The Effects of Bounding Syntactic Resources on Presburger LTL *

Stéphane Demri
LSV, ENS Cachan
CNRS, INRIA Saclay
demri@lsv.ens-cachan.fr

&

Régis Gascon
INRIA, I3S, CNRS
Univ. Nice Sophia Antipolis
rgascon@sophia.inria.fr

Abstract

LTL over Presburger constraints is the extension of LTL where the atomic formulae are quantifier-free Presburger formulae having as free variables the counters at different states of the model. This logic is known to admit undecidable satisfiability and model-checking problems. We study decidability and complexity issues for fragments of LTL with Presburger constraints obtained by restricting the syntactic resources of the formulae (the number of variables, the maximal distance between two states for which counters can be compared and, to a smaller extent, the set of Presburger constraints) while preserving the strength of the logical operators. We provide a complete picture refining known results from the literature. We show that model-checking and satisfiability problems for the fragments of LTL with difference constraints restricted to two variables and distance one and to one variable and distance two are highly undecidable, enlarging significantly the class of known undecidable fragments. On the positive side, we prove that the fragment restricted to one variable and to distance one augmented with propositional variables is PSPACE-complete. Since the atomic formulae can state quantitative properties on the counters, this extends some results about model-checking pushdown systems and one-counter automata. In order to establish the PSPACE upper bound, we show that the nonemptiness problem for Büchi one-counter automata taking values in $\mathbb{Z}$ and allowing zero tests and sign tests, is only NLOGSPACE-complete. Finally, we establish that model-checking one-counter automata with complete quantifier-free Presburger LTL restricted to one variable is also PSPACE-complete whereas the satisfiability problem is undecidable.

1 Introduction

Verification of infinite-state systems. Model-checking is a well-known approach to verifying behavioral properties of computing systems that has been

*Work supported by the Agence Nationale de la Recherche, grant ANR-06-SETIN-001

very successful in the verification of finite-state systems, see e.g. [CGP00]. The situation is different for infinite-state systems. Despite that numerous symbolic representations have been proposed to deal with such systems (see e.g. timed automata or counter automata), their formal verification remains a difficult problem. Many general formalisms referring to infinite-state systems have an undecidable model-checking problem. Sometimes, decidability can be regained by considering subproblems of the general problem, for example by restricting the number of counters/clocks/variables or by constraining their occurrences.

Counter systems. The class of counter automata is an example of such a formalism. Counter automata have many applications in formal verification. Their ubiquity stems from their use as operational models of numerous infinite-state systems, including for instance broadcast protocols [FL02] and programs with pointer variables (see [BFLS06, BBH$^+$06]). Even the case of a single counter has found some applications in the verification of cryptographic protocols [LLT05] and the validation of XML streams [CR04] (see also the context-free languages defined by one-counter automata in [BB90]). However, numerous model-checking problems for counter automata, such as reachability, are known to be undecidable. This does not end the story since many subclasses of counter automata admit a decidable reachability problem such as reversal-bounded multicounter machines [Iba78, ISD$^+$00] and flat counter systems [Boi98, CJ98, FL02]. These two classes of systems admit reachability sets effectively definable in Presburger arithmetic (assuming some additional conditions, unspecified herein).

Motivations. Extending the linear-time temporal logic LTL with Presburger constraints allows to specify quantitative properties of counter systems that go beyond simple reachability. Because this language is expressive enough to simulate Minsky machines [Min67], this extension is undecidable. Attempts to identify decidable fragments have already been made in [BEH95, CC00] by restricting the use of the temporal operators (see also [BH96]). In this paper, we are interested in the decidability and complexity of fragments of Presburger LTL obtained by restricting the following syntactic resources of the formulae: the set of Presburger constraints, the number of counters and the maximal distance between two states for which these counters can be compared. We do not restrict the use of the logical operators. Our investigation is based on the standard assumption that restricting the number of variables is a means of identifying decidable fragments of undecidable logics or to design counter/clock automata with decidable reachability problems (see examples for modal logics in [Hal95] or for alternating timed automata in [OW05, LW05]). Furthermore this kind of analysis helps to understand the complexity gaps between decidable problems, see complexity results for model-checking subproblems in [DS02, LMS04]. Our goal is therefore to identify decidable and undecidable fragments of Presburger LTL (both for model-checking and satisfiability problems) refining existing results from [CC00, DG08].

Our contribution. We define CLTL(DL) as a fragment of Presburger LTL where atomic formulae are difference constraints from quantifier-free Presburger arithmetic. Unlike the version of Presburger LTL defined in [BEH95], the models are ω-sequences of counter valuations. The underlying fragment of Presburger arithmetic in CLTL(DL) is identical to the one in the logic $\mathcal{L}_p$ from [CC00]. However, it is possible in CLTL(DL) to state constraints between counters at two non-consecutive states. For instance, "$\mathsf{XX}x = y$" means that the value of y at the current state is equal to the value of x two states further. We call X-length the maximal number of X operators prefixing a counter. The two main undecidability results shown in this paper are the following:

- satisfiability and model-checking for CLTL(DL) restricted to formulae of X-length two with at most one counter are Σ_1^1-complete (an exposition of the analytical hierarchy can be found in [Rog67]),
- satisfiability and model-checking for CLTL(DL) restricted to formulae of X-length one with at most two counters are Σ_1^1-complete.

The logic CLTL(DL) was already known to be undecidable since it has been shown in [CC00] that CLTL(DL) restricted to formulae of X-length one with at most three counters is undecidable. Similarly, undecidability of CLTL(DL) restricted to formulae with at most two counters has been shown in [DD07]. Hence, we push forward the decidability limits established by these results.

Then, we prove that model-checking and satisfiability problems for CLTL(DL) are PSPACE-complete when restricted to X-length one and to one counter. This result completes our classification of CLTL(DL) fragments restricting the number of counters and the X-length of the formulae. Actually, we establish this PSPACE-completeness result for an extension of CLTL(DL) including propositional variables and periodicity constraints of the form $x \equiv_k y + c$ (but still restricted to X-length one and to one counter). The addition of propositional variables allows to reduce the model-checking problem to the satisfiability problem, more precisely to encode control states in a formula, whereas the addition of periodicity constraints is performed to get as rich an underlying quantifier-free fragment of Presburger arithmetic as possible. Several problems involving one-counter automata can be encoded in these decidable fragments. These problems are related to different applications such as verification of cryptographic protocols [LLT05], validation of XML streams (string representations of XML documents) [CR04], or resolution of the identification problem [WTT04]. These results complete our precise taxonomy of CLTL(DL) fragments with respect to decidability.

Our decidability results about CLTL(DL) fragments restricting the syntactic ressources of formulas are optimal with respect to the constraints used. Satisfiability for LTL with quantifier-free Presburger constraints (properly including difference constraints) is undecidable even if restricted to X-length one and to one counter. However, we show that one can regain decidability by restricting the class of models. We prove that the model checking problem for the one variable fragment of this logic is PSPACE-complete when the class of models is

one-counter automata (with updates in $\mathbb{Z}$).

To prove all our decidability and complexity results, we follow a standard automata-based approach [VW94, KVW00] but we introduce an original symbolic representation of models that can be recognized by a fine-tuned class of one-counter automata (instead of standard Büchi automata). A nice property of this method is that it can be generalized to various LTL extensions that define ω-regular classes of models. The class of automata that we use have integer counters which are updated by adding -1, 0 or 1, zero-tests and sign-tests, and accept regular ω-languages. We show that the nonemptiness problem for such automata is NLOGSPACE-complete. This extends what is known about Büchi automata and variants of one-counter automata [VW94, LLT05].

Related work. Decidability and complexity issues for LTL variants with Presburger constraints can be found in [BEH95, BH96, CC00]. This type of logical formalisms has been also considered in Artificial Intelligence (AI) and Knowledge Represention (KR), for instance in description logics with concrete domains in [Lut04] and in logics of space and time in [BC02]. Unlike these works, we study systematically the effects of restricting the number of variables, the X-length and, in a less systematic way, the set of constraints. In the fragments we consider, we preserve the logical operators but we put restrictions on the form of atomic formulae. This is in sharp contrast, for instance, with the flat fragment shown decidable in [CC00]. The complexity bound we obtain contrasts with the bound established in [CC00] which is obtained by reduction to satisfiability of Presburger arithmetic and is therefore rather high.

One-counter automata are interesting operational models. Unlike multi-counter automata, they have nice computational properties, see for instance complexity results about behavioural equivalences in [Kuč00, JKMS04] (see also [BHM03]). Moreover, one-counter automata are equivalent to pushdown systems with a singleton stack alphabet. Therefore the results on these systems can help to refine some results about pushdown systems. For instance, the model-checking problem for one-counter automata with the modal μ-calculus has been shown to be in PSPACE [Ser06] whereas the model-checking problems for pushdown automata over the modal μ-calculus and the linear μ-calculus are in EXPTIME (see [Wal01] and [BEM97], respectively). It is worth mentioning that in these logics the atomic formulae can only speak about the control states and not about the content of the counter. This is a major difference with our formalism. We refine these results by showing that model-checking one-counter automata over linear μ-calculus is PSPACE-complete.

In [FWW97], a version of CTL* with atomic formulae containing control states and regular conditions on the current stack is shown to admit a decidable model-checking problem over pushdown systems. This result is improved in [EKS03] where EXPTIME-completeness is established. We establish that the restriction to LTL and one-counter automata is PSPACE-complete, refining the above result. However, our model is more expressive than pushdown systems

with a single letter pushdown alphabet. For instance, it allows incrementation modulo some integer or transition of the form $\mathsf{X}x \leq x + 1$ that can be seen as lossy transitions.

Structure of the paper. In Section 2, we introduce the fragments of Presburger arithmetic we consider and give a general definition for the extension of LTL with Presburger constraints between terms representing values of the variables at different states of the execution. We also define the relevant satisfiability and model-checking problems we consider. For the extension of LTL with the difference constraint language DL, we establish undecidability results for the fragments with X-length one and two variables, and X-length two and one variable. In Section 3 we consider an extension of CLTL(DL) with periodicity constraints and propositional variables and we show how to translate a one-variable formula of this logic with X-length equal to one into a one-counter automaton. This translation is the basis for the PSPACE upper bound results established in this paper and requires a finite abstraction of the models defined in the same section. Based on the same approach, we show in Section 4 that model-checking one-$\mathbb{Z}$-counter automata for LTL with full quantifier-free Presburger formulae restricted to one variable is PSPACE-complete. The two PSPACE upper bounds are obtained thanks to the NLOGSPACE-membership of the nonemptiness problem for a particular subclass of one-counter automata shown in Section 5. Finally, Section 6 contains concluding remarks and open problems.

This paper is a completed version of [DG07].

2 Temporal logics, automata and Presburger constraints

2.1 Linear-time temporal logics

Constraint languages. Let $\mathrm{VAR} = \{x_0, x_1, \ldots\}$ be a countably infinite set of variables. We consider several fragments of Presburger Arithmetic (PA). The Difference Logic DL is defined by constraints of the form

$$\alpha ::= x \sim y + d \mid x \sim d \mid \alpha \wedge \alpha \mid \neg\alpha$$

where $x, y \in \mathrm{VAR}$, $d \in \mathbb{Z}$ and $\sim \in \{<, >, \leq, \geq, =\}$. We denote by DL^+ the extension of DL with periodicity constraints of the form $x \equiv_k c$ or $x \equiv_k y + c$ ($c \in \mathbb{N}$ and $k \in \mathbb{N} \setminus \{0, 1\}$). Finally, QFP is the quantifier-free fragment of PA, which can be defined by the following grammar:

$$\alpha ::= \sum_{i \in I} a_i x_i \sim d \mid \sum_{i \in I} a_i x_i \equiv_k c \mid \alpha \wedge \alpha \mid \neg\alpha$$

where $a_i \in \mathbb{Z} \setminus \{0\}$ and I is a finite set of indices. Obviously, we have the following inclusions between these languages: $\mathrm{DL} \subseteq \mathrm{DL}^+ \subseteq \mathrm{QFP}$. Given a

valuation $v : \mathrm{VAR} \to \mathbb{Z}$, the satisfaction relation $v \models \alpha$ is defined in the obvious way. For instance, $v \models x \equiv_k y$ iff there exists $z \in \mathbb{Z}$ such that $v(x) = v(y) + kz$. We assume that all integers are encoded in binary.

Linear-time temporal logics. Given a constraint language L (typically DL, DL^+ or QFP), we define the logic CLTL(L) as the extension of LTL [GPSS80] where the propositional variables are refined to atomic constraints from L over expressions representing different states of the variables. The formulae of CLTL(L) are defined by the grammar:

$$\phi ::= \alpha\theta \mid \phi \wedge \phi \mid \neg\phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi$$

where α is a constraint in L and θ is a substitution of all the free variables which possibly occur in α by terms of the form $\mathsf{X}^l x_j$. The symbols X and U are respectively the classical operators "next" and "until" of LTL. We also use the standard notations $\mathsf{F}\phi$ and $\mathsf{G}\phi$ as the abbreviations for $\top \mathsf{U} \phi$ and $\neg \mathsf{F} \neg \phi$.

We call *term* a variable x_i prefixed by a certain number l of X symbols, shortly denoted by $\mathsf{X}^l x_i$. The encoding of $\mathsf{X}^l x_i$ requires $\mathcal{O}(l + log(i))$ bits. Indeed, the symbol "X" in $\mathsf{X}^l x_i$ is not a logical operator but a simple syntactic means to refer to the value of x_i at the lth next state.

Given a CLTL(L) formula ϕ, we define its X-*length* $|\phi|_{\mathsf{X}}$ as the maximal number l such that a term of the form $\mathsf{X}^l x$ occurs in ϕ. Intuitively, the X-length defines the size of a frame of consecutive states that can be compared in the formula. We denote by $\mathrm{CLTL}_k^l(L)$ the restriction of CLTL(L) to formulae with at most k variables and X-length bounded by l.

The models of CLTL(L) are ω-sequences of valuations $\sigma : \mathbb{N} \to (\mathrm{VAR} \to \mathbb{Z})$ and the satisfaction relation is defined as in LTL except for atomic formulae:

- $\sigma, i \models \alpha[x_1 \leftarrow \mathsf{X}^{l_1} x_{j_1}, ..., x_n \leftarrow \mathsf{X}^{l_n} x_{j_n}]$ iff $(\sigma(i + l_1)(x_{j_1}), ..., \sigma(i + l_n)(x_{j_n})) \models \alpha$ in PA,
- $\sigma, i \models \phi \wedge \phi'$ iff $\sigma, i \models \phi$ and $\sigma, i \models \phi'$,
- $\sigma, i \models \neg\phi$ iff $\sigma, i \not\models \phi$,
- $\sigma, i \models \mathsf{X}\phi$ iff $\sigma, i + 1 \models \phi$,
- $\sigma, i \models \phi \mathsf{U} \phi'$ iff there exists $j \geq i$ such that $\sigma, j \models \phi'$ and for every $i \leq k < j$, we have $\sigma, k \models \phi$.

We abbreviate $\sigma, 0 \models \phi$ to $\sigma \models \phi$. The symbol $\models$, used at the level of the constraint language, is overloaded but this will not lead to any confusion. As usual, a formula $\phi \in \mathrm{CLTL}(L)$ is satisfiable iff there exists a model σ such that $\sigma \models \phi$. Note that the satisfiability problem for CLTL(QFP) can be placed easily in the class Σ_1^1 from the analytical hierarchy (see e.g. [Rog67]). Indeed, any model of CLTL(QFP) restricted to n variables can be encoded by functions $f_1, \ldots, f_n : \mathbb{N} \to \mathbb{N}$. A first-order predicate on $f_1, \ldots, f_n$ which expresses that $\sigma, 0 \models \phi$ (ϕ contains at most n variables) is routine to construct by structural recursion on ϕ. We conclude that satisfiability for ϕ can be expressed by a Σ_1^1-sentence.

Constraint automata. The model-checking problem we consider for CLTL(L) involves a class of automata whose transitions are labeled with constraints of L. A *one-step constraint* is an atomic formula of the form $\alpha[x_1 \leftarrow \mathsf{X}^{l_1} x_{j_1}, \ldots, x_n \leftarrow \mathsf{X}^{l_n} x_{j_n}]$ such that $l_1, \ldots, l_n \leq 1$. One-step constraints express constraints on the values of the variables at the current state and the next state only. A *k-variable L-automaton* $\mathcal{A}$ is a structure $\langle Q, \delta, I, F \rangle$ such that Q is a finite set of states, $I \subseteq Q$ is a subset of initial states, $F \subseteq Q$ is a subset of final states and $\delta \subseteq Q \times A \times Q$ where A is a finite subset of the set $1SC_k(L)$ of Boolean combinations of one-step constraints from L built over the variables $\{x_1, \ldots, x_k\}$. We use the notation $q \xrightarrow{\alpha} q'$ as an abbreviation for $\langle q, \alpha, q' \rangle \in \delta$. Note that L-automata have no input alphabet. We do not use them as language acceptors but we are rather interested in the behaviors of such systems.

A *configuration* of $\mathcal{A}$ is a tuple $\langle q, \overline{c} \rangle \in Q \times \mathbb{Z}^k$. We denote by $\overline{c}(i)$ the i^{th} value of $\overline{c}$. The one-step transition relation $\rightarrow$ is defined as follows: $\langle q, \overline{c} \rangle \rightarrow \langle q', \overline{c'} \rangle \stackrel{\text{def}}{\Leftrightarrow}$ there exists $\langle q, \alpha, q' \rangle \in \delta$ such that if for every $i \in \{1, \ldots k\}$ the term x_i takes the value $\overline{c}(i)$ and $\mathsf{X}x_i$ takes the value $\overline{c'}(i)$, then α holds true. We write $\langle q, \overline{c} \rangle \xrightarrow{\alpha} \langle q', \overline{c'} \rangle$ whenever we need to exhibit the constraint α on the transition. The symbol $\rightarrow$ is overloaded but this does not lead to any confusion. A finite *path* w is a sequence of the form $\{0, \ldots, n\} \rightarrow (Q \times \mathbb{Z}^k)$ such that for every $i \in \{0, \ldots, n-1\}$ we have $w(i) \rightarrow w(i+1)$. Infinite paths are defined in a similar way, by considering infinite sequences of the form $\mathbb{N} \rightarrow (Q \times \mathbb{Z}^k)$. We denote by $\rightarrow^*$ the reflexive and transitive closure of $\rightarrow$, i.e. $\langle q, \overline{c} \rangle \rightarrow^* \langle q', \overline{c'} \rangle$ iff there is a finite path from $\langle q, \overline{c} \rangle$ to $\langle q', \overline{c'} \rangle$. An accepting run for $\mathcal{A}$ is an infinite path w such that $w(0) \in I \times \mathbb{Z}^k$ and the set $\{i \in \mathbb{N} : w(i) \in F \times \mathbb{Z}^k\}$ is infinite (standard Büchi acceptance condition). We write $\mathrm{L}^{\mathrm{symb}}(\mathcal{A})$ to denote the set of ω-words accepted by $\mathcal{A}$ viewed as an automaton over the alphabet A. A CLTL(L) model σ *realizes* an ω-word $\alpha_0 \alpha_1 \cdots$ over A iff for every $i \geq 0$, we have $\sigma, i \models \alpha_i$.

Given a CLTL(L) formula ϕ and an L-automaton $\mathcal{A}$, the *model-checking problem* for CLTL(L) is to check whether there is a CLTL(L) model σ that realizes some word of $\mathrm{L}^{\mathrm{symb}}(\mathcal{A})$ and such that $\sigma, 0 \models \phi$. We denote this by $\mathcal{A} \models \phi$. In other words, $\mathcal{A} \models \phi$ iff there is an accepting run of $\mathcal{A}$ such that the corresponding valuation sequence, obtained by removing the states, satisfies ϕ. States are removed since the atomic formulae of the logics only speak about counters. This existential version of the problem simplifies forthcoming developments since we also deal with satisfiability. Results about the universal version can be withdrawn from those presented herein. For the restriction to $\mathrm{CLTL}^l_k(L)$, the model-checking problem takes as inputs a $\mathrm{CLTL}^l_k(L)$ formula and a k-variable L-automaton.

Counter automata. In the rest of the paper, we mainly consider DL^+-automata or subclasses that can simulate non-deterministic Minsky machines. We introduce below subclasses of DL-automata on which we will restrict in some places the model-checking problem.

A *k-$\mathbb{Z}$-counter automaton* is a restricted DL-automaton such that for every

transition $q \xrightarrow{\alpha} q'$ in the automaton, the constraint α is a conjunction of the form below

$$\bigwedge_{i\in\{1\ldots k\}} \alpha_{test^i} \wedge \bigwedge_{i\in\{1\ldots k\}} \alpha_{update^i}$$

with

- $\alpha_{test^i} \in \{\top\} \cup \{x_i \sim 0 \mid \sim \in \{<, >, =, \neq\}\}$,
- $\alpha_{update^i} \in \{\mathsf{X}x_i = x_i + u \mid u \in \mathbb{Z}\}$

for every $i \in \{1 \ldots k\}$. Moreover, we require that the initial values of the counters are equal to zero (with a zero test on every transition from an initial state). Obviously, k-$\mathbb{Z}$-counter automata form a proper subclass of k-variables DL-automata that admit also constraints of the form $\mathsf{X}x_i > x_j$ or $\mathsf{X}x_i < x_j + d$.

For ease of presentation, the elements of $\{\top\} \cup \{x_i \sim 0 \mid \sim \in \{<, >, =, \neq\}\}$ are encoded by $\{\top, <, >, =, \neq\}$, the elements of $\{\mathsf{X}x_i = x_i + u \mid u \in \mathbb{Z}\}$ by $\mathbb{Z}$ and we order the constraints according to an arbitrary ordering of the variables (from x_1 to x_k). For instance, the transition

$$q \xrightarrow{\top \wedge x_2=0 \wedge \mathsf{X}x_1=x_1 \wedge \mathsf{X}x_2=x_2-1} q'$$

is encoded by

$$q \xrightarrow{\top,=,0,-1} q'.$$

A k-$\mathbb{N}$-counter automaton is defined similarly except that we only consider non-negative values for the counters. So, we require that $\bigwedge_{i\in\{1\ldots k\}} x_i \geq 0$ is also part of the constraints on every transition. When explicitly dealing with $\mathbb{N}$-counter automata, we omit this additional constraint on the transitions.

We only refer in the remaining to automata with at most two counters. In Section 3, we define an automata-based approach which differs from [VW94] by the use of one-$\mathbb{Z}$-counter automata where the updates are restricted to $\{-1, 0, 1\}$, instead of classical Büchi automata. Such automata are called *simple*. Hence, counter automata are used as operational models (inputs of the model-checking problem) and also as language acceptors for adapting the automata-based approach from [VW94]. Proving the existence of accepting runs for simple one-$\mathbb{Z}$-counter automata is not immediate since we are dealing with Büchi acceptance condition, the counter is interpreted in $\mathbb{Z}$ and zero/sign tests are allowed (see Section 5). As far as two-$\mathbb{N}$-counter automata are concerned, the existence of an accepting run is Σ_1^1-hard since the recurrence problem for nondeterministic two counter Minsky machines, known to be Σ_1^1-complete [AH94, Lemma 8], can easily be reduced to this problem.

2.2 Improving undecidability boundaries

Satisfiability for CLTL(DL) is undecidable since we can easily encode the executions of a multi-counter Minsky machine with a CLTL(DL) formula. The proof of [CC00, Theorem 3] provides that CLTL_3^1(DL) satisfiability is already

Σ_1^1-hard. We refine this result by showing that one variable and X-length two or two variables and X-length one is enough for high undecidability. The idea of the two proofs below is standard and consists in encoding directly in the formula the instructions of the Minsky machines. Depending on the available syntactic resources, configurations of the machine are represented differently. Even though the proofs are not very difficult, we believe that it simply illustrates that having strictly less than three variables does not necessarily decrease the complexity of the logic.

Theorem 1 *Satisfiability for* $\mathrm{CLTL}_1^2(\mathrm{DL})$ *is* Σ_1^1*-complete.*

Proof. We show that the existence of an accepting run for a two-$\mathbb{N}$-counter automaton can be reduced to a satisfiability question in $\mathrm{CLTL}_1^2(\mathrm{DL})$. This is sufficient to get Σ_1^1-hardness because such counter automata can easily simulate nondeterministic two counter Minsky machines whose recurrence problem is Σ_1^1-hard.

First we show that for every two-$\mathbb{N}$-counter automaton $\mathcal{A}$, there is a two-$\mathbb{N}$-counter automaton $\mathcal{A}'$ computable in logarithmic space in $|\mathcal{A}|$ such that $\mathcal{A}$ has an accepting run iff $\mathcal{A}'$ has an accepting run and none of the transitions of $\mathcal{A}'$ is of the form $q \xrightarrow{test^1, test^2, 0, 0} q'$. In other words, at least one counter changes its value at every step of $\mathcal{A}'$. Let $\mathcal{A} = \langle Q, \delta, I, F\rangle$ be a two-$\mathbb{N}$-counter automaton. The two-$\mathbb{N}$-counter automaton $\mathcal{A}' = \langle Q', \delta', I', F'\rangle$ is defined as follows:

- $Q' \stackrel{\text{def}}{=} Q \cup \{t \in \delta : t = q \xrightarrow{test^1, test^2, 0, 0} q'\}$,
- $I' \stackrel{\text{def}}{=} I$ and $F' = F$,
- δ' is defined from δ by replacing each transition $t = q \xrightarrow{test^1, test^2, 0, 0} q' \in Q' \setminus Q$ by $q \xrightarrow{test^1, test^2, +1, 0} t$ and $t \xrightarrow{\top, \top, -1, 0} q'$.

Now let $\mathcal{A} = \langle Q, \delta, I, F\rangle$ be a two-$\mathbb{N}$-counter automaton such that every transition of δ changes the value of at least one counter. We pose $Q = \{q_1, \ldots, q_n\}$, $I = \{q_{a_1}, \ldots, q_{a_m}\}$, $F = \{q_{b_1}, \ldots, q_{b_{m'}}\}$ and the variables used in the transition relation of $\mathcal{A}$ are denoted by x_1 and x_2. A configuration of the form $\langle q_i, c_1, c_2\rangle$ is encoded by a sequence of $2i$ states repeating i times the pair $c_1, (c_1 + c_2 + 1)$. We recall that a $\mathrm{CLTL}_1^2(\mathrm{DL})$ model is simply an ω-sequence of integers. A new configuration is detected when four consecutive values c, d, c', d' are found with either $c \neq c'$ or $d \neq d'$ (i.e. when the value of a counter changes).

- The formula ϕ_{ch} states the change of configuration:

$$\phi_{ch} = x < \mathsf{X}x \ \wedge \ (x \neq \mathsf{X}^2 x \ \vee \ \mathsf{X}(x \neq \mathsf{X}^2 x))$$

- Let Before_i state that we are just before the configuration with control state q_i:

$$\mathrm{Before}_i \stackrel{\text{def}}{=} \phi_{ch} \ \wedge \ \mathsf{X}^2\Big(\bigwedge_{0 \leq j < i-1} \mathsf{X}^{2j}(x = \mathsf{X}^2 x \ \wedge \ \mathsf{X}(x = \mathsf{X}^2 x)) \ \wedge \ \mathsf{X}^{2(i-1)}\phi_{ch}\Big)$$

- Initial configuration:

$$\phi_{init} \stackrel{\text{def}}{=} x = 0 \ \wedge \ \mathsf{X}x = 1 \ \wedge \bigvee_{1 \leq i \leq m} \Big(\bigwedge_{0 \leq j < a_i - 1} \mathsf{X}^{2j}(x = \mathsf{X}^2 x \ \wedge \ \mathsf{X}(x = \mathsf{X}^2 x)) \wedge \bigvee_{\langle q_{a_i}, \alpha, q_{j'} \rangle \in \delta} \mathsf{X}^{2(a_i - 1)}(\phi' \ \wedge \ \text{Before}_{j'}) \Big)$$

- Recurring elements of F: $\phi_{rec} \stackrel{\text{def}}{=} \bigvee_{1 \leq i \leq m'} \mathsf{GF}\text{Before}_{b_i}$.
- Simulation of the run:

$$\phi_{run} \stackrel{\text{def}}{=} \mathsf{G} \bigwedge_{1 \leq i \leq n} (\text{Before}_i \Rightarrow \bigvee_{\langle q_i, \alpha, q_j \rangle \in \delta} \mathsf{X}^{2i}(\alpha' \ \wedge \ \text{Before}_j))$$

  where α' is obtained from α

  - by replacing $x_1 = 0$ by $x = 0$,
  - by replacing $x_2 = 0$ by $\mathsf{X}x = x + 1$,
  - by replacing $\mathsf{X}x_1 = x_1 + d_1$ by $\mathsf{X}^2 x = x + d_1$ for every $d_1 \in \{-1, 0, 1\}$,
  - by replacing $\mathsf{X}x_2 = x_2 + d_2$ by

$$\bigwedge_{d_1 \in \{-1,0,1\}} (\mathsf{X}^2 x = x + d_1) \Rightarrow \mathsf{X}(\mathsf{X}^2 x = x + (d_1 + d_2))$$

    for every $d_2 \in \{-1, 0, 1\}$.

It is easy to show that $\mathcal{A}$ has an accepting run iff the formula $\phi_{init} \wedge \phi_{run} \wedge \phi_{rec}$ is satisfiable. □

Theorem 2 *Satisfiability for* $\text{CLTL}^1_2(\text{DL})$ *is* Σ^1_1*-complete.*

Proof. We prove this result by reducing the satisfiability problem for $\text{CLTL}^2_1(\text{DL})$ to the satisfiability problem for $\text{CLTL}^1_2(\text{DL})$. Basically, the idea is to encode two states from a $\text{CLTL}^2_1(\text{DL})$ model into a single state in a $\text{CLTL}^1_2(\text{DL})$ model. For instance, the $\text{CLTL}^2_1(\text{DL})$ model below

$$x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdots$$

is encoded as the following $\text{CLTL}^1_2(\text{DL})$ model

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \cdots$$

For the rest of this proof, we consider a $\text{CLTL}^2_1(\text{DL})$ model whose unique variable is x and we show how it can be encoded by a $\text{CLTL}^2_1(\text{DL})$ model with variables denoted by y_0 and y_1. We define a map $f : \text{CLTL}^2_1(\text{DL}) \times \{0, 1\} \rightarrow \text{CLTL}^1_2(\text{DL})$ such that $f(\phi, i)$ enforces the variable x in ϕ to be interpreted by y_i.

- $f(\mathsf{R}(\mathsf{X}^{j_1}x,\ldots,\mathsf{X}^{j_s}x),i) = \mathsf{R}(\mathsf{X}^{j_1}x,\ldots,\mathsf{X}^{j_s}x)[\mathsf{X}^{a}x \leftarrow \mathsf{X}^{a'}y_{b'}]$
  where $\mathsf{R}(\mathsf{X}^{j_1}x,\ldots,\mathsf{X}^{j_k}x)[\mathsf{X}^{a}x \leftarrow \mathsf{X}^{a'}x_{b'}]$ is obtained from $\mathsf{R}(\mathsf{X}^{j_1}x,\ldots,\mathsf{X}^{j_s}x)$ by replacing every occurrence of $\mathsf{X}^{a}x$ by $\mathsf{X}^{a'}y_{b'}$ where a,a' and b' are such that $(a+i) = 2a' + b'$ with $a' > 0$ and $b' \in \{0,1\}$.[1]
- f is homomorphic for the Boolean operators,
- $f(\mathsf{X}\phi, 0) = f(\phi, 1)$,
- $f(\mathsf{X}\phi, 1) = \mathsf{X}f(\phi, 0)$,
- $f(\phi\mathsf{U}\psi, 0) = f(\psi, 0) \vee (f(\phi, 0) \wedge (f(\psi, 1) \vee ((f(\phi, 1) \wedge \mathsf{X}(\phi'\mathsf{U}\psi'))))$ and
  $f(\phi\mathsf{U}\psi, 1) = f(\psi, 1) \vee ((f(\phi, 1) \wedge \mathsf{X}(\phi'\mathsf{U}\psi'))))$ where
  - $\phi' = f(\phi, 0) \wedge f(\phi, 1)$,
  - $\psi' = f(\psi, 0) \vee (f(\phi, 0) \wedge f(\psi, 1))$.

One can easily show that ϕ is $\mathrm{CLTL}_1^2(\mathrm{DL})$ satisfiable iff $f(\phi, 0)$ is $\mathrm{CLTL}_1^2(\mathrm{DL})$ satisfiable. □

The satisfiability problem can be reduced to the model-checking problem since $\phi \in \mathrm{CLTL}(\mathrm{DL})$ is satisfiable iff $\mathcal{A}_\top \models \phi$ where $\mathcal{A}_\top$ is the one-state DL-automaton $\langle\{q\}, \delta, \{q\}, \{q\}\rangle$ with unique transition $q \xrightarrow{\top} q$ (i.e. every model realizes the execution of $\mathcal{A}_\top$). We obtain the following corollary.

Corollary 1 *The model-checking problems for* $\mathrm{CLTL}_2^1(\mathrm{DL})$ *and* $\mathrm{CLTL}_1^2(\mathrm{DL})$ *are* Σ_1^1*-complete.*

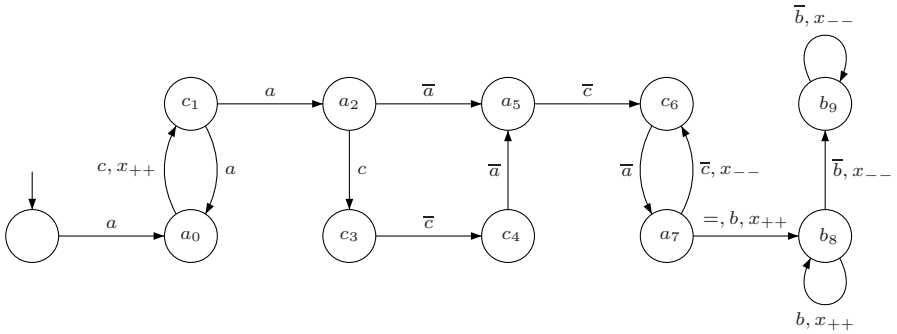
The Σ_1^1 upper bound is obtained by reducing model-checking to satisfiability for CLTL(DL) along the lines of [SC85] (this technique is also used in the proof of Lemma 1). Indeed, in order to simulate a propositional variable, we use a constraint of the form $x = 0$ assuming that x is not used already for other purposes. By close inspection of the proofs of Theorems 1 and 2, one can even show that satisfiability and model-checking for $\mathrm{CLTL}_2^1(\mathrm{DL})$ and $\mathrm{CLTL}_1^2(\mathrm{DL})$ restricted to the temporal operator F (instead of until) are also Σ_1^1-hard.

3 PSPACE-completeness of $\mathbf{CLTL_1^1(DL^+)}$ with propositional variables

To complete the results of Section 2, we show that satisfiability for $\mathrm{CLTL}_1^1(\mathrm{DL})$ and model-checking $\mathrm{CLTL}_1^1(\mathrm{DL})$ formulae over one-variable DL-automata are both PSPACE-complete, thus refining the EXPTIME bound established for pushdown systems [FWW97]. To do so, we establish a PSPACE upper bound for satisfiability of the richer logic $\mathrm{CLTL}_1^1(\mathrm{DL}^+, \mathrm{PROP})$ that includes periodicity constraints of the form $x \equiv_k y + c$, $x \equiv_k c$ ($c \in \mathbb{N}$, $k \in \mathbb{N} \setminus \{0,1\}$) and propositional variables that can be viewed as specific variables with their values restricted to be in $\{0,1\}$. We get as a corollary that the model-checking problem

[1] Note that $a' > 0$ and $b' \in \{0,1\}$ imply that a' and b' are unique.

for $\mathrm{CLTL}_1^1(\mathrm{DL})$ is in PSPACE by reducing this problem in logarithmic space to the satisfiability problem for $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$. The above-mentioned results complete our classification. Furthermore, many problems on one-counter automata/nets can be encoded in $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$. These problems come from several applications: verification of cryptographic protocols [LLT05], validation of XML streams (string representations of XML documents) [CR04], and resolution of the identification problem described in [WTT04]. For example, the class of one-variable DL-automata properly contains the one-counter automata used to validate XML streams against a recursive DTD in [CR04, Sect.5]. Below is the one-counter automaton recognizing the language $\{(ac)^n a(\varepsilon|c\overline{c})\overline{a}(\overline{ca})^n b^m \overline{b}^m : n, m \geq 1\}$ where x is a counter and the alphabet is partitioned into a set of opening tags $\{a, b, c\}$ and a set of corresponding closing tags $\{\overline{a}, \overline{b}, \overline{c}\}$. The symbol "=" for the transition between a_7 and b_8 refers to a zero-test and we assume that the acceptance condition of the automaton is defined with respect to the final configuration $\langle b_9, 0\rangle$.



Checking if a word belongs to the language recognized by this automaton, a key problem in [CR04, Sect.5], can be expressed in our formalism. The fragment $\mathrm{CLTL}_1^1(\mathrm{DL}^+)$ can also express concisely richer standard properties, for instance non-trivial safety properties of the form $\mathsf{G}(x < 2^m)$ and liveness properties such as $\mathsf{G}(x \equiv_{2^n} 0 \Rightarrow \mathsf{F}(x \equiv_{3^m} 1))$. We also recall that DL-automata are strictly more expressive than one-counter automata since the transitions are not restricted to incrementationss and decrementations only. Herein, we provide an optimal PSPACE upper bound refining the decidability result from [FWW97] for the model checking of $\mathrm{CTL}^\star$ properties over pushdown systems.

3.1 Adding propositional variables

Let $\mathrm{PROP} = \{p_1, p_2, \ldots\}$ be a countably infinite set of propositional variables. We define the logic $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$ as the extension of $\mathrm{CLTL}(\mathrm{DL}^+)$ obtained by adding propositional variables at the atomic level. As a consequence, the models of $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$ are pairs of the form $\langle \sigma_1, \sigma_2 \rangle$ such that $\sigma_1 : \mathbb{N} \to 2^{\mathrm{PROP}}$ is a standard LTL model and $\sigma_2 : \mathbb{N} \to (\mathrm{VAR} \to \mathbb{Z})$ is a $\mathrm{CLTL}(\mathrm{DL}^+)$ model. The satisfaction relation is defined as for CLTL(DL) except at the atomic level:

- for every proposition $p \in \mathrm{PROP}$ we have $\langle \sigma_1, \sigma_2 \rangle, i \models p \stackrel{\text{def}}{\Leftrightarrow} p \in \sigma_1(i)$ and

- for every DL^+-constraint α we have $\langle \sigma_1, \sigma_2 \rangle, i \models \alpha \stackrel{\text{def}}{\Leftrightarrow} \sigma_2(i) \models \alpha$ (using the satisfaction relation of $\mathrm{CLTL}(\mathrm{DL}^+)$).

There is no restriction on the propositional variables in the fragments of the form $\mathrm{CLTL}^l_k(\mathrm{DL}^+, \mathrm{PROP})$.

3.2 Model-checking one-counter (DL^+)-automata

The presence of propositional variables in $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$ makes the reduction from the model-checking problem for $\mathrm{CLTL}(\mathrm{DL}^+)$ to the satisfiability problem for $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$ easy.

Lemma 1 *There is a logspace reduction from the model-checking problem for* $\mathrm{CLTL}(\mathrm{DL}^+)$ *to the satisfiability problem for* $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$.

Proof. The reduction is similar to the reduction from model-checking to satisfiability for standard LTL [SC85]. Let ϕ be a $\mathrm{CLTL}(\mathrm{DL}^+)$ formula and $\mathcal{A} = \langle Q, \delta, I, F \rangle$ be a DL^+-automaton with $\delta \subseteq Q \times 1SC_k \times Q$. We describe below the construction of a $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$ formula $\phi_{\mathcal{A}}$ such that $\mathcal{A} \models \phi$ iff $\phi \wedge \phi_{\mathcal{A}}$ is $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$ satisfiable. For every location q_i in $Q = \{q_1, \ldots, q_n\}$, we introduce a propositional variable p_i. First, we state that a unique propositional variable holds true at each position:

$$\phi_{uni} \stackrel{\text{def}}{=} \bigvee_{i \in \{1,\ldots,n\}} (p_i \wedge \bigwedge_{j \in \{1,\ldots,n\} \setminus \{i\}} \neg p_j).$$

The transition relation can be encoded as follows:

$$\phi_{next} \stackrel{\text{def}}{=} \bigwedge_{i \in \{1,\ldots,n\}} (p_i \Rightarrow \bigvee_{\langle q_i, \alpha, q_j \rangle \in \delta} (\alpha \wedge \mathsf{X} p_j)).$$

Finally, initial configurations and the accepting condition are expressed by the formulae below:

$$\phi_{init} \stackrel{\text{def}}{=} \bigvee_{q_i \in I} p_i, \qquad \phi_{acc} \stackrel{\text{def}}{=} \mathsf{GF}(\bigvee_{q_i \in F} p_i).$$

The formula $\phi_{\mathcal{A}}$ is $\phi_{\mathcal{A}} \stackrel{\text{def}}{=} \phi_{init} \wedge \mathsf{G}(\phi_{uni} \wedge \phi_{next}) \wedge \phi_{acc}$. □

As a corollary, we have the following reduction for the model-checking problem of $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$.

Corollary 2 *The model-checking problem for* $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ *can be reduced in logspace to the satisfiability problem for* $\mathrm{CLTL}^1_1(\mathrm{DL}^+, \mathrm{PROP})$.

Proof. The model-checking problem for $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ can be reduced to a satisfiability problem by using the reduction of Lemma 1. The formula built with this construction always belongs to $\mathrm{CLTL}^1_1(\mathrm{DL}^+, \mathrm{PROP})$. Indeed, the guards from a one-variable (DL^+)-automaton are one-step constraints from $1SC_1$ which corresponds to the set of $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ atomic constraints. □

We dedicate the remaining of this section to prove decidability of the satisfiability problem for $\mathrm{CLTL}^1_1(\mathrm{DL}^+, \mathrm{PROP})$, which is partially based on the abstraction of models.

3.3 Symbolic models

Symbolic valuations. In order to build an automaton that recognizes the symbolic models of a $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ formula, we first introduce a symbolic representation of valuations. Without any loss of generality, we can assume that all the atomic constraints involving both x and $\mathsf{X}x$ in a $\mathrm{CLTL}^1_1(\mathrm{DL}^+, \mathrm{PROP})$ formula (and in any set of constraints used in the sequel) are of the form $\mathsf{X}x \sim x+d$ and $\mathsf{X}x \equiv_k x + c$ where $d \in \mathbb{Z}$, $c \in \mathbb{N}$ and $k \in \mathbb{N} \setminus \{0, 1\}$. Given a finite set X of one-step constraints from $\mathrm{CLTL}(\mathrm{DL}^+)$ built over the variable x, we consider the following syntactic resources.

- $\mathrm{CONS}_x = \{d_{\min}, \ldots, d_{-1}, d_0, d_1, \ldots, d_{\max}\}$ is the set of constants d occurring in X in constraints of the form $x \sim d$ and $\mathsf{X}x \sim d$. We assume that $d_{\min} < \cdots < d_{-1} < d_0 < d_1 < \cdots < d_{\max}$.
- $\mathrm{CONS}_{\mathrm{step}} = \{e_{\min'}, \ldots, e_{-1}, e_0, e_1, \ldots, e_{\max'}\}$ is the set of constants e occurring in X in constraints of the form $\mathsf{X}x \sim x + e$. We assume that $e_{\min'} < \cdots < e_{-1} < e_0 < e_1 < \cdots < e_{\max'}$.
- K is the least common multiple (lcm) of all the integers $k \in \mathbb{N} \setminus \{0, 1\}$ such that a relation of the form $\equiv_k$ occurs in X.

Without loss of generality, we can assume that $d_0 = e_0 = 0$, $d_{\max} \geq 0$, $e_{\max'} \geq 0$, $d_{\min} \leq 0$ and $e_{\min'} \leq 0$.

We define an abstraction for valuations that is similar to regions for timed automata [AD94] and we shall prove that this abstraction fits exactly our goal. A map $\{x, \mathsf{X}x\} \to \mathbb{Z}$ (that can also be viewed as a pair $\langle z_1, z_2\rangle \in \mathbb{Z}^2$) is represented by a tuple $sv = \langle \alpha_x, \alpha_m, \alpha'_x, \alpha'_m, \alpha_s\rangle \in \mathrm{C}_x \times \mathrm{Mod}_x \times \mathrm{C}_{\mathsf{X}x} \times \mathrm{Mod}_{\mathsf{X}x} \times \mathrm{C}_{\mathrm{step}}$ (depending on X) such that

- C_x is composed of constraints of the form below
  - $d_i < x \wedge x < d_{i+1}$ for $i \in \{\min, \ldots, \max -1\}$,
  - $x = d_i$ for $i \in \{\min, \ldots, \max\}$,
  - $x < d_{\min}$ and $d_{\max} < x$,
- Mod_x is composed of the constraints of the form
  - $x \equiv_K c$ for $c \in \{0, \ldots, K-1\}$,
- $\mathrm{C}_{\mathsf{X}x}$ and $\mathrm{Mod}_{\mathsf{X}x}$ are defined similarly to C_x and Mod_x respectively, by replacing "x" by "$\mathsf{X}x$" in the constraints used,
- $\mathrm{C}_{\mathrm{step}}$ is composed of constraints of the form below
  - $x + e_i < \mathsf{X}x \wedge \mathsf{X}x < x + e_{i+1}$ for $i \in \{\min', \ldots, \max' -1\}$,

– $\mathsf{X}x = x + e_i$ for $i \in \{\min', \ldots, \max'\}$,
– $\mathsf{X}x < x + e_{\min'}$ and $x + e_{\max'} < \mathsf{X}x$.

We call the tuple *sv symbolic valuation* and we write $\mathrm{SV}(X)$ to denote the set of symbolic valuations with respect to the set X. By extension, $\mathrm{SV}(\phi)$ denotes the set of symbolic valuations with respect to the set of atomic constraints of the $\mathrm{CLTL}^1_1(\mathrm{DL}^+, \mathrm{PROP})$ formula ϕ. Note that the size of the set $\mathrm{SV}(\phi)$ is exponential in the size of ϕ (but each element has polynomial size with respect to $|\phi|$).

Let $v : \{x, \mathsf{X}x\} \to \mathbb{Z}$ be a valuation and sv be a symbolic valuation. We note $v \models sv$ iff v satisfies all the constraints in sv. The following result shows the correctness of our symbolic valuation abstraction.

Lemma 2 *Let X be a finite set of one-step constraints from $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ built over the variable x.*

(I) *For every map $v : \{x, \mathsf{X}x\} \to \mathbb{Z}$ there is a unique symbolic valuation in $SV(X)$ denoted by $sv(v)$ such that $v \models sv(v)$.*

(II) *For every pair of maps $v, v' : \{x, \mathsf{X}x\} \to \mathbb{Z}$ such that $sv(v) = sv(v')$ $(sv(v), sv(v') \in SV(X))$ and for every $\alpha \in X$, we have $v \models \alpha$ iff $v' \models \alpha$.*

Proof. **(I)** Given $sv \in \mathrm{SV}(X)$, let V_{sv} be the set of pairs $\langle z_1, z_2 \rangle \in \mathbb{Z}^2$ such that $\langle z_1, z_2 \rangle \models sv$. By definition of symbolic valuations, it is obvious that $\{V_{sv} : sv \in \mathrm{SV}(X), V_{sv} \neq \emptyset\}$ is a partition of $\mathbb{Z}^2$.

(II) Let v and v' be valuations such that $sv(v) = sv(v') = \langle \alpha_x, \alpha_m, \alpha'_x, \alpha'_m, \alpha_s \rangle$ and suppose that $v \models \alpha$. We proceed by induction on the structure of α (we omit the cases with Boolean connectives).

- If α is of the form $x = d$ then $v(x) = d$ and α_x must be equal to α. Indeed, we have $d \in \mathrm{CONS}_x$ by definition of CONS_x and so the only constraint of C_x satisfied by v is $x = d$ (see the definition of C_x). Since v' also satisfies α_x, we have $v' \models \alpha$.
- If α is of the form $x < d$ then $v(x) < d$ and α_x can be equal either to $x = d'$ with $d' < d$ or to $d'' < x \wedge x < d'$ with $d' \leq d$. In both cases, it is easy to check that if $v' \models \alpha_x$ then we also have $v' \models \alpha$.
- If α is of the form $\mathsf{X}x \sim d$ (resp. $\mathsf{X}x \sim x + d$) where $\sim \in \{<, =\}$, the proof is similar to the two above cases, using the constraint α'_x (resp. α_s).
- Let α be of the form $x \equiv_k c$. We suppose that the constraint α_m is equal to $x \equiv_K c'$. By construction of K, k divides K and so we have $\alpha_m \Rightarrow (x \equiv_k c'_r)$ is valid in Presburger arithmetic where $c'_r \in \mathbb{N}$ is the remainder of the division of c' by k. As c and c'_r belong to $\{0, \ldots, k-1\}$ and v satisfies both α and α_m, c must be equal to c'_r. Since $v' \models \alpha_m$ and $\alpha_m \Rightarrow \alpha$ is valid, we have $v' \models \alpha$.

- If α is of the form $\mathsf{X}x \equiv_k c$ (resp. $\mathsf{X}x \equiv_k x + c$) the proof is similar to the previous case, using the constraint α'_m (resp. the conjunction $\alpha_m \wedge \alpha'_m$).

□

Given an atomic formula α from $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$, we note $sv \models_{\mathrm{symb}} \alpha$ iff for every valuation v such that $sv(v) = sv$ we have $v \models \alpha$. Note that this notion is well-defined thanks to Lemma 2.

Our symbolic representation of $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ models relies on sequences of symbolic valuations. We say that an infinite sequence $\rho : \mathbb{N} \to \mathrm{SV}(\phi)$ of symbolic valuations with respect to ϕ is satisfiable iff there is a $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ model $\sigma : \mathbb{N} \to \mathbb{Z}$ such that for every $i \in \mathbb{N}$ we have $\sigma, i \models \rho(i)$. In this case, we write $\sigma \models \rho$. A *symbolic model* with respect to ϕ is a pair $\langle \sigma_1, \rho \rangle$ such that $\sigma_1 : \mathbb{N} \to 2^{\mathrm{PROP}}$ and $\rho : \mathbb{N} \to \mathrm{SV}(\phi)$. We extend the symbolic satisfaction relation $\models_{\mathrm{symb}}$ to symbolic models in a natural way. The definition is identical to the satisfaction relation of $\mathrm{CLTL}(\mathrm{DL}^+, \mathrm{PROP})$ except for atomic constraints: $\langle \sigma, \rho \rangle, i \models_{\mathrm{symb}} \alpha \stackrel{\mathrm{def}}{\Leftrightarrow} \rho(i) \models_{\mathrm{symb}} \alpha$.

3.4 Automata-based approach

We now show that given a $\mathrm{CLTL}^1_1(\mathrm{DL}^+, \mathrm{PROP})$ formula ϕ, one can build an automaton $\mathcal{A}_\phi$ recognizing the set of symbolic models satisfying ϕ. To define this automaton we extend the definition of simple one-$\mathbb{Z}$-counter automata (where updates are restricted to $\{-1, 0, 1\}$) with an alphabet Σ and ε-transitions. The transitions are decorated by elements from $\Sigma \cup \{\varepsilon\}$ and the constraints are the same as in one-$\mathbb{Z}$-counter automata. Adding an alphabet is motivated by the need to consider the automata as language acceptors. So, we define

- $\mathrm{L}'(\mathcal{A}) = \{\sigma : \mathbb{N} \to (\Sigma \cup \{\varepsilon\}) \ : \$ there is an accepting run $w : \mathbb{N} \to (Q \times \mathbb{Z})$ such that $\forall i,\ w(i) \xrightarrow{\sigma(i)} w(i+1)\}$,
- $\mathrm{L}(\mathcal{A}) = \{\sigma^{\setminus \varepsilon} \ : \ \sigma \in \mathrm{L}'(\mathcal{A})$ and $|\sigma^{\setminus \varepsilon}| = \infty\}$,

where $\sigma^{\setminus \varepsilon}$ is obtained from σ by erasing all the occurrences of the letter ε. The set $\mathrm{L}(\mathcal{A})$ is the language accepted by $\mathcal{A}$ and corresponds to the accepting runs of $\mathrm{L}'(\mathcal{A})$ where non-ε-transitions are fired infinitely often. The construction of $\mathcal{A}_\phi$ relies on the following observation.

Lemma 3 *A* $\mathrm{CLTL}^1_1(\mathrm{DL}^+, \mathrm{PROP})$ *formula* ϕ *is satisfiable iff there exist a symbolic model* $\langle \sigma_1, \rho \rangle$ *such that* $\langle \sigma_1, \rho \rangle \models_{\mathrm{symb}} \phi$ *and a* $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ *model* σ_2 *such that* $\sigma_2 \models \rho$.

Proof. If ϕ is satisfiable then we consider a model $\langle \sigma_1, \sigma_2 \rangle$ that satisfies it. Let $\rho : \mathbb{N} \to \mathrm{SV}(\phi)$ be the symbolic model such that $\rho(i) = sv(v_i)$ where the valuation $v_i : \{x, \mathsf{X}x\} \to \mathbb{Z}$ is defined by $v_i(x) = \sigma_2(i)$ and $v_i(\mathsf{X}x) = \sigma_2(i+1)$ for every $i \in \mathbb{N}$. By construction, we have $\sigma_2 \models \rho$. Using Lemma 2(II) we can show that for every v such that $sv(v) = \rho(i)$ we have $\sigma_2(i) \models \alpha$ iff $v \models \alpha$ for every atomic subformula α of ϕ. By definition of the symbolic satisfaction relation, this

implies that if $\sigma_2 \models \alpha$ then $\rho \models_{\text{symb}} \alpha$. Consequently, $\langle \sigma_1, \rho \rangle \models_{\text{symb}} \phi$ because the symbolic satisfaction relation differs from CLTL(DL^+, PROP) satisfaction relation only in the case of atomic constraints.

Conversely, suppose that $\langle \sigma_1, \rho \rangle \models_{\text{symb}} \phi$ and $\sigma_2 \models \rho$ for some σ_1, σ_2 and ρ. Since for every $i \in \mathbb{N}$ we have $\sigma_2, i \models \rho(i)$, the symbolic valuation $\rho(i)$ corresponds to the abstraction of the valuation v_i such that $v_i(x) = \sigma_2(i)$ and $v_i(\mathsf{X}x) = \sigma_2(i+1)$. By definition of $\models_{\text{symb}}$, this implies that for every atomic subformula α of ϕ, if $\rho(i) \models_{\text{symb}} \alpha$ then $\sigma_2, i \models \alpha$. Thus, we can show that $\langle \sigma_1, \rho \rangle \models_{\text{symb}} \phi$ and $\sigma_2 \models \rho$ imply $\langle \sigma_1, \sigma_2 \rangle \models \phi$ by using the fact that CLTL(DL^+, PROP) satisfaction relation is different from the symbolic satisfaction only for atomic constraints. □

So, we define $\mathcal{A}_\phi$ as the intersection of two automata $\mathcal{A}_{\text{symb}}$ and $\mathcal{A}_{\text{sat}}$ such that $\text{L}(\mathcal{A}_{\text{symb}})$ is the set of symbolic models that symbolically satisfy ϕ. Furthermore, $\text{L}(\mathcal{A}_{\text{sat}})$ is the set of symbolic models of the form $\langle \sigma_1, \rho \rangle$ such that ρ is satisfiable. Both automata are simple one-$\mathbb{Z}$-counter automata over the alphabet $\Sigma = (2^{\text{PROP}} \times \text{SV}(\phi))$ but $\mathcal{A}_{\text{symb}}$ is essentially a finite-state automaton (the counter is not used).

Main steps to get the PSPACE upper bound. The automaton $\mathcal{A}_{\text{symb}}$ is built as in [VW94] for LTL except at the atomic level. We define $cl(\phi)$ the closure of ϕ with a slight modification to consider both atomic constraints and propositional variables. Let us briefly recall that the closure set $cl(\phi)$ is the smallest set containing ϕ, closed under subformulae, negations (double negations are eliminated) and such that if $\psi \mathsf{U} \psi' \in cl(\phi)$, then $\mathsf{X}(\psi \mathsf{U} \psi') \in cl(\phi)$. A set $X \subseteq cl(\phi)$ is an atom whenever it satisfies the usual conditions for subformulae whose outermost connective is Boolean and, we have $\psi \mathsf{U} \psi' \in X$ iff ($\psi' \in X$ or $(\psi, \mathsf{X}(\psi \mathsf{U} \psi') \in X)$) whenever $\psi \mathsf{U} \psi' \in cl(\phi)$. Let $\mathcal{A}'_{\text{symb}}$ be the generalized Büchi automaton defined as the structure $\langle Q, \delta, I, F \rangle$ such that:

- Q is the set of atoms of ϕ,
- $I = \{X \in Q : \phi \in X\}$,
- $X \xrightarrow{\langle P, sv \rangle, \top, 0} Y$ iff

  (prop.) $P = X \cap \text{PROP}$,
  (constr.) for every atomic formula α in X, $sv \models_{\text{symb}} \alpha$,
  (1-step) for every $\mathsf{X}\psi \in cl(\phi)$, $\mathsf{X}\psi \in X$ iff $\psi \in Y$,

- Let $\{\psi_1 \mathsf{U} \phi_1, \ldots, \psi_n \mathsf{U} \phi_n\}$ be the set of until formulae in $cl(\phi)$. We pose $F = \{F_1, \ldots, F_n\}$ where $F_i = \{X \in Q : \psi_i \mathsf{U} \phi_i \notin X \text{ or } \phi_i \in X\}$ for every $i \in \{1, \ldots, n\}$.

The automaton $\mathcal{A}_{\text{symb}}$ is a non-generalized Büchi automaton equivalent to $\mathcal{A}'_{\text{symb}}$. It can be built in logarithmic space in the size of $\mathcal{A}'_{\text{symb}}$. Note that the counter is useless in this construction.

Section 3.5 is dedicated to the lengthy construction of $\mathcal{A}_{\text{sat}}$. The final automaton is obtained by synchronizing $\mathcal{A}_{\text{symb}}$ and $\mathcal{A}_{\text{sat}}$ in the following way. Let us pose $\mathcal{A}_{\text{symb}} = \langle Q_{sy}, \delta_{sy}, I_{sy}, F_{sy}\rangle$ and $\mathcal{A}_{\text{sat}} = \langle Q_{sa}, \delta_{sa}, I_{sa}, F_{sa}\rangle$. The automaton $\mathcal{A}_{\text{sat}}$ has ε-transitions while $\mathcal{A}_{\text{symb}}$ has not. The ε-transitions of $\mathcal{A}_{\text{sat}}$ can be fired independently of $\mathcal{A}_{\text{symb}}$. Otherwise, one can make a move in both $\mathcal{A}_{\text{symb}}$ and $\mathcal{A}_{\text{sat}}$ when the same letter is read. In this case, the counter is updated according to the transition in $\mathcal{A}_{\text{sat}}$ because the counter is useless in $\mathcal{A}_{\text{symb}}$. Formally, the automaton $\mathcal{A}_\phi = \langle Q, \delta, I, F\rangle$ is defined by:

- $Q = Q_{sy} \times Q_{sa}$,
- $I = I_{sy} \times I_{sa}$,
- $F = F_{sy} \times Q_{sa}$ (since we will have $Q_{sa} = F_{sa}$),
- $\langle q_1, q_2\rangle \xrightarrow{\varepsilon,t,u} \langle q_1', q_2'\rangle \in \delta \stackrel{\text{def}}{\Leftrightarrow} q_1 = q_1'$ and $q_2 \xrightarrow{\varepsilon,t,u} q_2' \in \delta_{sa}$,
- $\langle q_1, q_2\rangle \xrightarrow{\langle P,sv\rangle,t,u} \langle q_1', q_2'\rangle \in \delta \stackrel{\text{def}}{\Leftrightarrow} q_1 \xrightarrow{\langle P,sv\rangle,\top,0} q_1' \in \delta_{sy}$ and $q_2 \xrightarrow{\langle P,sv\rangle,t,u} q_2' \in \delta_{sa}$.

We will show at the end of this section that $\mathcal{A}_\phi$ can be effectively built from ϕ in polynomial space thanks to the way $\mathcal{A}_{\text{sat}}$ can be built. Moreover the nonemptiness problem for one-$\mathbb{Z}$-counter automata with alphabet and ε-transitions is shown to be NLOGSPACE-complete in Section 5. By combining these two results, we obtain the result below.

Theorem 3 *The satisfiability problem for the fragment* $\text{CLTL}_1^1(\text{DL}^+, \text{PROP})$ *is* PSPACE*-complete.*

Proof. The size of the automaton $\mathcal{A}_\phi$ is exponential in the size of the formula ϕ (denoted by $|\phi|$) and it can be built in polynomial space with respect to $|\phi|$. Since the nonemptiness problem for the resulting automaton is in NLOGSPACE and the composition of a logarithmic space function with a polynomial space function is a polynomial space function [BDG88, Lemma 3.35], we obtain a whole procedure in nondeterministic polynomial space. Using Savitch's theorem, we can deduce that the problem is in PSPACE.

The PSPACE-hardness is given by the PSPACE-hardness of LTL. The logic LTL is subsumed by the fragment $\text{CLTL}_1^1(\text{DL}^+, \text{PROP})$ because the number of propositional variables in this fragment is not concerned by the syntactic restrictions. □

We can also deduce the following corollaries.

Corollary 3 *The satisfiability and model-checking problems for* $\text{CLTL}_1^1(\text{DL})$ *are* PSPACE*-complete.*

Proof. These problems can easily be shown to be in PSPACE using Theorem 3 and Lemma 1. In order to prove the PSPACE-hardness, we reduce the satisfiability problem for the one-variable fragment of LTL (shown to be to PSPACE-complete in [DS02, Corollary 3.2]) to the satisfiability problem for $\text{CLTL}_1^1(\text{DL})$. Given an

LTL formula ϕ with a unique propositional variable p, we simply replace each occurrence of p by $x = 0$, providing a formula ϕ' in $\mathrm{CLTL}^1_1(\mathrm{DL})$. It is obvious that ϕ is satisfiable iff ϕ' is satisfiable. □

Corollary 4 *The one-variable fragment of the counter logic $\mathcal{L}_p$ [CC00] has a* PSPACE*-complete satisfiability problem.*

Another corollary of Theorem 3 is that model-checking one-clock discrete timed automata from [DPK03] with $\mathrm{CLTL}^1_1(\mathrm{DL}^+)$ can be done in PSPACE which contrasts with the undecidability results from [DPK03, Section 6].

A nice feature of this approach is that the temporal logic part is separated from the constraint part. As a consequence, this method can be tailored to any extension with Presburger constraints of a temporal logic whose operators are definable in the Monadic Second Order logic (MSO), thanks to [GK03]. One just needs to replace the automaton $\mathcal{A}_{\mathrm{symb}}$ by an appropriate automaton recognizing the models of the input formula in the corresponding propositional logic (instead of the LTL automaton). By [GK03], this automaton can be built in polynomial space and so we preserve our complexity bound for this kind of extensions. This includes extensions of LTL with past-time operators, with automaton-based temporal operators [Wol83], or fixpoints operators [Var88]. All these extensions are denoted by xCLTL in the following corollary.

Corollary 5 *The satisfiability and model-checking problems for* $\mathrm{xCLTL}^1_1(\mathrm{DL})$ *are* PSPACE*-complete.*

As a consequence, the satisfiability and model-checking of the linear μ-calculus extended with DL-constraints is PSPACE-complete when the above syntactic restrictions are made. This refines the result stated in [BEM97].

3.5 Construction of $\mathcal{A}_{\mathrm{sat}}$

In the rest of this section, we describe the construction of the automaton $\mathcal{A}_{\mathrm{sat}}$ recognizing exactly the set of symbolic models $\langle \sigma_1, \rho \rangle$ such that ρ is satisfiable. We recall that the set of constants CONS_x used in the symbolic representation of the models contains the following elements: $d_0 = 0$, $d_{\max} \geq 0$ and $d_{\min} \leq 0$.

The alphabet of $\mathcal{A}_{\mathrm{sat}}$ is $2^{\mathrm{PROP}} \times \mathrm{SV}(\phi)$ but since the set of propositional variables is not constrained in $\mathcal{A}_{\mathrm{sat}}$, we omit them in the technical developments below. This means that for every non-epsilon transition of the form $q \xrightarrow{sv,t,u} q'$ defined in the rest of this section, we mean to consider all the transitions $q \xrightarrow{\langle X, sv \rangle, t, u} q'$ for some $X \in 2^{\mathrm{PROP}}$.

The construction of $\mathcal{A}_{\mathrm{sat}}$ is done in a modular fashion. The automaton $\mathcal{A}_{\mathrm{sat}}$ is made of a network of components/gadgets and its size is exponential with respect to the size of the input formula ϕ. A *component* is defined as a simple one-$\mathbb{Z}$-counter automaton $\langle \Sigma, Q, \delta, I, F \rangle$ such that

- I and F are singletons,

- δ is a subset of $(Q \setminus F) \times \{\varepsilon\} \times \{\top, =, \neq, >, <\} \times \{-1, 0, 1\} \times (Q \setminus I)$.

The unique state in I (resp. F) is called the *input* (resp. *output*) state of the component. Components are connected in the network by defining transitions between input states and output states. Each component in $\mathcal{A}_{\text{sat}}$ has the function either to check a property of the counter from constraints in C_x or to update the counter according to constraints in Mod_x or $\mathrm{Mod}_{\mathsf{X}x} \times \mathrm{C}_{\text{step}}$ (viewed as a conjunction). We define below the components $\mathcal{A}_{\alpha,sv}$ for some $\alpha \in \mathrm{C}_x \cup \mathrm{Mod}_x \cup (\mathrm{Mod}_{\mathsf{X}x} \times \mathrm{C}_{\text{step}})$ and $sv \in \mathrm{SV}(\phi)$. We write $q_{in}^{\alpha,sv}$ (resp. $q_{out}^{\alpha,sv}$) to denote the input (resp. output) state of $\mathcal{A}_{\alpha,sv}$. However, when the context is clear we shortly write q_{in} and q_{out}. Each component $\mathcal{A}_{\alpha,sv}$ enforces that the next symbolic valuation that is guessed is precisely sv. For every $sv = \langle \alpha_x, \alpha_m, \alpha'_x, \alpha'_m, \alpha_s \rangle \in \mathrm{SV}(\phi)$, we define the following components:

- $\mathcal{A}_{\alpha_x,sv}$ is such that for every $c \in \mathbb{Z}$, $\langle q_{in}^{\alpha_x,sv}, c\rangle \rightarrow^* \langle q_{out}^{\alpha_x,sv}, c'\rangle$ iff $c = c'$ and $[x \leftarrow c] \models \alpha_x$. This component checks that value of the counter c satisfies α_x. Figure 1 contains a graphical representation of components $\mathcal{A}_{x=d_i,sv}$ and $\mathcal{A}_{d_i<x<d_{i+1},sv}$ when $d_i \geq 0$. Components with $d_i \leq 0$ can be defined analogously.

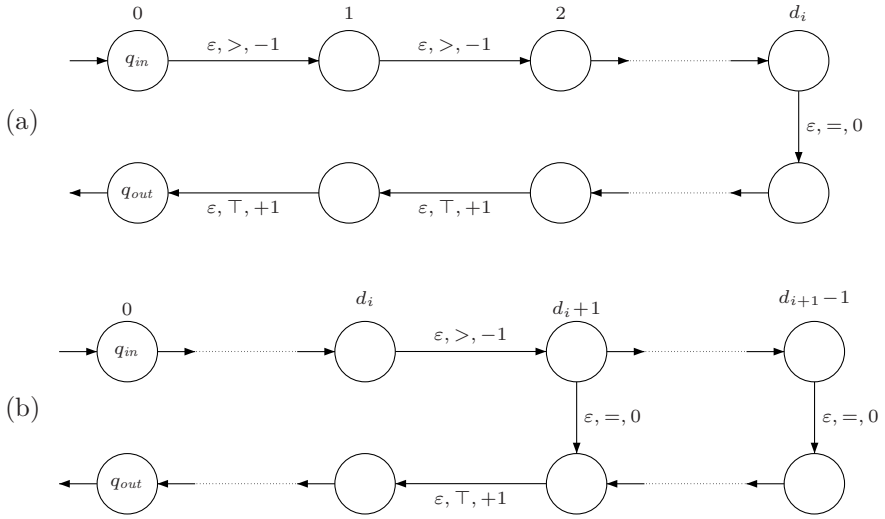


Figure 1: Components $\mathcal{A}_{x=d_i,sv}$ (a) and $\mathcal{A}_{d_i<x<d_{i+1},sv}$ (b)

- $\mathcal{A}_{\langle \alpha'_m, \alpha_s\rangle,sv}$ is such that for every $c \in \mathbb{Z}$, $[x \leftarrow c] \models \alpha_m$ and $\langle q_{in}^{\langle \alpha'_m, \alpha_s\rangle,sv}, c\rangle \rightarrow^* \langle q_{out}^{\langle \alpha'_m, \alpha_s\rangle,sv}, c'\rangle$ iff $[x \leftarrow c, \mathsf{X}x \leftarrow c'] \models \alpha'_m \wedge \alpha_s$. This component updates the counter according to $\langle \alpha'_m, \alpha_s\rangle$. Figure 2 contains a graphical representation of the component $\mathcal{A}_{\langle \alpha'_m, \alpha_s\rangle,sv}$ with $\alpha_m = x \equiv_2 1$, $\alpha'_m = \mathsf{X}x \equiv_2 0$ and $\alpha_s = x < \mathsf{X}x < x + 7$. To build $\mathcal{A}_{\langle \alpha'_m, \alpha_s\rangle,sv}$, we determine on-the-fly (using α_m, α'_m and α_s) that $\mathsf{X}x = x + i$ for some $i \in \{1, 3, 5\}$.
- $\mathcal{A}_{\alpha_m,sv}$ is such that for every $c \in \mathbb{Z}$, $\langle q_{in}^{\alpha_m,sv}, 0\rangle \rightarrow^* \langle q_{out}^{\alpha_m,sv}, c\rangle$ iff $[x \leftarrow c] \models \alpha_m$. This component updates the counter from 0 to a value satisfying α_m

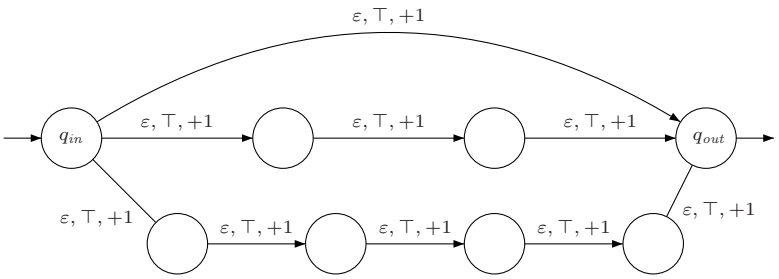


Figure 2: Component $\mathcal{A}_{\langle \alpha'_m, \alpha_s \rangle, sv}$

(only used at the beginning of the run). Figure 3 contains a graphical representation of some component $\mathcal{A}_{x \equiv_K c, sv}$.

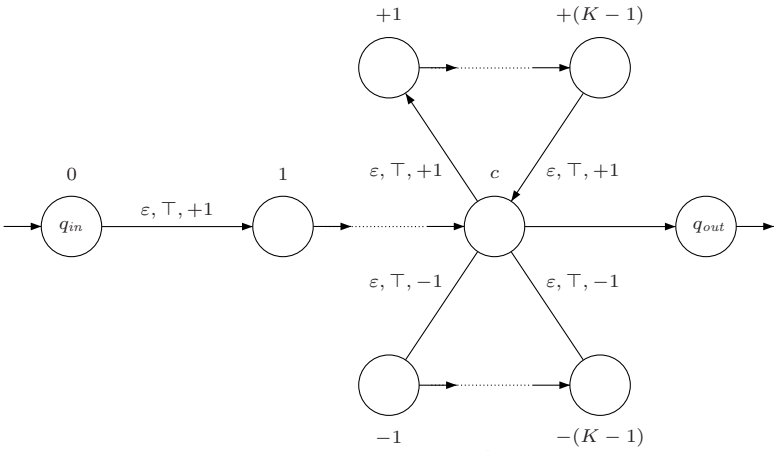


Figure 3: Component $\mathcal{A}_{x \equiv_K c, sv}$

The automaton $\mathcal{A}_{\text{sat}} = \langle \Sigma, Q, \delta, I, F \rangle$ is defined as the "disjoint union" of the above-mentioned components with an additional initial state s_0, $F = Q$ and with the following additional transitions.

- For every $sv = \langle \alpha_x, \alpha_m, \alpha'_x, \alpha'_m, \alpha_s \rangle \in \text{SV}(\phi)$, we have a transition

$$s_0 \xrightarrow{\varepsilon, \top, 0} q_{in}^{\alpha_m, sv} \in \delta.$$

  These transitions correspond to the choice of the first symbolic valuation. After this choice, the value of the counter has to be updated in order to satisfy α_m. This update is done by the component $\mathcal{A}_{\alpha_m, sv}$.

- For every $sv = \langle \alpha_x, \alpha_m, \alpha'_x, \alpha'_m, \alpha_s \rangle \in \text{SV}(\phi)$, we have a transition

$$q_{out}^{\alpha_m, sv} \xrightarrow{\varepsilon, \top, 0} q_{in}^{\alpha_x, sv} \in \delta.$$

When the control state is $q_{out}^{\alpha_m,sv}$ during any execution, we know that the counter satisfies α_m. The next step is to check that the constraint α_x is satisfied. So, we impose to continue the run by entering in the component $\mathcal{A}_{\alpha_x,sv}$.

- For every $sv = \langle \alpha_x, \alpha_m, \alpha'_x, \alpha'_m, \alpha_s \rangle \in \mathrm{SV}(\phi)$, we have a transition

$$q_{out}^{\alpha_x,sv} \xrightarrow{\varepsilon,\top,0} q_{in}^{\langle \alpha'_m,\alpha_s\rangle,sv} \in \delta.$$

When the current control state is $q_{out}^{\alpha_x,sv}$, the counter satisfies α_x and it is now time to update it according to the constraints α'_m and α_s, imposing constraints on the counter for the next position of symbolic model. Therefore, the only way for the run to continue is to enter further in the component $\mathcal{A}_{\langle \alpha'_m,\alpha_s\rangle,sv}$.

- For all pairs of symbolic valuations $sv_1 = \langle (\alpha_x)_1, (\alpha_m)_1, (\alpha'_x)_1, (\alpha'_m)_1, (\alpha_s)_1 \rangle \in \mathrm{SV}(\phi)$ and $sv_2 = \langle (\alpha_x)_2, (\alpha_m)_2, (\alpha'_x)_2, (\alpha'_m)_2, (\alpha_s)_2 \rangle \in \mathrm{SV}(\phi)$ such that
  - $(\alpha'_x)_1[\mathsf{X}x \leftarrow x] = (\alpha_x)_2$ and
  - $(\alpha'_m)_1[\mathsf{X}x \leftarrow x] = (\alpha_m)_2$

  we have a transition

$$q_{out}^{\langle (\alpha'_m)_1,(\alpha_s)_1\rangle,sv_1} \xrightarrow{sv_1,\top,0} q_{in}^{(\alpha_x)_2,sv_2} \in \delta.$$

At this step, the letter sv_1 can be read since all the verifications have been successful. The only way for the run to continue is to enter further in a component of the form $\mathcal{A}_{(\alpha_x)_2,sv_2}$. The new symbolic valuation sv_2 that is guessed has to agree with sv_1 on some constraints. Note that since $(\alpha'_m)_1$ have been verified by $\mathcal{A}_{\langle (\alpha'_m)_1,(\alpha_s)_1\rangle,sv_1}$ and $(\alpha'_m)_1[\mathsf{X}x \leftarrow x] = (\alpha_m)_2$, we can go directly to the input state in $\mathcal{A}_{(\alpha_x)_2,sv_2}$.

By construction of the components the following property is satisfied.

Lemma 4 *For all symbolic valuations $sv_1 = \langle (\alpha_x)_1, (\alpha_m)_1, (\alpha'_x)_1, (\alpha'_m)_1, (\alpha_s)_1 \rangle \in \mathrm{SV}(\phi)$ and $sv_2 = \langle (\alpha_x)_2, (\alpha_m)_2, (\alpha'_x)_2, (\alpha'_m)_2, (\alpha_s)_2 \rangle \in \mathrm{SV}(\phi)$, and for all $c, c' \in \mathbb{Z}$, the propositions below are equivalent:*

(I) $[x \leftarrow c] \models (\alpha_m)_1$ *and* $\langle q_{in}^{(\alpha_x)_1,sv_1}, c\rangle \xrightarrow{\varepsilon}{}^{*} \langle q_{in}^{\langle (\alpha'_m)_1,(\alpha_s)_1\rangle,sv_1}, c\rangle \xrightarrow{\varepsilon}{}^{*}$
$\langle q_{out}^{\langle (\alpha'_m)_1,(\alpha_s)_1\rangle,sv_1}, c'\rangle \xrightarrow{sv_1} \langle q_{in}^{(\alpha_x)_2,sv_2}, c'\rangle \xrightarrow{\varepsilon}{}^{*} \langle q_{out}^{(\alpha_x)_2,sv_2}, c'\rangle$,

(II) $[x \leftarrow c, \mathsf{X}x \leftarrow c'] \models sv_1$.

We can now state the main property: the set of satisfiable symbolic models (with respect to a formula ϕ) is recognized by the simple one-$\mathbb{Z}$-counter automaton $\mathcal{A}_{\mathrm{sat}}$.

Lemma 5 $\mathrm{L}(\mathcal{A}_{\mathrm{sat}})$ *is exactly the set of satisfiable symbolic models.*

Proof. Let $\langle \sigma, \rho \rangle$ be a satisfiable symbolic model and for all $i \in \mathbb{N}$, $\rho(i) = \langle (\alpha_x)_i, (\alpha_m)_i, (\alpha'_x)_i, (\alpha'_m)_i, (\alpha_s)_i \rangle$. So there is a $\mathrm{CLTL}^1_1(\mathrm{DL})$ model $\sigma' : \mathbb{N} \to \mathbb{Z}$ such that $\sigma' \models \rho$. We can show that $\langle \sigma, \rho \rangle \in \mathrm{L}(\mathcal{A}_{\mathrm{sat}})$ since there is an accepting run of the form

$$s_0 \xrightarrow{\varepsilon} \langle q_{in}^{(\alpha_m)_0, \rho(0)}, \sigma'(0) \rangle \xrightarrow{\varepsilon}{}^* \langle q_{in}^{(\alpha_x)_0, \rho(0)}, \sigma'(0) \rangle \xrightarrow{\varepsilon}{}^* \langle q_{in}^{\langle (\alpha'_m)_0, (\alpha_s)_0 \rangle, \rho(0)}, \sigma'(0) \rangle \xrightarrow{\varepsilon}{}^*$$
$$\langle q_{out}^{\langle (\alpha'_m)_0, (\alpha_s)_0 \rangle, \rho(0)}, \sigma'(1) \rangle \xrightarrow{\rho(0)} \langle q_{in}^{(\alpha_x)_1, \rho(1)}, \sigma'(1) \rangle \xrightarrow{\varepsilon}{}^* \langle q_{in}^{\langle (\alpha'_m)_1, (\alpha_s)_1 \rangle, \rho(1)}, \sigma'(1) \rangle \xrightarrow{\varepsilon}{}^*$$
$$\langle q_{out}^{\langle (\alpha'_m)_1, (\alpha_s)_1 \rangle, \rho(1)}, \sigma'(2) \rangle \xrightarrow{\rho(1)} \langle q_{in}^{(\alpha_x)_2, \rho(2)}, \sigma'(2) \rangle \dots$$

Now suppose that $\langle \sigma, \rho \rangle \in \mathrm{L}(\mathcal{A}_{\mathrm{sat}})$. By construction of the network of components in $\mathcal{A}_{\mathrm{sat}}$, there is an accepting run necessarily of the form

$$s_0 \xrightarrow{\varepsilon} \langle q_{in}^{(\alpha_m)_0, sv_0}, c_0 \rangle \xrightarrow{\varepsilon}{}^* \langle q_{in}^{(\alpha_x)_0, sv_0}, c_0 \rangle \xrightarrow{\varepsilon}{}^* \langle q_{in}^{\langle (\alpha'_m)_0, (\alpha_s)_0 \rangle, sv_0}, c_0 \rangle \xrightarrow{\varepsilon}{}^*$$
$$\langle q_{out}^{\langle (\alpha'_m)_0, (\alpha_s)_0 \rangle, sv_0}, c_1 \rangle \xrightarrow{\rho(0)} \langle q_{in}^{(\alpha_x)_1, sv_1}, c_1 \rangle \xrightarrow{\varepsilon}{}^* \langle q_{in}^{\langle (\alpha'_m)_1, (\alpha_s)_1 \rangle, sv_1}, c_1 \rangle \xrightarrow{\varepsilon}{}^*$$
$$\langle q_{out}^{\langle (\alpha'_m)_1, (\alpha_s)_1 \rangle, sv_1}, c_2 \rangle \xrightarrow{\rho(1)} \langle q_{in}^{(\alpha_x)_2, sv_1}, c_2 \rangle \dots$$

By construction of each component and by Lemma 4, we have $[x \leftarrow c_0] \models (\alpha_m)_0$ and for every $i \in \mathbb{N}$, $[x \leftarrow c_i, \mathsf{X}x \leftarrow c_{i+1}] \models (\alpha_x)_i \wedge (\alpha_m)_i \wedge (\alpha'_x)_i \wedge (\alpha'_m)_i \wedge (\alpha_s)_i$. So the model $\sigma' : \mathbb{N} \to \mathbb{Z}$ such that $\sigma'(i) = c_i$ satisfies ρ, i.e. $\sigma' \models \rho$. This means precisely that $\langle \sigma, \rho \rangle$ is a satisfiable symbolic model. □

4 Model-checking one-$\mathbb{Z}$-counter automata

A natural question is whether the decidability results of the previous section are optimal with respect to the fragments of Presburger arithmetic we have considered. Lemma 6 below states that we do not preserve decidability when extending the constraint language to quantifier-free Presburger arithmetic.

Lemma 6 *Satisfiability for* $\mathrm{CLTL}^1_1(\mathrm{QFP})$ *and model-checking* $\mathrm{CLTL}^1_1(\mathrm{QFP})$ *formulae over 1-variable* QFP*-automata are* Σ^1_1*-complete.*

This follows directly from [Min67, Section 14.2] about one counter machines with multiplication and division by constants (plus Σ^1_1-hardness from [AH94, Lemma 8]). Intuitively, the introduction of constraints of the form $ax + by = 0$ where $a, b \in \mathbb{Z}$ allow to encode a configuration $\langle q_i, c_1, c_2 \rangle$ of a Minsky machine, where q_i is the i^{th} control state, by the value $2^{c_1} 3^{c_2} 5^i$. Zero tests, incrementations and decrementations can be encoded with constraints of the form $x \equiv_2 0$, $x \equiv_3 0$, $\mathsf{X}x = 2x$ (incrementation of the first counter) etc.

As a consequence, the model-checking problem as it is defined in Section 2 is undecidable even for the fragment restricted to one variable and X-length one too. In this section, the strategy to regain decidability consists in restricting the class of models to one-$\mathbb{Z}$-counter automata. Model-checking becomes decidable (even in PSPACE) for LTL with *full quantifier-free Presburger constraints* restricted to *one variable* but with *no restriction on the* X*-length*. Indeed, the behavior of the unique counter in such automata is more constrained than in

QFP-automata. However, the examples of applications given at the beginning of the previous section still hold. This result is in sharp contrast with the results of the previous sections since the logic is much more expressive than CLTL(DL$^+$).

Let $\mathcal{A} = \langle Q_\mathcal{A}, \delta_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A} \rangle$ be a one-$\mathbb{Z}$-counter automaton whose set of updates is of the form $\mathsf{X}x = x + u$ with $u \in \{u_{\min}, u_{\min}+1, \ldots, u_{\max}\}$. Without loss of generality, we can assume that $u_{\min} = -u_{\max}$. Given a CLTL$_1^\omega$(QFP) formula ϕ such that $|\phi|_\mathsf{X} = l$, we consider the following syntactic resources:

- K is the lcm of the integers k such that $\equiv_k$ occurs in ϕ ($k \in \mathbb{N} \setminus \{0,1\}$),
- CONS is the set of constants $d \in \mathbb{Z}$ such that $\sum a_i \mathsf{X}^i x \sim d$ occurs in ϕ,
- $M = \max\{|d| : d \in \text{CONS}\}$ is the maximal absolute value of the elements of CONS,
- COEF is the set of constants $a_i \in \mathbb{Z} \setminus \{0\}$ such that $\sum a_i \mathsf{X}^i x \sim d$ occurs in ϕ.

Without any loss of generality, we can assume that $a_{\min} = -a_{\max}$ where $a_{\min}$ and $a_{\max}$ are respectively the minimal and the maximal element of COEF. For technical reasons (see the proof of Lemma 7), we define CONS$(\mathcal{A}, \phi)$ as the set $\{d_{\min}, \ldots, d_{\max}\}$ such that

$$d_{\max} = -d_{\min} = M + \frac{l(l+1)}{2} a_{\max} u_{\max}.$$

We define a symbolic valuation with respect to $\mathcal{A}$ and ϕ as an element of the set $\text{SV}(\mathcal{A}, \phi) = \text{C}_x \times \text{Mod}_x \times \text{C}^1_{\text{step}} \times \cdots \times \text{C}^l_{\text{step}}$ such that

- C_x is the set composed of the constraints $x < d_{\min}$, $d_{\max} < x$, and $x = d$ for every $d \in \text{CONS}(\mathcal{A}, \phi)$.
- Mod_x contains the constraint $x \equiv_K c$ for every $c \in \{0, \ldots, K-1\}$.
- C^i_{step} contains the constraint $\mathsf{X}^i x = \mathsf{X}^{i-1} x + u$ for every $u \in \{u_{\min}, \ldots, u_{\max}\}$.

Lemma 7 *Let ϕ be a* CLTL$_1^\omega$(QFP) *formula such that $|\phi|_\mathsf{X} = l$.*

(I) *For every path $v = \langle q_0, c_0 \rangle \ldots \langle q_l, c_l \rangle$ of $\mathcal{A}$ there is a unique symbolic valuation $sv(v) \in SV(\mathcal{A}, \phi)$ such that $v \models sv(v)$.*

(II) *For all valuations $v, v' : \{x, \mathsf{X}x, \ldots, \mathsf{X}^l x\} \to \mathbb{Z}$ such that $sv(v) = sv(v')$ ($sv(v) = sv(v') \in SV(\mathcal{A}, \phi)$) and for every atomic subformula α of ϕ, $v \models \alpha$ iff $v' \models \alpha$.*

In the statement **(II)** we had to suppose that the valuations v and v' correspond to some path in $\mathcal{A}$. Actually, this implies that the valuation v is such that for every $i \in \{0, \ldots, l-1\}$ we have $|v(\mathsf{X}^{i+1}x) - v(\mathsf{X}^i x)| \leq u_{\max}$ (idem for v').

Proof. **(I)** The argument is similar to the proof of Lemma 2(I) since the definition of symbolic valuations induces a partition of the set of valuations obtained from runs of $\mathcal{A}$.

(II) We proceed by induction on the structure of the formula. Assuming that for every $1 \leq i \leq l$, the constraint $\mathsf{X}^i x = \mathsf{X}^{i-1} x + u_i$ is in $sv(v)$, a term of the form $b_0 x + b_1 \mathsf{X} x + b_2 \mathsf{X}^2 x \cdots + b_l \mathsf{X}^l x$ (some b_i might be equal to zero) is equivalent to

$$(\sum_{j=0}^{j=l} b_j) x + \sum_{i=1}^{i=l} b_i (\sum_{j=1}^{j=i} u_j)$$

Note that by definition of $d_{\max} = -d_{\min} = M + \frac{l(l+1)}{2} a_{\max} u_{\max}$, for every atomic constraint of the form $b_0 x + b_1 \mathsf{X} x + b_2 \mathsf{X}^2 x \cdots + b_l \mathsf{X}^l x \sim d$, the value $d - \sum_{i=1}^{i=l} b_i (\sum_{j=1}^{j=i} u_j)$ is also in $\mathrm{CONS}(\mathcal{A}, \phi)$. Indeed, for every $i \in \{1, \ldots, l\}$ we have $u_{\min} \leq u_l \leq u_{\max}$ and $a_{\min} \leq b_i \leq a_{\max}$. We recall that by definition $a_{\min} = -a_{\max}$ and $u_{\min} = -u_{\max}$. Moreover, when $(\sum_{j=0}^{l} b_j)$ is nonzero, $\frac{\sum_{i=1}^{i=l} b_i (\sum_{j=1}^{j=i} u_j)}{(\sum_{j=0}^{l} b_j)}$ belongs to the range of constants in $\mathrm{CONS}(\mathcal{A}, \phi)$. Thus, we can reduce every atomic constraint $\alpha \in \phi$ without Boolean connective in the following way.

- A constraint of the form $\sum a_i \mathsf{X}^i x \sim d$ can be reduced to constraints of the form $x \sim d'$ with $d_{\min} \leq d' \leq d_{\max}$ and since $sv(v) = sv(v')$ we can conclude.
- A constraint of the form $\sum a_i \mathsf{X}^i x \equiv_k c$ can be reduced to a constraint of the form $ax \equiv_k c'$ with $c' \in \{0, \ldots, k-1\}$. The set of solutions is of the form $x \equiv_{k'} c''$ for $k' = gcd(a, k)$ (computable in polynomial time). As k' divides K, $x \equiv_K c_x$ implies $x \equiv_{k'} c'_x$ for a unique $c'_x \in \{0, \ldots, k'-1\}$. We can conclude by using the hypothesis $sv(v) = sv(v')$.

The step with Boolean connectives is by an easy verification. Note that the substitution can be done in polynomial time (the size of $d_{\max}$ is polynomial with respect to $|\phi|$) and checking that an atomic constraint is satisfied by a symbolic valuation can be checked in polynomial time. □

We define a symbolic satisfaction relation as in Section 3.3: $sv \models_{\mathrm{symb}} \alpha$ iff for every valuation $v : \{x, \mathsf{X}x, \ldots, \mathsf{X}^l x\} \to \mathbb{Z}$ (corresponding to a path with $l+1$ configurations in $\mathcal{A}$) such that $sv(v) = sv$ we have $v \models \alpha$. We can naturally extend this relation to symbolic valuation sequences.

Lemma 8 *Let ϕ be a $\mathrm{CLTL}_1^{\omega}(\mathrm{QFP})$ formula and $\mathcal{A}$ be a one-$\mathbb{Z}$-counter automaton. $\mathcal{A} \models \phi$ iff there exist a symbolic model $\rho \in SV(\mathcal{A}, \phi)$ such that $\rho \models_{\mathrm{symb}} \phi$ and an accepting run $\langle q_0, c_0 \rangle, \langle q_1, c_1 \rangle, \langle q_2, c_3 \rangle, \ldots$ of $\mathcal{A}$ such that $c_0, c_1, c_2 \ldots \models \rho$.*

Proof. In a nutshell, if ϕ is satisfied by a run then the symbolic valuation sequence $\rho : \mathbb{N} \to \mathrm{SV}(\mathcal{A}, \phi)$ such that for every position $i \in \mathbb{N}$ the symbolic valuation $\rho(i)$ is the abstraction of the subrun of length l starting at position i symbolically satisfies ϕ (see Lemma 7). Conversely, we can prove that if ρ symbolically satisfies ϕ, then every run of $\mathcal{A}$ satisfying ρ also satisfies ϕ. Indeed, for every i, the subrun of this run starting at position i satisfies $\rho(i)$ and therefore also satisfies the subformulae that are symbolically satisfied by $\rho(i)$. □

Again, we build an automaton $\mathcal{A}_\phi$ as the intersection $\mathcal{A}_{\text{symb}} \cap \mathcal{A}_{\text{sat}}$ such that $\mathcal{A}_{\text{symb}}$ recognizes the set of symbolic models satisfying ϕ and $\mathcal{A}_{\text{sat}}$ recognizes the set of symbolic models abstracting an accepting run of $\mathcal{A}$. The definition of $\mathcal{A}_{\text{symb}}$ and the synchronization between $\mathcal{A}_{\text{symb}}$ and $\mathcal{A}_{\text{sat}}$ are similar to what is done in Section 3.4 considering the alphabet $\Sigma = \text{SV}(\mathcal{A}, \phi)$ (with ε-transitions) and the corresponding relation $\models_{\text{symb}}$. Lemma 9 below is a pivot result for proving Theorem 4.

Lemma 9 *Given a formula ϕ and a one-$\mathbb{Z}$-counter automaton $\mathcal{A}$, one can build a simple one-$\mathbb{Z}$-counter automaton $\mathcal{A}_{\text{sat}}$ over the alphabet $\Sigma = 2^{\text{PROP}} \times \text{SV}(\mathcal{A}, \phi)$ such that $\text{L}(\mathcal{A}_{\text{sat}})$ is the set of satisfiable symbolic models w.r.t ϕ and $\mathcal{A}$. Moreover, $\mathcal{A}_\phi$ can be effectively built from ϕ and $\mathcal{A}$ in polynomial space thanks to the way $\mathcal{A}_{\text{sat}}$ is defined.*

Proof. Let $\mathcal{A} = \langle Q_\mathcal{A}, \delta_\mathcal{A}, I_\mathcal{A}, F_\mathcal{A} \rangle$ be a one $\mathbb{Z}$-counter automaton and ϕ be a $\text{CLTL}_1^\omega(\text{QFP})$ formula such that $|\phi|_{\mathsf{X}} = l$. The automaton $\mathcal{A}_{\text{sat}}$ is defined over the alphabet $\Sigma = SV(\mathcal{A}, \phi)$. As in Section 3.5, the construction is modular. We denote by $Q_\mathcal{A} \times Q$ the set of states of $\mathcal{A}_{\text{sat}}$ where Q is a set of auxiliary states used in the construction of components similar to components of Section 3.5. For every $sv = \langle \alpha_x, \alpha_m, \alpha_s^1, \ldots, \alpha_s^l \rangle$ and $q_a \in Q_\mathcal{A}$, we define the following components:

- $\mathcal{A}^{q_a}_{\alpha_m, sv}$ such that for every $c \in \mathbb{Z}$, $\langle q_a, q_{in}^{\alpha_m, sv}, 0 \rangle \rightarrow^* \langle q_a, q_{out}^{\alpha_m, sv}, c \rangle$ iff $[x \leftarrow c] \models \alpha_m$.
- $\mathcal{A}^{q_a}_{\alpha_x, sv}$ such that for every $c \in \mathbb{Z}$, $\langle q_a, q_{in}^{\alpha_x, sv}, c \rangle \rightarrow^* \langle q_a, q_{out}^{\alpha_x, sv}, c' \rangle$ iff $c = c'$ and $[x \leftarrow c] \models \alpha_x$.
- We also need to define another kind of components that update the counter. For every $d \in \text{CONS}(\mathcal{A}, \phi)$, the component $\mathcal{A}^{q_a}_{\alpha_s^1, sv}$ is such that for every $c \in \mathbb{Z}$, $\langle q_a, q_{in}^{\alpha_s^1, sv}, c \rangle \rightarrow^* \langle q_a, q_{out}^{\alpha_s^1, sv}, c' \rangle$ iff $[x \leftarrow c, \mathsf{X}x \leftarrow c'] \models \alpha_s^1$.

These different components are connected as follows:

- The set of initial states is composed of the states of the form $\langle q_0, q_{in}^{\alpha_m, sv} \rangle$ for every $sv = \langle \alpha_x, \alpha_m, \alpha_s^1, \ldots, \alpha_s^l \rangle \in \text{SV}(\phi, \mathcal{A})$ and $q_0 \in I_\mathcal{A}$.
- For every $sv = \langle \alpha_x, \alpha_m, \alpha_s^1, \ldots, \alpha_s^l \rangle \in \text{SV}(\phi, \mathcal{A})$ and $q_0 \in I_\mathcal{A}$, we have $\langle q_0, q_{out}^{\alpha_m, sv} \rangle \xrightarrow{\varepsilon, \top, 0} \langle q_0, q_{in}^{\alpha_x, sv} \rangle \in \delta$.
- For all $sv = \langle \alpha_x, \alpha_m, \alpha_s^1, \ldots, \alpha_s^l \rangle \in \text{SV}(\phi, \mathcal{A})$, $\langle q_a, q_{out}^{\alpha_x, sv} \rangle \xrightarrow{\varepsilon, \top, 0} \langle q_a, q_{in}^{\alpha_s^1, sv} \rangle$ belongs to δ.
- For all $sv_1 = \langle (\alpha_x)_1, (\alpha_m)_1, (\alpha_s^1)_1, \ldots, (\alpha_s^l)_1 \rangle \in \text{SV}(\phi, \mathcal{A})$, $sv_2 = \langle (\alpha_x)_2, (\alpha_m)_2, (\alpha_s^1)_2, \ldots, (\alpha_s^l)_2 \rangle \in \text{SV}(\phi, \mathcal{A})$ and $q_a, q'_a \in Q_\mathcal{A}$ such that
  - $(\alpha_s^1)_1$ is equal to $\mathsf{X}x = x + d$,
  - $q_a \xrightarrow{d} q'_a \in \delta_\mathcal{A}$,

– $(\alpha^1_s)_1 \wedge (\alpha^1_m)_1 \Rightarrow (\alpha^1_m)_2$ is valid in Presburger arithmetic (checkable in polynomial-time),
– for every $1 \leq i \leq l-1$, $(\alpha^i_s)_2 = (\alpha^{i+1}_s)_1[\mathsf{X}^{i+1}x \leftarrow \mathsf{X}^i x, \mathsf{X}^i x \leftarrow \mathsf{X}^{i-1}x]$,

we have $\langle q_a, q_{out}^{\alpha^1_s, sv_1}\rangle \xrightarrow{sv_1, \top, 0} \langle q'_a, q_{in}^{(\alpha_x)_2, sv_2}\rangle \in \delta$.

- the set of final states is $\{\langle q_f, q_{out}^{\alpha_x, sv}\rangle | q_f \in F_{\mathcal{A}}\}$.

By construction, for every path in $\mathcal{A}_{\text{sat}}$ of the form $\cdots \langle q_i, q_{in}^{(\alpha_x)_i, sv_i}, c_i\rangle \rightarrow \cdots \xrightarrow{sv_i} \langle q_{i+1}, q_{in}^{(\alpha_x)_{i+1}, sv_{i+1}}, c_{i+1}\rangle \rightarrow \cdots \xrightarrow{sv_{i+1}} \cdots \xrightarrow{sv_{i+l-1}} \langle q_{i+l}, q_{in}^{(\alpha_x)_{i+l}, sv_{i+l}}, c_{i+l}\rangle \cdots$, we have $c_i, \ldots, c_{i+l} \models sv_i$. □

We can conclude about the complexity of the model-checking problem.

Theorem 4 *Model-checking* $\text{CLTL}^{\omega}_1(\text{QFP})$ *formulae over one-*$\mathbb{Z}$*-counter automata is* PSPACE*-complete.*

Proof. The PSPACE upper bound is corollary of Lemma 9 and Theorem 6. The PSPACE-hardness can be obtained by using the result of [DS02] about the model-checking of LTL restricted to one propositional variable. □

This result can also be generalized to LTL extensions with MSO-definable operators (denoted by $\text{xCLTL}^1_1(\text{QFP})$ below).

Theorem 5 *Model-checking* $\text{xCLTL}^1_1(\text{QFP})$ *formulae over one-*$\mathbb{Z}$*-counter automata is* PSPACE*-complete.*

5 Complexity of the nonemptiness problem for simple one-$\mathbb{Z}$-counter automata

This last technical section is dedicated to prove the result used in Theorems 3 and 4 stating that the nonemptiness problem for simple one-$\mathbb{Z}$-counter automata with alphabet can be decided in NLOGSPACE. This is an interesting result for its own sake even though it is instrumental to establish our PSPACE upper bounds. The proof is divided in two main parts. We first define several reductions to restrict the class of automata we have to handle. Then, we show that the existence of accepting runs is equivalent to the existence of finite runs of polynomial length. Even when it is not explicitly mentioned, in the rest of this section the counter automata are always simple counter automata (the updates of the counter are among $\{-1, 0, 1\}$).

5.1 Reduction to one-$\mathbb{N}$-counter automata

We first show how to reduce in logarithmic space the nonemptiness problem for simple one-$\mathbb{Z}$-counter automata with alphabet to the existence of an accepting run in one-$\mathbb{N}$-counter automata (without alphabet). We proceed with several

reductions. The first step is to eliminate the alphabet and the ε-transitions. The main difficulty is to avoid accepting runs where ε-transitions are fired forever after a certain position.

Lemma 10 *Checking whether* $\mathrm{L}(\mathcal{A})$ *is non-empty for one-*$\mathbb{Z}$*-counter automata* $\mathcal{A}$ *with alphabet can be reduced in logarithmic space to the existence of an accepting run in one-*$\mathbb{Z}$*-counter automata (without alphabet).*

Proof. We show that given a one-$\mathbb{Z}$-counter automaton $\mathcal{A}$ with alphabet Σ, it is possible to compute in logarithmic space a one-$\mathbb{Z}$-counter automaton $\mathcal{A}'$ without alphabet such that $\mathrm{L}(\mathcal{A}) \neq \emptyset$ iff $\mathcal{A}'$ has an accepting run. The idea of the proof consists in defining $\mathcal{A}'$ as two copies of $\mathcal{A}$. The copy that contains the final states is reachable only by reading a non-epsilon transition.

Given the one-$\mathbb{Z}$-counter automaton $\mathcal{A} = \langle Q, \delta, I, F\rangle$, the one-$\mathbb{Z}$-counter automaton $\mathcal{A}' = \langle Q', \delta', I', F'\rangle$ is defined as follows:

- $Q' = Q \times \{1, 2\}$, $I' = I \times \{1\}$, $F' = F \times \{2\}$,
- The transitions in δ' are defined from transitions in δ:
  - for every $q \xrightarrow{a,t,u} q' \in \delta$ such that $a \in \Sigma$ ($u \in \{-1, 0, +1\}$) we have $\langle q, i\rangle \xrightarrow{t,u} \langle q', 2\rangle$ for $i \in \{1, 2\}$,
  - for every $q \xrightarrow{\varepsilon,t,u} q' \in \delta$ we have
    - if $q \in F$ then $\langle q, i\rangle \xrightarrow{t,u} \langle q', 1\rangle \in \delta'$ for $i \in \{1, 2\}$,
    - if $q \notin F$ then $\langle q, i\rangle \xrightarrow{t,u} \langle q', i\rangle \in \delta'$ for $i \in \{1, 2\}$.

It is easy to show that $\mathrm{L}(\mathcal{A}) \neq \emptyset$ iff $\mathcal{A}'$ has an accepting run. □

We now show that the existence of an accepting run for one-$\mathbb{Z}$-counter automata can be reduced in logarithmic space to the same problem for the simpler class of one-$\mathbb{N}$-counter automata.

Lemma 11 *The existence of an accepting run in one-*$\mathbb{Z}$*-counter automata can be reduced in logarithmic space to the existence of an accepting run in one-*$\mathbb{N}$*-counter automata.*

Proof. We show that given a one-$\mathbb{Z}$-counter automaton $\mathcal{A}$, one can compute in logarithmic space a one-$\mathbb{N}$-counter automaton $\mathcal{A}'$ such that $\mathcal{A}$ has an accepting run iff $\mathcal{A}'$ has an accepting run. The idea of the proof is simply to define $\mathcal{A}'$ as two copies of $\mathcal{A}$, one copy when the counter is positive (see the states in $Q \times \{+\}$ below) and another one when the counter is negative (see the states in $Q \times \{-\}$ below). The move from one copy to another is done when the counter is equal to zero and the sign of the counter changes. These two different copies allow to simulate tests of the form $x < 0$ and $x > 0$.

Given the one-$\mathbb{Z}$-counter automaton $\mathcal{A} = \langle Q, \delta, I, F\rangle$, the one-$\mathbb{N}$-counter automaton $\mathcal{A}' = \langle Q', \delta', I', F'\rangle$ is defined as follows:

- $Q' = Q \times \{+, -\}$, $I' = I \times \{+\}$, $F' = F \times \{+, -\}$,

- The transitions in δ' are defined from transitions in δ:
  - $q \xrightarrow{t,0} q' \in \delta$ and $t \in \{=, \neq\}$ imply $\langle q, - \rangle \xrightarrow{t,0} \langle q', - \rangle \in \delta'$ and $\langle q, + \rangle \xrightarrow{t,0} \langle q', + \rangle \in \delta'$,
  - $q \xrightarrow{>,0} q' \in \delta$ implies $\langle q, + \rangle \xrightarrow{\neq,0} \langle q', + \rangle \in \delta'$,
  - $q \xrightarrow{<,0} q' \in \delta$ implies $\langle q, - \rangle \xrightarrow{\neq,0} \langle q', - \rangle \in \delta'$,
  - $q \xrightarrow{=,-1} q' \in \delta$ implies $\langle q, + \rangle \xrightarrow{=,+1} \langle q', - \rangle \in \delta'$ and $\langle q, - \rangle \xrightarrow{=,+1} \langle q', - \rangle \in \delta'$,
  - $q \xrightarrow{>,-1} q' \in \delta$ implies $\langle q, + \rangle \xrightarrow{\neq,-1} \langle q', + \rangle \in \delta'$,
  - $q \xrightarrow{<,-1} q' \in \delta$ implies $\langle q, - \rangle \xrightarrow{\neq,+1} \langle q', - \rangle \in \delta'$,
  - $q \xrightarrow{\top,-1} q' \in \delta$ implies $\langle q, + \rangle \xrightarrow{\neq,-1} \langle q', + \rangle \in \delta'$, $\langle q, + \rangle \xrightarrow{=,+1} \langle q', - \rangle \in \delta'$ and $\langle q, - \rangle \xrightarrow{\top,+1} \langle q', - \rangle \in \delta'$,
  - $q \xrightarrow{\neq,-1} q' \in \delta$ implies $\langle q, + \rangle \xrightarrow{\neq,-1} \langle q', + \rangle \in \delta'$ and $\langle q, - \rangle \xrightarrow{\neq,+1} \langle q', - \rangle \in \delta'$,
  - the remaining cases with incrementations by one are treated in a similar fashion.

It is easy to show that $\mathcal{A}$ has an accepting run iff $\mathcal{A}'$ has an accepting run. □

Finally, we can simplify the problem once more, by restricting the class of one-$\mathbb{N}$-counter automata we have to consider.

Lemma 12 *The existence of some accepting run in one-$\mathbb{N}$-counter automata can be reduced in logarithmic space to the existence of some accepting run in one-$\mathbb{N}$-counter automata with no test of the form $x \neq 0$.*

The simple proof is based on the replacement of each transition of the form $q \xrightarrow{\neq,u} q'$ by the sequence $q \xrightarrow{\top,-1} q_1 \xrightarrow{\top,+1} q_2 \xrightarrow{\top,u} q'$ where q_1 and q_2 are new states.

5.2 Zero-test free paths

We recall that the composition of logarithmic space reductions is a logarithmic space reduction and the composition of a logarithmic space reduction with a nondeterministic logarithmic space test can be done in nondeterministic logarithmic space. For these reasons, *we are restricting ourselves in the following to one-$\mathbb{N}$-counter automata with the tests $x = 0$ and $\top$*, that we can obtain using the reductions we have just described.

We say that an infinite path $w : \mathbb{N} \to (Q \times \mathbb{N})$ for $\mathcal{A}$ is *zero-test free* iff for every $i \in \mathbb{N}$ (with $w(i) = \langle q_i, c_i \rangle$), there is a transition $q_i \xrightarrow{\top,u_i} q_{i+1}$ and $c_{i+1} = c_i + u_i$. Finite zero-test free paths are defined similarly. In other words, the absence of zero-tests implies that the only "test" is $\top$. We define below several properties on zero-test free paths that will be useful in the following.

Lemma 13 *Let $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a finite path such that $\langle q_2, c_2\rangle, \ldots, \langle q_n, c_n\rangle$ is zero-test free. There exists a finite path $w' = \langle q'_1, c'_1\rangle, \ldots, \langle q'_{n'}, c'_{n'}\rangle$ such that $q'_1 = q_1$, $q'_{n'} = q_n$, $c'_1 = c_1$ and $n' \leq |Q|^2 + 2|Q| + 1$.*

Proof. First, remark that if there exist two indices $i, j > 1$ such that $q_i = q_j$ and $c_i \geq c_j$ then the subpath $\langle q_i, c_i\rangle, \ldots, \langle q_j, c_j\rangle$ can be deleted from w, still providing a path since $\langle q_2, c_2\rangle, \ldots, \langle q_n, c_n\rangle$ is zero-test free. The values of the counter have to be updated adequately but since the new values are greater, every transition can be fired. Note that this remark is not true if the path contains zero-tests. So we can assume without any loss of generality that if there exist $1 < i < j \leq n$ such that $q_i = q_j$, then $c_i < c_j$.

If there are no $1 < i < j \leq n$ such that $q_i = q_j$, then $n \leq |Q|$ and so we can take $w' = w$.

Otherwise, there exists a unique pair of indices $\langle i_0, j_0\rangle$ such that

- $i_0 < j_0$,
- $q_{i_0} = q_{j_0}$,
- $q_1, \ldots, q_{i_0}, \ldots, q_{j_0-1}$ is a sequence of distinct control states.

These indices correspond to the first control state that is repeated in the path. By hypothesis, we must have $N = c_{j_0} - c_{i_0} > 0$. Moreover, it is easy to check that $i_0 \leq |Q|$ and $j_0 - i_0 \leq |Q|$ because the control states before the position j_0 are distinct.

Since $\langle q_{j_0}, c_{j_0}\rangle, \ldots, \langle q_n, c_n\rangle$ is a zero-test free path of $\mathcal{A}$, there is a path $q'_1 \ldots q'_m$ of length at most $|Q|$ such that $q'_1 = q_{j_0}$, $q'_m = q_n$ in the directed graph $G = \langle Q, \{\langle r, s\rangle : \langle r, \top, u, s\rangle \in \delta\}\rangle$ of reachable control states without zero-test transitions. Thus, we can use the following observation:

If there is a path in G from a state q to a state q' whose length is c then one can reach the control state q' from any configuration $\langle q, c'\rangle$ such that $c' \geq c$ in $\mathcal{A}$.

Indeed, one can fire all the transition corresponding to this path in $\mathcal{A}$ (the worst case is c decrements by one) since it is zero-test free.

So we build the finite path w' as the path composed of the following successive subpaths:

- $\langle q_1, c_1\rangle, \ldots, \langle q_{i_0-1}, c_{i_0-1}\rangle$, $\langle q_{i_0}, c_{i_0}\rangle, \ldots, \langle q_{j_0-1}, c_{j_0-1}\rangle$,
- $\langle q_{i_0}, c_{i_0} + N\rangle, \ldots, \langle q_{j_0-1}, c_{j_0-1} + N\rangle$,
- $\langle q_{i_0}, c_{i_0} + 2N\rangle, \ldots, \langle q_{j_0-1}, c_{j_0-1} + 2N\rangle$,
- $\ldots$
- $\langle q_{i_0}, c_{i_0} + (|Q| - 1)N\rangle, \ldots, \langle q_{j_0-1}, c_{j_0-1} + (|Q| - 1)N\rangle$,
- $\langle q_{i_0}, c_{i_0} + |Q|N\rangle, \langle q'_2, y_2\rangle, \ldots, \langle q'_m, y_m\rangle$.

This path is well-defined since $q_{i_0} = q_{j_0}$, $c_{j_0} = c_{i_0} + N$ and the value of the counter is great enough to fire the last part of the path (the part corresponding to the path in G). The restriction of w' to control states is presented below:

$$\overbrace{q_1 \dots q_{i_0-1}}^{\text{length } \leq |Q|}(\overbrace{q_{i_0} q_{i_0+1} \dots q_{j_0-1} q_{j_0}}^{\text{length } \leq |Q|})^{|Q|} \overbrace{q'_2 \dots q'_m}^{\text{length } \leq |Q|}.$$

Obviously, the length of this path is bounded by $|Q|^2 + 2|Q| + 1$ □

It is also possible to easily adapt the above proof by requiring in the statement of Lemma 13 that c'_m is greater than some constant K. In the proof, it is sufficient to repeat the sequence of transitions between q_{i_0} and q_{j_0}, $|Q| + K$ times.

We now treat the case of infinite zero-test free path where a final state occurs infinitely often. Our goal is to extract a cycle whose initial state is this final state. Indeed, such a cycle allows to build a run satisfying the Büchi acceptance condition.

Lemma 14 *Let $w : \mathbb{N} \to (Q \times \mathbb{N})$ be an infinite zero-test free path where there is a final state $q_f \in F$ such that $\{i \in \mathbb{N} : w(i)$ is of the form $\langle q_f, j\rangle\}$ is infinite. There exist $i \in \mathbb{N}$ and a finite path $w' = \langle q_1, c'_1\rangle, \dots, \langle q_n, c'_n\rangle$ such that $n > 1$, $\langle q_1, c'_1\rangle = w(i)$ with $q_1 = q_f$, $q_n = q_f$, $c'_n \geq c'_1$ and $n \leq 2|Q|^2 + 3|Q|$.*

Proof. Let $w(i) = \langle q_i, c_i\rangle$ for every $i \in \mathbb{N}$. Since $<$ is a well-quasi ordering over the elements of $\mathbb{N}$, there exist i_0 and j_0 such that:

- $i_0 < j_0$,
- $q_{i_0} = q_{j_0} = q_f$ and
- $c_{i_0} \leq c_{j_0}$.

The finite path $w'' = w(i_0), \dots, w(j_0)$ satisfies all the properties of w' except that $j_0 - i_0$ can be greater than the desired bound. At this point, we proceed similarly to the proof of Lemma 13. Without any loss of generality, we can assume that if there exist $i_0 < i < j \leq j_0$ such that $q_i = q_j$ then $c_i < c_j$ (otherwise, the subpath can be deleted).

If there are no $i_0 < i < j \leq j_0$ such that $q_i = q_j$ then the length of the path is smaller than $|Q|$ and we can take $w' = w''$.

Otherwise, there is a unique pair $\langle i_1, j_1\rangle$ such that

- $i_0 \leq i_1 < j_1 \leq j_0$,
- $q_{i_1} = q_{j_1}$,
- $q_{i_0}, \dots, q_{i_1}, \dots, q_{j_1-1}$ is a sequence of distinct control states.

By hypothesis, we have $N = c_{j_1} - c_{i_1} > 0$ and one can check that $i_1 \leq |Q|$ and $j_1 - i_1 \leq |Q|$. Since $\langle q_{j_1}, c_{j_1}\rangle, \dots, \langle q_{j_0}, c_{j_0}\rangle$ is a zero-test free path, there is a path

$q'_1 \dots q'_m$ of length at most $|Q|$ in the graph $G = \langle Q, \{\langle r, s\rangle : \langle r, \top, u, s\rangle \in \delta\}\rangle$ such that $q'_1 = q_{j_0}$, $q'_m = q_n$. The same observation about the graph G as in Lemma 13 allows us to build the path w' composed of the following successive subpaths:

- $\langle q_{i_0}, c_{i_0}\rangle, \dots, \langle q_{i_1-1}, c_{i_1-1}\rangle$, $\langle q_{i_1}, c_{i_1}\rangle, \dots, \langle q_{j_1-1}, c_{j_1-1}\rangle$,
- $\langle q_{i_1}, c_{i_1} + N\rangle, \dots, \langle q_{j_1-1}, c_{j_1-1} + N\rangle$,
- $\langle q_{i_1}, c_{i_1} + 2N\rangle, \dots, \langle q_{j_1-1}, c_{j_1-1} + 2N\rangle$,
- $\dots$
- $\langle q_{i_1}, c_{i_1} + 2|Q|N\rangle, \dots, \langle q_{j_1}, c_{j_1-1} + 2|Q|N\rangle$,
- $\langle q_{i_1}, c_{i_1} + (2|Q| + 1)N\rangle, \langle q'_2, c'_2\rangle, \dots, \langle q'_m, c'_m\rangle$.

This path is well-defined since $q_{i_1} = q_{j_1}$, $c_{j_1} = c_{i_1} + N$ and the value of the counter is large enough to fire the last part corresponding to the path in G. The restriction of w' to control states is presented below:

$$\overbrace{q_1 \dots q_{i_0-1}}^{\text{length } \leq |Q|}(\overbrace{q_{i_0} q_{i_0+1} \dots q_{j_0-1} q_{j_0}}^{\text{length } \leq |Q|})^{2|Q|+1} \overbrace{q'_2 \dots q'_m}^{\text{length } \leq |Q|}.$$

The length of this path is clearly bounded by $2|Q|^2 + 3|Q|$. Moreover, $c'_m > c_{i_0}$ since the first and last part of the path decrement the counter at most by $2|Q|$ whereas the repetition of the center loop increments it by $2|Q|N$. □

5.3 Paths with zero-tests

It remains to consider paths containing zero-tests. We first state a result that is a consequence of [LLT04, Lemma 42].

Lemma 15 *[LLT04] Let $w = \langle q_1, c_1\rangle, \dots, \langle q_n, c_n\rangle$ be a finite path of $\mathcal{A}$ such that $\langle q_2, c_2\rangle, \dots, \langle q_n, c_n\rangle$ is zero-test free and $c_n = c_1 = 0$. There exists a finite path $w' = \langle q'_1, c'_1\rangle, \dots, \langle q'_{n'}, c'_{n'}\rangle$ such that $q'_1 = q_1$, $q'_{n'} = q_n$, $c'_{n'} = c'_1 = 0$ and for every $i \in \{1, \dots, n'\}$, $c'_i \leq |Q|^3 + |Q|^2$.*

Proof. The statement of the lemma is a variant of [LLT04, Lemma 42] and the proof below is a slight adaptation of its proof.

First, we show the following property.

(⋆) Let $w = \langle q_1, c_1\rangle, \dots, \langle q_n, c_n\rangle$ be a path such that $c_n = c_1 + |Q|$. Then, there exists a subpath w' of w such that $w' = \langle q, c\rangle \to^* \langle q, c + d\rangle$ for some $0 < d \leq |Q|$ and $q \in Q$.

Indeed, there is a sequence $1 = i_1 < i_2 < i_3 < \dots < i_{|Q|+1} \leq n$ such that $c_{i_{j+1}} = c_{i_j} + 1$ for $1 \leq j \leq |Q|$. Consequently, there are $l < l'$ such that $q_{i_l} = q_{i_{l'}}$ and $\langle q_{i_l}, c_{i_l}\rangle \to^* \langle q_{i_l}, c_{i_l} + d\rangle$ with $d = c_{i_{l'}} - c_{i_l}$ and $0 < d = l' - l \leq |Q|$. In a similar way, one can show the property below.

($\star\star$) Let $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a path such that $c_n + |Q| = c_1$. Then, there exists a subpath w' of w such that $w' = \langle q, c+d\rangle \rightarrow^* \langle q, c\rangle$ for some $0 < d \leq |Q|$ and $q \in Q$.

Let $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a path of $\mathcal{A}$ such that $\langle q_2, c_2\rangle, \ldots, \langle q_n, c_n\rangle$ is zero-test free and $c_n = c_1 = 0$. Let γ be the value $|Q|$ (γ will be used also in the proof of Lemma 17). The overweight of the path w, denoted by $ov(w)$, is defined as the sum $\sum_{i=1}^{n} max(c_i - (|Q| \times \gamma^2 + |Q|^2), 0)$. The proof consists in transforming w into a path w' with identical first and last configurations and $ov(w') < ov(w)$, whenever there is $1 < j < n$ such that $c_j > |Q| \times \gamma^2 + |Q|^2$.

Given a finite path $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ with $ov(w) > 0$, let $\langle L, L'\rangle$ be the unique pair of indices such that L is the smallest index such that $c_L = |Q| \times \gamma^2 + |Q|^2$ and $c_{L+1} = |Q| \times \gamma^2 + |Q|^2 + 1$, and L' is the smallest index such that $c_{L'-1} = |Q| \times \gamma^2 + |Q|^2 + 1$ and $c_{L'} = |Q| \times \gamma^2 + |Q|^2$.

The proof is by induction on $ov(w)$. The base case with $ov(w) = 0$ is obvious. Now, suppose that $ov(w) > 0$. There exist $\beta < L \leq L' < \beta'$ such that for $\beta \leq i \leq \beta'$, $c_i \geq |Q|^2$ and $c_\beta = c_{\beta'} = |Q|^2$. Figure 4 illustrates the behaviour of the counter. Consequently, there exist $\beta = \beta_1 < \beta_2 < \ldots < \beta_{\gamma^2+1} = L$ and

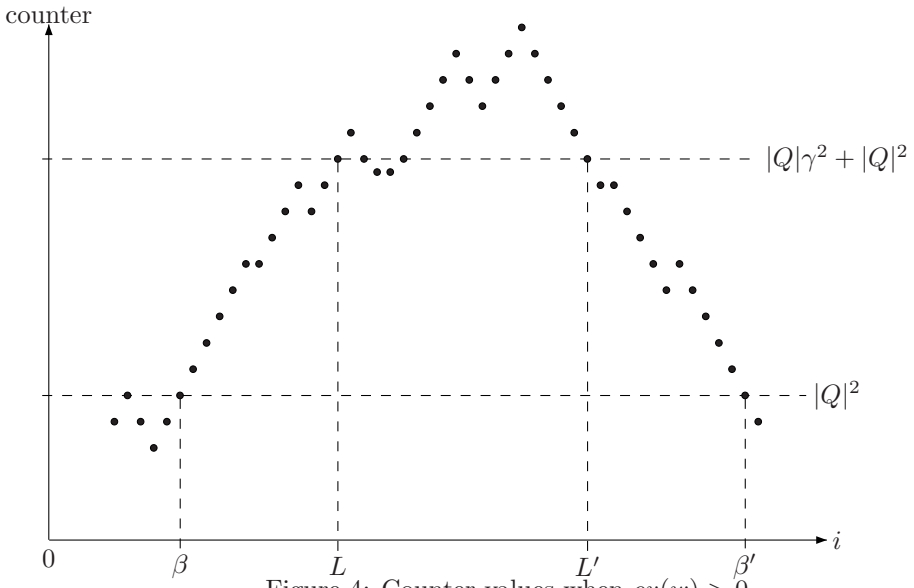


Figure 4: Counter values when $ov(w) > 0$

$L' = \beta'_1 < \beta'_2 < \ldots < \beta'_{\gamma^2+1} = \beta'$ such that for $1 \leq j \leq \gamma^2$, $c_{\beta_{j+1}} = c_{\beta_j} + |Q|$ and $c_{\beta'_j} = c_{\beta'_{j+1}} + |Q|$. By using ($\star$) and ($\star\star$), we get that for $1 \leq j \leq \gamma^2$,

- there exists an ascending subpath $\langle q, c\rangle \rightarrow^* \langle q, c + d_j\rangle$ of the path $\langle q_{\beta_j}, c_{\beta_j}\rangle \rightarrow^* \langle q_{\beta_{j+1}}, c_{\beta_{j+1}}\rangle$ for some $0 < d_j \leq |Q|$,
- there exists a descending subpath $\langle q, c + d'_j\rangle \rightarrow^* \langle q, c\rangle$ of the path $\langle q_{\beta'_j}, c_{\beta'_j}\rangle \rightarrow^* \langle q_{\beta'_{j+1}}, c_{\beta'_{j+1}}\rangle$ for some $0 < d'_j \leq |Q|$,

There exists $d \in \{d_1, \ldots, d_{\gamma^2}\}$ such that d is equal to at least γ values d_j and there exists $d' \in \{d'_1, \ldots, d'_{\gamma^2}\}$ such that d' is equal to at least γ values d'_j.

(Remove) The path w' is obtained from w by removing d' ascending subpaths with increasing effect d and by removing d descending subpaths with decreasing effect d'.

It is easy to see that $ov(w') < ov(w)$. Observe that removing such subpaths causes no problem since $0 < d \times d' \leq |Q|^2$ and $c_\beta = c'_\beta = |Q|^2$. □

In Lemma 16 below, the initial path is not necessarily zero-test free. This is a direct consequence of the result stated above.

Lemma 16 *Let $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a finite path of $\mathcal{A}$ such that $c_n = c_1 = 0$. There exists a finite path $w' = \langle q'_1, c'_1\rangle, \ldots, \langle q'_{n'}, c'_{n'}\rangle$ satisfying $q'_1 = q_1$, $q'_{n'} = q_n$, $c'_{n'} = c'_1 = 0$ and $n' \leq |Q|^5 + |Q|^4$.*

Proof. Let $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a finite path of $\mathcal{A}$ such that $c_n = c_1 = 0$. Let $1 = i_1 < i_2 < \ldots < i_m = n$ be such that $c_{i_1} = c_{i_2} = \cdots = c_{i_m} = 0$ and no other configuration has the counter equal to zero. Note that subpaths of the form $\langle q, 0\rangle \cdots \langle q, 0\rangle$ can be withdrawn from w. Hence, without loss of generality, we can assume that $m \leq |Q|$.

For every $1 \leq j < m$, the subpath $\langle q_{i_j}, c_{i_j}\rangle \cdots \langle q_{i_{j+1}}, c_{i_{j+1}}\rangle$ satisfies the conditions of Lemma 15. So for every $1 \leq j < m$, there exists a path w_j such that $w_j = \langle q_{i_j}, 0\rangle \cdots \langle q_{i_{j+1}}, 0\rangle$ with all the values of the counter smaller or equal to $|Q|^3 + |Q|^2$. Since subpaths of the form $\langle q, c\rangle, \ldots, \langle q, c\rangle$ can also be withdrawn from w_j (see remark in Lemma 13), we can assume that the length of w_j is smaller than $|Q| \times (|Q|^3 + |Q|^2)$. By concatenating the paths w_1, …, w_{m-1} we can easily obtained a path $w' = \langle q'_1, c'_1\rangle, \ldots, \langle q'_{n'}, c'_{n'}\rangle$ satisfying $q'_1 = q_1$, $q'_{n'} = q_n$, $c'_{n'} = c'_1 = 0$ and $n' \leq (m-1) \times |Q| \times (|Q|^3 + |Q|^2) \leq |Q|^5 + |Q|^4$. □

We finally consider accepting runs such that zero-tests are repeated infinitely often. As stated in Lemma 14, we want to extract a cycle with a final state whose length is bounded by some polynomial.

Lemma 17 *Let $q_f \in F$ and $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a finite path such that $\langle q_2, c_2\rangle, \ldots, \langle q_n, c_n\rangle$ is zero-test free, $c_n = c_1 = 0$ and $q_f \in \{q_1, \ldots, q_n\}$. There is a finite path $w' = \langle q'_1, c'_1\rangle, \ldots, \langle q'_{n'}, c'_{n'}\rangle$ such that $q'_1 = q_1$, $q'_{n'} = q_n$, $c'_{n'} = c'_1 = 0$, $q_f \in \{q'_1, \ldots, q'_n\}$ and for every $i \in \{1, \ldots, n'\}$, $c'_i \leq |Q|^3 + 3|Q|^2 + |Q|$.*

Proof. The proof is similar to the proof of Lemma 15 except that when transforming a path, we require that q_f remains in the path w'. In order to get the present proof, from the proof of Lemma 15, it is sufficient to take $\gamma = |Q| + 1$ and then to observe that in the part (Remove), there are $\gamma = |Q| + 1$ subpaths for d and d' respectively, we can guarantee there is a way to remove subpaths while preserving the presence of q_f. □

Lemma 18 *Let $q_f \in F$ and $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a finite path such that $c_n = c_1 = 0$ and $q_f \in \{q_1, \ldots, q_n\}$. There is a finite path w' of the form $\langle q'_1, c'_1\rangle, \ldots, \langle q'_{n'}, c'_{n'}\rangle$ satisfying $q'_1 = q_1$, $q'_{n'} = q_n$, $c'_{n'} = c'_1 = 0$, $q_f \in \{q'_1, \ldots, q'_{n'}\}$ and $n' \leq 2|Q|^5 + 3|Q|^4 + 3|Q|^3 + |Q|^2$.*

Proof. Let $q_f \in F$ and $w = \langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ be a finite path such that $c_n = c_1 = 0$ and $q_f \in \{q_1, \ldots, q_n\}$. Let $1 = i_1 < i_2 < \ldots < i_m = n$ be such that $c_{i_1} = \cdots = c_{i_m} = 0$ and no other configuration has the counter equal to zero. There is $k \in \{1, \ldots, m-1\}$ such that a configuration of the form $\langle q_f, c\rangle$ occurs in the subpath $\langle q_{i_k}, c_{i_k}\rangle, \ldots, \langle q_{i_{k+1}}, c_{i_{k+1}}\rangle$. The paths $\langle q_1, c_1\rangle, \ldots, \langle q_{i_{k-1}}, c_{i_{k-1}}\rangle$ and $\langle q_{i_{k+1}}, c_{i_{k+1}}\rangle, \ldots, \langle q_{i_m}, c_{i_m}\rangle$ satisfy the hypothesis of Lemma 16. So, there exist paths $w_1 = \langle q_1, 0\rangle, \ldots, \langle q_{i_{k-1}}, 0\rangle$ and $w_3 = \langle q_{i_{k+1}}, 0\rangle, \ldots, \langle q_{i_m}, 0\rangle$ of length at most $|Q|^5 + |Q|^4$.

Moreover, the path $\langle q_{i_k}, c_{i_k}\rangle, \ldots, \langle q_{i_{k+1}}, c_{i_{k+1}}\rangle$ satisfies the hypotheses of Lemma 17 and therefore there is a path $w_2 = \langle q_{i_k}, 0\rangle, \ldots, \langle q_{i_{k+1}}, 0\rangle$ of length at most $|Q| \times (|Q|^3 + 3|Q|^2 + |Q|)$ that contains a configuration of the form $\langle q_f, c'\rangle$. By concatenating w_1, w_2 and w_3, we obtain a path w' satisfying the desired properties. □

5.4 Size bound for accepting runs

We now have all the elements to establish a bound on the size of a witness for the existence of an accepting run in one-$\mathbb{N}$-counter automata without tests of the form $x \neq 0$.

Lemma 19 *Let $P_1(x) = x^5 + x^4 + x^3 + \frac{9}{2}x^2 + 5x + 1$ and $P_2(x) = 3x^5 + 4x^4 + 3x^3 + x^2$ be two polynomials. For every one-$\mathbb{N}$-counter automaton $\mathcal{A}$ with no test of the form $x \neq 0$, $\mathcal{A}$ has an accepting run iff one of the propositions below holds true:*

- **($\star$)** *There exist a finite path $\langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ with $n \leq P_1(|Q|)$ and $i' < n$ such that $q_{i'} = q_n \in F$, $c_{i'} \leq c_n$, $q_1 \in I$, $c_1 = 0$ and $\langle q_{i'}, c_{i'}\rangle, \ldots, \langle q_n, c_n\rangle$ is a finite zero-test free path.*
- **($\star\star$)** *There is a finite path $\langle q_1, c_1\rangle, \ldots, \langle q_n, c_n\rangle$ with $n \leq P_2(|Q|)$ such that there exists $i' < n$ with $q_{i'} = q_n$, $c_{i'} = c_n = 0$, $\{q_{i'}, \ldots, q_n\} \cap F \neq \emptyset$, $q_1 \in I$, $c_1 = 0$ and $\langle q_{i'}, c_{i'}\rangle, \ldots, \langle q_n, c_n\rangle$ is a finite path that is not zero-test free.*

Proof. If either ($\star$) or ($\star\star$) holds true, then an accepting run can be easily defined by repeating infinitely the finite path $\langle q_{i'}, x_{i'}\rangle, \ldots, \langle q_n, x_n\rangle$.

Conversely, let us suppose that there exits an accepting run $w : \mathbb{N} \to (Q \times \mathbb{N})$ of $\mathcal{A}$ performing finitely many zero-tests. We pose $w(i) = \langle q_i, c_i\rangle$ for every $i \in \mathbb{N}$. By hypothesis, there exist $0 < i_0 \leq i_1 < i_2$ such that

- $c_{i_0 - 1} = 0$ and no zero-tests are performed after the position i_0 (we omit the particular case when w is zero-test free),
- $q_{i_1} = q_{i_2} \in F$ and $c_{i_1} \leq c_{i_2}$.

By Lemma 16, there is a path $w_1 = \langle q_0, 0\rangle \cdots \langle q_{i_0-1}, 0\rangle$ whose length is bounded by $|Q|^5 + |Q|^4$. Moreover, Lemma 14 implies that there exists a path $w_3 = \langle q_{i_1}, c\rangle, \ldots, \langle q_{i_2}, c'\rangle$ such that $c \leq c'$ whose length is bounded by $2|Q|^2 + 3|Q|$. Since one step in the path can decrement the counter by at most one unit, the i^{th} configuration $\langle q'', c''\rangle$ in w_3 satisfies $c'' \geq c - i$. Hence, the sequence $w_3' = \langle q_{i_1}, |Q|^2 + \ulcorner \frac{3}{2}|Q|\urcorner\rangle, \ldots, \langle q_{i_2}, c' + (|Q|^2 + \ulcorner \frac{3}{2}|Q|\urcorner - c)\rangle$ obtained from w_3 by adding $(|Q|^2 + \ulcorner \frac{3}{2}|Q|\urcorner - c)$ is also a valid finite path (the value of the counter is never negative).

It remains to concatenate these two paths. By using arguments similar to the proof of Lemma 13 (see the paragraph after its proof), one can show that there exists a path $w_2 = \langle q_{i_0-1}, 0\rangle \ldots \langle q_{i_1}, x\rangle$ such that $x \geq |Q|^2 + \ulcorner \frac{3}{2}|Q|\urcorner$ whose length is bounded by

$$(2 + |Q|^2 + \ulcorner \frac{3}{2}|Q|\urcorner) \times |Q| + |Q|^2.$$

Since $x \geq |Q|^2 + \ulcorner \frac{3}{2}|Q|\urcorner$, it will be possible to reproduce the sequence of transitions between q_{i_1} and q_{i_2}. One just need to repeat the loop a sufficient amount of times. By concatenating w_1, w_2 and w_3' we get a path satisfying $(\star)$.

Now we suppose that every accepting run of the automaton $\mathcal{A}$ contains infinitely many zero-tests. We consider such an accepting run $w : \mathbb{N} \to (Q \times \mathbb{N})$ and we still note $w(i) = \langle q_i, c_i\rangle$ for every $i \in \mathbb{N}$. There exists in w two indices $0 < i_0 < i_1$ such that

- $q_{i_0} = q_{i_1}$ and $c_{i_0} = c_{i_1} = 0$,
- a final state $q_f \in F$ occurs in the subpath $w' = \langle q_{i_0+1}, c_{i_0+1}\rangle, \ldots, \langle q_{i_1}, c_{i_1}\rangle$.

Such i_0, i_1 exist because both final states and zero-tests are repeated infinitely often in w. By Lemma 16, there is a path $w_1 = \langle q_0, 0\rangle, \ldots, \langle q_{i_0}, 0\rangle$ of length smaller than $|Q|^5 + |Q|^4$. Moreover, Lemma 18 implies that there is a path $w_2 = \langle q_1', c_1'\rangle, \ldots, \langle q_j', c_j'\rangle$ whose length is bounded by $2|Q|^5 + 3|Q|^4 + 3|Q|^3 + |Q|^2$ such that

- $q_1' = q_j' = q_{i_0}$, $c_1' = c_j' = 0$,
- there is $j' \in \{1, \ldots, n\}$ such that $q_{j'}' \in F$.

Observe that w_2 is not zero-test free, otherwise the infinite path described below is accepting, which contradicts our hypothesis since it is zero-test free:

$$\langle q_0, 0\rangle, \ldots, \langle q_{i_0}, 0\rangle, (\langle q_2', c_2'\rangle, \ldots, \langle q_{j'}', c_{j'}'\rangle, \ldots, \langle q_j', 0\rangle)^\omega.$$

By concatenating w_1 and w_2 we get a path w' satisfying $(\star\star)$. □

So we can establish the main result of this section.

Theorem 6 *Checking whether* $\mathrm{L}(\mathcal{A})$ *is non-empty for one-*$\mathbb{Z}$*-counter automata* $\mathcal{A}$ *with alphabet is* NLOGSPACE*-complete.*

Proof. Since the composition of logarithmic space reductions is a logarithmic space reduction, this problem can be reduced in logarithmic space to the existence of accepting runs for one-$\mathbb{N}$-counter automata with no test of the form $x \neq 0$. Lemma 19 states that the existence of accepting runs for such one-$\mathbb{N}$-counter automata can be tested in non-deterministic logarithmic space, by distinguishing the accepting run performing finitely and infinitely many zero-tests. Indeed, verifying that a path of polynomial length verifies the conditions of Lemma 19 can be done in logarithmic space.

The composition of a logarithmic space reduction with a nondeterministic logarithmic space test can be done in nondeterministic logarithmic space. Hence, the problem of checking whether $\mathrm{L}(\mathcal{A})$ is non-empty is in NLOGSPACE. The NLOGSPACE-hardness can be obtained by reducing the graph accessibility problem. □

Note that, if we extend one-$\mathbb{Z}$-counter automata by allowing updates of the form $\mathsf{X}x = x + u$ for some $u \in \mathbb{Z}$ (u encoded in binary), then the problem of the existence of accepting runs becomes NP-hard. This can be easily shown by a reduction from the NP-complete problem SUBSET-SUM; a very similar proof is given in [RY86, Theorem 3.5] for the boundedness problem of vector addition systems with states restricted to one counter. Moreover, the problem is in PSPACE since this problem can be reduced in polynomial space to our original problem. However, we ignore the precise optimal complexity of this extended problem. Recently, an NP upper bound has been established for one-$\mathbb{N}$-counter automata with succinct updates [HKOW09].

6 Conclusion

We have studied the decidability and complexity of LTL with Presburger constraints by restricting the number of variables, the X-length of the formulae and the set of constraints. Figure 5 summarizes the complexity of satisfiability, model-checking over DL-automata and model-checking over k-$\mathbb{Z}$-counter automata for most LTL-like specification languages considered herein. As a technical lemma, we have proved that the nonemptiness problem for one-counter Büchi automata taking values in $\mathbb{Z}$ and allowing zero-tests and sign-tests is NLOGSPACE-complete. This result is interesting for its own sake.

These results design new decidable subproblems of undecidable problems from [CC00, DPK03] and establish new decidability boundaries for Presburger LTL. Apart from the completion of our classification, the most positive results concern one-counter automata/nets, see applications in [CR04, WTT04, LLT05]. The PSPACE upper bound for model-checking one-$\mathbb{Z}$-counter automata over $\mathrm{CLTL}^{\omega}_{1}(\mathrm{QFP})$, or even over its linear μ-calculus extension, refines results from [FWW97, BEM97, Wal01, Ser06] that concern more general systems and languages.

	MC (DL)	SAT	MC (CA)
$\mathrm{CLTL}_3^1(\mathrm{DL})$	undec. [Min67]	undec. [CC00]	undec. [Min67]
$\mathrm{CLTL}_2^\omega(\mathrm{DL})$	undec. [Min67]	undec. [DD07]	undec. [Min67]
$\mathrm{CLTL}_1^2(\mathrm{DL})$	undec. Cor. 1	undec. Theo. 1	PSPACE-c. Theo. 4 + [DS02]
$\mathrm{CLTL}_2^1(\mathrm{DL})$	undec. [Min67]	undec. Theo. 1	undec. [Min67]
$\mathrm{CLTL}_1^1(\mathrm{DL} \text{ or } \mathrm{DL}^+)$	PSPACE-c. Theo. 3	PSPACE-c. Theo. 3	PSPACE-c. Theo. 4 + [DS02]
$\mathrm{CLTL}_1^1(\mathrm{QFP})$	undec. Lem. 6	undec. Lem. 6	PSPACE-c. Theo. 4 + [DS02]
$\mathrm{CLTL}_1^\omega(\mathrm{QFP})$	undec. Lem. 6	undec. Lem. 6	PSPACE-c. Theo. 4 + [DS02]

Figure 5: Summary

Acknowledgments. We would like to thank the anonymous referees for numerous helpful remarks and suggestions allowing us to significantly improve this article.

References

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

[AH94] R. Alur and Th. Henzinger. A really temporal logic. *Journal of the Association for Computing Machinery*, 41(1):181–204, 1994.

[BB90] J. Berstel and L. Boasson. Context-free languages. In *Handbook of Theoretical Computer Science, Volume B, Formal models and semantics*, pages 59–102. Elsevier, 1990.

[BBH+06] A. Bouajjani, M. Bozga, P. Habermehl, R. Iosif, P. Moro, and T. Vojnar. Programs with lists are counter automata. In *CAV'06*, volume 4144 of *Lecture Notes in Computer Science*, pages 517–531. Springer, 2006.

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

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer, 2nd edition, 1988.

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

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model checking. In *CONCUR'97*, volume 1243 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 1997.

[BFLS06] S. Bardin, A. Finkel, E. Lozes, and A. Sangnier. From pointer systems to counter systems using shape analysis. In *AVIS'06*, 2006.

[BH96] A. Bouajjani and P. Habermehl. Constrained properties, semilinear sets, and Petri nets. In *CONCUR'96*, volume 1119 of *Lecture Notes in Computer Science*, pages 481–497. Springer, 1996.

[BHM03] A. Bouajjani, P. Habermehl, and R. Mayr. Automatic verification of recursive procedures with one integer parameter. *Theoretical Computer Science*, 295(1-3):85–106, 2003.

[Boi98] B. Boigelot. *Symbolic methods for exploring infinite state spaces*. PhD thesis, Université de Liège, 1998.

[CC00] H. Comon and V. Cortier. Flatness is not a weakness. In *CSL'00*, volume 1862 of *Lecture Notes in Computer Science*, pages 262–276. Springer, 2000.

[CGP00] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 2000.

[CJ98] H. Comon and Y. Jurski. Multiple counters automata, safety analysis and Presburger arithmetic. In *CAV'98*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 1998.

[CR04] C. Chitic and D. Rosu. On validation of XML streams using finite state machines. In *WebDB*, pages 85–90, 2004.

[DD07] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. *Information and Computation*, 205(3):380–415, 2007.

[DG07] S. Demri and R. Gascon. The effects of bounding syntactic resources on Presburger LTL (extended abstract). In *TIME'07*, pages 94–104. IEEE, 2007.

[DG08] S. Demri and R. Gascon. Verification of qualitative Z constraints. *Theoretical Computer Science*, 409(1):24–40, 2008.

[DPK03] Z. Dang, P. San Pietro, and R. Kemmerer. Presburger liveness verification of discrete timed automata. *Theoretical Computer Science*, 299:413–438, 2003.

[DS02] S. Demri and Ph. Schnoebelen. The complexity of propositional linear temporal logics in simple cases. *Information and Computation*, 174(1):84–103, 2002.

[EKS03] J. Esparza, A. Kučera, and S. Schwoon. Model checking LTL with regular valuations for pushdown systems. *Information and Computation*, 186(2):355–376, 2003.

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

[FWW97] A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems (extended abstract). In *INFINITY'97*, volume 9 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1997.

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

[GPSS80] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *7th Annual ACM Symposium on Principles of Programming Languages*, pages 163–173. ACM Press, 1980.

[Hal95] J. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 75(2):361–372, 1995.

[HKOW09] C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *CONCUR'09*, Lecture Notes in Computer Science. Springer, 2009. To appear.

[Iba78] O. Ibarra. Reversal-bounded multicounter machines and their decision problems. *Journal of the Association for Computing Machinery*, 25(1):116–133, 1978.

[ISD+00] O. Ibarra, J. Su, Z. Dang, T. Bultan, and A. Kemmerer. Counter machines: Decidable properties and applications to verification problems. In *MFCS'00*, volume 1893 of *Lecture Notes in Computer Science*, pages 426–435. Springer, 2000.

[JKMS04] P. Jančar, A. Kučera, F. Moller, and Z. Sawa. DP lower bounds for equivalence-checking and model-checking of one-counter automata. *Information and Computation*, 188(1):1–19, 2004.

[Kuč00] A. Kučera. Efficient verification algorithms for one-counter processes. In *ICALP'00*, volume 1853 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2000.

[KVW00] O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the Association for Computing Machinery*, 47(2):312–360, 2000.

[LLT04] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. Research Report LSV-04-16, Laboratoire Spécification et Vérification, ENS Cachan, France, November 2004. 69 pages.

[LLT05] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In *RTA'05*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2005.

[LMS04] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking timed automata with one or two clocks. In *CONCUR'04*, volume 3170 of *Lecture Notes in Computer Science*, pages 387–401. Springer, 2004.

[Lut04] C. Lutz. NEXPTIME-complete description logics with concrete domains. *ACM Transactions on Computational Logic*, 5(4):669–705, 2004.

[LW05] S. Lasota and I. Walukiewicz. Alternating timed automata. In *FOSSACS'05*, volume 3441 of *Lecture Notes in Computer Science*. Springer, 2005.

[Min67] M. Minsky. *Computation, Finite and Infinite Machines*. Prentice Hall, 1967.

[OW05] J. Ouaknine and J. Worrell. On the decidability of metric temporal logic. In *LICS'05*, pages 188–197. IEEE, 2005.

[Rog67] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill Book Company, 1967.

[RY86] L. Rosier and H.-C. Yen. A multiparameter analysis of the boundedness problem for vector addition systems. *Journal of Computer and System Sciences*, 32:105–135, 1986.

[SC85] A. Sistla and E. Clarke. The complexity of propositional linear temporal logic. *Journal of the Association for Computing Machinery*, 32(3):733–749, 1985.

[Ser06] O. Serre. Parity games played on transition graphs of one-counter processes. In *FOSSACS'06*, volume 3921 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2006.

[Var88] M. Vardi. A temporal fixpoint calculus. In *POPL'88*, pages 250–259. ACM, 1988.

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

[Wal01] I. Walukiewicz. Pushdown processes: games and model-checking. *Information and Computation*, 164(2):234–263, 2001.

[Wol83] P. Wolper. Temporal logic can be more expressive. *Information and Computation*, 56:72–99, 1983.

[WTT04] M. Wakatsuki, K. Teraguchi, and E. Tomita. Polynomial time identification of strict deterministic restricted one-counter automata in some class from positive data. In *ICGI'04*, volume 3264 of *Lecture Notes in Artificial Intelligence*, pages 260–272. Springer, 2004.