S. Demri, R. Lazić and D. Nowak

On the freeze quantifier in constraint LTL: decidability and complexity

Research Report LSV-05-03, April 2005

Laboratoire Spécification et Vérification

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

On the freeze quantifier in Constraint LTL: decidability and complexity

Stéphane Demri[1*] Ranko Lazić[2†] David Nowak[1‡]

[1] LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: {demri,nowak}@lsv.ens-cachan.fr

[2] Department of Computer Science, University of Warwick
Coventry CV4 7AL, United Kingdom
email: lazic@dcs.warwick.ac.uk

Abstract

*Constraint LTL, a generalization of LTL over Presburger constraints, is often used as a formal language to specify the behavior of operational models with constraints. The freeze quantifier can be part of the language, as in some real-time logics, but this variable-binding mechanism is quite general and ubiquitous in many logical languages (first-order temporal logics, hybrid logics, logics for sequence diagrams, navigation logics, etc.). We show that Constraint LTL over the simple domain $\langle \mathbb{N}, = \rangle$ augmented with the freeze operator is undecidable which is a surprising result regarding the poor language for constraints (only equality tests). Many versions of freeze-free Constraint LTL are decidable over domains with qualitative predicates and our undecidability result actually establishes Σ^1_1-completeness. On the positive side, we provide complexity results when the domain is finite (*EXPSPACE*-completeness) or when the formulae are flat in a sense introduced in the paper. Our undecidability results are quite sharp (i.e. with restrictions on the number of variables) and all our complexity characterizations insure completeness with respect to some complexity class (mainly *PSPACE *and *EXPSPACE*).*

Track 3: Temporal Logic in Computer Science
Topics: specification and verification of systems, verification of infinite-state systems, reasoning about transition systems, real-time logics.

*Supported by ACI "Sécurité et Informatique" CORTOS.
†Supported by an invited professorship from ENS Cachan, and by grants from the EPSRC (GR/S52759/01) and the Intel Corporation. Also affiliated to the Mathematical Institute, Serbian Academy of Sciences and Arts, Belgrade.
‡Supported by ACI "Sécurité et Informatique" PERSÉE.

1 Introduction

Model-checking for infinite-state systems. Temporal logics are well-studied formalisms to specify the behavior of finite-state systems and the computational complexity of the model-checking problems is nowadays well-known, see e.g. a survey in [37]. However, many systems such as communication protocols have an infinite amount of configurations and usually the techniques for the finite case cannot be applied directly. For numerous infinite-state systems, the model-checking problem for the linear-time temporal logic LTL can be easily shown to be undecidable (counter automata, hybrid automata and more general constraint automata [35, Chapter 6]). Actually, simpler problems such as reachability are already undecidable. However, remarkable classes of infinite-state systems admit decidable model-checking problems such as the timed automata [1] and subclasses of counter automata [31, 5, 6, 21]. For instance, fragments of LTL with Presburger constraints have been shown decidable over appropriate counter automata [13, 19]. In order to push further the decidability border, one way consists in considering larger classes of operational models, see e.g. [31]. Alternatively, enriching the specification language is another possibility. In the paper we are interested in studying systematically the extensions of versions of LTL over concrete domains by adding the so-called freeze quantifier and to analyze the consequences in terms of decidability and computational complexity.

A variable-binding mechanism. The freeze quantifier in real-time logics has been introduced by Alur and Henzinger in the logic TPTL, see e.g. [2]. The formula $x \cdot \phi(x)$ binds the variable x to the time t of the current state: $x \cdot \phi(x)$ is semantically equivalent to $\phi(t)$. Alternatively, in the explicit clock approach [27], there is an explicit clock variable t and even though in this approach the freeze variable-binding mechanism is possible, the logical formalisms from [2] and [27] are incomparable. In this paper, we want to extend some of the decidable logics from [13, 19, 18] to admit the freeze quantifier: $\downarrow_{y=x} \phi(y)$ holds true at a state iff $\phi(y)$ holds true at the same state with y taking the value of x. Here, y can be in the scope of temporal operators. A crucial difference with the logics in [2, 27] rests on the fact that the variable x may not be monotonic. We focus on decidability and complexity issues when the language of constraints (at the atomic level of the logics) is very simple in order to isolate properly the very effects of the freeze quantifier. We know for instance that LTL over integer periodicity constraints augmented with the freeze operator is EXPSPACE-complete [18].

The above-mentioned variable-binding mechanism that allows the binding of logical variables to objects is very general and it has been used in the literature for various purposes. Details will be provided along the paper (see e.g. Sects. 2.3 and 4.3).

Our contribution. In the paper, we analyze decidability and complexity issues of Constraint LTL augmented with the freeze operator. The temporal operators we consider are restricted to the standard future-time operators "until" and "next" (no past-time operators). $\text{CLTL}^{\downarrow}(\mathcal{D})$ denotes such a logic over the concrete domain $\mathcal{D}$. A concrete domain is composed of a non-empty set equipped with a family of relations. The atomic formulae of $\text{CLTL}^{\downarrow}(\mathcal{D})$ are based on constraints over $\mathcal{D}$ with the ability to compare values of variables at states of bounded distance (see details in the body of the paper) as done in [41, 4, 19, 25]. First, we show that when the underlying domain $\mathcal{D}$ is finite, $\text{CLTL}^{\downarrow}(\mathcal{D})$ satisfiability is in EXPSPACE. If moreover $\mathcal{D}$ has at least two elements with the equality predicate, then $\text{CLTL}^{\downarrow}(\mathcal{D})$ is EXPSPACE-hard. As a corollary, $\text{CLTL}^{\downarrow}(D, =)$ satisfiability is EXPSPACE-complete when $|D| \geq 2$ and D is finite (Sect. 3.2). This witnesses an exponential blow-up since satisfiability for the freeze-free fragment $\text{CLTL}(\mathcal{D})$ when $\mathcal{D}$ is finite can be easily shown in PSPACE as plain LTL.

When the domain D is infinite, we show that $\text{CLTL}^{\downarrow}(D, =)$ is undecidable which is the main result of the paper (Sect. 4). This is quite surprising since the language of constraints is poor (only equality tests) and only future-time operators are used unlike what is shown in [18, Sect. 7] with past-time operators. Our proof, based on a reduction from the recurring problem for 2-counter machines, refines this result: $\text{CLTL}^{\downarrow}(D, =)$ is Σ_1^1-complete even if only one flexible variable and two rigid variables (used to record the values of flexible variables) are involved. Hence, in spite of the very basic Presburger constraints in $\text{CLTL}^{\downarrow}(\mathbb{N}, =)$, satisfiability is Σ_1^1-complete. Even if the language of constraints is minimal, decidability of $\text{CLTL}^{\downarrow}(\mathcal{D})$ can be obtained either at the cost of syntactic restrictions or by assuming semantical constraints (as in the logic TPTL [2] where the freeze quantifier can only record the value of a monotonic variable, namely time).

In order to regain decidability, we introduce the flat fragment of $\text{CLTL}^{\downarrow}(\mathcal{D})$ which contains the freeze-free fragment $\text{CLTL}(\mathcal{D})$ and we show that there is a logspace reduction from the flat fragment of $\text{CLTL}^{\downarrow}(\mathcal{D})$ into $\text{CLTL}(\mathcal{D})$ assuming that the equality predicate belongs to $\mathcal{D}$. As a corollary, we obtain that the flat fragments of $\text{CLTL}^{\downarrow}(\mathbb{Z}, <, =)$ and $\text{CLTL}^{\downarrow}(\mathbb{R}, <, =)$ are PSPACE-complete (Sect. 3.2). Flat fragments of plain LTL versions have been studied in [14, 13] (see also in [32, Sect. 5] the design of a flat logical temporal language for model-checking pushdown machines) and our definition of flatness takes advantage in a non-trivial way of the polarity of "until" subformulae occurring in a formula.

Along the paper, we explicitly consider the satisfiability problem but as shown in Sect. 2.2, our results extend to the model-checking problem of the logics we consider. Moreover, the language of $\text{CLTL}^{\downarrow}(\mathcal{D})$ extends naturally what is done for the freeze-free fragment $\text{CLTL}(\mathcal{D})$ and we show that $\text{CLTL}^{\downarrow}(\mathcal{D})$ increases strictly the expressive power (Proposition 1). However, we prove that significant fragments of $\text{CLTL}^{\downarrow}(\mathcal{D})$ are as expressive as the full language, for instance by recording only values of flexible variables at the current state or by allowing only rigid variables in atomic formulae.

Finally, apart from the technical contributions of the paper, we provide a comparison of several works dealing with freeze-like operators such as in first-order quantification, in timed LTL, in hybrid logics with reference pointers, to quote a few examples.

Related work. Complexity results for Constraint LTL over concrete domains can be found in [41, 4, 19, 25, 18] whereas decidability and complexity issues for LTL over Presburger constraints have been studied for instance in [8, 13, 18]. Most decision procedures in the above-mentioned works are automata-based whereas undecidability proofs often rely on an easy encoding of the halting problem for Minsky machines.

Similar issues for real-time and modal logics equipped with the freeze operator have been considered in [2, 28, 27, 11]. In spite of its rich language of constraints, TPTL model-checking is decidable [2] because of the restricted use of the freeze operator. By contrast, the following variants are undecidable: without the monotonicity condition on time sequences or, with the addition of the multiplication by 2 or, by replacing the time domain $\mathbb{N}$ by $\mathbb{Q}$ (see also in [28] the encoding of classical logic into some half-order modal logic). On the side of Constraint LTL, LTL over integer periodicity constraints augmented with the freeze operator is shown EXPSPACE-complete [18] but CLTL($\mathbb{N}, <, =$) with past-time operator $\mathbf{F^{-1}}$ and $\downarrow$ is undecidable [18].

Variable-binding mechanism similar to the freeze quantifier can be found in hybrid logics, see e.g. [26, 3, 22] where $\downarrow_x \ \phi(x)$ holds true iff $\phi(x)$ holds true when the propositional variable x is interpreted as a singleton containing the current state. The downarrow binder in such hybrid logics records the value of the current state. Similarly, in temporal logic with forgettable past [34], the effect of the **Now** operator is that the origin of time takes the value of the current state: the states before the current state are forgotten. Identical mechanisms are used in navigation logics for object structures, see e.g. [16] and in half-order dynamic temporal logics interpreted over traces from sequence diagrams [12].

First-order temporal logics [17, 42, 29, 24] can also simulate the freeze quantifier which is not surprising since freeze quantification is first-order in nature. In Sect. 4.3 we provide more details about the way to encode CLTL$^{\downarrow}(\mathbb{N}, =)$ into first-order temporal logic $\mathcal{TL}$ over the linear structure $\langle \mathbb{N}, < \rangle$ (with equality) introduced in [24, Chapter 11].

In [9, 10], data languages are defined as sets of finite data words in $(\Sigma \times D)^*$ where Σ is a finite alphabet and D is an infinite domain, generalizing the concept of timed words. Models of the logic CLTL$^{\downarrow}(\mathbb{N}, =)$ restricted to a single flexible variable are indeed infinite data words over a singleton alphabet. Similar models are studied in [15, 7] with motivations stemming from query languages for semistructured data. The only built-in constraint is equality between data, denoted by $\sim$. First-order logic over such finite structures restricted to the predicate $=, <$ (on positions) and $\sim$ is undecidable with three variables [15, 7] and is equivalent to multicounter automata with two variables [7]. Observe that the above formalisms are generally able to encode past-time operators unlike the logics presented in this paper.

Structure of the paper. In Sect. 2, we present the variants of Constraint LTL with the freeze quantifier, expressivity issues as well as the satisfiability and model-checking problems of interest. In Sect. 3, we show decidability and complexity results when the underlying concrete domain is finite or when the flat fragment is considered. In sect. 4, we show that CLTL$^{\downarrow}(\mathbb{N}, =)$ is Σ_1^1-complete. Sect. 5 concludes the paper by enumerating a few open problems about decidability (restrictions over the logical language, restrictions over the intepretation of variables).

2 Constraint LTL with the freeze quantifier

2.1 Syntax and semantics

A constraint system is a set, called the domain, with a family of relations on this set. Let $\mathcal{D} = (D, (R_i)_{i \in I})$ be a constraint system. We define the logic CLTL$^{\downarrow}(\mathcal{D})$ by giving its syntax and semantics.

Syntax. Let $\mathsf{FleVarSet}$ and $\mathsf{RigVarSet}$ be countable sets of variables which are respectively called *flexible variables* and *rigid variables*. Terms are given by the grammar: $t \ ::= \ \underbrace{\mathbf{X} \cdots \mathbf{X}}_{n \text{ times}} x \mid y$ where x is in $\mathsf{FleVarSet}$ and y is in $\mathsf{RigVarSet}$. We use $\mathbf{X}^n$ as an abbreviation for $\underbrace{\mathbf{X} \cdots \mathbf{X}}_{n \text{ times}}$. Formulae are given by the grammar:

$$\begin{array}{lcl} c & ::= & R(t_1, \ldots, t_n) \\ \phi & ::= & c \mid \neg \phi \mid \phi_1 \wedge \phi_2 \mid \\ & & \mathbf{X}\phi \mid \phi_1 \mathbf{U} \phi_2 \mid \downarrow_{y = \mathbf{X}^n x} \phi \end{array}$$

where R ranges over the predicate symbols associated to the relations in $(R_i)_{i \in I}$, x over $\mathsf{FleVarSet}$, and y over $\mathsf{RigVarSet}$. Note that we use the same symbol $\mathbf{X}$ for denoting either the n^{th} next value $\mathbf{X}^n x$ of the variable x or the formula $\mathbf{X}\phi$. We define the Boolean constants, and the temporal operators 'eventually' and 'always', as the following abbreviations: $\top \stackrel{\text{def}}{=} R(t_1, \ldots, t_n) \vee \neg R(t_1, \ldots, t_n)$, $\mathbf{F}\phi \stackrel{\text{def}}{=} \top \mathbf{U} \phi$, $\perp \stackrel{\text{def}}{=} R(t_1, \ldots, t_n) \wedge \neg R(t_1, \ldots, t_n)$, and $\mathbf{G}\phi \stackrel{\text{def}}{=} \neg \mathbf{F} \neg \phi$.

Let $\mathsf{FleVars}(\phi)$ and $\mathsf{RigVars}(\phi)$ denote the sets of all flexible and rigid (respectively) variables which occur in ϕ.

Freeze-free fragment. CLTL($\mathcal{D}$) is the fragment of CLTL$^{\downarrow}(\mathcal{D})$ with no rigid variables and hence without freeze quantifier.

Flat fragment. We say that the occurrence of a subformula in a formula is *positive* if it occurs under an even number of negations, otherwise it is *negative*. The *flat fragment of CLTL*$^{\downarrow}(\mathcal{D})$ is the restriction of CLTL$^{\downarrow}(\mathcal{D})$ where, for any subformula $\phi_1 \mathbf{U} \phi_2$, if it is positive then $\downarrow$ does not occur in ϕ_1, and if it is negative then $\downarrow$ does not occur in ϕ_2.

More precisely, a formula φ of the flat fragment of CLTL$^{\downarrow}(\mathcal{D})$ is given by the grammar:

$$\begin{array}{rcl} \varphi & ::= & c \mid \neg \varphi^- \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \psi \mathbf{U} \varphi \mid \downarrow_{y = \mathbf{X}^n x} \varphi \\ \varphi^- & ::= & c \mid \neg \varphi \mid \varphi_1^- \wedge \varphi_2^- \mid \mathbf{X}\varphi^- \mid \varphi^- \mathbf{U} \psi \mid \downarrow_{y = \mathbf{X}^n x} \varphi^- \\ \psi & ::= & c \mid \neg \psi \mid \psi_1 \wedge \psi_2 \mid \mathbf{X}\psi \mid \psi_1 \mathbf{U} \psi_2 \end{array}$$

Subformulae φ are positive, whereas subformulae φ^- are negative.

Semantics. A model $\sigma : \mathbb{N} \to (\mathsf{FleVarSet} \to D)$ is a sequence of mappings from $\mathsf{FleVarSet}$ to D. For any $i \in \mathbb{N}$, we write σ^i for the model defined by $\sigma^i(j) = \sigma(i+j)$ for every $j \geq 0$. An environment ρ is a mapping from $\mathsf{RigVarSet}$ to D. We write $\rho[x \mapsto v]$ for the environment mapping x to v, and any other variable y to $\rho(y)$. The semantics of terms is given by:

$$[\![\mathbf{X}^n x]\!]_{\sigma,\rho} = \sigma(n)(x) \quad \text{if } x \text{ is in } \mathsf{FleVarSet}$$
$$[\![y]\!]_{\sigma,\rho} = \rho(y) \quad \text{if } y \text{ is in } \mathsf{RigVarSet}$$

The semantics of formulae is given by the following satisfaction relation :

- $\sigma \models_\rho R(t_1, \ldots, t_n)$ iff $([\![t_1]\!]_{\sigma,\rho}, \ldots, [\![t_2]\!]_{\sigma,\rho}) \in R$;
- $\sigma \models_\rho \neg\phi$ iff $\sigma \not\models_\rho \phi$;
- $\sigma \models_\rho \phi_1 \wedge \phi_2$ iff $\sigma \models_\rho \phi_1$ and $\sigma \models_\rho \phi_2$;
- $\sigma \models_\rho \mathbf{X}\phi$ iff $\sigma^1 \models_\rho \phi$;
- $\sigma \models_\rho \phi_1 \mathbf{U} \phi_2$ iff there exists i such that $\sigma^i \models_\rho \phi_2$ and for all $j < i$, $\sigma^j \models_\rho \phi_1$;
- $\sigma \models_\rho \downarrow_{y=\mathbf{X}^n x} \phi$ iff $\sigma \models_{\rho[y \mapsto \sigma(n)(x)]} \phi$,

where we write $\sigma \not\models_\rho \phi$ for not $\sigma \models_\rho \phi$. Note that we use the same symbol R for denoting a relation symbol and its meaning as a relation. Assuming that the domain $\mathcal{D}$ is non-trivial (with at least two elements and non-trivial relations), propositional variables can be easily encoded by constraint terms.

2.2 Satisfiability and model-checking problems

We recall below the problems we are interested in.

Satisfiability problem for CLTL$^{\downarrow}(\mathcal{D})$:
instance: a CLTL$^{\downarrow}(\mathcal{D})$ formula ϕ,
question: is there a model σ and an environment ρ such that $\sigma \models_\rho \phi$?

Without loss of generality we can assume that no free rigid variable occurs in ϕ which means that ρ is not essential above. As usual, the occurrence of a rigid variable x is free in ϕ if it is not in the scope of a freeze quantifier with rigid variable x. Similarly, the model-checking problem rests on $\mathcal{D}$-automata which are constraints automata. A $\mathcal{D}$-automaton $\mathcal{A}$ is simply a Büchi automaton over the infinite alphabet composed of Boolean combinations of atomic CLTL$^{\downarrow}(\mathcal{D})$ formulae with terms of the form x and $\mathbf{X}x$ ($x \in \mathsf{FleVarSet}$). In a $\mathcal{D}$-automaton, letters on transitions induce constraints between the variables of the current state and the variables of the next state as done in [13]. Alternatively, labelling the transitions by CLTL$^{\downarrow}(\mathcal{D})$ formulae (as done in [40]) would not modify essentially the decidability status of model-checking problems considered in this paper.

Model-checking problem for CLTL$^{\downarrow}(\mathcal{D})$:
instance: A $\mathcal{D}$-automaton $\mathcal{A}$ and a CLTL$^{\downarrow}(\mathcal{D})$ formula ϕ,
question: are there a symbolic ω-word $v = \phi_0, \phi_1, \ldots$ accepted by $\mathcal{A}$, a model σ (a realization of v) and an environment ρ such that $\sigma \models_\rho \phi$ and for every $i \geq 0$, $\sigma^i \models_\rho \phi_i$?

It is not difficult to show that as soon as $\mathcal{D}$ is non-trivial the satisfiability problem and the model-checking problem are reducible to each other in logspace following techniques from [38]. In the sequel, we prove results for the satisfiability problems but one has to keep in mind that our results extend to the model-checking problem.

2.3 Expressive power

TPTL. The class of logics CLTL$^{\downarrow}(\mathcal{D})$ defined above is quite general and it is not difficult to show that the real-time logic TPTL is exactly the fragment of the logic CLTL$^{\downarrow}(\mathcal{D})$ where

- $D = \mathbb{N}$ and the only flexible variable is t (time);
- the predicates of $\mathcal{D}$ are the following:
  - $(x \leq c)_{c \in \mathbb{Z}}$, $(x \leq y + c)_{c \in \mathbb{Z}}$,
  - $(x \equiv_d c)_{c,d \in \mathbb{N}}$, $(x \equiv_d y + c)_{c,d \in \mathbb{N}}$

  where $\equiv_d$ is equality modulo d;
- the formulae are of the form $\mathbf{G}(t \leq \mathbf{X}t) \wedge \mathbf{GF}(t < \mathbf{X}t) \wedge \phi$ with the freeze quantifier used with bindings of the form $\downarrow_{x=t}$.

The decidability of TPTL [2] is mainly due to the following semantical restriction: t is monotonic.

The freeze operator strictly increases the expressive power. The addition of the freeze quantifier really enhances the expressive power of CLTL($\mathcal{D}$). For instance, the formula $\phi^x_\infty \stackrel{\text{def}}{=} \mathbf{G} \downarrow_{y=x} \mathbf{XG}x \neq y$ states that the variable x never takes twice the same value in a linear-time model. This is interesting for the verification of cryptographic protocols, nonces are variables that have to be fresh, i.e. they cannot take twice the same value. Similarly, in the context of spatio-temporal logics, Wolter and Zakharyashev [41, Sect. 7] advocate the need to consider operators expressing constraints of the form $\bigwedge_{i \in \mathbb{N}} R(x, \mathbf{X}^i y)$ and $\bigvee_{i \in \mathbb{N}} R(x, \mathbf{X}^i y)$. For instance, $\bigwedge_{i \in \mathbb{N}} R(x, \mathbf{X}^i y)$ can be expressed simply in our formalism by the formula $\downarrow_{x'=x} \mathbf{G}R(x', y)$. This formula is in the flat fragment, which implies for instance nice computational properties, see e.g. Sect. 3.2.

In order to formally show that the freeze operator is powerful, we show that CLTL$^{\downarrow}(\mathbb{N},=)$ is strictly more expressive than its freeze-free fragment CLTL$(\mathbb{N},=)$. There is a formula ϕ in CLTL$^{\downarrow}(\mathbb{N},=)$ (with no free rigid variable) for which there is no formula ψ in CLTL$(\mathbb{N},=)$ such that for all models σ and environments ρ, $\sigma \models_\rho \phi$ iff $\sigma \models_\rho \psi$. Since no free rigid variable occurs in ϕ and ψ, the environment ρ is irrelevant here and we write $\sigma \models \phi$ instead of $\sigma \models_\rho \phi$. First, we show the following property.

Lemma 1 *Every satisfiable formula in CLTL$(\mathbb{N},=)$ has a model with a finite amount of values in the whole model.*

Proof. Let ϕ be a formula in CLTL$(\mathbb{N},=)$ with variables in $\{x_1,\ldots,x_n\}$ and k be equal to 1 plus the maximal j such that $\mathbf{X}^j x_i$ occurs in ϕ for some flexible variable x_i. Let C be the finite set of constraints of the form $\mathbf{X}^{j_1} x_{i_1} = \mathbf{X}^{j_2} x_{i_2}$ with $0 \leq j_1, j_2 \leq k-1$ and $i_1, i_2 \in \{1,\ldots,n\}$.

We define a total ordering on $\{1,\ \ldots, n\} \times \mathbb{N}$ as follows: $\langle i,j\rangle < \langle i',j\rangle$ iff $j < j'$ or ($j = j'$ and $i < i'$). Given a model $\sigma : \mathbb{N} \rightarrow (\mathsf{FleVarSet} \rightarrow \mathbb{N})$, we build a model $\sigma' : \mathbb{N} \rightarrow (\mathsf{FleVarSet} \rightarrow \{1,\ldots,k\times n\})$ such that $\sigma \models \phi$ iff $\sigma' \models \phi$.

If x is a flexible variable not occurring in ϕ, $\sigma'(i)(x) = 1$ for every $i \geq 0$. Otherwise $\sigma'(0)(x_1) = 1$ ($\langle 1,0\rangle$ is minimal wrt $<$). Now suppose that for every $\langle i',j'\rangle < \langle i,j\rangle$, $\sigma'(j')(x_{i'})$ has been already defined. We shall define $\sigma'(j)(x_i)$. If for some $\langle i',j'\rangle$ in $\{\langle i'',j''\rangle : 0 \leq j-j'' \leq k-1,\ 1 \leq i'' \leq n,\ \langle i'',j''\rangle < \langle i,j\rangle\}$, $\sigma(j')(x_{i'}) = \sigma(j)(x_i)$ then $\sigma'(j)(x_i)$ takes the value $\sigma(j')(x_{i'})$. Otherwise, $\sigma'(j)(x_i)$ takes an arbitrary value from the set

$$\begin{array}{l}\{1,\ldots,k\times n\}\setminus\\ \{\sigma(j'')(x_{i''}) : \ 0 \leq j-j'' \leq k-1,\ 1 \leq i'' \leq n,\\ \qquad\qquad\qquad \langle i'',j''\rangle < \langle i,j\rangle\}\end{array}$$

which is always possible since the second set has strictly less that $k \times n$ elements. One can show that for all $c \in C$ and $i \geq 0$, $\sigma'^{i} \models c$ iff $\sigma^i \models c$. Hence, $\sigma \models \phi$ iff $\sigma' \models \phi$. $\Box$

Proposition 1 *There is no formula in CLTL$(\mathbb{N},=)$ equivalent to $\phi^x_\infty \in$ CLTL$^{\downarrow}(\mathbb{N},=)$.*

Indeed, every satisfiable formula in CLTL$(\mathbb{N},=)$ admits a model in which the variable x takes a finite amount of values by Lemma 1.

Equivalent syntactic restrictions. We now show that expressiveness of CLTL$^{\downarrow}(\mathcal{D})$ does not change if we restrict the freeze quantifier to refer only to flexible variables in the current state, or if we restrict atomic formulae to contain only rigid variables, or with both restrictions. Therefore, those restrictions could have been incorporated into the definition of the logic. However, we chose to allow terms of the form $\mathbf{X}^n x$ with flexible x in atomic formulae in order to have CLTL$(\mathcal{D})$ as the freeze-free fragment; and to allow the freeze quantifier to refer to the future so that formulae would be closed under substitution of terms.

Proposition 2 *For any formula ϕ of CLTL$^{\downarrow}(\mathcal{D})$, there exists an equivalent formula ϕ' such that:*

(i) *any occurence of $\downarrow$ in ϕ' is of the form $\downarrow_{y=x}$;*

(ii) $\mathsf{FleVars}(\phi') = \mathsf{FleVars}(\phi)$*;*

(iii) $\mathsf{RigVars}(\phi') = \mathsf{RigVars}(\phi)$.

Proof. By structural induction on ϕ, it suffices to prove the statement for formulae of the form $\downarrow_{y=\mathbf{X}^n x}\ \phi'$ where ϕ' satisfies (i).

This can be done by induction on n. The base case $n = 0$ is trivial. For the inductive step, we use structural induction on ϕ'. The most difficult case is $\phi' = \phi'_1 \mathbf{U} \phi'_2$. We then have

$$\begin{array}{ll} & \downarrow_{y=\mathbf{X}^{n+1}x}\ \phi' \\ \equiv & \downarrow_{y=\mathbf{X}^{n+1}x}\ \phi'_2 \vee (\phi'_1 \wedge \mathbf{X}\phi') \\ \equiv & (\downarrow_{y=\mathbf{X}^{n+1}x}\ \phi'_2) \vee ((\downarrow_{y=\mathbf{X}^{n+1}x}\ \phi'_1) \wedge \mathbf{X} \downarrow_{y=\mathbf{X}^{n}x}\ \phi') \end{array}$$

and the induction hypotheses apply to each of the three freeze subformulae. $\Box$

Proposition 3 *For any formula ϕ of CLTL$^{\downarrow}(\mathcal{D})$, there exists an equivalent formula ϕ' such that:*

- *atomic formulae in ϕ' contain only rigid variables;*
- *if any occurence of $\downarrow$ in ϕ is of the form $\downarrow_{y=x}$, then the same is true of ϕ';*
- $\mathsf{FleVars}(\phi') = \mathsf{FleVars}(\phi)$*;*
- *if k is the maximum number, over all atomic formulae in ϕ, of distinct terms of the form $\mathbf{X}^n x$ with $x \in$* $\mathsf{FleVarSet}$*, then* $|\mathsf{RigVars}(\phi')| \leq |\mathsf{RigVars}(\phi)| + k$.

Proof. ϕ' is constructed from ϕ by translating only atomic subformulae of ϕ. The translation is as in the following example. $R(\mathbf{X}^2 x_1, y_1, \mathbf{X}^3 x_2, \mathbf{X}^2 x_3, x_4, y_2, x_4)$, where $x_i \in \mathsf{FleVarSet}$ and $y_i \in \mathsf{RigVarSet}$, is translated to $\downarrow_{y_3=x_4}$ $\mathbf{X}^2 \downarrow_{y_4=x_1}\downarrow_{y_5=x_3}\ \mathbf{X}^1 \downarrow_{y_6=x_2}\ R(y_4,y_1,y_6,y_5,y_3,y_2,y_3)$ where $y_3,\ldots,y_6$ are fresh rigid variables. $\Box$

3 Decidability results

3.1 Finite domain case

In this section, we basically show that, when $\mathcal{D}$ is finite (with at least two elements) and contains the equality predicate, CLTL$^{\downarrow}(\mathcal{D})$ is EXPSPACE-complete. In Theorem 1 below, we establish that EXPSPACE-hardness is very common when the freeze quantifier is present.

Theorem 1 *Let $\mathcal{D}$ be a constraint system with equality such that the underlying domain D contains at least two elements. The satisfiability problem for CLTL$^{\downarrow}(\mathcal{D})$ is* EXPSPACE-*hard.*

Proof. We prove this result by a reduction from an EXPSPACE-complete tiling problem (see e.g. [39]). A tile is a unit square of one of the several tile-types and the tiling problem we consider is specified by means of a finite set T of tile-type (say $T = \{t_1, \ldots, t_k\}$), two binary relations H and V over T and two distinguished tile-types t_{init}, $t_{final} \in T$. The tiling problem consists in determining whether, for a given number n in unary, the region $[0, \ldots, 2^n - 1] \times [0, \ldots, k-1]$ of the integer plane for some k can be tiled consistently with H and V, t_{init} is the left bottom tile, and t_{final} is the right upper tile.

Given an instance $I = \langle T, t_{init}, t_{final}, n \rangle$ of the tiling problem, we build a CLTL$^{\downarrow}(\mathcal{D})$ formula ϕ_I such that $I = \langle T, t_{init}, t_{final}, n \rangle$ has a solution iff ϕ_I is CLTL$^{\downarrow}(\mathcal{D})$ satisfiable.

We consider the following flexible variables:

- $c_1, \ldots, c_n$ are variables that allow to count until 2^n and x_0, x_1 are variables that will play the role of 0 and 1, respectively; there are corresponding rigid variables $c'_1, \ldots, c'_n$; each element $\langle \alpha, i \rangle$ of a row $[0, \ldots, 2^n - 1] \times \{i\}$ such that the binary representation of α is $b_1 \ldots b_n$, satisfies $c_j = x_0$ iff $b_j = 0$ for every $j \in \{1, \ldots, n\}$;
- for $t \in T$, z_t^1, z_t^2 are variables such that $D_t := z_t^1 = z_t^2$ is the formula encoding the fact that at a certain position of the integer plane the tile t is present. There are also rigid variables $z_t^{1'}, z_t^{2'}$ and $D'_t := z_t^{1'} = z_t^{2'}$;
- end_1, end_2 such that $\mathtt{END} := end_1 = end_2$;

The formula ϕ_I is the conjunction of the following formulae:

- $\neg\mathtt{END} \wedge (\neg\mathtt{END}\mathbf{U}(c_1 = \cdots = c_n = x_0 \wedge \mathbf{G}\mathtt{END}))$
  (the region of the integer plane for the solution is finite);
- $\neg(x_0 = x_1) \wedge \mathbf{G}(x_0 = \mathbf{X}x_0 \wedge x_1 = \mathbf{X}x_1)$
  (x_0 and x_1 behave as different constants);
- $\mathbf{G}(\neg\mathtt{END} \Rightarrow \bigvee_{t \in T}(D_t \wedge \bigwedge_{t' \neq t} \neg D_{t'}))$
  (exactly one tile per element of the plane region);
- $\mathbf{F}(\bigwedge_{1 \leq i \leq n}(c_i = x_1) \wedge \neg\mathtt{END} \wedge D_{t_{final}} \wedge \mathbf{X}\mathtt{END})$
  (right upper tile);
- $\bigwedge_{1 \leq i \leq n}(c_i = x_0) \wedge D_{t_{init}}$
  (left bottom tile);
- $\mathbf{G}(\bigvee_{2 \leq i \leq n+1}((\bigwedge_{i \leq j \leq n} c_j = x_1) \wedge c_{i-1} = x_0 \wedge \neg\mathtt{END}) \Rightarrow (\bigwedge_{1 \leq j \leq i-2}(c_j = \mathbf{X}c_j) \wedge \mathbf{X}c_{i-1} = x_1 \wedge \bigwedge_{i \leq j \leq n}(\mathbf{X}c_j = x_0))))$
  (incrementation of the counters $c_1, \ldots, c_n$);
- $\mathbf{G}((\neg\mathbf{X}\mathtt{END} \wedge c_1 = \cdots = c_n = x_1) \Rightarrow \mathbf{X}(c_1 = \cdots = c_n = x_0))$
  (limit condition for the incrementation of the counters $c_1, \ldots, c_n$);
- $\mathbf{G}(\overbrace{\neg(c_1 = \cdots = c_n = x_1)}^{\text{not the last element of a row}} \wedge \neg\mathtt{END} \Rightarrow \bigwedge_{t \in T}(D_t \Rightarrow \bigvee_{\langle t,t' \rangle \in H} \mathbf{X}D_{t'}))$
  (horizontal consistency);
- $$\mathbf{G}(\neg\mathtt{END} \wedge \overbrace{\mathbf{F}(\mathbf{X}\neg\mathtt{END} \wedge c_1 = \ldots = c_n = x_1)}^{\text{not on the last row}} \Rightarrow$$
  $\downarrow_{c'_1 = c_1} \cdots \downarrow_{c'_n = c_n} \downarrow_{z_{t_1}^{1'} = z_{t_1}^1} \downarrow_{z_{t_1}^{2'} = z_{t_1}^2} \cdots \downarrow_{z_{t_k}^{1'} = z_{t_k}^1} \downarrow_{z_{t_k}^{2'} = z_{t_k}^2} \mathbf{X}((\neg(c'_1 = c_1 \wedge \cdots \wedge c'_n = c_n))\mathbf{U}(c'_1 = c_1 \wedge \cdots \wedge c'_n = c_n \wedge \bigwedge_{t \in T}(D'_t \Rightarrow \bigvee_{\langle t,t' \rangle \in V} \mathbf{X}D_{t'})))$
  (vertical consistency).

It is not difficult to show that the instance $I = \langle T, t_{init}, t_{final}, n \rangle$ has a solution iff ϕ_I is CLTL$^{\downarrow}(\mathcal{D})$ satisfiable. $\square$

This is reminiscent to the EXPSPACE-hardness of Timed Propositional Temporal Logic (TPTL) [2, Theorem 2], PLTL+Now (NLTL) [34, Proposition 4.7] and a variant of the guarded fragment with transitivity [33, Theorem 2]. Our EXPSPACE-hardness proof is in the same vein since basically in CLTL$^{\downarrow}(\mathcal{D})$ we are able to count till 2^n using only a number of resources polynomial in n and we can compare the truth value of atomic formulae in states of "temporal distance" exactly 2^n, whence the reduction of a famous EXPSPACE-complete tiling problem.

Our proof is a slight variant of the proof of [18, Theorem 6]: instead of using integer periodicity constraints to count till 2^n, n binary counters are used. Observe also that the result formula is not flat because of the encoding of vertical consistency.

If we replace $\mathbf{U}$ by $\mathbf{F}$, then NEXPTIME-hardness can be shown by reducing the $n \times n$ tiling problem with n encoded in binary.

Finitess of $\mathcal{D}$ allows us to show the decidability of CLTL$^{\downarrow}(\mathcal{D})$.

Theorem 2 *Let $\mathcal{D}$ be a finite constraint system. The satisfiability problem for CLTL$^{\downarrow}(\mathcal{D})$ is in* EXPSPACE.

Proof. Assume that $D = \{d_1, \ldots, d_l\}$. We introduce an auxiliary constraint system $\mathcal{D}' = \langle D, P_1, \ldots, P_l \rangle$ such that $P_i = \{d_i\}$. For convenience, we write $x = d_i$ instead of $\mathsf{P}_i(x)$. We shall show how to reduce the satisfiability problem for CLTL$^{\downarrow}(\mathcal{D})$ into the satisfiability problem for CLTL$(\mathcal{D}')$. PSPACE-easiness of CLTL$(\mathcal{D}')$ is not very difficult to show and it is a direct consequence of [18, Theorem

4].
We introduce a translation T from CLTL$^{\downarrow}(\mathcal{D})$ formulae into CLTL$(\mathcal{D}')$ formulae defined as follows:

- T is homomorphic for the Boolean operators and the temporal operators;
- $\mathrm{T}(\mathsf{R}(\alpha_1, \ldots, \alpha_n)) = (\bigvee_{R(d_{i_1},\ldots,d_{i_n})}(\alpha_1 = d_{i_1} \wedge \cdots \wedge \alpha_n = d_{i_n}))$.

So far, the translation can be done in polynomial time and logarithmic space since $|D|^n$ is a constant of CLTL$^{\downarrow}(\mathcal{D})$. The last clause of T is related to the freeze quantifier:

$$\mathrm{T}(\downarrow_{x'=\alpha} \psi) = \bigwedge_{d_i \in D} (\alpha = d_i) \Rightarrow \mathrm{T}(\psi)^{x'=d_i},$$

where $\mathrm{T}(\psi)^{x'=d_i}$ is obtained from $\mathrm{T}(\psi)$ by replacing every occurrence of $x' = d_j$ with $j \neq i$ by $\perp$ and every occurrence of $x' = d_i$ by $\top$. This step requires an exponential blow up and therefore $|\mathrm{T}(\phi)|$ is exponential in $|\phi|$. It is easy to show that ϕ is CLTL$^{\downarrow}(\mathcal{D})$ satisfiable iff $\mathrm{T}(\phi)$ is CLTL$(\mathcal{D}')$ satisfiable. Since T may cause at most an exponential blow up and CLTL$(\mathcal{D}')$ is in PSPACE, we obtain that CLTL$^{\downarrow}(\mathcal{D})$ satisfiability is in EXPSPACE. □

Our proof can be easily adapted if the freeze quantifier is replaced by full existential quantifier $\exists$.

Corollary 1 *Let $\mathcal{D}$ be a finite constraint system with equality such that the underlying domain D contains at least two elements. The satisfiability problem for CLTL$^{\downarrow}(\mathcal{D})$ is* EXPSPACE*-complete.*

A formula $\phi \in$ CLTL$^{\downarrow}(\mathcal{D})$ is of $\downarrow$-depth k, for some $k \geq 0$ whenever every branch of the formula tree of ϕ has at most k freeze quantifiers. For example, the formula $\downarrow_{x'=x} (y = x')\mathbf{U} \downarrow_{x'=z} y = x'$. is of $\downarrow$-heigth 2.

Corollary 2 *Let $\mathcal{D}$ be a finite constraint system. For every $k \geq 0$, the satisfiability problem for CLTL$^{\downarrow}(\mathcal{D})$ restricted to formulae of $\downarrow$-height k is in* PSPACE.

3.2 Flat fragment between CLTL$(\mathcal{D})$ and CLTL$^{\downarrow}(\mathcal{D})$

The main result of this section is to show that the freeze quantifier in the flat fragment of CLTL$^{\downarrow}(\mathcal{D})$ can be encoded faithfully into CLTL$(\mathcal{D})$. The flatness concept is only related to occurrences of the freeze quantifier and for instance the formulae of the form ϕ^x_∞ do not belong to the flat fragment. By contrast, $\neg\phi^x_\infty$ belongs to the flat fragment of CLTL$^{\downarrow}(\mathbb{N},=)$. By Proposition 1, the flat fragment of CLTL$^{\downarrow}(\mathbb{N},=)$ is therefore strictly more expressive than CLTL$(\mathbb{N},=)$ since CLTL$(\mathbb{N},=)$ is closed under negation. However, as shown below, satisfiability for flat CLTL$^{\downarrow}(\mathbb{N},=)$ can be reduced in logarithmic space to satisfiability for CLTL$(\mathbb{N},=)$. By analogy, CTL* model-checking can be reduced to LTL model-checking [20] even though CTL* is more expressive than LTL.

We assume that the flexible variables of CLTL$^{\downarrow}(\mathcal{D})$ are $\{x_0, x_1, \ldots\}$ and the rigid variables of CLTL$^{\downarrow}(\mathcal{D})$ are $\{y_0, y_1, \ldots\}$. For the ease of presentation, we assume that the flexible variables of CLTL$(\mathcal{D})$ are composed of the following two disjoint sets: $\{x_0, x_1, \ldots\}$ and $\{y^{\mathrm{new}}_0, y^{\mathrm{new}}_1, \ldots\}$. We define below a map u from the flat fragment CLTL$^{\downarrow}(\mathcal{D})$ into CLTL$(\mathcal{D})$ that is homomorphic for the Boolean and temporal connectives and such that

- $u(c) \stackrel{\mathrm{def}}{=} c'$ where c' is obtained from c by replacing each rigid variable y_j by y^{new}_j,
- $u(\downarrow_{y=\mathbf{X}^n x} \psi) \stackrel{\mathrm{def}}{=} y^{\mathrm{new}} = \mathbf{X}^n x \quad \wedge \quad \mathbf{G}(y^{\mathrm{new}} = \mathbf{X} y^{\mathrm{new}}) \ \wedge \ u(\psi)$.

It is easy to show that $u(\phi)$ can be computed in logarithmic space in $|\phi|$.

Proposition 4 *Let $\mathcal{D}$ be a constraint system with equality. For any formula ϕ of the flat fragment of CLTL$^{\downarrow}(\mathcal{D})$, ϕ is CLTL$^{\downarrow}(\mathcal{D})$ satisfiable iff $u(\phi)$ is CLTL$(\mathcal{D})$ satisfiable.*

Proof. Given a model σ of CLTL$^{\downarrow}(\mathcal{D})$, an environment ρ and a formula ϕ we say that the model σ' of CLTL$(\mathcal{D})$ agrees with σ, ρ and ϕ iff for all $i, j \geq 0$, $\sigma(i)(x_j) = \sigma'(i)(x_j)$ and for all free rigid variable y_j in ϕ and $i \geq 0$, $\sigma'(i)(y^{\mathrm{new}}_j) = \rho(y_j)$.

We shall use the following properties:

- $u(\psi) = \psi$ if ψ belongs to CLTL$(\mathcal{D})$.
- If σ' agrees with σ, ρ and ψ then $(\sigma')^i$ agrees with σ^i, ρ and ψ for every $i \geq 0$.

Given the occurrence of a subformula ψ in ϕ with positive [resp. negative] polarity, we write the sign s_ψ to denote the empty string [resp. $\neg$]. By abusing notation, we do not distinguish subformulae from occurrences.

We shall show by structural induction that for any occurrence of a subformula ψ in ϕ, for all models σ of CLTL$^{\downarrow}(\mathcal{D})$ and environment ρ, $\sigma \models_\rho s_\psi \ \psi$ iff there is σ' that agrees with σ, ρ and ψ such that $\sigma' \models s_\psi \ u(\psi)$. Statement of the lemma is then immediate.

The base case with atomic formulae and the cases in the induction step with $\neg$, $\wedge$ and $\mathbf{X}$ are by an easy verification. By way of example, we treat the case with $\psi = \neg\psi'$ with negative polarity. So ψ' occurs with positive polarity. Let σ be a model and ρ be an environment such that $\sigma \models_\rho \neg\neg\psi'$. The statements below are equivalent:

- $\sigma \models_\rho \neg\neg\psi'$,

- $\sigma \models_\rho \psi'$,
- there is σ' that agrees with σ, ρ and ψ' such that $\sigma' \models u(\psi')$ (by (IH) and change of polarity),
- there is σ' that agrees with σ, ρ and ψ' such that $\sigma' \models \neg u(\neg\psi')$ (by definition of u).

Let us treat the remaining cases.
Case 1: $\psi = \psi_1 \mathbf{U} \psi_2$ with positive polarity.
Since ϕ belongs to the flat fragment, we have $\psi_1 = u(\psi_1)$. Let σ be a model and ρ be an environment such that $\sigma \models_\rho \psi$. The statements below are equivalent:

- $\sigma \models_\rho \psi$,
- there is $i \geq 0$ such that $\sigma^i \models_\rho \psi_2$ and for every $j < i$, $\sigma^j \models_\rho \psi_1$,
- there is σ' that agrees with σ, ρ and ψ_2 such that $(\sigma')^i \models u(\psi_2)$ and for every $j < i$, $(\sigma')^j \models u(\psi_1)$ (by (IH), $\psi_1 = u(\psi_1)$ and, σ and σ' agree on flexible variables of ψ_1),
- there is σ' that agrees with σ, ρ and ψ such that $\sigma' \models u(\psi_1)\mathbf{U}u(\psi_2)$ (ψ_1 has no free rigid variable).

Case 2: $\psi = \psi_1 \mathbf{U} \psi_2$ with negative polarity.
Since ϕ belongs to the flat fragment, we have $\psi_2 = u(\psi_2)$ and both ψ_1 and ψ_2 have negative polarity. Let σ be a model and ρ be an environment such that $\sigma \models_\rho \psi$. The statements below are equivalent:

- $\sigma \models_\rho \psi$,
- either there is $j \geq 0$ such that $\sigma^j \models_\rho \neg\psi_1$ and for every $j \leq i$, $\sigma^i \models_\rho \neg\psi_2$ or for every $i \geq 0$, $\sigma^i \models_\rho \neg\psi_2$,
- either there is σ' that agrees with σ, ρ and ψ_1 such that there is $j \geq 0$ such that $(\sigma')^j \models \neg u(\psi_1)$ and for every $j \leq i$, $(\sigma')^i \models \neg u(\psi_2)$ (by (IH) and $\psi_2 = u(\psi_2)$) or there is σ' that agrees with σ, ρ and ψ_2 such that for every $i \geq 0$, $(\sigma')^i \models \neg u(\psi_2)$ (by (IH)),
- there is σ' that agrees with σ, ρ and $\psi_1 \mathbf{U} \psi_2$ such that either there is $j \geq 0$ such that $(\sigma')^j \models \neg u(\psi_1)$ and for every $j \leq i$, $(\sigma')^i \models \neg u(\psi_2)$ or for every $i \geq 0$, $(\sigma')^i \models \neg u(\psi_2)$ (ψ_2 has no free rigid variables),
- there is σ' that agrees with σ, ρ and $\psi_1 \mathbf{U} \psi_2$ such that $\sigma' \models \neg(u(\psi_1)\mathbf{U}u(\psi_2))$.

Case 3: $\psi = \downarrow_{y=\mathbf{X}^n x} \psi'$.
Let σ be a model and ρ be an environment for s_ψ and ψ. The statements below are equivalent:

- $\sigma \models_\rho s_\psi\ \psi$,
- $\sigma \models_{\rho[y \mapsto \sigma(n)(x)]} s_\psi\ \psi'$,
- there is σ' that agrees with σ, $\rho[y \mapsto \sigma(n)(x)]$ and ψ' such that $\sigma' \models s_\psi\ u(\psi')$ (by (IH)),
- there is σ' that agrees with σ, $\rho[y \mapsto \sigma(n)(x)]$ and ψ' such that $\sigma' \models s_\psi\ u(\psi')$ and $\sigma' \models \mathbf{G}(y^{\text{new}} = \mathbf{X}y^{\text{new}}) \wedge y^{\text{new}} = \mathbf{X}^n x$ (y free in ψ').
- there is σ' that agrees with σ, ρ and ψ such that $\sigma' \models s_\psi\ u(\psi') \wedge \mathbf{G}(y^{\text{new}} = \mathbf{X}y^{\text{new}}) \wedge y^{\text{new}} = \mathbf{X}^n x$ (ψ has less free rigid variable than ψ'). $\Box$

Corollary 3 *For every constraint system $\mathcal{D}$ equipped with equality predicate, decidability of CLTL($\mathcal{D}$) implies the decidability of the flat fragment of CLTL$^\downarrow$($\mathcal{D}$).*

Since CLTL($\langle \mathbb{Z}, <, = \rangle$), CLTL($\langle \mathbb{N}, <, = \rangle$) and CLTL($\langle \mathbb{R}, <, = \rangle$) are PSPACE-complete [19], we can establish the following corollary.

Corollary 4 *Flat fragments of CLTL$^\downarrow$($\langle \mathbb{Z}, <, = \rangle$), CLTL$^\downarrow$($\langle \mathbb{N}, <, = \rangle$), CLTL$^\downarrow$($\langle \mathbb{R}, <, = \rangle$), and CLTL$^\downarrow$($\mathcal{D}$) with $\mathcal{D}$ finite are* PSPACE*-complete.*

4 Undecidability results

In this section, we shall prove that, if the domain is infinite, and if we do not restrict to flat formulae, the satisfiability problem for CLTL$^\downarrow$($\mathcal{D}$) is undecidable even if we only have the equality predicate. More precisely, Theorem 3 below is a stronger result, stating that satisfiability is Σ^1_1-hard, even restricted to formulae with 1 flexible variable and at most 2 rigid variables. (An exposition of the analytical hierarchy can be found in [36].) A corollary of Σ^1_1-hardness is that the logic cannot be recursively axiomatized.

4.1 Comparison with other undecidability results

In [2, Theorem 5], Σ^1_1-hardness of satisfiability for TPTL without the monotonicity condition on time sequences is established. By Propositions 2 and 3, CLTL$^\downarrow$($\mathbb{N}, =$) restricted to one flexible variable can be seen as the fragment of TPTL where there are no atomic propositions, and where the only operation on time is equality. Moreover, it is straightforward to see that Theorem 3 below still holds when satisfiability is restricted to models which contain infinitely many values, which is equivalent to the progress condition when the domain is $\mathbb{N}$. Therefore, a corollary of our result is the following strengthening of [2, Theorem 5]: satisfiability for TPTL without the monotonicity condition remains Σ^1_1 even without atomic propositions and with only equality constraints. (The proof of [2, Theorem 5] uses arithmetic on time values.)

As explained in Sect. 4.3, CLTL$^\downarrow$($\mathbb{N}, =$) can be naturally translated into first-order temporal logic $\mathcal{TL}$ over the linear structure $\langle \mathbb{N}, < \rangle$ with equality introduced in [24, Chapter 11]. Undecidability of the monodic fragment of this logic has been established in [42] by reducing the finite validity problem for classical logic known to be undecidable by Trakhtenbrot's Theorem. A refinement of this result is shown in [17] to the fragment restricted to two individual variables and monadic predicate symbols. The proof is based on a reduction from halting problem for Minsky machines by coding the values of counters by the cardinality of the interpretations of monadic predicate symbols.

The models of Quantified Propositional Temporal Logic with Repeating (RQPTL) introduced in [23] can be encoded by $\mathrm{CLTL}^{\downarrow}(\mathbb{N},=)$ formulae unlike the second-order quantification in the language. However, the variant logic RHLTL^n [23, Sect. 4] is equivalent to $\mathrm{CLTL}^{\downarrow}(\mathbb{N},=)$ with one flexible variable and n rigid variables except that RHLTL^n does not have $\mathbf{U}$ but has $\mathbf{F}$, $\mathbf{F^{-1}}$ and $\mathbf{X^{-1}}$. Theorem 3 and Σ^1_1-hardness of RHLTL^2 [23, Corollary 1] are complementary results.

Our encoding of configurations in 2-counter machines is similar to the one in [15, Sect. 7] to show undecidability of an emptiness problem for a class of two-ways register automata. As a corollary of [15, Sect. 7] and of the proof of Theorem 3 below, $\mathrm{CLTL}^{\downarrow}(\mathbb{N},=)$ augmented with the past-time operator "since" but restricted to a single rigid variable is Σ^1_1-complete.

4.2 Σ^1_1-completeness

The following proposition complements the main result in this section, and states that, for countable and computable constraint systems $\mathcal{D}$, satisfiability for $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ is in Σ^1_1. Hence, for a countably infinite domain, the problem in Theorem 3 is Σ^1_1-complete.

Proposition 5 *Suppose D is a countable set, and $(R_i)_{i\in I}$ is a family of computable relations on D. The satisfiability problem for $\mathrm{CLTL}^{\downarrow}(D,(R_i)_{i\in I})$ is in Σ^1_1.*

Proof. Let ϕ be a formula of $\mathrm{CLTL}^{\downarrow}(D,(R_i)_{i\in I})$. We can assume $\mathsf{FleVarSet} = \mathsf{FleVars}(\phi)$ and $\mathsf{RigVarSet} = \mathsf{RigVars}(\phi)$. Let $n = |\mathsf{FleVarSet}|$ and $m = |\mathsf{RigVarSet}|$. Any model $\sigma : \mathbb{N} \to (\mathsf{FleVarSet} \to D)$ can be encoded by functions $f_1,\ldots,f_n : \mathbb{N} \to \mathbb{N}$, and any environment $\rho : \mathsf{RigVarSet} \to D$ as an m-tuple $a_1,\ldots,a_m : \mathbb{N}$. A first-order predicate on $f_1, \ldots, f_n$ and $a_1, \ldots, a_m$ which expresses that $\sigma \models_\rho \phi$ is routine to construct by structural recursion on ϕ. We conclude that satisfiability of ϕ can be expressed by a Σ^1_1-sentence. □

We shall prove that the satisfiability problem for a fragment of $\mathrm{CLTL}^{\downarrow}(D,=)$ is Σ^1_1-hard by reducing from the Recurrence Problem for nondeterministic 2-counter machines, which was shown to be Σ^1_1-hard in [2, Section 4.1].

A *nondeterministic 2-counter machine* M consists of two counters C_1 and C_2, and a sequence of $n \geq 1$ instructions, each of which may increment or decrement one of the counters, or jump conditionally upon of the counters being zero. After the execution of a non-jump instruction, M proceeds nondeterministically to one of two specified instructions. Therefore, the l^{th} instruction is written as one of the following:

$$l:\ C_i := C_i + 1;\ \text{goto } l' \text{ or goto } l''$$
$$l:\ C_i := C_i - 1;\ \text{goto } l' \text{ or goto } l''$$
$$l:\ \text{if } C_i = 0 \text{ then goto } l' \text{ else goto } l''$$

We represent the configurations of M by triples $\langle l, c_1, c_2\rangle$, where $1 \leq l \leq n$, $c_1 \geq 0$, and $c_2 \geq 0$ are the current values of the location counter and the two counters C_1 and C_2, respectively. The consecution relation on configurations is defined in the obvious way, where decrementing 0 yields 0. A *computation* of M is an infinite sequence of related configurations, starting with the initial configuration $\langle 1,0,0\rangle$. The computation is *recurring* if it contains infinitely many configurations with the value of the location counter being 1.

The Recurrence Problem is to decide, given a nondeterministic 2-counter machine M, whether M has a recurring computation. This problem is Σ^1_1-hard.

Theorem 3 *Suppose D is an infinite set. The satisfiability problem for formulae ϕ of $\mathrm{CLTL}^{\downarrow}(D,=)$ such that $|\mathsf{FleVars}(\phi)| = 1$ and $|\mathsf{RigVars}(\phi)| \leq 2$ is Σ^1_1-hard.*

Proof. Suppose M is a nondeterministic 2-counter machine. We construct a formula ϕ_M of $\mathrm{CLTL}^{\downarrow}(D,=)$ such that $|\mathsf{FleVars}(\phi)| = 1$, $|\mathsf{RigVars}(\phi)| \leq 2$, and ϕ_M is satisfiable iff M has a recurring computation. The basis of the construction is an encoding of computations of nondeterministic 2-counter machines by models of $\mathrm{CLTL}^{\downarrow}(D,=)$ with one flexible variable, i.e. by infinite sequences of elements of D. As in the proofs of [2, Theorems 6 and 7], which show Σ^1_1-hardness of satisfiability of formulae of TPTL extended with either multiplication by 2 or dense time, we shall encode the value of a counter by a sequence of that length. However, much further work is needed in this proof because the only operation we have on elements of D is equality.

Let n be the number of instructions in M. We encode a configuration $\langle l, c_1, c_2\rangle$ by a sequence of elements of D of the form

$$ddd'd\underbrace{\ldots d' \ldots}_{n} f^1_1 \ldots f^1_{c_1} eee'e'' f^2_1 \ldots f^2_{c_2}$$

where:

(i) the only two pairs of consecutive elements which are equal are dd and ee, and also $f^2_{c_2}$ is distinct from the first element in the encoding of the next configuration;

(ii) $e \neq e''$;

(iii) after the first 4 elements, there is a sequence of n elements, and only the l^{th} equals d';

(iv) $f_1^i, \ldots, f_{c_i}^i$ are mutually distinct, for each i.

We write $\text{start}_{\text{d}\vee\text{e}}$ to denote the formula $x = \mathbf{X}^1 x$ stating that the current state is an occurrence of either dd or ee. We write start_{d} [resp. start_{e}] to denote the formula $\text{start}_{\text{d}\vee\text{e}} \wedge x = \mathbf{X}^3 x$ [resp. $\text{start}_{\text{d}\vee\text{e}} \wedge x \neq \mathbf{X}^3 x$] stating the current state is a first occurrence of d [resp. e] in dd [ee].

The formula ϕ_M is

$$\phi_n^{init} \wedge \phi_n^{glob} \wedge \phi_M^1 \wedge \cdots \wedge \phi_M^n \wedge \phi^{rec}$$

where the first two conjuncts state that the model is a concatenation of configuration encodings which satisfy (i)–(iv) above, and that it begins with an encoding of the initial configuration $\langle 1, 0, 0\rangle$. Their definitions are given in Figure 1.

For any $l \in \{1, \ldots, n\}$, ϕ_M^l states that, whenever the model contains an encoding of a configuration $\langle l, c_1, c_2\rangle$, then the next encoding is of a configuration which is obtained by executing the l^{th} instruction.

Consider the most complex case:

$$l: \; C_2 := C_2 - 1; \text{ goto } l' \text{ or goto } l''$$

The formula ϕ_M^l needs to state that, whenever the location counter is l, C_1 remains the same, C_2 either remains 0 or is decremented, and the next value of the location counter is either l' or l'':

$$\begin{array}{rcl}\phi_M^l & \stackrel{\text{def}}{=} & \mathbf{G}((\text{start}_{\text{d}} \wedge \mathbf{X}^2 x = \mathbf{X}^{l+3} x) \Rightarrow \\ & & \quad \mathbf{X}^{n+4}(\chi_{eq}^1 \wedge (\neg\text{start}_{\text{d}\vee\text{e}} \mathbf{U}(\text{start}_{\text{e}} \wedge \\ & & \quad\; \mathbf{X}^4(\chi_{dec}^2 \wedge (\neg\text{start}_{\text{d}\vee\text{e}} \mathbf{U}(\text{start}_{\text{d}} \wedge \\ & & \quad\;\; (\mathbf{X}^2 x = \mathbf{X}^{l'+3} x \vee \mathbf{X}^2 x = \mathbf{X}^{l''+3} x))))))))\end{array}$$

The formula χ_{dec}^2 given in Figure 2 specifies that, if the current value of C_2 is 0 or 1, then the next value of C_2 is 0; and if not, then the next encoding of the value of C_2 equals the current encoding with the last element removed. The latter is specified as the following conjunction:

(A) the first element of the current encoding equals the first element of the next encoding, and

(B) for any consecutive pair y and y' of elements in the current encoding such that y' is not the last element, the first occurence of y in the next encoding is followed by y', and

(C) the element before the last in the current encoding is the last element in the next encoding.

The formula χ_{eq}^1, which specifies that the value of C_1 remains the same, is defined similarly.

Definitions of ϕ_M^l for other forms of instruction use the same machinery. For incrementing a counter, it is not necessary to specify that the additional element in the next encoding is distinct from the rest, because that is ensured by ϕ_n^{glob}.

Finally, ϕ^{rec} states that the model encodes a recurring computation:

$$\phi^{rec} \stackrel{\text{def}}{=} \mathbf{GF}(\text{start}_{\text{d}} \wedge \mathbf{X}^2 x = \mathbf{X}^4 x) \qquad \square$$

By adapting the proof of Theorem 3, one can show that the variant of $\text{CLTL}^{\downarrow}(D, =)$ over models that are finite words as those considered in [9, 7] is also undecidable by encoding the halting problem for Minsky machines.

4.3 One rigid variable and monodic first-order temporal logics

The decidability status of $\text{CLTL}^{\downarrow}(\mathbb{N}, =)$ restricted to one rigid variable is still open (the proof of Theorem 3 uses exactly two rigid variables) which corresponds exactly to consider formulae of $\downarrow$-height 1. More precisely, $\text{CLTL}^{\downarrow}(\mathbb{N}, =)$ restricted to one rigid variable and one flexible variable is open: $\text{CLTL}^{\downarrow}(\mathbb{N}, =)$ restricted to one rigid variable can be reduced to this fragment. One way to show decidability of this fragment would be to reduce it to a decidable fragment of some first-order temporal logic. For instance, $\text{CLTL}^{\downarrow}(\mathbb{N}, =)$ satisfiability can be reduced to first-order temporal logic $\mathcal{TL}$ over the linear structure $\langle \mathbb{N}, < \rangle$ introduced in [24, Chapter 11]. Indeed, to each flexible variable x one associates a monadic predicate symbol P_x in such a way that P_x is interpreted as the singleton set containing the value of x and the translation of the formula $\downarrow_{x'=\mathbf{X}x} \; \phi$ is the $\mathcal{TL}$ formula $\exists x' \; \mathbf{X} P_x(x') \wedge \phi'$ where ϕ' is the translation of ϕ. The translation of the Boolean and temporal operators is performed homomorphically whereas $y = \mathbf{X} z$ with $y, z \in \mathsf{FleVarSet}$ is for instance translated into $\exists x \; P_y(x) \wedge \mathbf{X} P_z(x)$. One needs also to be able to express that at every state P_x is interpreted by a singleton which can be easily encoded by the formula $\mathbf{G}(\forall z, z' \; P_x(z) \wedge P_x(z') \Rightarrow z = z' \wedge \exists z \; P_x(z))$. It is then easy to check that the translation falls into the monodic fragment of $\mathcal{TL}$ whenever the $\text{CLTL}^{\downarrow}(\mathbb{N}, =)$ formula is of $\downarrow$-height 1. We recall that in the monodic fragment, any subformula of the form $\mathbf{X}\phi$, $\phi_1 \mathbf{U} \phi_2$, $\mathbf{F}\phi$, $\mathbf{G}\phi$ has at most one free individual variable.

Even though monodic first-order temporal logic over the linear structure $\langle \mathbb{N}, < \rangle$ is decidable [30], its extension with equality is not [42] and we need equality in the translation process in a substantial way. It is then easy to check that the translation falls into the monodic fragment of $\mathcal{TL}$ with only two individual variables and monadic predicate symbols whenever the $\text{CLTL}^{\downarrow}(\mathbb{N}, =)$ formula is of $\downarrow$-height 1.

$$
\begin{array}{lcl}
\phi_n^{init} & \stackrel{\text{def}}{=} & \mathrm{start_d} \wedge \mathbf{X}^2 x = \mathbf{X}^4 x \wedge \mathbf{X}^{n+4}(\mathrm{start_e} \wedge \mathbf{X}^4(\mathrm{start_{d\vee e}})) \\
\phi_n^{glob} & \stackrel{\text{def}}{=} & \mathbf{G}(\mathrm{start_d} \Rightarrow \psi_n^1 \wedge \mathrm{start_e} \Rightarrow \psi_n^2) \\
\psi_n^1 & \stackrel{\text{def}}{=} & \overbrace{\left(\bigwedge_{i=1}^{n+3} \mathbf{X}^i x \neq \mathbf{X}^{i+1} x\right)}^{\text{in } dd'd \ldots d' \ldots \text{ two consecutive values are distinct}} \wedge \\
& & \overbrace{\left(\bigvee_{l=1}^{n} \mathbf{X}^2 x = \mathbf{X}^{l+3} x \wedge \left(\bigwedge_{j=1}^{l-1} \mathbf{X}^2 x \neq \mathbf{X}^{j+3} x \wedge \bigwedge_{j=l+1}^{n} \mathbf{X}^2 x \neq \mathbf{X}^{j+3} x\right)\right)}^{\text{in } \ldots d' \ldots \text{ exactly one value equals } d'} \wedge \\
& & \overbrace{\mathbf{X}^{n+4}(\psi^{dist} \mathbf{U}\ \mathrm{start_e})}^{f_1^1 \ldots f_{c_1}^1 \text{ mutually distinct}} \\
\psi_n^2 & \stackrel{\text{def}}{=} & \left(\bigwedge_{i=1}^{3} \mathbf{X}^i x \neq \mathbf{X}^{i+1} x\right) \wedge \overbrace{\mathbf{X}^4(\psi^{dist} \mathbf{U}\ \mathrm{start_d})}^{f_1^2 \ldots f_{c_2}^2 \text{ mutually distinct}} \\
\psi^{dist} & \stackrel{\text{def}}{=} & \neg \mathrm{start_{d\vee e}} \wedge \downarrow_{y=x} \mathbf{X}((\neg \mathrm{start_{d\vee e}} \wedge x \neq y) \mathbf{U} \mathrm{start_{d\vee e}})
\end{array}
$$

Figure 1.

However, this very fragment of $\mathcal{TL}$ is also undecidable [17]. Hence, one way to show decidability of $\mathrm{CLTL}^{\downarrow}(\mathbb{N},=)$ restricted to one rigid variable would be to show the decidability of the monodic and monadic fragment of $\mathcal{TL}$ with equality, with only two individual variables and further restricted to formulae such that any subformula that contains two distinct free variables has no temporal operator (unlike formulae used in the undecidability proof in [17]). It would forbid formulae of the form $(\forall z, z'\ \mathbf{X}P(z) \wedge \mathbf{X}P(z'))$ that however still belong to the monodic fragment and it would allow formulae of the form $\mathbf{G}(\forall z, z'\ P_x(z) \wedge P_x(z') \Rightarrow z = z' \wedge \exists z\ P_x(z))$ needed to enforce that P_x is interpreted as a singleton.

5 Conclusion

In this paper, we have shown that adding the freeze operator to $\mathrm{CLTL}(\mathcal{D})$ leads to undecidability as soon as the underlying domain is infinite and the equality predicate is part of $\mathcal{D}$. As illustrated in the paper, most of related work dealing with undecidable logics having a binding-mechanism similar to the freeze quantification can encode past-time operators or has constraints richer than equality. The logic $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ is EXPSPACE-complete for most of finite $\mathcal{D}$. In order to design a specification language with LTL temporal operators and the freeze quantifier that admits a decidable model-checking problem, syntactic restrictions could be a reasonable solution. Typically, the existence of a logspace reduction from flat fragment of $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ into $\mathrm{CLTL}(\mathcal{D})$ when equality predicate is present leads us to believe that our flatness criterion is most relevant. However, some natural syntactic restrictions have not been considered in the paper and the decidability status of the fragments below is open (with D infinite):

- fragment of $\mathrm{CLTL}^{\downarrow}(D,=)$ where the operator $\mathbf{U}$ is restricted to $\mathbf{G}$,
- fragment of $\mathrm{CLTL}^{\downarrow}(D,=)$ with one rigid variable and one flexible variable,
- fragment of $\mathrm{CLTL}^{\downarrow}(D,=)$ restricted to formulae of the form $\phi_1 \wedge \phi_2$ where ϕ_2 is freeze-free and ϕ_1 is a conjunction of formulae of the form $\mathbf{G} \downarrow_{y=x} \mathbf{XG}x \neq y$: freeze operator is then only used to define nonces.

Alternatively, syntactic restrictions can be combined with restrictions on the interpretations of the variables as it is the case for TPTL [2]. For instance, which fragments of $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ are decidable assuming that the freeze operator is only used in formulae of the form $\downarrow_{y=x} \phi$ where x is bounded-reversal in the sense of [31]? Monotonic variables are in particular bounded-reversal. Finally, assuming that $\mathcal{D}$ does not contain the equality predicate and the underlying domain is infinite, it is not clear when $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ is decidable. For instance, the decidability status of $\mathrm{CLTL}^{\downarrow}(\{0,1\}^*, <)$ where $<$ is either the prefix relation or the subword relation is open. By contrast, when $\langle D, < \rangle$ is an infinite totally-ordered set, a consequence of our results is that $\mathrm{CLTL}^{\downarrow}(D,<)$ is undecidable since equality is definable.

Acknowledgements. We are grateful to Deepak D'Souza for useful discussions and to Frank Wolter for having directed us to related work.

References

[1] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[2] R. Alur and T. Henzinger. A really temporal logic. *Journal of the ACM*, 41(1):181–204, 1994.

[3] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In *Computer Science Logic*, volume 1683 of *Lecture Notes in Computer Science*, pages 307–321, 1999.

[4] P. Balbiani and J. Condotta. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *Frontiers of Combining Systems (FroCoS'02)*, volume 2309 of *Lecture Notes in Artificial Intelligence*, pages 162–173, 2002.

[5] B. Boigelot. *Symbolic methods for exploring infinite state spaces*. PhD thesis, Université de Liège, 1998.

[6] B. Boigelot and P. Wolper. Representing arithmetic constraints with finite automata: an overview. In *ICLP 2002*, volume 2401 of *Lecture Notes in Computer Science*, pages 1–19, 2002.

[7] M. Bojańczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on words with data. 2005. Submitted.

[8] A. Bouajjani, R. Echahed, and P. Habermehl. On the verification problem of nonregular properties for nonregular processes. In *Proc. 10th IEEE Symp. Logic in Computer Science (LICS '95)*, pages 123–133, 1995.

[9] P. Bouyer. A logical characterization of data languages. *Information Processing Letters*, 84(2):75–85, 2002.

[10] P. Bouyer, A. Petit, and D. Thérien. An algebraic approach to data languages and timed languages. *Information and Computation*, 182(2):137–162, 2003.

[11] T. Brihaye, V. Bruyère, and J. Raskin. Model-checking for weighted timed automata. In *FORMATS'04, Grenoble, France*, volume 3253 of *Lecture Notes in Computer Science*, pages 277–292, 2004.

[12] S. M. Cho, H. H. Kim, S. D. Cha, and D. H. Bae. A semantics of sequence diagrams. *Information Processing Letters*, 84:125–130, 2002.

[13] H. Comon and V. Cortier. Flatness is not a weakness. In *14 Int. Workshop Computer Science Logic*, volume 1862 of *Lecture Notes in Computer Science*, pages 262–276, 2000.

[14] D. Dams. Flat fragments of CTL and CTL*: separating the expressive and distinguishing powers. *Logic Journal of the IGPL*, 7(1):55–78, 1999.

[15] C. David. Mots et données infinies. Master's thesis, LIAFA, 2004. 45 pages.

[16] F. de Boer and R. van Eijk. Decidable navigation logics for object structures. In *CSL'01*, volume 2142 of *Lecture Notes in Computer Science*, pages 324–338, 2001.

[17] A. Degtyarev, M. Fisher, and A. Lisitsa. Equality and monodic first-order temporal logic. *Studia Logica*, 72:147–156, 2002.

[18] S. Demri. LTL over integer periodicity constraints. Technical Report LSV-04-6, LSV, February 2004. 35 pages. An extended abstract appeared in Proc. of FOSSACS'04.

[19] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. Technical Report LSV-03-11, LSV, 2003. 40 pages. An extended abstract appeared in Proc. of FSTTCS'02.

[20] A. Emerson and J. Halpern. 'sometimes' and 'not never' revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33:151–178, 1986.

[21] A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS'02, Kanpur*, volume 2256 of *Lecture Notes in Computer Science*, pages 145–156, 2002.

[22] M. Franceschet, M. de Rijke, and B.-H. Schlingloff. Hybrid logics on linear structures: Expressivity and complexity. In *TIME-ICTL 2003*, pages 164–171, 2003.

[23] T. French. Quantified propositional temporal logic with repeating states. In *TIME-ICTL 2003*, pages 155–165. IEEE Computer Society, 2003.

[24] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and practice*. Cambridge University Press, 2003.

[25] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. On the computational complexity of spatio-temporal logics. In *FLAIRS'03, St Augustine, Florida*, pages 460–464, 2003.

[26] V. Goranko. Hierarchies of modal and temporal logics with references pointers. *Journal of Logic, Language, and Information*, 5:1–24, 1996.

[27] E. Harel, O. Lichtenstein, and A. Pnueli. Explicit clock temporal logic. In *Proc. 5th IEEE Symp. Logic in Computer Science (LICS '90), Philadelphia, PA, USA, June 1990*, pages 400–413, 1990.

[28] T. Henzinger. Half-order modal logic: how to prove real-time properties. In *9th Annual ACM Symposium on Principles of Distributed Computing*, pages 281–296. ACM Press, 1990.

[29] I. Hodkinson, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. On the computational complexity of decidable fragments of first-order linear temporal logics. In *TIME-ICTL 2003*, pages 91–98, 2003.

[30] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134, 2000.

[31] O. Ibarra. Reversal-bounded multicounter machines and their decision problems. *Journal of the ACM*, 25(1):116–133, 1978.

[32] O. Ibarra and Z. Dang. On removing the stack from reachability constructions. In *International Symposium on Algorithms and Computation (ISAAC'01)*, volume 2223 of *Lecture Notes in Computer Science*, pages 244–256, 2001.

[33] E. Kieronski. EXPSPACE-complete variant of guarded fragment of transitivity. In *Proc. 19th Ann. Symp. Theoretical Aspects of Computer Science (STACS'2002), Antibes Juan-les-Pins, France, Mar. 2002*, volume 2285 of *Lecture Notes in Computer Science*, pages 608–619, 2002.

$$\chi^2_{dec} \stackrel{\text{def}}{=} \overbrace{((x = \mathbf{X}^1 x \vee \mathbf{X}^1 x = \mathbf{X}^2 x) \wedge (\neg \mathrm{start_e} \mathbf{U} (\mathrm{start_e} \wedge \mathbf{X}^4 (x = \mathbf{X}^1 x)))}^{0 \leq C_2 \leq 1 \text{ and the next value of } C_2 \text{ equals } 0}$$

$$\vee$$

$$(\overbrace{\neg(x = \mathbf{X}^1 x \vee \mathbf{X}^1 x = \mathbf{X}^2 x)}^{C_2 > 1} \wedge$$

$$\overbrace{(\downarrow_{y=x} \neg \mathrm{start_e}\ \mathbf{U}(\mathrm{start_e} \wedge \mathbf{X}^4(\neg \mathrm{start_d} \wedge x = y)))}^{\text{(A)}} \wedge$$

$$\overbrace{((\neg \mathbf{X}^2 \mathrm{start_{d \vee e}} \wedge \downarrow_{y=x} \mathbf{X} \downarrow_{y'=x} \neg \mathrm{start_e}\ \mathbf{U} \quad (\mathrm{start_e} \wedge \mathbf{X}^4 (x \neq y \mathbf{U} (x = y \wedge \mathbf{X}^1 x = y')))) \ \mathbf{U} \mathbf{X}^2 \mathrm{start_{d \vee e}})}^{\text{(B)}} \wedge$$

$$\overbrace{((\mathbf{X}^2 \neg \mathrm{start_d}) \mathbf{U} (\mathbf{X}^2 \mathrm{start_d} \wedge \downarrow_{y=x} \neg \mathrm{start_e} \mathbf{U} (\mathrm{start_e} \wedge \mathbf{X}^4 (x \neq y \mathbf{U}}^{\text{(C)}} \quad (x = y \wedge \neg \mathrm{start_d} \wedge \mathbf{X}^2 \mathrm{start_d})))))$$

Figure 2.

[34] F. Laroussinie, N. Markey, and P. Schnoebelen. Temporal logic with forgettable past. In *Proc. 17th IEEE Symp. Logic in Computer Science (LICS '02)*, pages 383–392, 2002.

[35] P. Revesz. *Introduction to Constraint Databases*. Springer, New York, 2002.

[36] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill Book Company, 1967.

[37] P. Schnoebelen. The complexity of temporal logic model checking. In *Advances in Modal Logic, vol. 4, selected papers from 4th Conf. Advances in Modal Logic (AiML'2002), Sep.-Oct. 2002, Toulouse, France*, pages 437–459, 2003.

[38] A. Sistla and E. Clarke. The complexity of propositional linear temporal logic. *Journal of the ACM*, 32(3):733–749, 1985.

[39] P. van Emde Boas. The convenience of tilings. In *Complexity, Logic, and recursion Theory*, volume 187 of *Lecture Notes in Pure and Applied Logic*, pages 331–363, 1997.

[40] P. Wolper. Temporal logic can be more expressive. *Information and Computation*, 56:72–99, 1983.

[41] F. Wolter and M. Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In *KR'00*, pages 3–14, 2000.

[42] F. Wolter and M. Zakharyaschev. Axiomatizing the monodic fragment of first-order temporal logic. *Annals of Pure and Applied Logic*, 118(133–145), 2002.