LTL over Integer Periodicity Constraints
(Extended Abstract)

Stéphane Demri

LSV/CNRS UMR 8643 & INRIA Futurs projet SECSI & ENS Cachan
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
demri@lsv.ens-cachan.fr

Abstract. Periodicity constraints are present in many logical formalisms, in fragments of Presburger LTL, in calendar logics, and in logics for access control, to quote a few examples. We introduce the logic PLTL^{mod}, an extension of Linear-Time Temporal Logic LTL with past-time operators whose atomic formulae are defined from a first-order constraint language dealing with periodicity. The underlying constraint language is a fragment of Presburger arithmetic shown to admit a PSPACE-complete satisfiability problem and we establish that PLTL^{mod} model-checking and satisfiability problems are in PSPACE as plain LTL. The logic PLTL^{mod} is a quite rich and concise language to express periodicity constraints. We show that adding logical quantification to PLTL^{mod} provides EXPSPACE-hard problems. As another application, we establish that the equivalence problem for extended single-string automata, known to express the equality of time granularities, is PSPACE-complete. The paper concludes by presenting a bunch of open problems related to fragments of Presburger LTL.

1 Introduction

Presburger Constraints. Presburger constraints are present in many logical formalisms including extensions of Linear-Time Logic LTL, see e.g. [AH94,Čer94] (and also [BEH95,CC00,BC02,DD02]). Formalisms with such constraints are also known to be well-suited for the specification and verification of infinite-state systems, see e.g. [BH99,WB00,FL02].

In the paper, we are interested in models of Presburger LTL that are ω-sequences of valuations for a given set VAR of integer variables taking their values in $\mathbb{Z}$ and the atomic formulae are Presburger arithmetic constraints with free variables in VAR. For instance, $\phi = \Box(\mathsf{X}x = x)$ states that the value of the variable x is constant over the time line where $\mathsf{X}x$ denotes the value of x at the next state. A model of ϕ is simply an ω-sequence in $(\mathbb{Z})^\omega$. The counterpart of the high expressive power of Presburger LTL rests on its undecidability, shown by a standard encoding of the halting problem for two-counter machines. However, to regain decidability one can either restrict the underlying constraint language, see e.g. [AH94, Sect. 3] and [DD02], or restrict the logical language, see e.g. a decidable flat fragment of Presburger LTL in [CC00]. Herein, we shall consider versions of LTL with Presburger constraints with the *full logical language* (mainly LTL with past-time operators sometimes augmented with first-order quantifiers) but with *strict fragments of Presburger arithmetic.*

Our motivation. Integer periodicity constraints, a special class of Presburger constraints, have found applications in many logical formalisms such as DATALOG with integer periodicity constraints [TC98], logical formalisms dealing with calendars, see e.g. [Ohl94,Wij00,CFP02], temporal reasoning in database access control [BBFS96,BBFS98], and reasoning about periodic time in generalized databases, see e.g. [NS92]. In view of the ubiquity of such constraints, the main motivation of the current work is to design a variant of LTL over a language for integer periodicity constraints that satisfies the following nice properties.

– The logical language contains at least LTL (no flatness restriction).
– The constraint language is expressive enough to capture most integer periodicity constraints used in calendar logics and in database access control. For instance, in [CFP02], the authors advocate the need to design an extension of LTL that expresses quantitative temporal requirements, such as periodicity constraints. We provide in the paper such an extension.
– Model-checking and satisfiability remain in PSPACE and possibly to adapt the technique with Büchi automata [VW94] to this new extension of LTL.

Last but not least, as a long-term project, we wish to understand what are the decidable fragments of Presburger LTL by restricting the constraint language but with the full logical language.

Our contribution. We introduce a decidable fragment of Presburger LTL that satisfies the above-mentioned requirements. Let us be a bit more precise.

1. We introduce a first-order theory of integer periodicity constraints IPC^{++} and we show its PSPACE-completeness (Sects. 2 and 3). This is a fragment of Presburger arithmetic that extends the one from [TC98].
2. We show the PSPACE-completeness of PLTL (LTL with past-time operators) over IPC^{++} (logic denoted by $\mathrm{PLTL}^{\mathrm{mod}}$ in the paper) by using Büchi automata (Sect. 4) in the line of [VW94].
3. We prove that adding the existential operator $\exists$ at the logical level ($\exists$ is already present at the constraint level) may lead to an exponential blow-up of the complexity (Sect. 5). We show that $\mathrm{PLTL}(\mathrm{IPC}^{+})$, a fragment of $\mathrm{PLTL}^{\mathrm{mod}}$, augmented with $\exists$ has a satisfiability problem in EXPSPACE and $\mathrm{PLTL}^{\mathrm{mod}}$ augmented with $\exists$ is EXPSPACE-hard.
4. As an application, we show the PSPACE-completeness of the equivalence problem for the extended single-string automata improving the complexity bound from [LM01, Sect. 5] (Sect. 6). Extended single-string automata are Büchi automata that recognize exactly one ω-word and guards involving periodicity constraints are present on the transitions. This formalism has been introduced as a concise means to define time granularities and the equivalence problem for such automata is central to check the equality of time granularities, see also [Wij00].

Because of lack of space, the proofs are omitted and can be found in [Dem03].

2 PLTL over Periodicity Constraints

2.1 Constraint Languages

Let $\mathrm{VAR} = \{x_0, x_1, \ldots\}$ be a countably infinite set of variables. The constraint language IPC is defined by the grammar $p ::= x \equiv_k y + c \mid x \equiv_k c \mid p \wedge p \mid \neg p$, where $k, c \in \mathbb{N}$. A simple periodicity constraint is a conjunction of constraints of the form either $x \equiv_k y + c$ or $x \equiv_k c$. Given $X \subseteq \{\exists, [], <, =\}$, we define an extension of IPC, namely IPC^X, by adding clauses to the definition of IPC:

- if $\exists \in X$, then the clause $\exists\ x\ p$ is added (existential quantification);
- if $[] \in X$, then the clause $x \equiv_k y + [c_1, c_2]$ with $c_1, c_2 \in \mathbb{N}$ is added;
- if $= \in X$, then the clause $x = y$ with $x, y \in \mathrm{VAR}$ is added;
- if $< \in X$, then the clauses $x < c \mid x > c \mid x = c$ with $x \in \mathrm{VAR}$ and $c \in \mathbb{Z}$ are added.

In the sequel, IPC^+ denotes $\mathrm{IPC}^{\{\exists, [], <\}}$ and IPC^{++} denotes $\mathrm{IPC}^{\{\exists, [], <, =\}}$, which is the richer constraint language considered in the paper. IPC^{++} is the extension of the language of the first-order theory of integer periodicity constraints introduced in [TC98] but with the inclusion of negation as in [BBFS96]. A semi-simple periodicity constraint is a conjunction between a simple periodicity constraint and a conjunction of atomic constraints of the form $x \sim c$ with $\sim \in \{<, >, =\}$. The interpretation of the constraints is standard (v is a map $v : \mathrm{VAR} \to \mathbb{Z}$):

- $v \models x \sim c \overset{\text{def}}{\Leftrightarrow} v(x) \sim c$ with $\sim \in \{<, >, =\}$; $v \models x = y \overset{\text{def}}{\Leftrightarrow} v(x) = v(y)$;
- $v \models x \equiv_k c \overset{\text{def}}{\Leftrightarrow} v(x)$ is equal to c modulo k;
- $v \models x \equiv_k y + c \overset{\text{def}}{\Leftrightarrow} v(x) - v(y)$ is equal to c modulo k;
- $v \models x \equiv_k y + [c_1, c_2] \overset{\text{def}}{\Leftrightarrow} v(x) - v(y)$ is equal to c modulo k for some $c_1 \leq c \leq c_2$;
- $v \models p \wedge p' \overset{\text{def}}{\Leftrightarrow} v \models p$ and $v \models p'$; $v \models \neg p \overset{\text{def}}{\Leftrightarrow}$ not $v \models p$;
- $v \models \exists\ x\ p \overset{\text{def}}{\Leftrightarrow}$ there is $c \in \mathbb{Z}$ s.t. $v[x \leftarrow c] \models p$ where $v[x \leftarrow c](x') = v(x')$ if $x' \neq x$, and $v[x \leftarrow c](x) = c$.

Given p in IPC^{++} with free variables $x_1, \ldots, x_k$ (in the order of enumeration of the variables), $\mathrm{sol}(p)$ denotes the set of k-tuples $\langle n_1, \ldots, n_k \rangle \in \mathbb{Z}^k$ such that $[x_1 \leftarrow n_1, \ldots, x_k \leftarrow n_k] \models p$. Given a constraint language L, the L-satisfiability problem is to decide given a constraint $p \in \mathrm{L}$ whether $\mathrm{sol}(p)$ is non-empty. Without any loss of generality, we assume that p contains at least one free variable (otherwise consider $(x_1 \equiv_1 0) \wedge p$ and x_1 does not occur in p) and in p a variable cannot occur both free and bounded.

The expressive power of a constraint language L is measured by the set $\{\mathrm{sol}(p) : p \in \mathrm{L}\}$. For instance, $\mathrm{IPC}^{\{\exists, <\}}$ is as expressive as IPC^+ since $x \equiv_k y + [c_1, c_2]$ is equivalent to $\bigvee_{c_1 \leq c \leq c_2} x \equiv_k y + c$. However, because all the natural numbers are encoded in binary, IPC^+ may be more concise than $\mathrm{IPC}^{\{\exists, <\}}$. The introduction of the concise atomic constraints of the form $x \equiv_k y + [c_1, c_2]$ is motivated by the existence of such constraints in the calendar logic from [Ohl94].

2.2 Definition of PLTL$^{\mathrm{mod}}$

The atomic formulae of PLTL$^{\mathrm{mod}}$ are expressions of the form $p[x_1 \leftarrow \mathsf{X}^{i_1} x_{j_1}, \ldots, x_k \leftarrow \mathsf{X}^{i_k} x_{j_k}]$ where p is a constraint of IPC^{++} with free variables $x_1, \ldots, x_k$ (in the order of enumeration of the variables) and $p[x_1 \leftarrow \mathsf{X}^{i_1} x_{j_1}, \ldots, x_k \leftarrow \mathsf{X}^{i_k} x_{j_k}]$ is obtained from p by replacing every occurrence of x_u by x_{j_u} preceded by i_u next symbols for $1 \leq u \leq k$. For instance, the formula $x \equiv_2 0 \wedge \Box(\mathsf{X}x \equiv_2 x+1)$ states that the value of x is even on states of even indices. Otherwise stated, the atomic formulae of PLTL$^{\mathrm{mod}}$ are the constraints of IPC^{++} except that the variables are of the form $\mathsf{X}^j x_i$. The formulae of PLTL$^{\mathrm{mod}}$ are defined by the grammar $\phi ::= p[x_1 \leftarrow \mathsf{X}^{i_1} x_{j_1}, \ldots, x_k \leftarrow \mathsf{X}^{i_k} x_{j_k}] \mid \neg\phi \mid \phi \wedge \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}^{-1}\phi \mid \phi \mathsf{S} \phi$, where p belongs to IPC^{++}. As usual, X is the next-time operator, X^{-1} is the previous past-time operator, U is the until operator, and S is the since past-time operator. We write PLTL(L) to denote the variant of PLTL$^{\mathrm{mod}}$ where the atomic formulae are built from the constraint language L: PLTL$^{\mathrm{mod}}$ is simply PLTL(IPC^{++}). We write LTL(L) to denote the restriction of PLTL(L) to the future-time operators X and U. We include past-time operators in the logic in order to capture the conciseness of LTL with past considered in [CFP02]. However, the addition of a finite amount of MSO-definable temporal operators still guarantees the (forthcoming) PSPACE upper bound thanks to [GK03].

A model σ for PLTL$^{\mathrm{mod}}$ is an ω-sequence of valuations of the form $\sigma : \mathbb{N} \times \mathrm{VAR} \rightarrow \mathbb{Z}$. The satisfiability relation $\models$ is inductively defined below:

- $\sigma, i \models p[x_1 \leftarrow \mathsf{X}^{i_1} x_{j_1}, \ldots, x_k \leftarrow \mathsf{X}^{i_k} x_{j_k}]$ iff $[x_1 \leftarrow \sigma(i+i_1, x_{j_1}), \ldots, x_k \leftarrow \sigma(i+i_k, x_{j_k})] \models p$ (for IPC^{++});
- $\sigma, i \models \phi \wedge \phi'$ iff $\sigma, i \models \phi$ and $\sigma, i \models \phi'$; $\sigma, i \models \neg\phi$ iff not $\sigma, i \models \phi$;
- $\sigma, i \models \mathsf{X}\phi$ iff $\sigma, i+1 \models \phi$; $\sigma, i \models \mathsf{X}^{-1}\phi$ iff $i > 0$ and $\sigma, i-1 \models \phi$;
- $\sigma, i \models \phi \mathsf{U} \phi'$ iff there is $j \geq i$ s.t. $\sigma, j \models \phi'$ and for every $i \leq k < j$, $\sigma, k \models \phi$;
- $\sigma, i \models \phi \mathsf{S} \phi'$ iff there is $0 \leq j \leq i$ s.t. $\sigma, j \models \phi'$ and for every $j < k \leq i$, $\sigma, k \models \phi$.

A very important aspect of PLTL$^{\mathrm{mod}}$ rests on the fact that the values of variables at different states can be compared. We use the standard abbreviations $\Box\phi, \ldots$ The satisfiability problem for PLTL$^{\mathrm{mod}}$ is to decide given a formula ϕ whether there is σ such that $\sigma, 0 \models \phi$. A few other remarks are in order. No propositional variables are part of PLTL$^{\mathrm{mod}}$ but they can be easily simulated. Furthermore, we can simulate the access to past values of variables. For instance, $\mathsf{X}^{-2}x = x$ can be translated into $\mathsf{X}^{-1}\mathsf{X}^{-1}\top \wedge \mathsf{X}^{-1}\mathsf{X}^{-1}(x = \mathsf{X}^2 x)$ assuming that if $\mathsf{X}^{-2}x$ is undefined, then the atomic constraint is interpreted by false. When complexity issues are considered, all the integers are encoded in binary representation.

PLTL$^{\mathrm{mod}}$ is a quite rich and concise language to express periodicity constraints. Formulae of PLTL$^{\mathrm{mod}}$ encode calendars and slices from [NS92], and formulae of the form $[\tau]\phi$ from [Ohl94] where $[\tau]\phi$ is interpreted by "for every point of the interval τ, the formula ϕ holds" can be encoded by $\Box(\tau' \Rightarrow \phi)$. Here τ' is a constraint in IPC^{++} encoding τ. Unlike what is done in [Ohl94], no exponential-time reduction to propositional calculus (PC) is performed. PLTL$^{\mathrm{mod}}$ allows more efficient reasoning than an expensive translation into PC (see details in

Sect. 4). Furthermore, we provide a quantitative version of LTL that meet the requirements from [CFP02] in order to deal with periodicity constraints.

2.3 Model-Checking

The languages of the form PLTL(L) are of course well-designed to perform model-checking of counter automata, similarly to what is done in [Čer94,DD03]. Given a constraint language L, a PLTL(L)-automaton is a Büchi automaton $\mathcal{A}$ over the alphabet of PLTL(L) formulas: transitions are of the form $q \xrightarrow{\phi} q'$. To each ω-word $w = \phi_0\phi_1\cdots$ accepted by $\mathcal{A}$, we associate a model σ which satisfies $\sigma, i \models \phi_i$ for $i \geq 0$. Let $l(\mathcal{A})$ denote the set $l(\mathcal{A}) = \{\sigma : \mathbb{N} \times \mathrm{VAR} \rightarrow \mathbb{Z} \mid \exists w$ accepted by $\mathcal{A}$ such that $\sigma, i \models w(i)$ for each $i\}$. The model-checking problem for PLTL(L) is defined as follows: given a PLTL(L)-automaton $\mathcal{A}$ and a PLTL(L) formula ϕ, is there a $\sigma \in l(\mathcal{A})$ such that $\sigma \models \phi$?

Theorem 1. *The model-checking and satisfiability problems for* $\mathrm{PLTL}^{\mathrm{mod}}$ *are inter-reducible with respect to logspace transformations.*

The proof is similar to the proof of [DD03, Theorem 8.3]. That is why in the sequel, only satisfiability problems are explicitly treated.

3 First-Order Theory of Integer Periodicity Constraints

Given p in IPC^{++} with free variables $x_1, \ldots, x_k$, we shall construct a finite partition of $\mathbb{Z}^k$ such that (1) every region can be represented by a semi-simple periodicity constraint, and (2) for all k-tuples $\overline{z}$ and $\overline{z'}$ in a given region of the partition, $\overline{z} \in \mathrm{sol}(p)$ iff $\overline{z'} \in \mathrm{sol}(p)$. In this way, we are able to finitely represent the set of solutions $\mathrm{sol}(p)$ and such a representation is easy to manipulate since it can be viewed as a disjunction of semi-simple periodicity constraints. This is actually a standard requirement when an infinite set of tuples has to be finitely abstracted, see e.g. the clock regions for timed automata in [AD94].

3.1 Quantifier Elimination

Quantifier elimination (QE) is a known method to show decidability of logical theories, see e.g. [Pre29,KK67]. In this section, we establish such a property to prove the PSPACE upper bound of the IPC^{++}-satisfiability problem. Let p be a constraint in IPC^{++} such that

- $c_1 < \ldots < c_n$ are the constants in p occurring in constraints of the form $x \sim c$ with $\sim \in \{<, >, =\}$; we also fix $c_0 = -\infty$ and $c_{n+1} = +\infty$;
- $k_1, \ldots, k_u$ are the natural numbers occurring in constraints of the form $x \equiv_k y + [d_1, d_2]$; we fix K to be the least common multiple of $1, k_1, \ldots, k_u$, denoted by $lcm(1, k_1, \ldots, k_u)$. K is in $2^{\mathcal{O}(|p|)}$ where $|p|$ is the size of p for some reasonably succinct encoding.

Given p, we define an equivalence relation $\sim_p \subseteq \mathbb{Z} \times \mathbb{Z}$ as follows: $z \sim_p z'$ $\overset{\text{def}}{\Leftrightarrow}$ (1) for all $i \leq j \in \{0, \ldots, n+1\}$, $c_i \leq z \leq c_j$ iff $c_i \leq z' \leq c_j$, and (2) for every $l \in \{0, \ldots, K-1\}$, $z \equiv_K l$ iff $z' \equiv_K l$. Hence, the number of equivalence classes of $\sim_p$ is bounded by $(n+1) \times K$, that is in $2^{\mathcal{O}(|p|)}$. The idea behind the definition of $\sim_p$ is simply that $z \sim_p z'$ iff z and z' cannot be distinguished by constraints of IPC$^+$ that use only $c_1, \ldots, c_n$ and $k_1, \ldots, k_u$. For instance, it is easy to check that for every $j \in \{1, \ldots, n\}$, $\{c_j\}$ is an equivalence class of $\sim_p$. The relation $\sim_p$ extended to tuples will not be a simple component-wise extension because of the presence equality in IPC^{++}. For $k \geq 1$, we say that $\langle z_1, \ldots, z_k \rangle = \overline{z} \sim_p^k \overline{z'} = \langle z'_1, \ldots, z'_k \rangle$ iff for every $i \in \{1, \ldots, k\}$, $z_i \sim_p z'_i$, and for all $i, j \in \{1, \ldots, k\}$, $z_i = z_j$ iff $z'_i = z'_j$. If $x_1, \ldots, x_k$ are the free variables in p, we write $\overline{z} \sim_p \overline{z'}$ instead of $\overline{z} \sim_p^k \overline{z'}$. The number of equivalence classes of $\sim_p$ (on k-tuples) is bounded by $(n+1) \times K \times 2^{k^2}$.

Lemma 1. *Let p be a constraint in* IPC^{++} *with k free variables and $\overline{z}, \overline{z'} \in \mathbb{Z}^k$. $\overline{z} \in \mathrm{sol}(p)$ and $\overline{z} \sim_p \overline{z'}$ imply $\overline{z'} \in \mathrm{sol}(p)$.*

The proof can be found in [Dem03]. Each equivalence class of $\sim_p$ on $\mathbb{Z}$ can be represented by a triple $\langle i, j, l \rangle$ with $i, j \in \{0, \ldots, n+1\}$ and $l \in \{0, \ldots, K-1\}$ such that (1) $i \leq j \leq i+1$, (2) if $i = j$ and $c_i \equiv_K l$ then $\langle i, j, l \rangle$ represents the equivalence class $\{c_i\}$, and (3) if $j = i+1$, then $\langle i, j, l \rangle$ represents the equivalence class $\{z \in \mathbb{Z} : c_i < z < c_{i+1}, \text{ and } z \equiv_K l\}$ if this set is non empty. We introduce the map $[\cdot] : \mathbb{Z} \rightarrow \{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\}$ such that $[z]$ is the equivalence class of $\sim_p$ containing z. For instance, if $c_i \equiv_K 0$, then $[c_i] = \langle i, i, 0 \rangle$. By extension, given Y a non-empty finite subset of $\mathbb{N}$ of cardinality k representing a set of variable indices, we introduce the map $[\cdot]^Y : \mathbb{Z}^k \rightarrow (\{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\})^k \times \mathcal{P}(Y^2)$ such that $[\langle z_1, \ldots, z_k \rangle]^Y = \langle \langle [z_1], \ldots, [z_k] \rangle, \{\langle J_i, J_j \rangle \in Y^2 : z_i = z_j\} \rangle$, where $Y = \{J_1, \ldots, J_k\}$ and $J_1 < \ldots < J_k$. If p has free variables $x_1, \ldots, x_k$, the finite set $(\{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\})^k \times \mathcal{P}(\{1, \ldots, k\}^2)$ will represent the equivalence classes of $\sim_p$ on k-tuples.
If p contains k free variables $x_1, \ldots, x_k$, we write D_p to denote the domain $(\{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\})^k \times \mathcal{P}(\{1, \ldots, k\}^2)$ and D_p^{sat} to denote the set $\{[\overline{z}]^{\{1, \ldots, k\}} \in D_p : \overline{z} \in \mathrm{sol}(p)\}$. The set D_p is indeed a finite abstraction of the infinite domain $\mathbb{Z}^k$ with respect to the constraint p and D_p^{sat} is a finite representation of the possibly infinite set $\mathrm{sol}(p)$. In the sequel, we show how an element of D_p^{sat} can be represented by a semi-simple periodicity constraint.

To each $\langle i, j, l \rangle \in \{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\}$, and variable index $\alpha \in \mathbb{N}$, we associate a semi-simple periodicity constraint IPC$^<(\langle i, j, l \rangle, \alpha)$ in IPC$^{\{<\}}$ with free variable x_α defined as follows:

$$\mathrm{IPC}^<(\langle i, j, l \rangle, \alpha) = (x_\alpha \equiv_K l) \wedge \begin{cases} x_\alpha = c_i \text{ if } i = j, \\ (c_i < x_\alpha) \wedge (x_\alpha < c_j) \\ \text{if } j = i+1,\ i \neq 0, \text{ and } j \neq n+1, \\ x_\alpha < c_1 \text{ if } i = 0 \text{ and } j = 1, \\ c_n < x_\alpha \text{ if } i = n \text{ and } j = n+1, \\ \text{undefined otherwise.} \end{cases}$$

We are now able to show that IPC^{++} satisfies (QE) by appropriately extending the map $\mathrm{IPC}^{<}$. To each $\langle\langle t_1, \ldots, t_k\rangle, X\rangle \in D_p$ we associate a semi-simple periodicity constraints $\mathrm{IPC}^{++}(\langle\langle t_1, \ldots, t_k\rangle, X\rangle)$ defined by

$$(\bigwedge_{1 \leq i \leq k} \mathrm{IPC}^{<}(t_i, i)) \wedge (\bigwedge_{\langle i,j\rangle \in X} x_i = x_j) \wedge (\bigwedge_{\langle i,j\rangle \notin X} \neg(x_i = x_j)).$$

The following lemma (not difficult to show) makes explicit the relationship between the constraints generated by the map $\mathrm{IPC}^{++}(\cdot)$ and the map $[\cdot]^{\{1,\ldots,k\}}$.

Lemma 2. *For all $\langle z_1, \ldots, z_k\rangle \in \mathbb{Z}^k$ and $u \in D_p$, we have $[x_1 \leftarrow z_1, \ldots, x_k \leftarrow z_k] \models \mathrm{IPC}^{++}(u)$ iff $[\langle z_1, \ldots, z_k\rangle]^{\{1,\ldots,k\}} = u$.*

Theorem 2. IPC^{++} *admits quantifier elimination.*

Proof. Let p be a constraint in IPC^{++} with free variables $x_1, \ldots, x_k$. We define below a constraint p' in IPC^{++} such that $\mathrm{sol}(p) = \mathrm{sol}(p')$:

$$p' = \bigvee_{\langle\langle t_1,\ldots,t_k\rangle, X\rangle \in D_p^{sat}} \mathrm{IPC}^{++}(\langle\langle t_1, \ldots, t_k\rangle, X\rangle).$$

Equality between $\mathrm{sol}(p)$ and $\mathrm{sol}(p')$ can be proved by using Lemma 2.

3.2 PSPACE-Complete Satisfiability Problem

We establish that IPC^{++}-satisfiability is decidable in polynomial space.

Theorem 3. IPC^{++}*-satisfiability is* PSPACE*-complete.*

Proof. (idea) PSPACE-hardness is immediate by reducing QBF. Satisfiability in PSPACE can be shown via a procedure similar to first-order model-checking [CM77], see details in [Dem03]. The PSPACE upper bound is obtained since the recursion depth of the procedure is polynomial and quantification over exponential size sets is performed, which requires only polynomial space.

The PSPACE-completeness of IPC^{++}-satisfiability does not play in favor of the tractability of this first-order theory, especially if one compares it with NLOGSPACE consistency problems. However, Presburger arithmetic is of much higher complexity and PSPACE-hardness is the optimal lower bound one can expect for PSPACE-complete PLTL over fragments of Presburger arithmetic.

Corollary 1. *Let p be a constraint in* IPC^{++}*. Checking whether $u \in D_p$ belongs to D_p^{sat} can be done in* PSPACE*.*

Finally (QE) holds and requires only polynomial space.

Corollary 2. *Given a constraint p in* IPC^{++}*, one can compute an equivalent quantifier-free p' in polynomial space in $|p|$ (but $|p'|$ is in $\mathcal{O}(2^{|p|})$).*

This is a mere consequence of the proof of Theorem 2, Corollary 1, and the fact that the elements of D_p can be enumerated using polynomial space in $|p|$.

4 Complexity of PLTL$^{\mathrm{mod}}$

Let ϕ be a PLTL$^{\mathrm{mod}}$ formula with free variables $x_1, \ldots, x_s$, constants $c_1 < \ldots < c_n$ ($c_0 = -\infty$ and $c_{n+1} = +\infty$), and natural numbers $k_1, \ldots, k_u$ occurring in the context of $\equiv$-atomic formulae and their lcm is K. Without any loss of generality, we can assume that these sets of integers are non-empty. Let $\mathsf{X}(\phi)$ be one plus the greatest i for some term $\mathsf{X}^i x_j$ occurring in ϕ. For instance, $\mathsf{X}(\phi) = 2$ with $\phi = \Box(\mathsf{X}x \equiv_4 x+1)$. In the sequel, we pose $l = \mathsf{X}(\phi)$. l is the maximal number of consecutive states necessary to evaluate an atomic subformula of ϕ. We provide below a procedure to decide satisfiability of ϕ using only polynomial space in $|\phi|$.

4.1 Abstraction of PLTL$^{\mathrm{mod}}$ Models

A model σ of ϕ is a structure $\sigma : \mathbb{N} \times \{x_1, \ldots, x_s\} \rightarrow \mathbb{Z}$ such that $\sigma, 0 \models \phi$. However, each $\sigma(i) : \{x_1, \ldots, x_s\} \rightarrow \mathbb{Z}$ can take an infinite amount of values. By contrast, for classical LTL, there is a finite amount of intepretations over a finite set of propositional variables. That is why, we abstract such valuations as elements of a finite set, more precisely the set $(\{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\})^k \times \mathcal{P}(\{1, \ldots, k\}^2)$ with $k = s \times l$. The rest of this section is dedicated to such abstractions by using Sect. 3.

Another way to understand a function $\sigma : \mathbb{N} \times \{x_1, \ldots, x_s\} \rightarrow \mathbb{Z}$ with the PLTL$^{\mathrm{mod}}$ semantics, is to view it as a structure $\sigma' : \mathbb{N} \times (\{x_1, \ldots, x_s\} \times \{0, \ldots, l-1\}) \rightarrow \mathbb{Z}$ such that (C1) for all $i \in \mathbb{N}$, $\alpha \in \{1, \ldots, s\}$, and $\beta \in \{1, \ldots, l-1\}$, $\sigma'(i, \langle x_\alpha, \beta \rangle) = \sigma'(i+1, \langle x_\alpha, \beta - 1 \rangle)$. In that way, the pair $\langle x_\alpha, \beta \rangle$ plays the rôle of $\mathsf{X}^\beta x_\alpha$. So far, the profile of σ' depends on ϕ by the value l and the number of variables s but one has also to relate σ' with σ. The condition (C2) below does the job: (C2) for all $i \in \mathbb{N}$ and $\alpha \in \{1, \ldots, s\}$, $\sigma'(i, \langle x_j, 0 \rangle) = \sigma(i, x_j)$.

The lemma states the relevance of this encoding.

Lemma 3. *ϕ is satisfiable iff there is a structure $\sigma' : \mathbb{N} \times (\{x_1, \ldots, x_s\} \times \{0, \ldots, l-1\}) \rightarrow \mathbb{Z}$ satisfying (C1) such that $\sigma', 0 \models \phi'$ where ϕ' is obtained from ϕ by replacing every occurrence of $\mathsf{X}^\beta x_\alpha$ by $\langle x_\alpha, \beta \rangle$.*

In Lemma 3 above, we assume that $\sigma', i \models p[x_1 \leftarrow \langle x_{j_1}, \beta_1 \rangle, \ldots, x_d \leftarrow \langle x_{j_d}, \beta_d \rangle]$ holds true with $p \in$ IPC^{++} and p has free variables $x_1, \ldots, x_d$ whenever $[x_1 \leftarrow \sigma'(i, \langle x_{j_1}, \beta_1 \rangle), \ldots, x_d \leftarrow \sigma'(i, \langle x_{j_d}, \beta_d \rangle)] \models p$ in IPC^{++}. For the Boolean and temporal operators, the relation $\models$ on structures σ' is defined in the homomorphic way.

Let us now abstract the functions of the form $\sigma' : \mathbb{N} \times (\{x_1, \ldots, x_s\} \times \{0, \ldots, l-1\}) \rightarrow \mathbb{Z}$. We pose $k = s \times l$ and we write D_ϕ to denote the set $(\{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\})^k \times \mathcal{P}(\{1, \ldots, k\}^2)$ by similarity to the developments made in Sect. 3. D_ϕ^{sat} is defined as the subset of D_ϕ which is the image of $[\cdot]^{\{1, \ldots, k\}}$. In order to relate terms of the form $\mathsf{X}^\beta x_\alpha$ and variables x_i ($i \in \{1, \ldots, k\}$), we introduce the map $f : \{x_1, \ldots, x_s\} \times \{0, \ldots, l-1\} \rightarrow \{1, \ldots, k\}$ as the bijection defined by $f(\langle x_\alpha, \beta \rangle) = s \times \beta + \alpha$. The inverse function f^{-1} can be easily defined with the operations of the Euclidean division. Details are omitted here. One can check that $f^{-1}(1), f^{-1}(2), \ldots, f^{-1}(k)$ is precisely the sequence $\langle x_1, 0\rangle, \langle x_2, 0\rangle, \ldots, \langle x_s, 0\rangle, \langle x_1, 1\rangle, \ldots, \langle x_1, l-1\rangle, \langle x_2, l-1\rangle, \ldots, \langle x_s, l-1\rangle$, that is, first the variables at the current state are enumerated, then the variables at the next state are enumerated and so on.

Another way to understand a map $\sigma : \mathbb{N} \times (\{x_1, \ldots, x_s\} \times \{0, \ldots, l-1\}) \rightarrow \mathbb{Z}$ is to view it as a map $\sigma' : \mathbb{N} \rightarrow D_\phi^{\mathrm{sat}}$ such that **(C3)** for every $i \in \mathbb{N}$, if $\sigma'(i) = \langle\langle t_1, \ldots, t_k\rangle, X\rangle$ and $\sigma'(i+1) = \langle\langle t'_1, \ldots, t'_k\rangle, X'\rangle$ then

1. $\langle t_{s+1}, \ldots, t_k\rangle = \langle t'_1, \ldots, t'_{k-s}\rangle$ (shift of the values of s first variables)
2. $X \cap \{s+1, \ldots, k\}^2 = \{\langle u+s, v+s\rangle : \langle u, v\rangle \in X', u+s \leq k, v+s \leq k\}$ (preservation in X' of X restricted to the indices in $\{s+1, \ldots, k\}$).

One has also to relate σ' with σ. The condition (C4) below does the job. First we need a preliminary definition. Given $g : \{x_1, \ldots, x_s\} \times \{0, \ldots, l-1\} \rightarrow \mathbb{Z}$, we write g^k to denote the k-tuple $\langle g(f^{-1}(1)), \ldots, g(f^{-1}(k))\rangle$. g^k is simply a representation of g as a k-tuple of $\mathbb{Z}^k$ with $k = s \times l$. **(C4)** is then defined as the condition: for all $i \in \mathbb{N}$, $\sigma'(i) = [\sigma(i)^k]^{\{1, \cdots, k\}}$. The following lemma shows the relevance of this abstraction.

Lemma 4. *ϕ is satisfiable iff there is a structure $\sigma' : \mathbb{N} \rightarrow D_\phi^{\mathrm{sat}}$ satisfying (C3) such that $\sigma', 0 \models \phi'$ where ϕ' is obtained from ϕ by replacing every occurrence of $\mathsf{X}^\beta x_\alpha$ by $x_{f(\langle x_\alpha, \beta\rangle)}$.*

In Lemma 4 above, we assume that $\sigma', i \models p[x_1 \leftarrow x_{f(\langle x_{j_1}, \beta_1\rangle)}, \ldots, x_d \leftarrow x_{f(\langle x_{j_d}, \beta_d\rangle)}]$ holds true with $p \in \mathrm{IPC}^{++}$ and p has free variables $x_1, \ldots, x_d$ whenever $p[x_1 \leftarrow x_{f(\langle x_{j_1}, \beta_1\rangle)}, \ldots, x_d \leftarrow x_{f(\langle x_{j_d}, \beta_d\rangle)}] \wedge \mathrm{IPC}^{++}(\sigma'(i))$ is IPC^{++} satisfiable where $\mathrm{IPC}^{++}(.)$ is the map defined in Sect. 3.1. For the Boolean and temporal operators, the relation $\models$ on structures σ' is defined in the homomorphic way. The abstraction of $\mathrm{PLTL}^{\mathrm{mod}}$ models is now satisfying since the domain of σ' in Lemma 4 is finite and is of exponential cardinality in $|\phi|$.

4.2 Büchi Automata

Using the standard approach for LTL reducing model checking and satisfiability problems to the emptiness problem for Büchi automata [VW94], we construct a Büchi automaton $\mathcal{A}_\phi$ on the alphabet D_ϕ such that $\mathrm{L}(\mathcal{A}_\phi)$, the language recognized to $\mathcal{A}_\phi$, is non-empty iff ϕ is $\mathrm{PLTL}^{\mathrm{mod}}$ satisfiable. The automaton $\mathcal{A}_\phi$ is defined as the intersection of the following Büchi automata.

1. The automaton $\mathcal{A}_{D_\phi^{\mathrm{sat}}}$ recognizes all the ω-sequences in $(D_\phi^{\mathrm{sat}})^\omega$. $\mathcal{A}_{D_\phi^{\mathrm{sat}}}$ is the structure $\langle Q, Q_0, \rightarrow, F\rangle$ such that $Q = Q_0 = F = D_\phi$ and $u \xrightarrow{u''} u'$ iff $u = u''$ and $u \in D_\phi^{\mathrm{sat}}$. By Corollary 1, one can check in polynomial space in $|\phi|$ whether $u \xrightarrow{u''} u'$.

2. The automaton $\mathcal{A}_{(C3)}$ recognizes the ω-sequences satisfying (C3). $\mathcal{A}_{(C3)}$ is the structure $\langle Q, Q_0, \rightarrow, F\rangle$ such that $Q = Q_0 = F = D_\phi$ and $u \xrightarrow{u''} u'$ iff $u = u''$ and if $u = \langle\langle t_1, \ldots, t_k\rangle, X\rangle$ and $u' = \langle\langle t'_1, \ldots, t'_k\rangle, X'\rangle$ then $\langle t_{s+1}, \ldots, t_k\rangle = \langle t'_1, \ldots, t'_{k-s}\rangle$ and $X \cap \{s+1, \ldots, k\}^2 = \{\langle u+s, v+s\rangle : \langle u, v\rangle \in X', u+s \leq k, v+s \leq k\}$. One can check in polynomial time in $|\phi|$ whether $u \xrightarrow{u''} u'$.
3. The automaton $\mathcal{A}_{\mathrm{PLTL}}$ recognizes the ω-sequences in D_ϕ^{sat} that satisfying ϕ (with the extended version of the relation $\models$).

The rest of this section is dedicated to construct $\mathcal{A}_{\mathrm{PLTL}}$ based on developments from [LMS02] and on the abstraction introduced in Sect. 4.1. As usual, we define $cl(\phi)$, the closure of ϕ, as the smallest set of formulae such that

– $\{\phi, \mathsf{X}^{-1}\top, \top\} \subseteq cl(\phi)$ and $cl(\phi)$ is closed under subformulae;
– $cl(\phi)$ is closed under negation (we identify $\neg\neg\psi$ with ψ);
– $\psi \mathsf{U} \psi' \in cl(\phi)$ implies $\mathsf{X}(\psi \mathsf{U} \psi') \in cl(\phi)$; $\psi \mathsf{S} \psi' \in cl(\phi)$ implies $\mathsf{X}^{-1}(\psi \mathsf{S} \psi') \in cl(\phi)$.

The cardinality of $cl(\phi)$ is polynomial in $|\phi|$. We define an *atom* of ϕ to be a maximally consistent subset of $cl(\phi)$ defined as follows. X is an atom of ϕ iff

– $X \subseteq cl(\phi)$ and $\top \in X$;
– for every $\psi \in cl(\phi)$, $\psi \in X$ iff not $\neg\psi \in X$;
– for every $\psi \wedge \psi' \in cl(\phi)$, $\psi \wedge \psi' \in X$ iff $\psi \in X$ and $\psi' \in X$;
– for every $\psi \mathsf{U} \psi' \in cl(\phi)$, $\psi \mathsf{U} \psi' \in X$ iff either $\psi' \in X$ or $\{\psi, \mathsf{X}(\psi \mathsf{U} \psi')\} \subseteq X$;
– for every $\psi \mathsf{S} \psi' \in cl(\phi)$, $\psi \mathsf{S} \psi' \in X$ iff either $\psi' \in X$ or $\{\psi, \mathsf{X}^{-1}(\psi \mathsf{S} \psi')\} \subseteq X$;
– for every $\mathsf{X}^{-1}\psi \in cl(\phi)$, $\mathsf{X}^{-1}\psi \in X$ implies $\mathsf{X}^{-1}\top \in X$.

We can now define the generalized Büchi automaton $\mathcal{A}_{\mathrm{PLTL}} = (Q, Q_0, \longrightarrow, \mathcal{F})$ with $\mathcal{F} = \{F_1, \ldots, F_m\} \subseteq \mathcal{P}(Q)$. A run $\rho : \mathbb{N} \rightarrow Q$ is accepting according to $\mathcal{F}$ iff for each $i \in \{1, \ldots, m\}$, $\rho(j) \in F_i$ for infinitely many $j \in \mathbb{N}$. A generalized Büchi condition can be easily converted to a Büchi condition. The elements of $\mathcal{A}_{\mathrm{PLTL}}$ are defined as follows:

– $Q = \mathcal{P}(cl(\phi))$; $Q_0 = \{X \in Q : \{\phi, \neg\mathsf{X}^{-1}\top\} \subseteq X\}$.
– $X \xrightarrow{u} Y$ iff
  (ATOM) X and Y are atoms of ϕ.
  (IPC^{++}) for every atomic p in X, $p' \wedge \mathrm{IPC}^{++}(u)$ is IPC^{++}-satisfiable where p' is obtained from p by replacing the occurrences of $\mathsf{X}^\beta x_\alpha$ by $x_{f(\langle x_\alpha, \beta\rangle)}$.
  (NEXT) for each $\mathsf{X}\psi \in cl(\phi)$, $\mathsf{X}\psi \in X$ iff $\psi \in Y$.
  (PREVIOUS) for each $\mathsf{X}^{-1}\psi \in cl(\phi)$, $\mathsf{X}^{-1}\psi \in Y$ iff $\psi \in X$.
– Let $\{\psi_1 \mathsf{U} \varphi_1, \ldots, \psi_m \mathsf{U} \varphi_m\}$ be the set of until formulas in $cl(\phi)$. $\mathcal{F} = \{F_1, \ldots, F_m\}$ with for every $i \in \{1, \ldots, m\}$, $F_i = \{Z \in Q \mid \psi_i \mathsf{U} \varphi_i \notin Z$ or $\varphi_i \in Z\}$.

In $\mathcal{A}_{\mathrm{PLTL}}$, one can check whether $X \xrightarrow{u} Y$ holds true in polynomial space in $|\phi|$. The conditions (ATOM), (NEXT), and (PREVIOUS) can be checked in polynomial-time in $|\phi|$. However, the above condition (IPC^{++}) requires polynomial space by Corollary 1. The main difference with LTL with past remains in the condition at the atomic level, involving here an IPC^{++}-satisfiability check.

Lemma 5. *ϕ is satisfiable iff $\mathrm{L}(\mathcal{A}_\phi)$ is non-empty.*

This is a consequence of Lemma 4 and of the construction of Büchi automata from formulae in LTL with past [LMS02]. It is now standard to prove

Theorem 4. *Satisfiability for $\mathrm{PLTL}^{\mathrm{mod}}$ is in* PSPACE.

The PSPACE-hardness of $\mathrm{PLTL}^{\mathrm{mod}}$ is a consequence of the PSPACE-hardness of LTL [SC85]. This PSPACE upper bound is quite remarkable: PSPACE-completeness of satisfiability problems in [BC02,DD03] has been mainly established for extensions of LTL over concrete domains with satisfiability problem in P (only).

5 Adding Logical First-Order Quantifiers

In this section, we investigate the complexity of $\mathrm{PLTL}(\mathrm{IPC}^+)$ augmented with the existential quantifier $\exists$, extension denoted by $\mathrm{PLTL}^\exists(\mathrm{IPC}^+)$. In the general case, first-order LTL is known to be highly undecidable [Aba89]. Decidability of $\mathrm{PLTL}^\exists(\mathrm{IPC}^+)$ is mainly due to the fact that the constraint language IPC^+ allows us to use an abstraction based on a finite domain (but whose size depends on the input formula). A similar argument cannot be used for $\mathrm{PLTL}^{\mathrm{mod}}$ augmented with the quantifier $\exists$ (denoted by $\mathrm{PLTL}^\exists(\mathrm{IPC}^{++})$) and the decidability status of this extension is unknown.

In order to define $\mathrm{PLTL}^\exists(\mathrm{IPC}^+)$, the definition of $\models$ is extended as follows: $\sigma, i \models \exists\ y\ \phi \overset{\text{def}}{\Leftrightarrow}$ there exists $n \in \mathbb{Z}$ such that $\sigma', i \models \phi$, where (1) for all $j \in \mathbb{N}$ and $x \in \mathrm{VAR} \setminus \{y\}$, $\sigma'(j,x) = \sigma(j,x)$, and (2) for every $j \in \mathbb{N}$, $\sigma'(j,y) = n$. Variables used with quantifiers are said to be global, the other ones are said to be local (it is not difficult to guarantee that a variable cannot be both local and global in a given formula). Otherwise stated, $\mathrm{PLTL}^\exists(\mathrm{IPC}^+)$ is the extension of $\mathrm{PLTL}(\mathrm{IPC}^+)$ where the temporal operators can be in the scope of $\exists$.

We write $\mathrm{PLTL}^\downarrow(\mathrm{IPC}^{++})$ to denote the fragment of $\mathrm{PLTL}^\exists(\mathrm{IPC}^{++})$ where the quantifier $\exists$ is used only in formulae of the form $\exists x'\ (x' = \mathsf{X}^i x) \wedge \phi$, with $i \geq 0$. We write $\downarrow_{x'=\mathsf{X}^i x}\ \phi$ instead of $\exists x'\ (x' = \mathsf{X}^i x) \wedge \phi$. The freeze quantifier $\downarrow$ that allows to bind the values of variables to a fixed value is a powerful binder used for instance in real-time logics [AH94] in order to capture the current value of a clock. The decidability status of decidable LTL over concrete domains from [DD03] but augmented with the freeze operator is still open. We treat a particular case with integer periodicity constraints for which decidability follows from decidability of $\mathrm{PLTL}^{\mathrm{mod}}$.

Lemma 6. *Satisfiability for $\mathrm{PLTL}^\downarrow(\mathrm{IPC}^{++})$ restricted to future-time operators and simple periodicity constraints is* EXPSPACE*-hard.*

The proof is based on a reduction of the 2^n-corridor tiling problem into $\mathrm{PLTL}^\downarrow(\mathrm{IPC}^{++})$ satisfiability, see [Dem03, Theorem 17]. As a corollary, satisfiability for $\mathrm{PLTL}^\exists(\mathrm{IPC}^{++})$ is also EXPSPACE-hard. A preliminary version of this paper (including [Dem03]) abusively stated the EXPSPACE-hardness of the logic $\mathrm{PLTL}^\exists(\mathrm{IPC}^+)$, this problem being still open.

Lemma 7. *Satisfiability for* $\mathrm{PLTL}^{\exists}(\mathrm{IPC}^{+})$ *is in* EXPSPACE.

Proof. (idea) Let ϕ be a $\mathrm{PLTL}^{\exists}(\mathrm{IPC}^{+})$ formula with (1) free variables $x_1, \ldots, x_k$; (2) $c_1 < \ldots < c_n$ are the constants in ϕ occurring in constraints of the form $x \sim c$ with $\sim \in \{<, >, =\}$; (3) $k_1, \ldots, k_u$ occurring in the context of $\equiv$-atomic formulae and their lcm is denoted by K. Let D be $\{\langle i,j,l \rangle \in \{0, \ldots, n+1\}^2 \times \{0, \ldots, K-1\} : \mathrm{IPC}^{<}(\langle i,j,l \rangle, 1))$ is satisfiable$\}$. To each $\langle i,j,l \rangle \in D$, we associate a constant $d_{\langle i,j,l \rangle}$ such that $|d_{\langle i,j,l \rangle}|$ is polynomial in $|\phi|$ and $[x_\alpha \leftarrow d_{\langle i,j,l \rangle}] \models \mathrm{IPC}^{<}(\langle i,j,l \rangle, \alpha))$. We reduce $\mathrm{PLTL}^{\exists}(\mathrm{IPC}^{+})$ satisfiability to $\mathrm{PLTL}(\mathrm{IPC}^{+})$ satisfiability. The translation t is the following: $t(p) = p$ for p atomic, t is homomorphic for the Boolean and temporal operators, $t(\exists\ x_\alpha\ \psi) = \bigvee_{\langle i,j,l \rangle \in D} t(\psi[x_\alpha \Leftarrow d_{\langle i,j,l \rangle}])$ where $\psi[x_\alpha \Leftarrow d_{\langle i,j,l \rangle}]$ denotes the formula obtained from ψ by replacing occurrences of x_α by $d_{\langle i,j,l \rangle}$ with adequate simplications. For instance $(x \equiv_3 x_\alpha + [1,2])[x_\alpha \Leftarrow 5]$ is equal to $x \equiv_3 0 \vee x \equiv_3 1$. ϕ is $\mathrm{PLTL}^{\exists}(\mathrm{IPC}^{+})$ satisfiable iff $t(\phi)$ is $\mathrm{PLTL}(\mathrm{IPC}^{+})$ satisfiable and $|t(\phi)|$ is in $2^{\mathcal{O}(|\phi|^2)}$.

The above translation does not work if we allow atomic constraints of the form $x = y$ (belonging to IPC^{++}) as in $\Box \downarrow_{x'=x} \mathsf{X}\Box(\neg(x = x'))$ that characterizes models where all the values for x are different. Such a formula is particularly interesting since in cryptographic protocols, nonces, ideally variables that never take twice the same value, are often used to guarantee freshness properties.

6 Application to Extended Single-String Automata

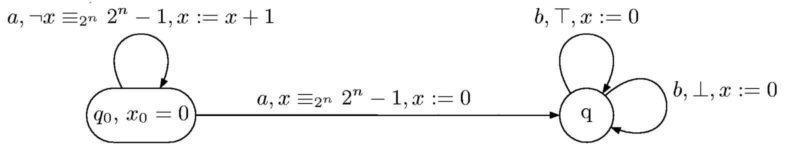
In this section, we characterize the complexity of the equivalence problem for extended single-string automata defined in [LM01, Sect. 5]. This problem is central to check whether two time granularities are equivalent (see also [Wij00]) when granularities are encoded with Büchi automata recognizing exactly one ω-word. Guards on transitions expressed by integer periodicity constraints and update maps on transitions provide conciseness of such contraint automata. We improve the known EXPSPACE upper bound from [LM01] into a PSPACE upper bound by reducing the equivalence problem to the model-checking problem for $\mathrm{PLTL}^{\mathrm{mod}}$-automata. Moreover, although a seemingly efficient algorithm is presented in [LM01], we show the PSPACE-hardness by reducing QBF.
Let IPC^{*} be the fragment of $\mathrm{IPC}^{\{\exists\}}$ containing Boolean combinations of atomic constraints of the form either $x \equiv_k c$ or $\exists z\ (x \equiv_k z\ \wedge\ y \equiv_{k'} z)$. Elements of IPC^{*} will be guards on transitions. An update map g for the variable x_i is of the form either $x_i := x_i + c$ or $x_i := c$ with $c \in \mathbb{Z}$. We write $\mathrm{UP}_{x_1,\ldots,x_n}$ to denote the set of update maps for the set $\{x_1, \ldots, x_n\}$ of variables.

An extended single-string automaton $\mathcal{A}$ (ESSA) over the finite set of variables $\{x_1, \ldots, x_n\}$ [LM01] is a structure $\langle Q, q_0, \overline{v_0}, \Sigma, \delta \rangle$ where

– Q is a finite set of states and $q_0 \in Q$ (initial state);
– $\overline{v_0} \in \mathbb{Z}^n$ (initial value of the variables); Σ is a finite alphabet;

– $\delta \subseteq Q \times \Sigma \times Q \times \mathrm{IPC}^* \times \mathcal{P}(\mathrm{UP}_{x_1,\ldots,x_n})$ and for every $q \in Q$, there are exactly two u such that $\langle q, u\rangle \in \delta$, say u_1 and u_2, and in that case u_1 is of the form $\langle a_1, q_1, p, X_1\rangle$, u_2 is of the form $\langle a_2, q_2, \neg p, X_2\rangle$ where p is a constraint in IPC^* built over variables in $\{x_1, \ldots, x_n\}$ and in both X_1 and X_2 exactly one update map for x_i is present.

The elements of δ are also denoted by $q \xrightarrow{a,p,X} q'$ (p is the guard and X is the global update map). A configuration is a member $\langle q, \overline{v}\rangle \in Q \times \mathbb{Z}^n$. We define the one-step relation $\xrightarrow{a}$ for $a \in \Sigma$ as follows: $\langle q, \overline{v}\rangle \xrightarrow{a} \langle q', \overline{v'}\rangle$ iff there is $\langle q, a, q', X, p\rangle \in \delta$ such that $[x_1 \leftarrow v_1, \ldots, x_n \leftarrow v_n] \models p$ (in IPC^{++}) and for every $g \in X$, (1) if g is $x_i := x_i + c$ then $v'_i = v_i + c$, and (2) if g is $x_i := c$ then $v'_i = c$. There is exactly one sequence $w = a_1 a_2 \ldots \in \Sigma^\omega$ such that $\langle q_0, \overline{v_0}\rangle \xrightarrow{a_1} \langle q_1, \overline{v_1}\rangle \xrightarrow{a_2} \ldots$. The unique ω-sequence generated from $\mathcal{A}$ is denoted by $w_{\mathcal{A}}$. The equivalence problem for ESSA consists in checking whether $w_{\mathcal{A}} = w_{\mathcal{A}'}$, given two ESSA $\mathcal{A}$ and $\mathcal{A}'$. The condition on δ is introduced in [LM01] to handle priorities between transitions. For instance, the ω-word associated with the ESSA below is $a^{2^n} \cdot b^\omega$:


$a, \neg x \equiv_{2^n} 2^n - 1, x := x + 1$
$b, \top, x := 0$
$b, \bot, x := 0$
$a, x \equiv_{2^n} 2^n - 1, x := 0$
$q_0,\ x_0 = 0$
q

Lemma 8. *The equivalence problem for ESSA can be solved in* PSPACE.

Proof. Given two ESSA $\mathcal{A}$ and $\mathcal{A}'$, we build an LTL($\mathrm{IPC}^{\{\exists\}}$)-automaton $\mathcal{B}$ in polynomial time such that $l(\mathcal{B})$ is non-empty iff $w_{\mathcal{A}} = w_{\mathcal{A}'}$. The LTL($\mathrm{IPC}^{\{\exists\}}$)-automaton $\mathcal{B}$ is indeed a kind of product of $\mathcal{A}$ and $\mathcal{A}'$, see details in [Dem03]. The PSPACE bound is then a corollary of Theorem 1 and Theorem 4.

One can also show that the equivalence problem for ESSA is PSPACE-hard even if the constraints occurring in transitions are either in $\{\top, \bot\}$ or literals built over atomic constraints of the form $x \equiv_k c$, the update maps are of the form either $x := x$ (identity) or $x := c$, and the alphabet Σ is binary.

Lemma 9. *The equivalence problem for ESSA is* PSPACE*-hard.*

The proof of Lemma 9 (see [Dem03]) entails that the problem remains PSPACE-hard when the only integer k in $\equiv_k$-guards occurring in $\mathcal{A}, \mathcal{A}'$ is 2. Similarly, the problem remains PSPACE-hard when only two distinct variables are used.

Theorem 5. *The equivalence problem for ESSA is* PSPACE*-complete.*

7 Concluding Remarks

We have introduced a first-order theory of periodicity constraints IPC^{++} whose satisfiability is PSPACE-complete and a version of LTL with past whose atomic formulae are constraints from IPC^{++} (with comparison of variables at different states). $\mathrm{PLTL}^{\mathrm{mod}}$ is a very concise logical formalism to deal with periodicity constraints. Nevertheless, we have shown that $\mathrm{PLTL}^{\mathrm{mod}}$ model-checking and satisfiability are PSPACE-complete and that PLTL$^{\exists}$(IPC^{+}), the extension PLTL(IPC^{+}) with the quantifier $\exists$ is in EXPSPACE. As an application, we have also proved that the equivalence problem for ESSA introduced in [LM01, Sect. 5] is PSPACE-complete, even if restricted to two variables.

In the table below, we recall the main results about LTL and PLTL over periodicity constraints and we indicate open problems related to them.

	LTL/PLTL	LTL/PLTL + $\downarrow$	LTL/PLTL + $\exists$
$\{x < y, x = y\}$	PSPACE-complete [DD02]	?/undecidable	undecidable
$\{x < y, x = y, x < c, x = c\}$	in EXPSPACE [DD03]	?/undecidable	undecidable
IPC + $\{x < y, x = y\}$	?	?/undecidable	undecidable
IPC^{+}	PSPACE-complete Theorem 4	in EXPSPACE Lemma 7	in EXPSPACE Lemma 7
IPC^{++}	PSPACE-complete Theorem 4	?	?

The question mark '?' refers to the decidability status. All undecidability results are (more or less straightforward) consequences of the fact that LTL over the contraints language allowing atomic constraint of the form $x = y$ and $x = y + 1$ is undecidable by simulation of two-counter machines. Among the open problems, we would like to emphasize that we ignore how to deal with $\downarrow$ in the presence of atomic constraints of the form $x = y$. The decidability status of LTL($\{x = y\}$) + $\downarrow$ restricted to formulae with a *unique local* variable is open. Finally, the difficulty with the decidability status of PLTL(IPC + $\{x < y, x = y\}$) is that LTL($\{x < y, x = y\}$) already characterizes non ω-regular sequences of constraints, see details in [DD03].

References

[Aba89] M. Abadi. The power of temporal proofs. *TCS*, 65:35–83, 1989.

[AD94] R. Alur and D. Dill. A theory of timed automata. *TCS*, 126:183–235, 1994.

[AH94] R. Alur and Th. Henzinger. A really temporal logic. *JACM*, 41(1):181–204, 1994.

[BBFS96] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. Supporting periodic authorizations and temporal reasoning in database access control. In *22nd VLDB, Bombay, India*, pages 472–483, 1996.

[BBFS98] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. An access control model supporting periodicity constraints and temporal reasoning. *ACM Transactions on Databases Systems*, 23(3):231–285, 1998.

[BC02] Ph. Balbiani and J.F. Condotta. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *FroCoS'02*, vol. 2309 of *LNAI*, pages 162–173. Springer, 2002.

[BEH95] A. Bouajjani, R. Echahed, and P. Habermehl. On the verification problem of nonregular properties for nonregular processes. In *LICS'95*, pages 123–133, 1995.

[BH99] A. Bouajjani and P. Habermehl. Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations. *TCS*, 221(1–2):211–250, 1999.

[CC00] H. Comon and V. Cortier. Flatness is not a weakness. In *CSL-14*, vol. 1862 of *LNCS*, pages 262–276. Springer-Verlag, 2000.

[Čer94] K. Čerāns. Deciding properties of integral relational automata. In *ICALP-21*, vol. 820 of *LNCS*, pages 35–46. Springer, 1994.

[CFP02] C. Combi, M. Franceschet, and A. Peron. A logical approach to represent and reason about calendars. In *Int. Symposium on Temporal Representation and Reasoning*, pages 134–140. IEEE Computer Society Press, 2002.

[CM77] A. Chandra and P. Merlin. Optimal implementation of conjunctive queries in relational databases. In *STOC-9*, pages 77–90, 1977.

[DD02] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. In *FST&TCS'02*, vol. 2556 of *LNCS*, pages 121–132. Springer, 2002.

[DD03] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. Technical Report LSV-03-11, LSV, August 2003. 40 pages. Submitted.

[Dem03] S. Demri. LTL over Integer Periodicity Constraints. Technical Report LSV-03-13, LSV, October 2003. 34 pages.

[FL02] A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS'02*, vol. 2256 of *LNCS*, pages 145–156. Springer, 2002.

[GHR94] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic - Mathematical Foundations and Computational Aspects, Volume 1.* OUP, 1994.

[GK03] P. Gastin and D. Kuske. Satisfiability and model checking for MSO-definable temporal logics are in PSPACE. In *CONCUR'03*, vol. 2761 of *LNCS*, pages 222–236. Springer, 2003.

[KK67] G. Kreisel and J.L. Krivine. *Elements of Mathematical Logic.* Studies in Logic and the Foundations of Mathematics. North-Holland, 1967.

[LM01] U. Dal Lago and A. Montanari. Calendars, time granularities, and automata. In *Int. Symposium on Spatial and Temporal Databases*, vol. 2121 of *LNCS*, pages 279–298. Springer, 2001.

[LMS02] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Temporal logic with forgettable past. In *LICS'02*, pages 383–392. IEEE Computer Society, 2002.

[NS92] M. Niezette and J. Stevenne. An efficient symbolic representation of periodic time. In *Proc. of the International Conference on Information and Knowledge Management*, vol. 752 of *LNCS*, pages 161–168. Springer, 1992.

[Ohl94] H.J. Ohlbach. Calendar logic. In *[GHR94]*, chapter 19. 1994.

[Pre29] M. Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes Rendus du premier congrès de mathématiciens des Pays Slaves, Warszawa*, pages 92–101, 1929.

[SC85] A. Sistla and E. Clarke. The complexity of propositional linear temporal logic. *JACM*, 32(3):733–749, 1985.

[TC98] D. Toman and J. Chomicki. DATALOG with integer periodicity constraints. *Journal of Logic Programming*, 35(3):263–290, 1998.

[VW94] M. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115:1–37, 1994.

[WB00] P. Wolper and B. Boigelot. On the construction of automata from linear arithmetic constraints. In *TACAS-6*, vol. 1785 of *LNCS*, pages 1–19. Springer, 2000.

[Wij00] J. Wijsen. A string based-model for infinite granularities. In *AAAI Workshop on Spatial and Temporal Granularity*, pages 9–16. AAAI Press, 2000.