When Model-Checking Freeze LTL over Counter Machines Becomes Decidable⋆

Stéphane Demri[1] and Arnaud Sangnier[2]

[1] LSV, ENS Cachan, CNRS, INRIA Saclay IdF, France
[2] Dipartimento di Informatica, Università di Torino, Italy

Abstract. We study the decidability status of model-checking freeze LTL over various subclasses of counter machines for which the reachability problem is known to be decidable (reversal-bounded counter machines, vector additions systems with states, flat counter machines, one-counter machines). In freeze LTL, a register can store a counter value and at some future position an equality test can be done between a register and a counter value. Herein, we complete an earlier work started on one-counter machines by considering other subclasses of counter machines, and especially the class of reversal-bounded counter machines. This gives us the opportuniy to provide a systematic classification that distinguishes determinism vs. nondeterminism and we consider subclasses of formulae by restricting the set of atomic formulae or/and the polarity of the occurrences of the freeze operators, leading to the flat fragment.

1 Introduction

Counter machines. Counter machines are ubiquitous computational models that provide a natural class of infinite-state transition systems, suitable for modeling various applications such as broadcast protocols [17], time granularities [10] and programs with pointer variables [6], to quote a few examples. They are also known to be closely related to data logics for which decision procedures can be designed relying on those for counter machines, see e.g. remarkable examples in [5,3]. When dealing with this class of models, most interesting reachability problems are undecidable but subclasses leading to decidability have been designed including reversal-bounded counter machines [25], one-counter machines [26], flat counter machines [18] and vector addition systems with states (see e.g. [32]).

Model-checking with Freeze LTL. In order to verify properties on counter machines, we aim at comparing counter values and we shall use the so-called *freeze* operator. The freeze quantifier in real-time logics has been introduced in the logic TPTL, see e.g. [1]. The formula $x \cdot \phi(x)$ binds the variable x to the time t of the current state: $x \cdot \phi(x)$ is semantically equivalent to $\phi(t)$. This variable-binding mechanism, quite natural when rephrased in first-order logic, is present in various logical formalisms including for example hybrid logics [22,2], freeze LTL [14] and predicate λ-abstraction [30]. Freeze

LTL is a powerful extension of LTL that allows to store counter values in registers. Infinitary satisfiability restricted to one register is already undecidable [14] just as model-checking for nondeterministic one-counter machines [15], which is quite unexpected since one-counter machines seem to be harmless operational models. Moreover, there is some hope that model-checking happens to be more tractable than satisfiability since more constraints are requested on models viewed as runs.

Our contribution. We carry on with the quest started in [15] to determine which classes of counter machines admit decidable model-checking with freeze LTL. In the paper, we consider the above-mentioned classes of counter machines for which the reachability problem is decidable. We provide an exhaustive analysis completing [15]; some results are obtained by adequately adapting known results to our framework or by designing simple reductions. However, at each position, we may have to deal with more than one counter values. Our main technical contributions allow us to establish the following results with a special focus on reversal-bounded counter machines.

- Model-checking freeze LTL (written $\mathrm{MC}^{\omega}(\mathrm{LTL}^{\downarrow})$) over deterministic vector addition systems with states and deterministic reversal-bounded counter machines is decidable (see Corollary 11). However, $\mathrm{MC}^{\omega}(\mathrm{LTL}^{\downarrow})$ over reversal-bounded counter machines is undecidable, even when restricted to one register (see Theorem 7).
- $\mathrm{MC}^{\omega}(\mathrm{LTL}^{\downarrow})$ restricted to flat formulae over reversal-bounded counter machines is decidable (see Corollary 17) as well as the restriction to positively flat formulae over one-counter machines (see Theorem 18), partly by taking advantage of recent results about parameterized one-counter machines [23].

A complete summary can be found in Section 8. As a nice by-product of the classification we made, we show a tight relationship between reachability problems for parameterized counter machines and model-checking counter machines over the flat fragment of freeze LTL (see Section 7.2). Besides, we believe that the principles underlying our undecidability proof for $\mathrm{MC}^{\omega}(\mathrm{LTL}^{\downarrow})$ over reversal-bounded counter machines could be reused for other problems on such counter machines.

Because of lack of space, omitted proofs can be found in [16].

2   Standard Classes of Counter Machines

In this section, we recall standard definitions about various classes of counter machines. We write $\mathbb{N}$ [resp. $\mathbb{Z}$] for the set of natural numbers [resp. integers]. Given a dimension $n \geq 1$ and $k \in \mathbb{Z}$, we write $\mathbf{k}$ to denote the vector with all values equal to k and $\mathbf{e}_i$ to denote the unit vector for $i \in \{1, \ldots, n\}$. We recall that a semilinear set of $\mathbb{N}^n$ is a finite union of linear sets. We often refer to Presburger arithmetic which consists of first-order logic over the structure $\langle \mathbb{N}, 0, \leq, + \rangle$ (and more generally over $\langle \mathbb{Z}, 0, \leq, + \rangle$) [31]. It is known that a subset of $\mathbb{N}^k$ is semilinear if and only if it is definable by a formula in Presburger arithmetic [20].

2.1   Counter Machines

A *counter machine* M is defined as a tuple $\langle n, Q, \Delta, q_0 \rangle$ where $n \geq 1$ is the *dimension* of M, Q is a finite set of *control states*, $\Delta \subseteq Q \times G \times A \times Q$ is a finite set of *transitions*

where $G = \{\texttt{zero}, \texttt{true}\}^n$ is the finite set of *guards* and $A = \{-1, 0, 1\}^n$ is the finite set of *actions*, and $q_0 \in Q$ is the *initial* control state. Given a counter machine M, we introduce the *transition system* $TS(M) = \langle Q \times \mathbb{N}^n, \rightarrow \rangle$ where $Q \times \mathbb{N}^n$ is the set of *configurations* and $\rightarrow \subseteq (Q \times \mathbb{N}^n) \times (Q \times \mathbb{N}^n)$ is the *transition relation*: for $\langle q, \mathbf{v} \rangle, \langle q', \mathbf{v}' \rangle \in Q \times \mathbb{N}^n$, we have $\langle q, \mathbf{v} \rangle \rightarrow \langle q', \mathbf{v}' \rangle \overset{\text{def}}{\Leftrightarrow}$ there exists a transition $t = \langle q, \mathbf{g}, \mathbf{a}, q' \rangle \in \Delta$ such that: $\mathbf{v}' = \mathbf{v} + \mathbf{a}$ and for $1 \leq c \leq n$, $\mathbf{g}(c) = \texttt{zero}$ implies $\mathbf{v}(c) = 0$. We write $\overset{*}{\rightarrow}$ to denote the reflexive and transitive closure of $\rightarrow$ and the reachability set of M is $\text{Reach}(M) \overset{\text{def}}{=} \{\langle q, \mathbf{v} \rangle \mid \langle q_0, \mathbf{0} \rangle \overset{*}{\rightarrow} \langle q, \mathbf{v} \rangle\}$. Observe that this reachability set implicitly depends on the initial configuration $\langle q_0, \mathbf{0} \rangle$: this is all what we need in the sequel. A finite (resp. infinite) *run* in $TS(M)$ is a finite (resp. infinite) sequence $\rho = \langle q_0, \mathbf{0} \rangle \rightarrow \langle q_1, \mathbf{v}_1 \rangle \rightarrow \dots$. A counter machine M is *deterministic* (also known as *single-path*) whenever for each $\langle q, \mathbf{v} \rangle \in \text{Reach}(M)$, there is at most one configuration $\langle q', \mathbf{v}' \rangle$ such that $\langle q, \mathbf{v} \rangle \rightarrow \langle q', \mathbf{v}' \rangle$. In the sequel, we shall use Minsky machines that form a special class of deterministic 2-counter machines.

We present below two types of decision problems when $\mathcal{C}$ is a class of counter machines. The *reachability problem* for the class $\mathcal{C}$ is defined as follows: given a machine $M \in \mathcal{C}$ and a configuration $\langle q, \mathbf{v} \rangle$, does $\langle q_0, \mathbf{0} \rangle \overset{*}{\rightarrow} \langle q, \mathbf{v} \rangle$? Similarly, the *generalized repeated reachability problem* for the class $\mathcal{C}$ is defined as follows: given a counter machine $M \in \mathcal{C}$ and N sets $F_1, \dots, F_N$ of control states, is there a run of M such that for $1 \leq i \leq N$, there is a control state in F_i that is repeated infinitely often?

1CM. One-counter machines are naturally defined as counter machines of dimension one. Various logical formalisms have been introduced to specify the behavior of one-counter machines, including Freeze LTL [15] and EF logic [21]. When one-counter machines are enriched by a finite alphabet (so that transitions are labelled), the universality problem is undecidable [26], witnessing that this simple operational model can lead to natural undecidable problems.

VASS. Vector addition systems with states (a.k.a. VASS) are known to be equivalent to Petri nets, see e.g. [32], and they correspond to counter machines without zero-tests, i.e. each guard has no component equal to $\texttt{zero}$. To be precise, we are a bit less liberal than the usual definition since we only consider actions in $\{-1, 0, 1\}^n$ (instead of $\mathbb{Z}^n$) but this does not make a real difference for all the developments made in this paper.

Flat counter machines. A directed graph $G = \langle V, E \rangle$ (with $V \subseteq E \times E$) is said to be *flat* whenever each vertex belongs to at most one cycle (path for which the initial and final vertices coincide). A counter machine $\langle n, Q, \Delta, q_0 \rangle$ is *flat* whenever (1) between two control states there is at most one transition and (2) the directed graph $\langle Q, \{\langle q, q' \rangle \in Q^2 : \langle q, \mathbf{g}, \mathbf{a}, q' \rangle \in \Delta\} \rangle$ is flat. Reachability problems have been considered for flat counter machines in [4,18]; for instance it is proved that flat counter machines have an effectively computable semilinear set [4,18], see also [8].

2.2 Reversal-Bounded Counter Machines

The class of *reversal-bounded* counter machines has been introduced in [25] by considering the following restriction: each counter performs only a bounded number of alternations between increasing and decreasing mode. This class of counter machines is particularly interesting because it has been shown that each reversal-bounded counter machine has a semilinear reachability set which can be effectively computed. We present

now a more general class of counter machines proposed in [19]. Given a bound $b \in \mathbb{N}$, we consider the number of alternations between increasing and decreasing mode when the value of the considered counter is above b; if for each counter this number of alternations is bounded by a constant $k \in \mathbb{N}$, we say that the counter machine is k-reversal-b-bounded. From now on, we say that a counter machine M is *reversal-bounded* if there exist $k, b \in \mathbb{N}$ such that M is k-reversal-b-bounded and in the sequel, when reversal-bounded counter machines are part of the instances of some decision problems, we assume that they come with their k and b. As mentioned in [19], the above-defined class of reversal-bounded counter machines contains those defined in [25] and it also contains the counter machines for which the set of reachable configurations is finite. To make the distinction, we will call the machines introduced in [25] *Ibarra* reversal-bounded counter machines.

In [19], the authors prove that the reachability problem is decidable for reversal-bounded counter machines (in fact their reachability set is also an effectively computable semilinear set) and in [33] it is proved that the generalized repeated reachability problem for this class of machines is also decidable when considering only one set of control states to be repeated infinitely often. The proof of this last result relies on the fact that this problem is decidable for Ibarra reversal-bounded counter machines [11]. Note that we can easily reduce the generalized reachability problem with $N \geq 1$ sets of control states to its restriction to only one set (the same way the emptiness problem for generalized Bûchi automata can be reduced to the emptiness problem for Bûchi automata).

Theorem 1. *The generalized repeated reachability problem for reversal-bounded counter machines is decidable.*

3   LTL with the Freeze Operator

In this section, we present a variant of temporal logic LTL with registers (also known as Freeze LTL) in order to reason about runs from counter machines. In [15], LTL with registers is used to specify properties about one-counter machines. The datum stored in a register is the current counter value and equality tests are performed between a register value and the current counter value. When dealing with counter machines, a register can store the value of a counter c and test it later against the value of counter c' with possibly $c \neq c'$. Below, we present different ways to restrict the equality tests between registers and counters.

Given a finite set Q of control states (possibly empty) and $n \geq 1$, the formulae of the logic $\mathrm{LTL}^{\downarrow}[Q, n]$ are defined as follows:

$$\phi ::= q \ \mid \ \uparrow_r^c \ \mid \ \neg\phi \ \mid \ \phi \wedge \phi \ \mid \ \phi \vee \phi \ \mid \ \phi\mathrm{U}\phi \ \mid \ \phi\mathrm{R}\phi \ \mid \ \mathrm{X}\phi \ \mid \ \downarrow_r^c \phi$$

where $q \in Q$, $c \in \{1, \ldots, n\}$ and $r \in (\mathbb{N} \setminus \{0\})$. Intuitively, the modality $\downarrow_r^c$ is used to store the value of the counter c into the register r; the atomic formula $\uparrow_r^c$ holds true if the value stored in the register r is equal to the current value of the counter c. An occurrence of $\uparrow_r^c$ within the scope of some freeze quantifier $\downarrow_r^c$ is bound by it; otherwise it is free. A sentence is a formula with no free occurrence of any $\uparrow_r^c$.

Models of $\text{LTL}^\downarrow[Q,n]$ are runs of transition systems from counter machines of dimension n and with a set of control states containing Q. Given a counter machine $\langle n, Q', \Delta, q_0 \rangle$ with $Q \subseteq Q'$ and a run ρ, we write $|\rho|$ to denote its *length* in $\omega + 1$ and the ith configuration ($0 \leq i < |\rho|$) is denoted by $\langle q_i, \mathbf{v}_i \rangle$. A *counter valuation* f is a finite partial map from $\mathbb{N} \setminus \{0\}$ to $\mathbb{N}$. Note that whenever $f(r)$ is undefined, the atomic formula $\uparrow_r^c$ is interpreted as false. Given a run ρ and a position $0 \leq i < |\rho|$, the satisfaction relation $\models$ is defined as follows (Boolean clauses are omitted):

$$
\begin{aligned}
\rho, i \models_f q \;&\overset{\text{def}}{\Leftrightarrow}\; q_i = q \\
\rho, i \models_f \uparrow_r^c \;&\overset{\text{def}}{\Leftrightarrow}\; r \in \mathrm{dom}(f) \text{ and } f(r) = \mathbf{v}_i(c) \\
\rho, i \models_f \mathtt{X}\phi \;&\overset{\text{def}}{\Leftrightarrow}\; i+1 < |\rho| \text{ and } \rho, i+1 \models_f \phi \\
\rho, i \models_f \phi_1 \mathtt{U} \phi_2 \;&\overset{\text{def}}{\Leftrightarrow}\; \text{for some } i \leq j < |\rho|,\; \rho, j \models_f \phi_2 \\
&\qquad \text{and for all } i \leq j' < j, \text{ we have } \rho, j' \models_f \phi_1 \\
\rho, i \models_f \phi_1 \mathtt{R} \phi_2 \;&\overset{\text{def}}{\Leftrightarrow}\; \text{for all } i \leq j < |\rho|,\; \rho, j \models_f \phi_2 \\
&\qquad \text{or for some } i \leq j < |\rho|,\; \rho, j \models_f \phi_1 \\
&\qquad \text{and for all } i \leq k \leq j,\; \rho, k \models_f \phi_2 \\
\rho, i \models_f \downarrow_r^c \phi \;&\overset{\text{def}}{\Leftrightarrow}\; \rho, i \models_{f[r \mapsto \mathbf{v}_i(c)]} \phi
\end{aligned}
$$

$f[r \mapsto \mathbf{v}_i(c)]$ denotes the register valuation equal to f except that the register r is mapped to $\mathbf{v}_i(c)$. In the sequel, we omit the subscript "f" in $\models_f$ when sentences are involved. We use the standard abbreviations for the temporal operators ($\mathtt{G}, \mathtt{F}, \dots$) and for the Boolean operators and constants ($\Rightarrow, \top, \bot, \dots$).

We defined below fragments of $\text{LTL}^\downarrow[Q,n]$ by restricting the use of the freeze operators. The *strict* fragment, written $\text{LTL}^{\downarrow,s}[Q,n]$, consists in associating a unique counter to each register (to store and to test). More precisely, a formula ϕ in $\text{LTL}^{\downarrow,s}[Q,n]$ verifies the following syntactic property: if $\downarrow_r^c \psi$ is a subformula of ϕ, then ϕ has not subformulae of the form either $\uparrow_r^{c'}$ or $\downarrow_r^{c'} \psi'$ with $c \neq c'$. We also write $\text{LTL}[Q]$ to denote the fragment of $\text{LTL}^\downarrow[Q,n]$ in which the atomic formulae of the form $\uparrow_r^c$ are forbidden (and therefore $\downarrow_r^c$ becomes also useless).

Model-checking problems. The infinitary (existential) model-checking problem over counter machines, written $\text{MC}^\omega(\text{LTL}^\downarrow[\cdot,\cdot])$, is defined as follows: given a counter machine $M = \langle n, Q', \Delta, q_0 \rangle$ and a sentence $\phi \in \text{LTL}^\downarrow[Q,n]$ with $Q \subseteq Q'$, is there an infinite run ρ such that $\rho, 0 \models \phi$? If the answer is "yes", we write $M \models^\omega \phi$. The subproblem of $\text{MC}^\omega(\text{LTL}^\downarrow[\cdot,\cdot])$ with formulae restricted to $\text{LTL}^{\downarrow,s}[Q,n]$ is written $\text{MC}^\omega(\text{LTL}^{\downarrow,s}[\cdot,\cdot])$. Given $n \geq 1$, we write $\text{MC}^\omega(\text{LTL}^\downarrow[\cdot,n])$ to denote the subproblem of $\text{MC}^\omega(\text{LTL}^\downarrow[\cdot,\cdot])$ with counter machines of dimension at most n. Similarly, we write $\text{MC}^\omega(\text{LTL}^\downarrow[\emptyset,\cdot])$ to denote the subproblem of $\text{MC}^\omega(\text{LTL}^\downarrow[\cdot,\cdot])$ with no atomic formula made of control states. Similar notations are used with other fragments of $\text{LTL}^\downarrow[Q,n]$. In this existential version of model checking, this problem can be viewed as a variant of satisfiability in which satisfaction of a formula can be only witnessed within a specific class of data words, namely the runs of the counter machine. Note that results for the universal version of model checking will follow easily from those for the existential version when considering fragments closed under negation or deterministic counter machines.

Flat formulae. We say that the occurrence of a subformula in a formula is *positive* if it occurs under an even number of negations, otherwise it is *negative*. Let $\mathcal{L}$ be a fragment of $\text{LTL}^{\downarrow}[Q, n]$. The *flat fragment of* $\mathcal{L}$, written flat-$\mathcal{L}$, is the restriction of $\mathcal{L}$ where, for any occurrence of $\phi_1 U \phi_2$ [resp. $\phi_2 R \phi_1$], if it is positive then the freeze operator $\downarrow$ does not occur in ϕ_1, and if it is negative then the freeze operator $\downarrow$ does not occur in ϕ_2. A formula is *positively flat* when it is flat and no occurrence of the freeze operator $\uparrow$ occurs in the scope of an odd number of negations. For example, the formula below belongs to the positively flat fragment and it states that sometimes there is a value of the counter 1 such that (1) infinitely often counter 2 takes that value if and only if infinitely often counter 3 takes that value and (2) from some future position, the counter 4 has always that value: $F \downarrow_1^1 [(GF \uparrow_1^2 \Leftrightarrow GF \uparrow_1^3) \wedge FG \uparrow_1^4]$. Considering flat fragments remains a standard means to regain decidability: for instance flat fragments of LTL variants have been studied in [9,7] (see also in [27, Section 5] the design of a flat logical temporal language for model-checking pushdown machines). Section 7 shall illustrate that flatness can lead to decidability but this is not always the case.

4   Preliminary Results

In this section, we present preliminary results that will be helpful to strenghten forthcoming results and we present results for flat counter machines and one-counter machines based on existing works. We shall study the effects of restricting the set of atomic formulae, for instance by allowing only atomic formulae that are control states [resp. that are of the form $\uparrow_r^c$].

4.1   Purification, or How to Get Rid of Control States

Control states can be viewed as an internal piece of information about the counter machines and therefore, it is interesting to understand whether the absence of control states among the set of atomic formulae (called herein *purification*) makes a difference. Lemma 2 below roughly shows that control states can be always encoded by patterns for various classes of counter machines.

Lemma 2. *Given a counter machine $M = \langle n, Q, \Delta, q_0 \rangle$ and a sentence ϕ in $\text{LTL}^{\downarrow}[Q, n]$, one can build in logspace a counter machine $M_P = \langle n + 1, Q_P, \Delta_P, q_0 \rangle$ and a formula $\phi_P \in \text{LTL}^{\downarrow}[\emptyset, n + 1]$ such that $M \models^{\omega} \phi$ iff $M_P \models^{\omega} \phi_P$. Moreover, M is deterministic [resp. reversal-bounded, flat] iff M_P is deterministic [resp. reversal-bounded, flat] and $\phi \in \text{LTL}^{\downarrow,s}[Q, n]$ iff $\phi_P \in \text{LTL}^{\downarrow,s}[\emptyset, n + 1]$.*

The proof consists in introducing an additional counter whose behavior in M_P encodes the control states from M. The reduction in the proof of Lemma 2 does not preserve the number of counters; however, a purification lemma can be also established for the class of one-counter machines as shown in [15]. By the way, the construction in [15] could be also adapted to encode control states by patterns however, it does not preserve reversal-boundedness.

4.2   Restricting the Atomic Formulae to Control States

Before considering decidability issues with the freeze operator, it is legitimate to wonder what happens when the atomic formulae are restricted to control states. We show below that for all subclasses of counter machines considered in this paper, this restriction leads to decidability (for flat counter machines, the proof is postponed to the next subsection). Basically, the proof is a consequence of the two following properties: LTL formulae can be translated into equivalent Büchi automata (see e.g. [35]) and repeated reachability problem is decidable for the concerned subclasses of counter machines.

Theorem 3. $\mathrm{MC}^\omega(\mathrm{LTL}[\cdot])$ *restricted to one-counter machines, VASS, and reversal-bounded counter machines is decidable.*

4.3   Existing Results for Two Subclasses

In this paper, we wish to provide a complete classification with respect to the above-mentioned subclasses. The two following results are known results recasted in our context. First, we observe that $\mathrm{LTL}^\downarrow[Q, n]$ can be viewed as a fragment of the temporal logic $\mathrm{FOCTL}^\star(\mathrm{Pr})$ [12] which extends the logic $\mathrm{CTL}^\star$ by allowing the use of Presburger formulae as atomic propositions to describe sets of configurations for a counter machine. Since model-checking $\mathrm{FOCTL}^\star(\mathrm{Pr})$ over flat counter machines is decidable [12], we establish the following theorem.

Theorem 4. $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot, \cdot])$ *restricted to flat counter machines is decidable.*

Moreover, in [15], the authors obtain the following results concerning the model-checking of LTL with registers over one-counter machines.

Theorem 5. *[15] (I)* $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot, 1])$ *is undecidable. (II)* $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot, 1])$ *restricted to deterministic one-counter machines is* PSPACE-*complete.*

5   Nondeterministic Counter Machines

Herein, we consider the model-checking problems over $\mathrm{LTL}^\downarrow[Q, n]$ for nondeterministic counter machines. We have seen that for the class of one-counter machines the problem is undecidable (see Theorem 5(I)) whereas it is decidable for flat counter machines (see Theorem 4). First, we observe that zero-tests can be easily encoded in $\mathrm{LTL}^\downarrow[Q, n]$ by first storing the initial value of counters in some register r_0 and then performing a zero-test on counter c with the atomic formula $\uparrow_{r_0}^c$.

Theorem 6. $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot, \cdot])$ *restricted to VASS and to positively flat formulae with at most one register is undecidable.*

The proof is based on a simple encoding of zero-tests. For what concerns reversal-bounded counter machines, we have the following result:

Theorem 7. $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot, 4])$ *restricted to reversal-bounded counter machines and to formulae with at most one register is undecidable.*

To prove this result, we present a reduction from the halting problem for Minsky machines; note that a similar reduction is used in [28] in order to prove that in reversal-bounded counter machines extended with equality tests between distinct counters, the reachability problem is undecidable.

Proof. (sketch) Let $M = \langle 2, Q, \Delta, q_0 \rangle$ be a Minsky machine (deterministic counter machine with two counters) and $q_F \in Q$ be a final control state with no transition from it. Without any loss of generality, we can assume that if $\langle q, \mathbf{g}, \mathbf{a}, q' \rangle \in \Delta$ performs a decrementation, then the transition is of the form $\langle q, \mathtt{true}, -\mathbf{e}_c, q' \rangle$ for some $c \in \{1, 2\}$. Moreover, for $q, q' \in Q$, the set $\{\langle \mathbf{g}, \mathbf{a} \rangle : \langle q, \mathbf{g}, \mathbf{a}, q' \rangle \in \Delta\}$ contains at most one element. Let us build the reversal-bounded counter machine $M = \langle 4, Q', \Delta', (q_0)_\emptyset \rangle$ as follows:

- $Q' = \{q_X : q \in Q, X \subseteq \{1, 2\}\}$ (X records on which counter of M zero-test is needed next),
- Δ' is the smallest set of transitions satisfying the conditions below:
  - for $X \subseteq \{1, 2\}$, $\langle (q_0)_\emptyset, \mathbf{true}, \mathbf{0}, (q_0)_X \rangle \in \Delta'$,
  - for all $\langle q, \mathbf{g}, \mathbf{a}, q' \rangle \in \Delta$, we have $\langle q_1, \mathbf{true}, \mathbf{a}', q_1' \rangle \in \Delta'$ assuming that
    - $* \ q_1 = q_X$ with $X = \{c \in \{1, 2\} : \mathbf{g}(c) = \mathtt{zero}\}$,
    - $*$ for $c \in \{1, 2\}$,
      - $\cdot$ $\mathbf{a}(c) = 1$ implies $\mathbf{a}'(c) = 1$ and $\mathbf{a}'(c + 2) = 0$,
      - $\cdot$ $\mathbf{a}(c) = -1$ implies $\mathbf{a}'(c) = 0$ and $\mathbf{a}'(c + 2) = 1$,
      - $\cdot$ $\mathbf{a}(c) = 0$ implies $\mathbf{a}'(c) = \mathbf{a}'(c + 2) = 0$.
  - for $X \subseteq \{1, 2\}$, $\langle (q_F)_X, \mathbf{true}, \mathbf{0}, (q_F)_X \rangle \in \Delta$ (final loops).

By construction, the counter machine M' is reversal-bounded since the four counters only increase. The idea behind this construction is that the first [resp. second] and the third [resp. fourth] counters of M' respectively count the number of incrementations and decrementations of the first [resp. second] counter of M. No zero-test is performed in M'; in order to simulate a zero-test in M, we would need to test equality between two counters, which is not allowed in our models. Consequently, we encode these equality tests by formulae.

Let us build a formula ϕ in LTL$^\downarrow[Q', 4]$ such that $M' \models^\omega \phi$ iff the control state q_F can be reached from the initial configuration of M. We consider the following auxiliary formulae ($c \in \{1, 2\}$):

$$\phi_c \overset{\text{def}}{=} \bigvee_{q \in Q} \bigvee_{\{c\} \subseteq X \subseteq \{1,2\}} q_X \quad \text{and} \quad \phi_q \overset{\text{def}}{=} \bigvee_{X \subseteq \{1,2\}} q_X.$$

We are now in position to define ϕ:

$$\phi \overset{\text{def}}{=} \mathsf{F}\phi_{q_F} \wedge \bigwedge_{c \in \{1,2\}} \mathsf{G}(\phi_c \Rightarrow \downarrow_1^c \uparrow_1^{c+2}) \wedge \bigwedge_{c \in \{1,2\}} \mathsf{G}\Big(\bigwedge_{\langle q, \mathbf{true}, -\mathbf{e}_c, q' \rangle \in \Delta} q_\emptyset \wedge \mathsf{X}\phi_{q'} \Rightarrow \downarrow_1^c \neg \uparrow_1^{c+2} \Big)$$

It remains to show that $M' \models^\omega \phi$ iff the control state q_F can be reached in M. $\square$

The result of Theorem 7 can be refined by showing the undecidability of the strict fragment $\mathrm{MC}^\omega(\mathrm{LTL}^{\downarrow,s}[\cdot, 4])$ restricted to reversal-bounded counter machines. Observe that we shall modify the above developments while we are dealing with a strict fragment for which each register is associated with a unique counter.

Theorem 8. $\mathrm{MC}^\omega(\mathrm{LTL}^{\downarrow,\mathrm{s}}[\cdot,4])$ *restricted to reversal-bounded counter machines is undecidable.*

The proof takes advantage of a refinement in the contruction of the counter machine M' from the proof of Theorem 7 and it is interesting for its own sake. So far, it is still open whether the problem is Σ_1^1-hard since we are "only" able to reduce the halting problem to it.

6   Deterministic Counter Machines

In this section, we restrict ourselves to classes of deterministic counter machines. A class $\mathcal{C}$ of deterministic counter machines has the *PA-property* $\overset{\mathrm{def}}{\Leftrightarrow}$ for each counter machine $M \in \mathcal{C}$, one can effectively build a formula $\phi_M(x_0, \ldots, x_{n+1})$ in Presburger arithmetic such that for all $j_0, \ldots, j_{n+1} \in \mathbb{N}$, $\langle j_0, \langle j_1, \ldots, j_n \rangle \rangle$ is the j_{n+1}th configuration of the unique run of M iff $\langle j_0, \ldots, j_{n+1} \rangle \models \phi_M(x_0, \ldots, x_{n+1})$ (assuming that M has dimension n and its set of control states is viewed as a finite subset of $\mathbb{N}$).

We show below that model-checking restricted to counter machines can be sometimes reduced to the decidable satisfiability problem for Presburger arithmetic.

Lemma 9. *Let $\mathcal{C}$ be a class of deterministic counter machines. If $\mathcal{C}$ has the PA-property, then the model-checking problem $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot,\cdot])$ over counter machines in $\mathcal{C}$ is decidable.*

The proof of Lemma 9 is based on an internalization of the satisfaction relation in Presburger arithmetic.

Lemma 10. *Deterministic reversal-bounded counter machines and deterministic VASS have the PA-property.*

Corollary 11. $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot,\cdot])$ *is decidable when restricted to deterministic reversal-bounded counter machines and deterministic VASS.*

Checking whether a VASS is deterministic can be decided by using instances of the covering problem (the problem is actually PSPACE-complete [24]). Checking whether a reversal-bounded counter machine is deterministic is also decidable adding a counter which counts each step and using the fact that the reachability set can be expressed in Presburger arithmetic. By contrast, checking whether a counter machine is reversal-bounded is undecidable [19].

7   Flat Freeze LTL

In this section, we consider the restriction of the model-checking problem to flat formulae only. By Theorem 4, we already know that $\mathrm{MC}^\omega(\mathrm{flat} - \mathrm{LTL}^\downarrow[\cdot,\cdot])$ restricted to flat counter machines is decidable and that $\mathrm{MC}^\omega(\mathrm{flat} - \mathrm{LTL}^\downarrow[\cdot,\cdot])$ restricted to VASS is undecidable (the proof of Theorem 6 involves only flat formulae). It is worth observing that flat $\mathrm{LTL}^\downarrow[Q,n]$ strictly contains $\mathrm{LTL}[Q]$, and therefore we refine below decidability results from Section 4.2.

7.1   A Detour to Counter Machines with Parameterized Tests

We introduce here parameterized counter machines in order to solve later model-checking problems restricted to flat formulae. First, let us fix some definitions. A *counter machine with parameterized tests* (shortly *parameterized* counter machine) is defined as a counter machine $M = \langle n, Q, \Delta, q_0, Z \rangle$ extended with a finite set Z of integer variables such that the guards $\mathbf{g}$ are among $(\{\texttt{zero}, \texttt{true}\} \cup \{=(z), \neq(z), >(z), <(z) \mid z \in Z\})^n$. A *concretization* C of M is a map C $: Z \rightarrow \mathbb{N}$. Given a parameterized counter machine M and a concretization C, we introduce the transition system $TS(M, \mathrm{C}) = \langle Q \times \mathbb{N}^n, \rightarrow \rangle$ where $\rightarrow \subseteq (Q \times \mathbb{N}^n) \times (Q \times \mathbb{N}^n)$ is defined as follows: for $\langle q, \mathbf{v} \rangle, \langle q', \mathbf{v}' \rangle \in Q \times \mathbb{N}^n$, we have $\langle q, \mathbf{v} \rangle \rightarrow \langle q', \mathbf{v}' \rangle \overset{\text{def}}{\Leftrightarrow}$ there exists a transition $t = \langle q, \mathbf{g}, \mathbf{a}, q' \rangle \in \Delta$ such that $\mathbf{v}' = \mathbf{v} + \mathbf{a}$, and for $1 \leq c \leq n$, $\mathbf{g}(c)$ equals $\texttt{zero}$ implies $\mathbf{v}(c) = 0$, $\mathbf{g}(c)$ is equal to $=(z)$ implies $\mathbf{v}(c) = \mathrm{C}(z)$, $\mathbf{g}(c)$ is equal to $\neq(z)$ implies $\mathbf{v}(c) \neq \mathrm{C}(z)$, $\mathbf{g}(c)$ is equal to $>(z)$ implies $\mathbf{v}(c) > \mathrm{C}(z)$ and, $\mathbf{g}(c)$ is equal to $<(z)$ implies $\mathbf{v}(c) < \mathrm{C}(z)$. A finite [resp. infinite] *run* in $TS(M, \mathrm{C})$ is a finite [resp. infinite] sequence $\rho = \langle q_0, \mathbf{0} \rangle \rightarrow \langle q_1, \mathbf{v}_1 \rangle \rightarrow \ldots$. The *parameterized reachability problem* for counter machines is defined as follows: given a parameterized counter machine M and a configuration $\langle q, \mathbf{v} \rangle$, is there a concretization C such that $\langle q_0, \mathbf{0} \rangle \overset{*}{\rightarrow} \langle q, \mathbf{v} \rangle$ in $TS(M, \mathrm{C})$? Even if the parameterized reachability problem is obviously undecidable, we will see in this section that some restrictions lead to decidability. We will say that a parameterized counter machine is *Ibarra reversal-bounded* if the classical counter machine obtained by replacing each parameterized test by $\texttt{true}$ is Ibarra reversal-bounded. We have then the following result.

Theorem 12. *[28] The parameterized reachability problem for Ibarra reversal-bounded parameterized counter machines is decidable.*

If a parameterized counter machine has no guard of the form either $\neq(z)$ or $<(z)$, we say it is *restricted*. In [23], parametric one-counter machines are defined as extensions of one-counter machines extended with actions consisting in incrementing or decrementing the unique counter with some parameterized integer constants. In [23], it is shown that the reachability problem for this class of one-counter machines is decidable. Here is a corollary.

Lemma 13. *The parameterized reachability problem for restricted parameterized one-counter machines is decidable.*

The proof of Lemma 13 consists in substituting each test of the form $=(z)$ by the following sequence of instructions: decrement by z, perform a zero-test and increment by z. In order to encode the test $>(z)$, we use the same technique except that we do not introduce a zero-test between the decrementation (in fact we also add a decrementation by 1 and an incrementation by 1) and the incrementation. Note that this method does not work if we allow guards of the form either $\neq(z)$ or $<(z)$, because the value of the counter cannot be negative, hence the decidability of the parameterized reachability problem for one-counter machines remains an open problem.

We introduce here a new problem which is needed to reduce the considered model-checking problem. The *parameterized generalized repeated reachability problem* for parameterized counter machines is defined as follows: given a parameterized counter machine M, N sets $F_1, \ldots, F_N$ of control states, are there a concretization C and an infinite run of $TS(M, C)$ such that for $1 \leq i \leq N$, one control state in F_i is repeated infinitely often?

From the previous theorem and lemma, we deduce the following corollary.

Corollary 14. *The parameterized generalized repeated reachability problem is decidable when considering Ibarra reversal-bounded parameterized counter machines and restricted parameterized one-counter machines.*

7.2   Flat Formulae and Parameterized Counter Machines

For $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot, \cdot])$ restricted to flat formulae, we have the following result.

Theorem 15. *There is a reduction from $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot, \cdot])$ restricted to flat formulae to the parameterized generalized repeated reachability problem for counter machines.*

Proof. (sketch) Let $M = \langle n, Q, \Delta, q_0 \rangle$ be a counter machine and ϕ be a flat sentence belonging to $\mathrm{LTL}^\downarrow[Q, n]$. Without any loss of generality, we can assume that ϕ is in negation normal form (which means that all the occurrences of negation appear only in front of atomic formulae). Moreover, we can assume that if $\downarrow_r^c \psi$ and $\downarrow_{r'}^{c'} \psi$ are distinct occurrences of subformulae in ϕ, then $r \neq r'$ (this may just linearly increase the number of registers). Consequently, if $\psi_1 U \psi_2$ [resp. $\psi_1 R \psi_2$] is a subformula of ϕ, then the freeze operator $\downarrow$ cannot occur in ψ_1 [resp. ψ_2]. We shall effectively build a parameterized counter machine $M' = \langle n, Q', \Delta', q_0, Z' \rangle$ and sets $F_1, \ldots, F_N \subseteq Q'$ for which there is a concretization C and an infinite run of $TS(M', C)$ such that for $1 \leq i \leq N$, one control state in F_i is repeated infinitely often iff $M \models^\omega \phi$.

Let us fix some notations. As usual, the formula ϕ can be encoded as a finite tree whose leaves are labelled by atomic formulae and internal nodes are labelled by (Boolean, temporal or freeze) connectives. Each node of the formula tree corresponds naturally to a subformula and the set of nodes can be viewed as a finite prefix-closed subset $\mathrm{occ}(\phi) \subseteq (\mathbb{N} \setminus \{0\})^*$ (finite sequence of natural numbers). Each element in $\mathrm{occ}(\phi)$ corresponds to the occurrence of a subformula in ϕ; hence two occurrences may correspond to the same subformula. The use of occurrences instead of subformulae is motivated by the need to provide formal and clear statements in which occurrences are crucial. For each occurrence $u \in \mathrm{occ}(\phi)$, we write $\phi(u)$ to denote the corresponding subformula in ϕ; for instance $\phi(\epsilon) = \phi$. Moreover, when u is a prefix of u', written $u \leq_{\mathrm{pre}} u'$, we know that $\phi(u')$ is a subformula of $\phi(u)$. We write $\mathrm{occ}^\downarrow(\phi)$ [resp. $\mathrm{occ}^\uparrow(\phi)$] to denote the set of occurrences corresponding to formulae whose outermost connective is of the form $\downarrow_r^c$ [resp. $\uparrow_r^c$]. Let $m = \mathrm{card}(\mathrm{occ}^\downarrow(\phi))$. Observe that if $m = 0$, then we are in the case of $\mathrm{MC}^\omega(\mathrm{LTL}[\cdot])$ which has been treated in Section 4.2. In the sequel, we assume that $m > 0$. Given $u \in \mathrm{occ}^\uparrow(\phi)$ with $\phi(u) = \uparrow_r^c$, we write $\mathrm{bind}(u)$ to denote the longest prefix of u (with respect to $\leq_{\mathrm{pre}}$) in $\mathrm{occ}^\downarrow(\phi)$ such that $\phi(\mathrm{bind}(u))$ is of the form $\downarrow_r^{c'} \psi$ (i.e., with the same register). An *atom* X is a subset of $\mathrm{occ}(\phi)$ satisfying the conditions below (we abusively use subformulae to denote occurrences corresponding to formulae

with the appropriate outermost connective): (1) if $\psi_1 \wedge \psi_2 \in X$, then $\psi_1, \psi_2 \in X$; (2) for all atomic formulae $\psi \in X$, $\{\psi, \neg\psi\} \not\subseteq X$; (3) if $\psi_1 \vee \psi_2 \in X$, then either $\psi_1 \in X$ or $\psi_2 \in X$; (4) if $\downarrow_r^c \psi \in X$, then $\psi \in X$. The set of *atoms* of ϕ is denoted by $\mathrm{AT}(\phi)$. A pair of atoms $\langle X, X' \rangle$ is said to be *one-step consistent* iff the conditions below hold true: (I) if $\psi_1 \mathtt{U} \psi_2 \in X$, then either $\psi_2 \in X$ or ($\psi_1 \in X$ and $\psi_1 \mathtt{U} \psi_2 \in X'$); (II) if $\psi_1 \mathtt{R} \psi_2 \in X$, then $\psi_2 \in X$ and ($\psi_1 \in X$ or $\psi_1 \mathtt{R} \psi_2 \in X'$); (III) if $\mathtt{X}\psi \in X$, then $\psi \in X'$; (IV) No atom X'' strictly included in X' satisfies the conditions (I)–(III) (by replacing X' by X''). We will now describe the construction of the parameterized counter machine M' which will use m integer variables $z_1, \ldots, z_m$. Intuitively, each integer variable will be used to store the value of a register. In order to make explicit this dependancy, we shall use a one-to-one map $reg : \mathrm{occ}^{\downarrow}(\phi) \to \{1, \ldots, m\}$. We define also a function $counter : \mathrm{occ}^{\downarrow}(\phi) \cup \mathrm{occ}^{\uparrow}(\phi) \to \{1, \ldots, n\}$ that indicates the counter involved in the subformula. Given $u \in \mathrm{occ}^{\downarrow}(\phi)$ such that $\phi(u) = \downarrow_r^c \psi$, we have $counter(u) = c$ and given $u \in \mathrm{occ}^{\uparrow}(\phi)$ such that $\phi(u) = \uparrow_r^c$, we have $counter(u) = c$. The set Q' of control states is equal to $\{q_0\} \uplus Q \times \mathrm{AT}(\phi)$ plus some auxiliary control states that are introduced to perform tests. The relation Δ' is defined as follows. First, $\langle q_0, \mathbf{true}, \mathbf{0}, \langle q_0, Y \rangle \rangle \in \Delta'$ whenever $\epsilon \in Y$ and no atom strictly included in Y contains ϵ (init). Then, for each transition $\langle q, \mathbf{g}, \mathbf{a}, q' \rangle \in \Delta$ there is in Δ' the following sequence of transitions $\langle q, Y \rangle \cdots q_1^{aux} \cdots q_T^{aux} \xrightarrow{\mathbf{g,a}} \langle q', Y' \rangle$ assuming that:

1. $\mathrm{occ}^{\downarrow}(\phi) \cap Y$ contains T_1 elements, say $u_1, \ldots, u_{T_1}$; $\mathrm{occ}^{\uparrow}(\phi) \cap Y$ contains T_2 elements, say $u_{T_1+1}, \ldots, u_{T_1+T_2}$; $\{u \in Y \mid u \cdot 1 \in \mathrm{occ}^{\uparrow}(\phi)$ and $\phi(u)$ is a negation$\}$ contains T_3 elements, say $u_{T_1+T_2+1}, \ldots, u_{T_1+T_2+T_3}$ with $T = T_1 + T_2 + T_3$,
2. $\langle Y, Y' \rangle$ is a one-step consistent pair,
3. $\{\phi(u) : u \in Y\} \cap Q \subseteq \{q\}$ and $\neg q \notin \{\phi(u) : u \in Y\}$,
4. for $i \in \{1, \ldots, T_1\}$ [resp. $i \in \{1, \ldots, T_2\}$, $i \in \{1, \ldots, T3\}$], before reaching the control state q_i^{aux} [resp. $q_{T_1+i}^{aux}$, $q_{T_1+T_2+i}^{aux}$], there exists a transition testing equality [resp. equality, inequality] between z_k and the counter $counter(u_i)$ [resp. $counter(u_{T_1+i})$, $counter(u_{T_1+T_2+i})$] with the identity $k = reg(u_i)$ [resp. $k = reg(\mathrm{bind}(u_{T_1+i}))$, $k = reg(\mathrm{bind}(u_{T_1+T_2+i}))$].

Finally, let $u_1, \ldots, u_N$ be the occurrences in $\mathrm{occ}(\phi)$ such that the outermost temporal connective of $\phi(u_i)$ is the until operator $\mathtt{U}$. Then, for $1 \leq i \leq N$, $F_i = \{\langle q, Y \rangle : u_i \notin Y$ or $(u_i \cdot 2) \in Y\}$. It remains to show that $M \models^\omega \phi$ iff there exist a concretization C and an infinite run of $TS(M', C)$ such that for $1 \leq i \leq N$, one control state in F_i is repeated infinitely often. □

7.3 Decidability Results

Remark that if the counter machine M is Ibarra reversal-bounded, then the parameterized counter machine M' built from M and the flat formula ϕ is Ibarra reversal-bounded. Using Corollary 14 and Theorem 15, we conclude that $\mathrm{MC}^\omega(\mathrm{LTL}^{\downarrow}[\cdot, \cdot])$ restricted to Ibarra reversal-bounded counter machines and to flat formulae is decidable. Furthermore this can be extended to the class of reversal-bounded counter machines, using Lemma 16 below.

Lemma 16. *There is an exponential-time reduction from* $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot,\cdot])$ *restricted to reversal-bounded counter machines into* $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot,\cdot])$ *restricted to Ibarra reversal-bounded counter machines. Furthermore this reduction preserves flatness of the formulae.*

Corollary 17. $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot,\cdot])$ *restricted to reversal-bounded counter machines and to flat formulae is decidable.*

Finally, assume the formula ϕ is a positively flat formula (see Section 3). For all atoms $Y \in \mathrm{AT}(\phi)$, the set $\{u \in Y \mid u \cdot 1 \in \mathrm{occ}^\uparrow(\phi)$ and $\phi(u)$ is a negation$\}$ is empty. So, in the construction of M' from M and ϕ, we only use parameterized tests of the form $=(z)$. Hence, if M is a one-counter machine and ϕ is a positively flat formula, we deduce that M' is a restricted parameterized one-counter machine. Using Corollary 14 and Theorem 15, we get the result below.

Theorem 18. $\mathrm{MC}^\omega(\mathrm{LTL}^\downarrow[\cdot,\cdot])$ *restricted to one-counter machines and to positively flat formulae is decidable.*

In order to extend Theorem 18 to the full flat fragment, one needs to perform inequality tests in parameterized one-counter machines, which is so far unclear how to perform while preserving decidability of the corresponding parameterized reachability problem. This generalization is left as an open problem.

8   Concluding Remarks

In this paper, we have studied the decidability status of model-checking freeze LTL over various subclasses of counter machines for which the reachability problem is known to be decidable. Our most remarkable technical contributions concern reversal-bounded counter machines and flat formulae. Besides, we have established an original link between reachability problems for parameterized counter machines and model-checking counter machines over the flat fragment of freeze LTL. The table below contains a summary of the main results (**D** stands for decidability, **U** for undecidability) in which the columns referred to restriction either on the counter machines or on the formulae. Sometimes, an additional restriction between parentheses is indicated in order to emphasize that the result holds true for a stricter fragment. Bibliographical references in the table indicate that the related result is mainly due to the referred work.

	Det.	**NDet.**	**Flat formulae**	**No $\uparrow_r^c$**
RB	D	U (strictness)	D	D
	Cor. 11	**Theo. 8**	**Cor. 17**	**[11]**
1CM	PSPACE-C.	U (1 reg.)	open │ D for pos. flatness	PSPACE-C.
	[15]	**[15]**	**Theo. 18**	**[34,13]**
Flat CM	D	D	D	D
		Theo. 4		
VASS	D	U (1 reg.)	U	D
	Cor. 11	**Theo. 6**	**Theo. 6**	**[29]**

Here are a few rules of thumb: determinism, flat counter machines and no freeze lead to decidability. However, flat formulae often guarantee decidability (except for VASS) whereas reversal-boundedness can lead to decidability (but the restriction with a single register leads to undecidability). Finally, throwing away the atomic formulae made of control states does not help for decidability. Even though we have established various decidability results, the complexity of the decision problems is far from being known, mainly because we use reductions to Presburger arithmetic. Such characterizations are part of future work.

References

1. Alur, R., Henzinger, T.: A really temporal logic. JACM 41(1), 181–204 (1994)
2. Areces, C., Blackburn, P., Marx, M.: A road-map on complexity for hybrid logics. In: Flum, J., Rodríguez-Artalejo, M. (eds.) CSL 1999. LNCS, vol. 1683, pp. 307–321. Springer, Heidelberg (1999)
3. Björklund, H., Bojańczyk, M.: Bounded depth data trees. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 862–874. Springer, Heidelberg (2007)
4. Boigelot, B.: Symbolic methods for exploring infinite state spaces. PhD thesis, Université de Liège (1998)
5. Bojańczyk, M., Muscholl, A., Schwentick, T., Segoufin, L., David, C.: Two-variable logic on words with data. In: LICS 2006, pp. 7–16. IEEE, Los Alamitos (2006)
6. Bouajjani, A., Bozga, M., Habermehl, P., Iosif, R., Moro, P., Vojnar, T.: Programs with lists are counter automata. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 517–531. Springer, Heidelberg (2006)
7. Bouyer, P., Markey, N., Ouaknine, J., Worrell, J.: The cost of punctuality. In: LICS 2007, pp. 109–118. IEEE, Los Alamitos (2007)
8. Bozga, M., Iosif, R., Lakhnech, Y.: Flat parametric counter automata. Fundamenta Informaticae 91(2), 275–303 (2009)
9. Comon, H., Cortier, V.: Flatness is not a weakness. In: Clote, P.G., Schwichtenberg, H. (eds.) CSL 2000. LNCS, vol. 1862, pp. 262–276. Springer, Heidelberg (2000)
10. Dal Lago, U., Montanari, A., Puppis, G.: On the equivalence of automaton-based representations of time granularities. In: TIME 2007, pp. 82–93. IEEE, Los Alamitos (2007)
11. Dang, Z., Ibarra, O., San Pietro, P.: Liveness verification of reversal-bounded multicounter machines with a free counter. In: Hariharan, R., Mukund, M., Vinay, V. (eds.) FSTTCS 2001. LNCS, vol. 2245, pp. 132–143. Springer, Heidelberg (2001)
12. Demri, S., Finkel, A., Goranko, V., van Drimmelen, G.: Towards a model-checker for counter systems. In: Graf, S., Zhang, W. (eds.) ATVA 2006. LNCS, vol. 4218, pp. 493–507. Springer, Heidelberg (2006)
13. Demri, S., Gascon, R.: The effects of bounding syntactic resources on Presburger LTL. Journal of Logic and Computation 19(6), 1541–1575 (2009)
14. Demri, S., Lazić, R., Nowak, D.: On the freeze quantifier in constraint LTL: decidability and complexity. I&C 205(1), 2–24 (2007)
15. Demri, S., Lazić, R., Sangnier, A.: Model checking freeze LTL over one-counter automata. In: Amadio, R.M. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 490–504. Springer, Heidelberg (2008)

16. Demri, S., Sangnier, A.: When Model-Checking Freeze LTL over Counter Machines Becomes Decidable. Research report, LSV, ENS Cachan (2010)
17. Esparza, J., Finkel, A., Mayr, R.: On the verification of broadcast protocols. In: LICS 1999, pp. 352–359. IEEE, Los Alamitos (1999)
18. Finkel, A., Leroux, J.: How to compose Presburger accelerations: Applications to broadcast protocols. In: FST&TCS 2002. LNCS, vol. 2256, pp. 145–156. Springer, Heidelberg (2002)
19. Finkel, A., Sangnier, A.: Reversal-bounded counter machines revisited. In: Ochmański, E., Tyszkiewicz, J. (eds.) MFCS 2008. LNCS, vol. 5162, pp. 323–334. Springer, Heidelberg (2008)
20. Ginsburg, S., Spanier, E.H.: Semigroups, Presburger formulas, and languages. Pacific Journal of Mathematics 16(2), 285–296 (1966)
21. Göller, S., Mayr, R., To, A.: On the computational complexity of verifying one-counter processes. In: LICS 2009, pp. 235–244. IEEE, Los Alamitos (2009)
22. Goranko, V.: Hierarchies of modal and temporal logics with references pointers. Journal of Logic, Language, and Information 5, 1–24 (1996)
23. Haase, C., Kreutzer, S., Ouaknine, J., Worrell, J.: Reachability in succinct and parametric one-counter automata. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009 - Concurrency Theory. LNCS, vol. 5710, pp. 369–383. Springer, Heidelberg (2009)
24. Howell, R., Jančar, P., Rosier, L.: Completeness results for single-path Petri nets. I&C 106(2), 253–265 (1993)
25. Ibarra, O.: Reversal-bounded multicounter machines and their decision problems. JACM 25(1), 116–133 (1978)
26. Ibarra, O.: Restricted one-counter machines with undecidable universe problems. Mathematical Systems Theory 13(181), 181–186 (1979)
27. Ibarra, O., Dang, Z.: On removing the stack from reachability constructions. In: Eades, P., Takaoka, T. (eds.) ISAAC 2001. LNCS, vol. 2223, pp. 244–256. Springer, Heidelberg (2001)
28. Ibarra, O., Su, J., Dang, Z., Bultan, T., Kemmerer, R.: Counter machines and verification problems. TCS 289(1), 165–189 (2002)
29. Jančar, P.: Decidability of a temporal logic problem for Petri nets. TCS 74(1), 71–93 (1990)
30. Lisitsa, A., Potapov, I.: Temporal logic with predicate λ-abstraction. In: TIME 2005, pp. 147–155. IEEE, Los Alamitos (2005)
31. Presburger, M.: Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In: Comptes Rendus du Premier Congrès de Mathématiciens des Pays Slaves, Warsaw, pp. 92–101 (1929)
32. Reutenauer, C.: The Mathematics of Petri nets. Masson and Prentice (1990)
33. Sangnier, A.: Vérification de systèmes avec compteurs et pointeurs. Thèse de doctorat, LSV, ENS Cachan, France (2008)
34. Sistla, A., Clarke, E.: The complexity of propositional linear temporal logic. JACM 32(3), 733–749 (1985)
35. Vardi, M., Wolper, P.: Reasoning about infinite computations. I&C 115, 1–37 (1994)