literal: more precisely, there exist $v < w$ between t and u such that $\lambda_v = \lambda_w$ and

- either $n_v > n_w$ and $n_i < 0$, for $v \leq i \leq w$,
- or $n_v < n_w$ and $n_i > 0$, for $v \leq i \leq w$.

In the former case $\mathfrak{d}$ left-stutters, and we perform the following operations on the suffix $\mathfrak{s}\colon (\lambda_w, n_w) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell)$ of $\mathfrak{d}$. First, we eliminate all applications of $\mathbf{(R_4)}$ in $\mathfrak{s}$: each suffix $(\Box_F q, 0) \Rightarrow_{\mathbf{(R_4)}} (\Box_F q, -1) \Rightarrow (\lambda_s, n_s) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell)$ is replaced by $(\Box_F q, 0) \Rightarrow (\lambda_s, n_s + 1) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell + 1)$; and similarly for $\Box_P$. If each time we eliminate the last application of $\mathbf{(R_4)}$ then the result is clearly a 0-derivation. Second, we remove all duplicating literals: each suffix $(\lambda_s, n_s) \Rightarrow \cdots \Rightarrow (\lambda_{s'}, n_{s'}) \Rightarrow (\lambda_{s'+1}, n_{s'+1}) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell)$ with $\lambda_s = \lambda_{s'}$ is replaced by $(\lambda_s, n_s) \Rightarrow (\lambda_{s'+1}, n_{s'+1}+k) \Rightarrow \cdots \Rightarrow (\lambda_\ell, n_\ell+k)$, where $k = n_s - n_{s'}$. This will give us a left-stuttering 0-derivation $\mathfrak{d}'$ of (λ, m), for some m. Since there are at most $|\varphi|$ distinct literals in $\mathfrak{s}$, we have $r(\mathfrak{d}') \leq 2|\varphi|$, thus satisfying the second option of $\mathbf{(C_2)}$; see Fig. 5.

In the latter case $\mathfrak{d}$ right-stutters, and we construct a 0-derivation $\mathfrak{d}'$ of (p, n') by cutting out the segment $(\lambda_{v+1}, n_{v+1}) \Rightarrow \cdots \Rightarrow (\lambda_w, n_w)$ from $\mathfrak{d}$ and 'shifting' the tail using the construction above: eliminate applications of $\mathbf{(R_4)}$ and then decrease all numbers by $n_w - n_v > 0$. We then consider the obtained $\mathfrak{d}'$ as the original $\mathfrak{d}$. As the length of the derivations decreases and $n' \leq n$, by applying this procedure sufficiently many times, we shall finally construct a 0-derivation of reach $\leq 2|\varphi|$ and satisfying either $\mathbf{(C_1)}$ or the first option of $\mathbf{(C_2)}$.

($\Leftarrow$) is left to the reader. ❑

In a similar way we can show how to efficiently check the condition $\Psi \Rightarrow^\forall p$:

Lemma 9 (checking $\Rightarrow^\forall$). *$\Psi \Rightarrow^0 (\lambda, n)$ holds for all $n \in \mathbb{Z}$ iff there are 0-derivations $\mathfrak{d}$ of (λ, m) and $\mathfrak{d}'$ of (λ, m') of reach at most $2|\varphi|$ such that one of the following conditions holds*:

$(\mathbf{C}'_1)$ $\mathfrak{d}$ *contains* $\Box_F$, $\mathfrak{d}'$ *contains* $\Box_P$ *and* $m \leq m' + 1$;
$(\mathbf{C}'_2)$ $\mathfrak{d}$ *contains* $\Box_F$ *and left-stutters*;
$(\mathbf{C}'_3)$ $\mathfrak{d}$ *contains* $\Box_P$ *and right-stutters.*

Proof. ($\Rightarrow$) Take a 0-derivation of $(q, 2|\varphi|+1)$. By Lemma 8, there is a derivation $\mathfrak{d}_0$ of (q, n_0) with $r(\mathfrak{d}_0) \leq 2|\varphi|$ satisfying either $(\mathbf{C}_2)$ or $(\mathbf{C}_3)$. If $\mathfrak{d}_0$ left- or right-stutters then we have $(\mathbf{C}'_2)$ or $(\mathbf{C}'_3)$, respectively. Otherwise, $\mathfrak{d}_0$ contains $\Box_F$ and we can construct a finite sequence of 0-derivations $\mathfrak{d}_0, \mathfrak{d}_1, \mathfrak{d}_2, \ldots, \mathfrak{d}_k$ of reach at most $2|\varphi|$, where each $\mathfrak{d}_i$ is a 0-derivation of (q, n_i) containing $\Box_F$, and such that $n_0 > n_1 > n_2 > \cdots > n_k$.

Suppose we have already constructed $\mathfrak{d}_i$. Since $\Psi \Rightarrow^0 (q, n)$, for all n, we have $\Psi \Rightarrow^0 (q, n_i - 1)$. By Lemma 8, there is a 0-derivation $\mathfrak{d}$ of (q, n_{i+1}), for some n_{i+1}, with one of $(\mathbf{C}_1)$–$(\mathbf{C}_3)$. If $(\mathbf{C}_2)$ and $\mathfrak{d}$ left-stutters or $(\mathbf{C}_3)$ and $\mathfrak{d}$ right-stutters then we obtain $(\mathbf{C}'_2)$ or $(\mathbf{C}'_3)$, respectively. If $(\mathbf{C}_2)$ and $\mathfrak{d}$ contains $\Box_F$ with $n_{i+1} \leq n_i - 1$ then $\mathfrak{d}$ becomes the next member $\mathfrak{d}_{i+1}$ in the sequence. If $(\mathbf{C}_3)$ and $\mathfrak{d}$ contains $\Box_P$ with $n_{i+1} \geq n_i - 1$ then $\mathfrak{d}_i$ and $\mathfrak{d}$ satisfy $(\mathbf{C}'_1)$. Otherwise, we have $(\mathbf{C}_1)$ with $n_{i+1} = n_i - 1$ (recall that $n_i > -2|\varphi|$). Consider three cases. If $\mathfrak{d}$ contains $\Box_F$ then $\mathfrak{d}$ becomes the next member $\mathfrak{d}_{i+1}$ in the sequence. If $\mathfrak{d}$ contains $\Box_P$ then $\mathfrak{d}_i$ and $\mathfrak{d}$ satisfy $(\mathbf{C}'_1)$. Otherwise, that is, if $\mathfrak{d}$ contains neither $\Box_P$ nor $\Box_F$, we must have $n_{i+1} = 0$ and $p \to^* q$, for some $p \in \Psi$. Then we have $n_i = 1$ and, as $\mathfrak{d}_i$ contains $\Box_F$, we can append $(q, 1) \Rightarrow_{(\mathbf{R}_5)} (\Box_F q, 0) \Rightarrow_{(\mathbf{R}_4)} (\Box_F q, -1) \Rightarrow_{(\mathbf{R}_3)} (q, 0)$ to $\mathfrak{d}$ to obtain the next member $\mathfrak{d}_{i+1}$ in the sequence.

($\Leftarrow$) is left to the reader. ❑

We are now in a position to prove the main result of this section.

Theorem 4. *The satisfiability problem for* $\mathsf{LTL}^{\Box}_{core}$*-formulas is in* NLogSpace.

Proof. An $\mathsf{LTL}^{\Box}_{core}$-formula $\varphi = \Psi \wedge \circledast \Phi^+ \wedge \boxplus \Phi^-$ is unsatisfiable iff Φ^- contains a clause $\neg\lambda_1 \vee \neg\lambda_2$ such that $\mathfrak{K}_\varphi, n \models \lambda_1 \wedge \lambda_2$, for some n with $|n| \leq K$. For each $\neg\lambda_1 \vee \neg\lambda_2$ in Φ^-, our algorithm guesses such an n (in binary) and, for both λ_1 and λ_2, checks whether $\Psi \Rightarrow^0 (\lambda_i, n)$ or $\Psi \Rightarrow^\forall \lambda_i$, which, by Lemmas 8 and 9, requires only logarithmic space. ❑

The initial clauses of $\mathsf{LTL}^{\Box}_{core}$-formulas φ are propositional variables. If we slightly extend the language to allow for initial core-clauses (without $\boxplus$), then the satisfiability problem becomes PTime-hard. This can be shown by reduction of satisfiability of propositional Horn formulas with clauses of the form p, $\neg p$ and $p \wedge q \to r$, which is known to be PTime-complete. Indeed, suppose $f = \bigwedge_{i=1}^{n} C_i$ is such a formula. We define a temporal formula φ_f to be the conjunction of all unary clauses of f with the following formulas, for each ternary clause C_i of the form $p \wedge q \to r$ in f:

$$c_i \quad \wedge \quad \boxplus(p \to \Box_F c_i) \quad \wedge \quad \boxplus(q \to \Box_P c_i) \quad \wedge \quad (\boxplus\, c_i \to r),$$

where c_i is a fresh variable. One can show that f is satisfiable iff φ_f is satisfiable.

We finish this section by an observation that if the language allows for non-Horn clauses (e.g., $p \vee q$) then the satisfiability problem becomes NP-hard:

Theorem 5. *The satisfiability problem for* $\mathsf{LTL}^{\Box}_{krom}$*-formulas is* NP*-hard.*

Proof. By reduction of graph 3-colourability. Given a graph $G = (V, E)$, consider the following $\mathsf{LTL}^{\Box}_{krom}$-formula φ_G with variables $p_0, \dots, p_4$ and $\overline{v}_i$, for $v_i \in V$:

$$p_0 \;\wedge\; \bigwedge\nolimits_{0 \le i \le 3} \boxplus(p_i \to \Box_F p_{i+1}) \;\wedge\; \bigwedge\nolimits_{v_i \in V} \boxplus(p_0 \to \neg\Box_F \overline{v}_i) \;\wedge\; \bigwedge\nolimits_{v_i \in V} \boxplus(p_4 \to \overline{v}_i) \;\wedge\; \bigwedge\nolimits_{(v_i, v_j) \in E} \boxplus(\overline{v}_i \vee \overline{v}_j).$$

Intuitively, the first four conjuncts of this formula choose, for each vertex v_i of the graph, a moment of time $1 \le n_i \le 3$; the last conjunct makes sure that $n_i \neq n_j$ in case v_i and v_j are connected by an edge in G. It can be easily shown that φ_G is satisfiable iff G is 3-colourable. ❑

5 Conclusion

We have investigated the computational complexity of the satisfiability problem for the fragments of LTL over $(\mathbb{Z}, <)$ given by the form of the clauses—*bool*, *horn*, *krom* and *core*—in the clausal normal form and the temporal operators available for constructing temporal literals. Apart from $\mathsf{LTL}^{\Box,\bigcirc}_{bool}$, whose formulas are equisatisfiable to formulas in the full LTL, only $\mathsf{LTL}^{\Box,\bigcirc}_{horn}$ has PSpace-complete satisfiability. For all other fragments, the complexity varies from NLogSpace to PTime and NP.

The idea to consider sub-Boolean fragments of LTL comes from description logic, where the *DL-Lite* family [9, 3] of logics has been designed and investigated with the aim of finding formalisms suitable for ontology-based data access (OBDA). It transpired that, despite their low complexity, *DL-Lite* logics were capable of representing basic conceptual data modelling constructs [8, 2], and gave rise to the W3C standard ontology language *OWL 2 QL* for OBDA. One possible application of the results obtained in this paper lies in temporal conceptual modelling and temporal OBDA [5]. Temporal description logics (and other many-dimensional logics) are notorious for their bad computational properties [17, 20]. We believe, however, that efficient practical reasoning can be achieved by considering sub-Boolean temporal extensions of *DL-Lite* logics; see [4] for first promising results.

References

1. T. Apostol. *Introduction to Analytic Number Theory.* Springer, 1976.
2. A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *Proc. of ER*, vol. 4801 of *LNCS*, pages 277–292. Springer, 2007.
3. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.
4. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Past and future of DL-Lite. In *Proc. of AAAI*, pages 243–248. 2010.

5. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Complexity of reasoning over temporal data models. In *Proc. of ER*, vol. 6412 of *LNCS*, pages 174–187. Springer, 2010.
6. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. The Complexity of Clausal Fragments of LTL. CoRR abs/1306.5088, 2013.
7. M. Bauland, T. Schneider, H. Schnoor, I. Schnoor, and H. Vollmer. The complexity of generalized satisfiability for linear temporal logic. *LMCS*, 5(1), 2009.
8. D. Berardi, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, 2005.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.
10. C.-C. Chen and I.-P. Lin. The computational complexity of satisfiability of temporal Horn formulas in propositional linear-time temporal logic. *Information Processing Letters*, 45(3):131–136, 1993.
11. M. Chrobak. Finite automata and unary languages. *Theoretical Computer Science*, 47(2):149–158, 1986.
12. S. Demri and Ph. Schnoebelen. The complexity of propositional linear temporal logics in simple cases. *Information and Computation*, 174(1):84–103, 2002.
13. C. Dixon, M. Fisher, and B. Konev. Tractable temporal reasoning. In *Proc. of IJCAI*, pages 318–323, 2007.
14. M. Fisher. A resolution method for temporal logic. In *Proc. of IJCAI*, pages 99–104. Morgan Kaufmann, 1991.
15. M. Fisher, C. Dixon, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computational Logic*, 2(1):12–56, 2001.
16. D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume 1. Oxford University Press, 1994.
17. D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Studies in Logic. Elsevier, 2003.
18. J. Halpern and J. Reif. The propositional dynamic logic of deterministic, well-structured programs. In *Proc. of FOCS*, pages 322–334. IEEE, 1981.
19. O. Lichtenstein, A. Pnueli, and L.D. Zuck. The glory of the past. In *Proc. of CLP*, vol. 193 of *LNCS*, pages 196–218. Springer, 1985.
20. C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *Proc. of TIME*, pages 3–14. IEEE Comp. Society, 2008.
21. N. Markey. Past is for free: on the complexity of verifying linear temporal properties with past. *Acta Informatica*, 40(6–7):431–458, 2004.
22. H. Ono and A. Nakamura. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica*, 39:325–333, 1980.
23. D. Plaisted. A decision procedure for combinations of propositional temporal logic and other specialized theories. *Journal of Automated Reasoning*, 2:171–190, 1986.
24. A. Rabinovich. Temporal logics over linear time domains are in PSPACE. In *Proc. of RP*, vol. 6227 of *LNCS*, pages 29–50. Springer, 2010.
25. M. Reynolds. The complexity of decision problems for linear temporal logics. *Journal of Studies in Logic*, 3(1):19–50, 2010.
26. A. Sistla and E. Clarke. The complexity of propositional linear temporal logics. In *Proc. of STOC*, pages 159–168. ACM, 1982.
27. L. Stockmeyer and A. Meyer. Word problems requiring exponential time: Preliminary report. In *Proc. of STOC*, pages 1–9. ACM, 1973.
28. A. W. To. Unary finite automata vs. arithmetic progressions. *Information Processing Letters*, 109(17):1010–1014, 2009.