Reasoning about Reversal-Bounded Counter Machines

To the honor of Professor Ewa Orłowska

Stéphane Demri[⋆]

LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay, France

Abstract. In this paper, we present a short survey on reversal-bounded counter machines. It focuses on the main techniques for model-checking such counter machines with specifications expressed with formulae from some linear-time temporal logic. All the decision procedures are designed by translation into Presburger arithmetic. We provide a proof that is alternative to Ibarra's original one for showing that reachability sets are effectively definable in Presburger arithmetic. Extensions to repeated control state reachability and to additional temporal properties are discussed in the paper. The article is written to the honor of Professor Ewa Orłowska and focuses on several topics that are developped in her works.

1 Introduction

Reversal-bounded counter machines. Counter machines are well-known infinite-state systems that have many applications in formal verification. Their ubiquity stems from their use as operational models for several purposes, including for instance for broadcast protocols [FL02] and for logics for data words, see e.g. [BL10,DFP16]. However, numerous model-checking problems for counter machines, such as reachability, are known to be undecidable. Nevertheless, many subclasses of counter machines admit a decidable reachability problem, such as reversal-bounded counter machines [Iba78] and flat counter machines [CJ98,Boi99,FL02]. These two classes of machines admit reachability sets effectively definable in Presburger arithmetic [Pre29] (assuming some additional conditions, unspecified herein). This has also significant consequences for other classes of machines, for instance the reachability relation for a discrete timed automaton can be accepted by a reversal-bounded counter machine [DIB$^+$00], leading to the translation of reachability problem into Presburger arithmetic. The use of Presburger arithmetic for formal verification has been advocated in [SJ80], see also [Fri00]. Moreover, another standard way to overcome undecidability is to consider subclasses of runs by bounding some features and to search for 'bounded runs' that may satisfy a desirable or an undesirable property. Here are standard examples for boundedness.

- Reversal-bounded counter machines have effectively semilinear reachability sets (bound on the number of reversals), see e.g. [Iba78] and this is the main subject of the paper.

[⋆] demri@lsv.fr

- Bounding the number of context switches is done for multi-pushdown systems in order to perform context-bounded model-checking, see e.g. [QR05,EG11], see also the weak multi-pushdown automata in [CHL13] that require a partial ordering on the control states.
- Bounding the distance of the reachable configurations has lead to bounded model-checking (BMC) and its decidability, see e.g. [BCC$^+$03].

Temporal reasoning and Presburger arithmetic. Reachability questions are not the only interesting ones and the verification of properties expressed in dedicated temporal logics is often desirable.

Temporal logics on vector addition systems with states (VASS) often lead to undecidable model-checking problems, see e.g. [HR89,Esp94,Esp98]. There are still exceptions for decidability. For instance, the control state repeated reachability problem for VASS is shown to be decidable in [Jan90] and this is generalised to full LTL [Hab97]. Another example for which temporal reasoning is possible concerns flat counter machines that are counter machines in which each control state belongs to at most one simple cycle (i.e., a cycle without any repetition of edges). Several classes of such flat operational devices have been identified and reachability sets have been shown effectively Presburger-definable for many of them, see e.g. [FO97,CJ98,Boi99,FL02]. This provides a decision procedure for the reachability problem, given a solver dedicated to Presburger arithmetic. Decidability results for verifying safety and reachability properties on flat counter machines have been obtained in [CJ98,FL02,BIK09]. For the verification of temporal properties, it is much more difficult to get sharp complexity characterisation. For instance, it is known that verifying flat counter machines with LTL enriched with arithmetical constraints is only NP-complete [DDS15]. In [DDS13], we characterised the computational complexity of model-checking problems on flat counter machines for several prominent linear-time specification languages.

Our contribution. In this paper, we present a short survey on reversal-bounded counter machines by providing several results related to model-checking and reachability. Even though in full generality these problems lead to undecidability, this is precisely the assumption on reversal-boundedness that allows to regain decidability, see e.g. the key results from [Iba78]. We focus on the main techniques for model-checking such counter machines with specifications expressed with formulae from some linear-time temporal logic. All the decision procedures are designed by translation into Presburger arithmetic [Pre29]. We provide a proof, partly based on [GI81], that is alternative to Ibarra's original one for showing that reachability sets are effectively definable in Presburger arithmetic [Iba78]. Extensions to repeated control state reachability and to additional temporal properties are discussed in the paper. So, we aim at providing a document that provides several key results for reversal-bounded counter machines and gives many bibliographical references to related works for further study.

Content of the paper. The paper is structured as follows. Section 2 is dedicated to preliminaries on counter machines and to related decision problems. Section 2.2 presents a short introduction to Presburger arithmetic, which shall be much useful when synthetising formulae from reversal-bounded counter machines. Section 3 presents the main result due to Ibarra: the reachability sets for reversal-bounded counter machines are effectively definable in Presburger arithmetic (and extensions related to control state repeated reachability). Section 4 lifts some of previous results to temporal reasoning.

The article is written to the honor of Professor Ewa Orłowska and focuses on several topics developped in her works and strongly influencial for many of us, such as logics for computer science (see e.g. [Orło85,FdCO85,Orło89]), automated deduction (see e.g. [Orło73]), reasoning about programs (see e.g. [Orło93]), temporal reasoning (see e.g. [Orło95,BGO06]), combination of logics (see e.g. [BO99]), faithful translations (see e.g. [Orło88]), to mention a few topics.

2 Preliminaries

2.1 Counter machines

We write $\mathbb{N}$ [resp. $\mathbb{Z}$] to denote the set of natural numbers [resp. integers] and $[i, j]$ to denote the set $\{k \in \mathbb{Z} : i \leq k \text{ and } k \leq j\}$ $(i, j \in \mathbb{Z})$. In the sequel, integers are encoded with a binary representation. For each $\boldsymbol{x} \in \mathbb{Z}^n$, $\boldsymbol{x}(i)$ denotes the i^{th} element of $\boldsymbol{x}$ for every $i \in [1, n]$. For a finite alphabet Σ, Σ^* represents the set of finite words over Σ, Σ^+ the set of finite non-empty words over Σ and Σ^ω the set of infinite words or ω-words over Σ.

The counter machines introduced below are finite-state automata augmented with counters (variables interpreted as natural numbers). Transitions are labelled by arithmetical constraints on counters (more specifically by pairs $\langle \mathsf{g}, \boldsymbol{a} \rangle$ where g is a guard and $\boldsymbol{a}$ is an update vector). The developments in the paper about reversal-boundedness are performed for this class of counter machines, unless otherwise stated. We consider *counter machines* $\mathcal{M} = \langle Q, T, C \rangle$ such that

- Q is a finite set of *control states*,
- C is a finite set of *counters* $\{\mathsf{x}_1, \ldots, \mathsf{x}_d\}$ for some $d \geq 1$,
- T is a finite set of *transitions* from $Q \times \varSigma \times Q$ where the operations in $\varSigma$ are defined as follows. Each operation in $\varSigma$ is defined as a pair $\langle \mathsf{g}, \boldsymbol{a} \rangle$ where $\boldsymbol{a} \in \mathbb{Z}^d$ is an update vector and g is a guard built over the following grammar:

$$\mathsf{g} ::= \top \mid \bot \mid \mathsf{x} \sim k \mid \mathsf{g} \wedge \mathsf{g} \mid \mathsf{g} \vee \mathsf{g} \mid \neg \mathsf{g}$$

  with $\mathsf{x} \in C$, $\sim \in \{\leq, \geq, =\}$ and $k \in \mathbb{N}$.

Below, we write $\mathcal{C}$ to denote this class of counter machines.

Figure 1 contains a counter machine $\mathcal{M}$ such that x_i++ [resp. x_i− −] stands for the pair $\langle \mathsf{g}, \boldsymbol{a} \rangle$ that increments [resp. decrements if possible] the counter x_i and keeps unchanged the other counters. The formula $\mathsf{x}_i = 0$? tests if counter x_i is equal to zero but it has no effect on the counters.

The elements $t = \langle q, \langle \mathsf{g}, \boldsymbol{a} \rangle, q' \rangle \in T$ are often represented by $q \xrightarrow{\langle \mathsf{g}, \boldsymbol{a} \rangle} q'$. A *configuration* is a pair $\langle q, \boldsymbol{x} \rangle \in Q \times \mathbb{N}^d$. Given two configurations $\langle q, \boldsymbol{x} \rangle$, $\langle q', \boldsymbol{x'} \rangle \in Q \times \mathbb{N}^d$ and a transition $t = q \xrightarrow{\langle \mathsf{g}, \boldsymbol{a} \rangle} q'$, we write $\langle q, \boldsymbol{x} \rangle \xrightarrow{t} \langle q', \boldsymbol{x'} \rangle$ whenever $\boldsymbol{x}$ satisfies g (in the most natural way) and $\boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{a}$.

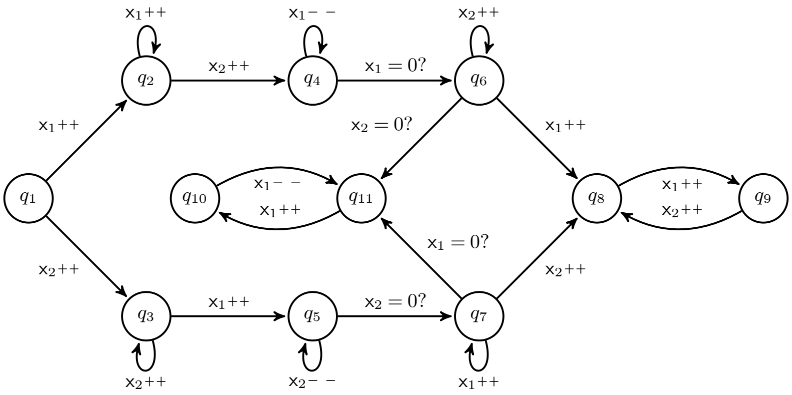

Fig. 1. A counter machine

Given a counter machine $\mathcal{M} = \langle Q, T, C\rangle$, its *transition system* $\mathfrak{T}(\mathcal{M}) = \langle \mathfrak{S}, \rightarrow\rangle$ is a graph with $\mathfrak{S} = Q \times \mathbb{N}^d$ and $\rightarrow \subseteq \mathfrak{S} \times \mathfrak{S}$ such that $\langle\langle q, \boldsymbol{x}\rangle, \langle q', \boldsymbol{x'}\rangle\rangle \in \rightarrow \overset{\text{def}}{\Leftrightarrow}$ there exists a transition $t \in T$ such that $\langle q, \boldsymbol{x}\rangle \xrightarrow{t} \langle q', \boldsymbol{x'}\rangle$. As usual, $\xrightarrow{*}$ denotes the reflexive and transitive closure of the binary relation $\rightarrow$. Figure 2 presents a simple counter machine (one counter and three control states) with its transition system.

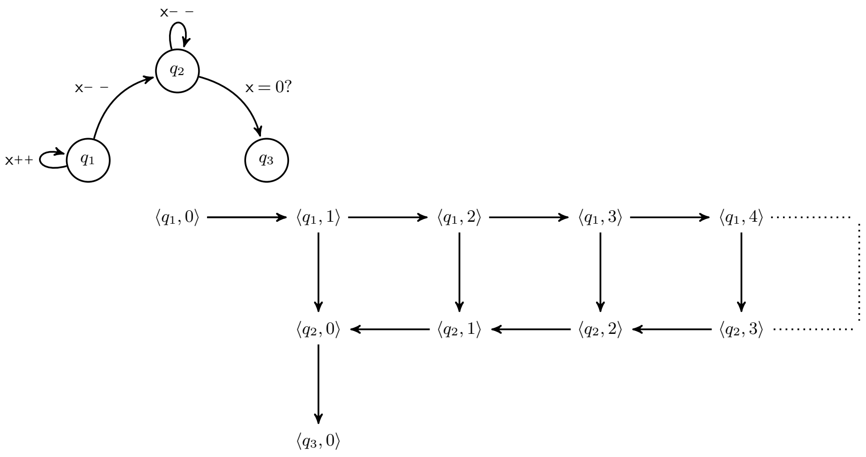

Fig. 2. A counter machine and its transition system

A *run* ρ is a non-empty (possibly infinite) sequence $\rho = \langle q_0, \boldsymbol{x_0}\rangle, \ldots, \langle q_k, \boldsymbol{x_k}\rangle, \ldots$ of configurations such that two consecutive configurations are in the relation $\rightarrow$ from $\mathfrak{T}(\mathcal{M})$.

Most verification problems on counter machines are known to be undecidable since they include Minsky machines [Min67, Chapter 11] (see also [Min61, Section 3]) that are Turing-complete, even if restricted to two counters [Min67, Chapter 14].

In this section, we recall several standard decision problems about counter machines. They are mainly related to reachability questions (problems related to temporal logics are introduced in Section 4).

REACHABILITY PROBLEM:

Input: a counter machine $\mathcal{M}$ and configurations $\langle q_0, \boldsymbol{x_0} \rangle$ and $\langle q_f, \boldsymbol{x_f} \rangle$.
Question: is there a finite run from $\langle q_0, \boldsymbol{x_0} \rangle$ to $\langle q_f, \boldsymbol{x_f} \rangle$?

CONTROL STATE REACHABILITY PROBLEM:

Input: a counter machine $\mathcal{M}$, a configuration $\langle q_0, \boldsymbol{x_0} \rangle$ and a state q_f.
Question: is there a finite run with initial configuration $\langle q_0, \boldsymbol{x_0} \rangle$ and whose final configuration has control state q_f?

CONTROL STATE REPEATED REACHABILITY PROBLEM:

Input: a counter machine $\mathcal{M}$, a configuration $\langle q_0, \boldsymbol{x_0} \rangle$ and a state q_f.
Question: is there an infinite run with initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ such that the control state q_f is repeated infinitely often?

A *vector addition system with states* [KM69] (VASS for short) is a counter machine such that all the transitions are of the form $q \xrightarrow{\langle \top, \boldsymbol{a} \rangle} q'$. So, a VASS can be represented by a tuple $\mathcal{M} = \langle Q, T, C \rangle$ where T is a finite subset of $Q \times \mathbb{Z}^d \times Q$. A famous result states that the reachability problem for VASS is decidable [May84,Kos82,Ler09]. It has been the subject of the book [Reu90] and its proof requires many non-trivial steps involving graph theory, logic and theory of well-quasi-orderings. Nevertheless, the exact complexity of the reachability problem is open: we know it is EXPSPACE-hard [Lip76] and the only known upper bound has been established recently in [LS15] (see also [Sch17]).

2.2 Presburger arithmetic

The arithmetic of the natural numbers with addition, nowadays known as Presburger arithmetic (PA), has been shown decidable in Presburger's master thesis supervised by A. Tarski, Warsaw University. The proof uses the approach by quantifier elimination (see e.g. [Pre29]) and this decidability result is regarded today as a key result in theoretical computer science.

Let VAR $= \{\mathsf{x}, \mathsf{y}, \mathsf{z}, \ldots\}$ be a countably infinite set of *variables*. *Terms* are expressions of the form $a_1\mathsf{x}_1 + \cdots + a_n\mathsf{x}_n + k$ where $a_1, \ldots, a_n$ are constant coefficients in $\mathbb{N}$, k is in $\mathbb{N}$ and the x_i's are variables. Variables and terms come with their interpretations when the variables are interpreted by natural numbers. A *valuation* $\mathfrak{v}$ is a map VAR $\rightarrow \mathbb{N}$ and it can be extended to the set of all terms as follows: $\mathfrak{v}(k) = k$, $\mathfrak{v}(a\mathsf{x}) = a \times \mathfrak{v}(\mathsf{x})$ and $\mathfrak{v}(\mathtt{t} + \mathtt{t}') = \mathfrak{v}(\mathtt{t}) + \mathfrak{v}(\mathtt{t}')$ for all terms $\mathtt{t}$ and $\mathtt{t}'$. *Formulae* are defined by the grammar below:

$$\varphi ::= \mathtt{t} \leq \mathtt{t}' \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists\, \mathsf{x}\, \varphi,$$

where $\mathtt{t}$ and $\mathtt{t}'$ are terms and $\mathsf{x} \in$ VAR. A formula φ is in the *linear fragment* $\overset{\text{def}}{\Leftrightarrow}$ φ is a Boolean combination of atomic formulae of the form $\mathtt{t} \leq \mathtt{t}'$. The semantics for formulae in (PA) is defined with the help of the satisfaction relation $\models$ that determines the conditions for the satisfaction of a formula under a given valuation (we omit the obvious Boolean clauses):

- $\mathfrak{v} \models \mathtt{t} \leq \mathtt{t}' \overset{\text{def}}{\Leftrightarrow} \mathfrak{v}(\mathtt{t}) \leq \mathfrak{v}(\mathtt{t}')$,
- $\mathfrak{v} \models \exists\, \mathsf{x}\, \varphi \overset{\text{def}}{\Leftrightarrow}$ there is $n \in \mathbb{N}$ such that $\mathfrak{v}[\mathsf{x} \mapsto n] \models \varphi$ where $\mathfrak{v}[\mathsf{x} \mapsto n]$ is equal to $\mathfrak{v}$ except that x is mapped to n.

Any formula $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ whose free variables are among $\mathsf{x}_1, \ldots, \mathsf{x}_n$, with $n \geq 1$, defines a set of n-tuples

$$[\![\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_n)]\!] \overset{\text{def}}{=} \{\langle \mathfrak{v}(\mathsf{x}_1), \ldots, \mathfrak{v}(\mathsf{x}_n)\rangle \in \mathbb{N}^n : \mathfrak{v} \models \varphi\},$$

which contains all the tuples that make true the formula φ by ignoring the irrelevant interpretation of the bound variables and by fixing an arbitrary ordering between the variables. For instance, $[\![\mathsf{x}_1 < \mathsf{x}_2]\!] = \{\langle n, n'\rangle \in \mathbb{N}^2 : n < n'\}$. Let φ be a formula $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ with $n \geq 1$ free variables $\mathsf{x}_1, \ldots, \mathsf{x}_n$. We say that $[\![\varphi]\!]$ is a *Presburger set*. A major result established in [GS66] states that the Presburger sets are precisely the semilinear sets, that can be defined as the finite unions of linear sets of $\mathbb{N}^d$. Each linear set is defined from a base vector and a finite set of periods. The *satisfiability problem* for (PA) is a decision problem that takes as input a formula φ and asks whether there is a valuation $\mathfrak{v}$ such that $\mathfrak{v} \models \varphi$. If such a valuation exists, we say that φ is *satisfiable*. Observe that given a formula φ whose free variables are among $\mathsf{x}_1, \ldots, \mathsf{x}_n$, the propositions below are equivalent: (1) φ is satisfiable, (2) $\exists\, \mathsf{x}_1, \ldots, \mathsf{x}_n\, \varphi$ is valid, (3) $\exists\, \mathsf{x}_1, \ldots, \mathsf{x}_n\, \varphi$ is equivalent to $\top$.

Theorem 1 (Decidability). *The satisfiability problem for (PA) is decidable.*

The decidability of Presburger arithmetic is due to [Pre29]. The restriction to quantifier-free formulae is NP-complete [Pap81] (see also [BT76,Sch86]) whereas an exact complexity charaterisation for the full logic is provided in [Ber80] (double exponential time on alternating Turing machines with linear amounts of alternations). Many other complexity results exist for fragments; for instance for any $i > 0$, the Σ_{i+1} fragment of (PA) is Σ_i^{EXP}-complete [Haa14] where the Σ_{i+1} fragment of (PA) consists of formulae with $i + 1$ quantifier alternations beginning with an existential quantifier and Σ_i^{EXP} is the ith level of the weak EXP hierarchy.

2.3 What is reversal-boundedness ?

As it is well-known, most reachability problems for counter machines are undecidable since counter machines are Turing-complete devices. A way to overcome this negative result is to restrict the class of runs for such machines so that decidability can be regained. An obvious way to restrict the runs is to require that the length of the runs is bounded by a value b. Other types of bound exist and in the class of *reversal-bounded counter machines* considered in the paper, the runs are restricted differently so that the number of *reversals* in a run is bounded by a value r. A reversal for a counter occurs in a run when there is an alternation from a nonincreasing mode to a nondecreasing mode and vice-versa.

For instance, in the sequence below, there are three reversals identified by an upper line:

$$0011223334444\overline{3}33222\overline{3}33444455555\overline{4}.$$

Similarly, the sequence 00111222223333334444 has no reversal. A counter machine is *reversal-bounded* whenever there is $r \geq 0$ such that for all the runs from a given initial configuration, every counter makes no more than r reversals. A formal definition will follow, but before going any further, it is worth pointing out a few peculiarities of this subclass. Firstly, reversal-boundedness is defined for initialised counter machines (a counter machine augmented with an initial configuration) and the bound r usually depends on the initial configuration. Secondly, this class is not defined from the full class of counter machines by imposing syntactic restrictions but rather semantically. For example, in flat counter machines [DDS15], the syntactic restriction consists in requiring that the control graph of the counter machine is flat, i.e., every control state belongs to at most one simple cycle. Similarly, VASS are counter machines for which the guards are equal to the truth constant.

A major property of such reversal-bounded counter machines is that reachability sets are computable Presburger sets (Theorem 11). We present a proof of this result in the paper that is different from the proof in [Iba78] and that partly relies on developments from [GI81]. Apart from presenting this essential property, the paper investigates other decidability/complexity issues summarised as follows.

1. The reachability problem for counter machines with a given bounded number of reversals is NEXPTIME-complete. Note that the decidability was already a consequence of the fact that reachability sets of reversal-bounded counter machines can be effectively represented by Presburger formulae.
2. The control state repeated reachability problem with bounded number of reversals and the $\exists$-Presburger infinitely often problem are shown decidable (and NEXPTIME-complete), see e.g. [DIP01].
3. All above-mentioned decidability problems are obtained with counter machines in which a counter value can only be compared to a constant and guards are closed under Boolean connectives. We also explain why the reachability problem with bounded number of reversals becomes undecidable if equalities and inequalities between counters are allowed in guards.
4. We present a sufficient condition for the decidability of model-checking reversal-bounded counter machines with specifications expressed in the linear-time temporal logic CLTL. The condition deals with reversal-boundedness for terms occurring in the input. NEXPTIME-completeness is actually stated.

The class of reversal-bounded counter machines has been introduced and studied in [Iba78], partly inspired by similar restrictions on multistack machines [BB74]. Theorem 11 that states that every reachability set of an initialised reversal-bounded counter machine is a Presburger set, is shown in [Iba78]. Reversal-bounded multipushdown machines that extend reversal-bounded counter machines with stacks have been also studied in [BB74,Iba78]. For instance, developments about multihead pushdown machines recognizing bounded languages [Iba74] can lead to semilinearity of reachability sets for initialised reversal-bounded counter machines. In terms of language acceptance, Parikh automata recognize the same languages as the languages accepted by reversal-bounded counter machines, see e.g. [KR02,CFM11] but herein we mainly focus on reachability-like properties.

The proof of Theorem 11 presented in the paper relies on developments from [GI81] and it uses a proof *à la Rackoff* [Rac78] (see Theorem 17). Even though the main intentions in [GI81] are related to optimal complexity upper bounds, semilinearity of reachability sets can be derived too. It is worth noting that the class of counter machines considered in the paper is a bit larger than the class considered in [Iba78]. Indeed, we allow comparisons between a counter value and any constant as well as any Boolean combination and the updates are those from VASS instead of being restricted to updates in $\{-1, 0, +1\}$.

The notion of reversal-boundedness from [Iba78] has been also relaxed, for example by allowing a free counter [Iba78,HR87] (i.e., one counter has no constraints on the number of reversals) or by counting the reversals only above a given bound [FS08,San08]. In both cases, semilinearity of the reachability sets is still preserved [Iba78,FS08]. The results in [GI81] are extended to the case with a single free counter in [HR87]. Reversal-bounded counter machines have been also studied as computational devices, see e.g. [Cha81]. Moreover, decidable reachability problems for parameterised reversal-bounded initialised counter machines can be found in [ISD$^+$02, Section 4].

2.4 A formal definition for reversal-boundedness

Let $\mathcal{M} = \langle Q, T, C \rangle$ be a counter machine in $\mathcal{C}$. We recall that a run ρ is defined as a sequence of configurations $c_0 = \langle q_0, \boldsymbol{x}_0 \rangle, \ldots, c_k, \ldots$ respecting the transitions in T (each c_i is in $Q \times \mathbb{N}^d$). We allow runs made of a unique configuration; similarly we may also add the transitions between the configurations to emphasize which transitions have been fired, as in the enriched run $c_0 \xrightarrow{t_1} c_1 \xrightarrow{t_2} c_2 \cdots$. The length of the run ρ, written $\mid \rho \mid$, is defined as the number of steps, typically $\mid c_0 \cdots c_k \mid = k$ and $\mid c_0 \mid = 0$. From a run

$$\rho = \langle q_0, \boldsymbol{x}_0 \rangle \xrightarrow{t_1} \langle q_1, \boldsymbol{x}_1 \rangle, \ldots$$

of $\mathcal{M}$, in order to describe the behavior of counters varying along ρ, we define a sequence of *mode vectors* $\mathfrak{md}_0, \mathfrak{md}_1, \ldots$ such that each $\mathfrak{md}_i$ belongs to $\{\uparrow, \downarrow\}^d$. Intuitively, each value in a mode vector records whether a counter is currently in an increasing phase or in a decreasing phase. We are now ready to define the sequence $\mathfrak{md}_0, \mathfrak{md}_1, \ldots$ associated with ρ.

- By convention, $\mathfrak{md}_0$ is the unique vector in $\{\uparrow\}^d$.
- For all $j \geq 0$ and for all $i \in [1, d]$, we have
  1. $\mathfrak{md}_{j+1}(i) \stackrel{\text{def}}{=} \mathfrak{md}_j(i)$ when $\boldsymbol{x}_j(i) = \boldsymbol{x}_{j+1}(i)$,
  2. $\mathfrak{md}_{j+1}(i) \stackrel{\text{def}}{=} \uparrow$ when $\boldsymbol{x}_{j+1}(i) - \boldsymbol{x}_j(i) > 0$,
  3. $\mathfrak{md}_{j+1}(i) \stackrel{\text{def}}{=} \downarrow$ when $\boldsymbol{x}_{j+1}(i) - \boldsymbol{x}_j(i) < 0$.

Now, let $Rev_i \stackrel{\text{def}}{=} \{j \in [0, \mid \rho \mid -1] : \mathfrak{md}_j(i) \neq \mathfrak{md}_{j+1}(i)\}$.

Definition 2. *Let $\mathcal{M} = \langle Q, T, C \rangle$ be a counter machine with $d \geq 1$ counters, $i \in [1, d]$ and $r \in \mathbb{N}$. The run ρ is* r-reversal-bounded $\overset{\text{def}}{\Leftrightarrow}$ *for every $i \in [1, d]$, we have $\mathrm{card}(Rev_i) \leq r$. An initialised counter machine $\langle \mathcal{M}, \langle q, \boldsymbol{x} \rangle \rangle$ is* r-reversal-bounded $\overset{\text{def}}{\Leftrightarrow}$ *every run from $\langle q, \boldsymbol{x} \rangle$ is r-reversal-bounded. An initialised counter machine $\langle \mathcal{M}, \langle q, \boldsymbol{x} \rangle \rangle$ is* reversal-bounded $\overset{\text{def}}{\Leftrightarrow}$ *there is some $r \geq 0$ such that every run from $\langle q, \boldsymbol{x} \rangle$ is r-reversal-bounded.*

Let us consider the counter machine $\mathcal{M}$ from Figure 1. The initialised counter machine $\langle \mathcal{M}, \langle q_1, \mathbf{0} \rangle \rangle$ is reversal-bounded. Actually, one can show that for all $\boldsymbol{y} \in \mathbb{N}^2$, $\langle \mathcal{M}, \langle q_1, \boldsymbol{y} \rangle \rangle$ is reversal-bounded too. Interestingly, let φ be the formula below:

$$\varphi = (\mathsf{x}_1 \geq 2 \wedge \mathsf{x}_2 \geq 1 \wedge \mathsf{x}_2 + 1 \geq \mathsf{x}_1) \vee (\mathsf{x}_2 \geq 2 \wedge \mathsf{x}_1 \geq 1 \wedge \mathsf{x}_1 + 1 \geq \mathsf{x}_2).$$

One can prove that the set of reachable configurations with control state q_9 from the initial configuration $\langle q_1, \mathbf{0} \rangle$ can be characterised by the formula φ as far as the counter values are concerned, i.e.

$$[\![\varphi]\!] = \{\boldsymbol{y} \in \mathbb{N}^2 : \langle q_1, \mathbf{0} \rangle \xrightarrow{*} \langle q_9, \boldsymbol{y} \rangle\}.$$

The purpose of Section 3 is to provide a method for computing the formula φ in the general case. Reversal-boundedness for counter machines is indeed very appealing because reachability sets are semilinear. Given an initialised counter machine $\langle \mathcal{M}, \langle q, \boldsymbol{x} \rangle \rangle$ that is r-reversal-bounded for some $r \geq 0$, for each state q', $\{\boldsymbol{y} \in \mathbb{N}^d : \langle q, \boldsymbol{x} \rangle \xrightarrow{*} \langle q', \boldsymbol{y} \rangle\}$ is a computable Presburger set, see Theorem 11. This means that one can compute effectively a Presburger formula that characterises precisely the reachable configurations whose state is q'.

A counter machine $\mathcal{M}$ is *uniformly reversal-bounded* iff there is $r \geq 0$ such that for every initial configuration, the initialised counter machine is r-reversal-bounded. One can check that the counter machine in Figure 1 is not uniformly reversal-bounded. Indeed, any configuration that reaches the control state q_{11} leads to non-reversal-boundedness because of the cycle between the control states q_{10} and q_{11} that increments and decrements the first counter. Similarly, any configuration with control state q_6 and such that the second counter is equal to zero leads to non-reversal-boundedness.

In the sequel, when we consider a uniformly reversal-bounded counter machine or a reversal-bounded initialised counter machine, it comes with a maximal number of reversals $r \geq 0$. Alternatively, given an arbitrary counter machine and a bound $r \geq 0$, it is possible to restrict it appropriately and to build a new counter machine such that each counter has at most r reversals on each run, possibly at the cost of increasing exponentially the cardinality of the set of control states (see the proof of Theorem 14). It is sufficient to take the product between $\mathcal{M}$ and a finite-state automaton with a number of control states in $\mathcal{O}(r^d)$ where d is the number of counters, the details are omitted.

3 Synthesis of Presburger formulae

In this section, we show that reachability sets in (initialised) reversal-bounded counter machines are computable Presburger sets. Moreover, when uniform reversal-boundedness is satisfied, one can show that the reachability relation is also computable and semilinear. Effectiveness refers here to the possibility to construct Presburger formulae defining exactly those sets or binary relations. We believe that the proof below is conceptually simple, which is the motivation to include it in the document.

Let $\mathcal{M} = \langle Q, T, C \rangle$ be a counter machine with d counters. Without any loss of generality, we can assume that the guards in $\mathcal{M}$ are negation-free. Indeed, $\neg(\mathsf{x} \geq k)$ is equivalent to $\mathsf{x} \leq k - 1$ when $k > 0$ (otherwise if $k = 0$, then it is equivalent to $\perp$). Similarly, $\neg(\mathsf{x} \leq k)$ is equivalent to $\mathsf{x} \geq k + 1$.

3.1 Paths, simple loops and effects

A *path* π is a finite sequence of transitions from T of the form

$$q_1 \xrightarrow{\langle \mathtt{g}_1, \boldsymbol{a}_1 \rangle} q_1', \ldots, q_n \xrightarrow{\langle \mathtt{g}_n, \boldsymbol{a}_n \rangle} q_n'$$

so that for every $i \in [1, n-1]$, we have $q_i' = q_{i+1}$. Let $\pi = t_1 \cdots t_n$ be a path such that each transition t_j has the update $\boldsymbol{a}_j \in \mathbb{Z}^d$. The *effect* of π is the tuple $\mathfrak{ef}(\pi) \overset{\text{def}}{=} \sum_j \boldsymbol{a}_j \in \mathbb{Z}^d$. A *simple loop sl* is a non-empty path that starts and ends by the same state and these are the only states that are repeated in sl. We say that sl *loops* on its first state (equal to its last state). The number of simple loops is therefore bounded by $\mathrm{card}(T)^{\mathrm{card}(Q)}$. We assume an arbitrary total linear ordering $\prec$ on simple loops.

We write $\mathfrak{sc}(\mathcal{M})$ to denote the maximal absolute value among the updates $\boldsymbol{a}$ in $\mathcal{M}$. The value $\mathfrak{sc}(\mathcal{M})$ is called the *scale* of $\mathcal{M}$. Assuming that the size of $\mathcal{M}$ is N, we have $\mathfrak{sc}(\mathcal{M}) \leq 2^N$ (all the integers in $\mathcal{M}$ are encoded with a binary representation).

Lemma 3. *The effect $\mathfrak{ef}(sl)$ of a simple loop sl is in*

$$[-\mathrm{card}(Q)\mathfrak{sc}(\mathcal{M}), \mathrm{card}(Q)\mathfrak{sc}(\mathcal{M})]^d.$$

This means also that the number of effects from simple loops in $\mathcal{M}$ is bounded by $(1 + 2 \times \mathrm{card}(Q)\mathfrak{sc}(\mathcal{M}))^d$.

3.2 Intervals

For defining reversal-bounded runs, we have seen that mode vectors play an important role since in such runs, the number of changes for mode vectors is bounded by definition. However, this has another consequence for the satisfaction of the guards. Indeed, assuming that the constants occurring in guards are $k_1 < k_2 < \cdots < k_K$, this defines a finite amount of intervals with these extremal values and the number of changes for intervals for each counter is bounded in reversal-bounded runs. Below, we introduce several notions about such intervals.

We write AG to denote the (finite) set of atomic guards of the form $\mathsf{x} \sim k$ occurring in $\mathcal{M}$ and $\mathrm{CST} = \{k_1, \ldots, k_K\}$ to denote the set of distinct constants k occurring in atomic guards of the form $\mathsf{x} \sim k$ in AG, possibly augmented with the value zero. We pose that $K = \mathrm{card}(\mathrm{CST})$. Below, we assume that $0 = k_1 < k_2 < \cdots < k_K$ and we write $\mathcal{I}$ to denote the set of intervals below:

$$\mathcal{I} \overset{\text{def}}{=} \{[k_1, k_1], [k_1 + 1, k_2 - 1], [k_2, k_2], [k_2 + 1, k_3 - 1], [k_3, k_3], \ldots,$$

$$[k_K, k_K], [k_K + 1, +\infty)\} \setminus \{\emptyset\}.$$

The empty set is possibly removed since $[k_j + 1, k_{j+1} - 1]$ is empty when $k_{j+1} = k_j + 1$. So, $\mathcal{I}$ contains at most $2K$ intervals and at least $K + 1$ intervals. Furthermore, we consider a natural discrete linear ordering $\leq$ on the intervals in $\mathcal{I}$ so that

$$[k_1, k_1] \ \leq \ [k_1 + 1, k_2 - 1] \leq [k_2, k_2] \leq [k_2 + 1, k_3 - 1] \leq [k_2, k_2] \leq \ldots \leq$$

$$\leq [k_K, k_K] \leq [k_K + 1, +\infty)\}.$$

The above relationships should be understood to hold only if the intervals are nonempty. An *interval map* $\mathfrak{im}$ is a map of the form $\mathfrak{im} : C \to \mathcal{I}$ understood as a symbolic way to represent counter values. This abstracts a map $C \to \mathbb{N}$ by only taking into account to which elements of $\mathcal{I}$ each counter value belongs to, which is sufficient to compare counter values against constants from CST.

Given a guard g built from AG and an interval map $\mathfrak{im}$, we write $\mathfrak{im} \vdash \mathsf{g}$ with the following inductive definition:

$$\begin{array}{lcl}
\mathfrak{im} \vdash \mathsf{g}_1 \vee \mathsf{g}_2 & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{im} \vdash \mathsf{g}_1 \text{ or } \mathfrak{im} \vdash \mathsf{g}_2 \\
\mathfrak{im} \vdash \mathsf{g}_1 \wedge \mathsf{g}_2 & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{im} \vdash \mathsf{g}_1 \text{ and } \mathfrak{im} \vdash \mathsf{g}_2 \\
\mathfrak{im} \vdash \mathsf{x} = k & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{im}(\mathsf{x}) = [k, k] \\
\mathfrak{im} \vdash \mathsf{x} \geq k & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{im}(\mathsf{x}) \subseteq [k, +\infty) \\
\mathfrak{im} \vdash \mathsf{x} \leq k & \overset{\text{def}}{\Leftrightarrow} & \mathfrak{im}(\mathsf{x}) \subseteq [0, k]
\end{array}$$

There is no clause for negated guards because we have seen that negation can be discarded without causing any harm.

The relation $\vdash$ is simply a symbolic satisfaction relation between interval maps and guards. Since interval maps and guards are built over the same set CST of constants, completeness is obtained as stated in the properties below:

($\mathcal{P}_1$) $\mathfrak{im} \vdash \mathsf{g}$ can be checked in polynomial time in the sum of the respective sizes of $\mathfrak{im}$ and g (for some reasonably succinct encoding in which natural numbers can be encoded in binary).

($\mathcal{P}_2$) $\mathfrak{im} \vdash \mathsf{g}$ iff (for all $\mathfrak{f} : C \to \mathbb{N}$ and for all $\mathsf{x} \in C$, we have $\mathfrak{f}(\mathsf{x}) \in \mathfrak{im}(\mathsf{x})$) implies $\mathfrak{f} \models \mathsf{g}$ (in Presburger arithmetic).

A *guarded mode* $\mathfrak{gmd}$ is a pair $\langle \mathfrak{im}, \mathfrak{md} \rangle$ where $\mathfrak{im}$ is an interval map and $\mathfrak{md} \in \{\uparrow, \downarrow\}^d$. A transition $t = q \xrightarrow{\langle \mathsf{g}, \boldsymbol{a} \rangle} q'$ is *compatible* with $\mathfrak{gmd} = \langle \mathfrak{im}, \mathfrak{md} \rangle \overset{\text{def}}{\Leftrightarrow}$

1. $\mathfrak{im} \vdash \mathsf{g}$,
2. for every $i \in [1, d]$,
   - $\mathfrak{md}(i) = \uparrow$ implies $\boldsymbol{a}(i) \geq 0$,
   - $\mathfrak{md}(i) = \downarrow$ implies $\boldsymbol{a}(i) \leq 0$.

3.3 Extended paths: a schema to denote a class of paths

A path $\pi = t_1 \cdots t_n$ is a finite sequence of transitions that can be fired or not from a given configuration. It is often useful to describe a (possibly infinite) family of paths and a natural choice consists in using regular expressions on the finite alphabet of transitions. For instance, semilinear regular expressions (SLRE) are quite convenient to perform accelerations on labelled transition systems, see e.g. [FIS03]. Below, we introduce the notion of extended path, so that a finite number of such structures is quite close to SLREs. An extended path is basically a finite repetition of a path followed by a set of simple loops or a set of simple loops followed by a path so that the control states match between paths and simple loops. A formal definition is provided below. An *extended path* (or a *path schema*) $\mathbf{P}$ is an expression of the form below:

$$\pi_0 \; S_1 \; \pi_1 \; \cdots \; S_\alpha \; \pi_\alpha$$

where the S_i's are (finite) non-empty sets of simple loops, the π_i's are non-empty paths and

1. if S occurs just before a path π, then all the simple loops in S loop on the first state of π,
2. similarly, if S occurs just after a path π, then all the simple loops in S loop on the last state of π.

An extended path generalises the notion of path in which simple loops in the sets S_i's can be visited an arbitrary non-zero number of times while respecting the arbitrary linear ordering on simple loops. A guarded mode induces a restriction of a counter machine by considering only a subset of transitions from $\mathcal{M}$, namely those that are compatible with the guarded mode. We say that a path [resp. simple loop, extended path] is *compatible* with the guarded mode $\mathfrak{gmd} = \langle \mathfrak{im}, \mathfrak{md} \rangle \overset{\text{def}}{\Leftrightarrow}$ all its transitions are compatible with $\mathfrak{gmd}$.

Given an extended path $\mathbf{P}$, we introduce a few more notions.

- The *skeleton* of $\mathbf{P}$ is the path $\pi_0 \cdots \pi_\alpha$.
- Given a set of simple loops $S = \{sl_1, \ldots, sl_m\}$ with $sl_1 \prec \cdots \prec sl_m$, we write $e(S)$ to denote the regular expression

$$(sl_1)^+ \cdots (sl_m)^+.$$

  So, each simple loop is taken at least once. Indeed, we want to make explicit that each simple loop sl is used in S, otherwise it is always possible to exclude sl from S, leading to another legitimate extended path (assuming that $S \setminus \{sl\}$ is non-empty). We write $e(\mathbf{P})$ to denote the regular expression defined as follows:

$$\pi_0 \cdot e(S_1) \cdots e(S_\alpha) \cdot \pi_\alpha.$$

Furthermore, we write $\mathrm{L}(e)$ to denote the language (with alphabet T) generated by the regular expression e. One can observe that $\mathrm{L}(e(\mathbf{P}))$ is a regular language for any extended path $\mathbf{P}$. For the sake of simplicity, we write $\mathrm{L}(\mathbf{P})$ instead of $\mathrm{L}(e(\mathbf{P}))$.

Example 1. Let us consider the counter machine in Figure 3. From the five transitions (we have omitted to specify what are the guards and update vectors since this is irrelevant), we can build the following extended path

$$\mathbf{P} \overset{\text{def}}{=} t_0 \cdot t_1 \cdot \{t_1, t_2\} \cdot t_3 \cdot \{t_4, t_5\} \cdot t_4 \cdot t_5 \cdot t_5.$$

Indeed, t_1, t_2, t_4 and t_5 are simple loops, and $t_0 t_1$, t_3 and $t_4 t_5 t_5$ are paths. Assuming that $t_1 \prec t_2$ and $t_5 \prec t_4$, we get that

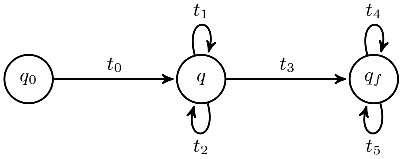

Fig. 3. A three-state counter machine

$$e(\mathbf{P}) = t_0 \cdot t_1 \cdot t_1^+ t_2^+ \cdot t_3 \cdot t_5^+ t_4^+ \cdot t_4 \cdot t_5 \cdot t_5.$$

Let $\mathbf{P}$ be an extended path compatible with a guarded mode $\mathfrak{gmd}$ and $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \xrightarrow{t_1} \cdots \xrightarrow{t_\ell} \langle q_\ell, \boldsymbol{x}_\ell \rangle$ be a finite run. The run ρ *respects* the extended path $\mathbf{P} \overset{\text{def}}{\Leftrightarrow} \pi = t_1 \cdots t_\ell \in \mathrm{L}(\mathbf{P})$ and for all $i \in [0, \ell - 1]$ and all $j \in [1, d]$, we have $\boldsymbol{x}_i(j) \in \mathfrak{im}(\mathsf{x}_j)$. Note that it may happen that for some $j \in [1, d]$, $\boldsymbol{x}_\ell(j)$ may not belong to $\mathfrak{im}(\mathsf{x}_j)$.

3.4 Runs in normal form

Given a non-empty r-reversal-bounded run $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \xrightarrow{t_1} \cdots \xrightarrow{t_\ell} \langle q_\ell, \boldsymbol{x}_\ell \rangle$, we aim at showing that the path $\pi = t_1 \cdots t_\ell$ can be transformed into π' leading to the run ρ', and π' belongs to a bounded regular language (the initial and last configurations of ρ are preserved too). To do so, we divide the run ρ into several subruns such that the number of reversals on each subrun is zero (i.e., a reversal can only occur when passing from one subrun to a next one) and moreover, all the counter values of each subrun satisfy exactly the same atomic guards. That is why we have introduced the notion of guarded mode since it contains an interval map and a mode for each counter in C.

A *global phase* is a finite sequence of transitions such that there is a mode vector $\mathfrak{md} \in \{\uparrow, \downarrow\}^d$ for which each transition in it is compatible with $\langle \mathfrak{im}, \mathfrak{md} \rangle$, for some interval map $\mathfrak{im}$. So, in a run respecting a global phase, the number of reversals is zero for all the counters and a global phase depends only on a single mode vector.

Lemma 4. *Any r-reversal-bounded run $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \cdots \langle q_\ell, \boldsymbol{x}_\ell \rangle$ can be divided as a sequence of subruns $\rho = \rho_1 \cdot \rho_2 \cdots \rho_L$ such that each ρ_i respects a global phase and* $L \leq (d \times r) + 1$.

The proof is by an easy verification. Note that $\rho \cdot \rho'$ is a run, assuming that the last configuration of ρ is equal to the first configuration of ρ' and the concatenation merges these two occurrences. A *local phase* is a finite sequence of transitions such that there is a guarded mode $\langle \mathfrak{im}, \mathfrak{md} \rangle$ for which each transition in it is compatible with $\langle \mathfrak{im}, \mathfrak{md} \rangle$. Hence, in a run respecting a local phase, not only the number of reversals is zero for all the counters but also the counter values satisfy the same atomic guards.

Lemma 5. *Any r-reversal-bounded run $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \cdots \langle q_\ell, \boldsymbol{x}_\ell \rangle$ can be divided as a sequence of subruns $\rho = \rho_1 \cdot \rho_2 \cdots \rho_{L'}$ such that each ρ_i respects a local phase and $L' \leq ((d \times r) + 1) \times 2Kd$.*

Proof. By Lemma 4, we have seen that ρ can be divided in at most $(d \times r) + 1$ subruns respecting a global phase. It remains to show that each such subrun can be divided in at most $2Kd$ subruns respecting a local phase. Actually, this is due to the following property.

Let $\boldsymbol{a} \in \mathbb{Z}^d$ be an update vector. We define the binary relation $\preceq_{\boldsymbol{a}}$ on the set of interval maps so that $\mathfrak{im} \preceq_{\boldsymbol{a}} \mathfrak{im}' \overset{\text{def}}{\Leftrightarrow}$ for every $i \in [1, d]$, we have

- $\mathfrak{im}(\mathsf{x}_i) \leq \mathfrak{im}'(\mathsf{x}_i)$ if $\boldsymbol{a}(i) \geq 0$,
- $\mathfrak{im}'(\mathsf{x}_i) \leq \mathfrak{im}(\mathsf{x}_i)$ if $\boldsymbol{a}(i) \leq 0$,
- $\mathfrak{im}'(\mathsf{x}_i) = \mathfrak{im}(\mathsf{x}_i)$ if $\boldsymbol{a}(i) = 0$.

We write $\mathfrak{im} \prec_{\boldsymbol{a}} \mathfrak{im}'$ when $\mathfrak{im} \preceq_{\boldsymbol{a}} \mathfrak{im}'$ and $\mathfrak{im} \neq \mathfrak{im}'$. The property $(\mathcal{P}_3)$ below states that sequences of strictly increasing interval maps have polynomially bounded length, even though the number of interval maps is in $\mathcal{O}(K^d)$.

$(\mathcal{P}_3)$ Let $\boldsymbol{a} \in \mathbb{Z}^d$ and $\mathfrak{im}_1 \prec_{\boldsymbol{a}} \mathfrak{im}_2 \prec_{\boldsymbol{a}} \cdots \prec_{\boldsymbol{a}} \mathfrak{im}_\beta$. Then, $\beta \leq 2Kd$.

Indeed, in a subrun respecting a global phase, each counter is compared against at most K constants and all the counters have a monotone behaviour (in increasing mode or in decreasing mode). Hence, each counter during the global phase can visit at most $2K$ distinct intervals in $\mathcal{I}$, whence the bound $2Kd$ for the maximal number of local phases.

Below, a *sequence* of extended paths is understood as being of the form $\mathbf{P}_1 \cdots \mathbf{P}_{L'}$ with the proviso that each $\mathbf{P}_i$ is an extended path compatible with some guarded mode and the expression $\mathbf{P}_1 \cdots \mathbf{P}_{L'}$ can be also viewed as an extended path by itself (possibly by concatenating paths), i.e. it is compatible with the control graph of $\mathcal{M}$.

Lemma 6. *Any r-reversal-bounded run $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \cdots \langle q_\ell, \boldsymbol{x}_\ell \rangle$ respects a sequence of extended paths $\mathbf{P}_1 \cdots \mathbf{P}_{L'}$ with $L' \leq ((d \times r) + 1) \times 2Kd$.*

A *small* extended path $\pi_0 \ S_1 \ \pi_1 \ \cdots \ \pi_{\alpha-1} \ S_\alpha \ \pi_\alpha$, with $\alpha \geq 1$, is an extended path such that

1. π_α have at most $\mathrm{card}(Q) + 2$ transitions,
2. $\pi_0, \ldots, \pi_{\alpha-1}$ have at most $\mathrm{card}(Q)$ transitions,
3. for each state $q \in Q$, there is at most one set S containing simple loops on q.

So, the length of the skeleton is bounded by $\mathrm{card}(Q)^2 + \mathrm{card}(Q) + 2$. Note that the set of small extended paths is finite, even though its cardinality can be exponential in the size of $\mathcal{M}$. We also consider *degenerated* small extended paths made of paths of length at most $2\mathrm{card}(Q)+2$ (i.e. without any set S_γ). Usually, this case is omitted in the proofs since it can be easily obtained from the non-degenerated case (i.e. when $\alpha \geq 1$).

Lemma 7. *Let $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \cdots \langle q_\ell, \boldsymbol{x}_\ell \rangle$ be a run respecting an extended path $\mathbf{P}$ compatible with some guarded mode $\mathfrak{gmd}$. Then, there is small extended path $\mathbf{P}'$ compatible with $\mathfrak{gmd}$ and a run ρ' with the same initial and final configurations such that ρ' respects $\mathbf{P}'$.*

The proof provides an algorithm to transform $\mathbf{P}$ to $\mathbf{P}'$ by replacing sequences of transitions by simple loops and termination is guaranteed because a given simple loop can be used more than once.

Proof. Let $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \xrightarrow{t_1} \cdots \xrightarrow{t_\ell} \langle q_\ell, \boldsymbol{x}_\ell \rangle$ be a run of $\mathcal{M}$ respecting an extended path $\mathbf{P}$ compatible with some guarded mode $\mathfrak{gmd}$. So, this means that $\pi = t_1 \cdots t_\ell \in \mathrm{L}(\mathbf{P})$ and for all $i \in [0, \ell - 1]$ and all $j \in [1, d]$, we have $\boldsymbol{x}_i(j) \in \mathtt{im}(\mathsf{x}_j)$. We shall build a small extended path $\mathbf{P}'$ such that $\mathbf{P}'$ is compatible with $\mathfrak{gmd}$ and there is a run ρ' respecting $\mathbf{P}'$ that starts and ends by the same configurations as ρ.

To do so, we define a sequence of extended paths $\mathbf{P}_0$, $\mathbf{P}_1$, ..., $\mathbf{P}_\beta$ (iteratively) such that

- all the $\mathbf{P}_i$'s are extended paths compatible with $\mathfrak{gmd}$ and there is a run ρ_i respecting $\mathbf{P}_i$ that starts and ends by the same configurations,
- $\mathbf{P}_0$ is equal to $t_1 \cdots t_\ell$ viewed as an extended path,
- $\mathbf{P}_\beta$ is a small extended path,
- $\mathbf{P}_{i+1}$ is obtained from $\mathbf{P}_i$ by removing a simple loop on a control state, say q, and possibly adding it to a set of simple loops S already in $\mathbf{P}_i$ or by creating one if none exists.

At the end of this process, the extended path $\mathbf{P}_\beta$ is small and there is a run ρ_β respecting $\mathbf{P}_\beta$ that starts by $\langle q_0, \boldsymbol{x}_0 \rangle$ and ends by $\langle q_\ell, \boldsymbol{x}_\ell \rangle$.

It remains to explain how to build $\mathbf{P}_{i+1}$ from $\mathbf{P}_i$. We assume that $\mathbf{P}_i$ has the form below

$$\pi_0 \; S_1 \; \pi_1 \; \cdots \; S_\alpha \; \pi_\alpha$$

where

(1.) $\alpha \leq \mathrm{card}(Q)$,
(2.) $\pi_0, \ldots, \pi_{\alpha-1}$ have at most $\mathrm{card}(Q)$ transitions,
(3.) for each state $q \in Q$, there is at most one set S_γ, $\gamma \in [1, \alpha]$, containing simple loops on q.

Obviously, $\mathbf{P}_0$ verifies these conditions since it is degenerated. The extended path $\mathbf{P}_{i+1}$ shall satisfy the same condition except that we require that the length of the final path of $\mathbf{P}_{i+1}$ strictly decreases. Now, let us define $\mathbf{P}_{i+1}$ from $\mathbf{P}_i$.

Case 1: $\mathbf{P}_i$ is a small extended path. We are done and $\mathbf{P}_i$ is the final extended path of the sequence.

Case 2: $\pi_\alpha = \pi \cdot sl \cdot \pi'$ where sl is a simple loop on q, $\pi \neq \varepsilon$, $\pi' \neq \varepsilon$ and S_γ already contains simple loops on q ($\gamma \leq \alpha$). Then, $\mathbf{P}_{i+1}$ is equal to the extended path below:

$$\pi_0 \ \cdots \ S_{\gamma-1} \, \pi_{\gamma-1} \, (S_\gamma \cup \{sl\}) \ \cdots \ \pi_{\alpha-1} \, S_\alpha \, (\pi\pi').$$

Case 3: $\pi_\alpha = \pi \cdot sl \cdot \pi'$ where sl is a simple loop on q and the first one occurring in $\pi \cdot sl$, $\pi \neq \varepsilon$, $\pi' \neq \varepsilon$ and no S_γ already contains simple loops on q. Then, $\mathbf{P}_{i+1}$ is equal to the extended path below:

$$\pi_0 \ \cdots \ S_\alpha \, \pi \, \{sl\} \, \pi'.$$

In that case, we create a new set of simple loop(s).

It remains to show that there is a run ρ_{i+1} respecting $\mathbf{P}_{i+1}$ that starts by $\langle q_0, \boldsymbol{x}_0 \rangle$ and ends by $\langle q_\ell, \boldsymbol{x}_\ell \rangle$. Satisfaction of the conditions (1.)–(3.) above is by an easy verification. In order to show the former property, we use the fact that all the transitions in $\mathbf{P}_{i+1}$ are compatible with $\mathfrak{gmd}$ (by construction), the counter values have a monotone behaviour (increasing mode or decreasing mode) and the atomic guards are convex (by definition).

We deal with the Case 2 below, Case 3 admits a similar analysis, which is left as an exercise. Let ρ_i be a run respecting $\mathbf{P}_i$, starting by the configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ and ending by the configuration $\langle q_\ell, \boldsymbol{x}_\ell \rangle$. The extended path $\mathbf{P}_i$ is of the form

$$\pi_0 \, S_1 \, \pi_1 \ \cdots \ S_\alpha \, (\pi \cdot sl \cdot \pi').$$

Suppose that $S_\gamma = S_\gamma^1 \uplus S_\gamma^2$, and for all $sl' \in S_\gamma^1$ [resp. $sl' \in S_\gamma^2$], we have $sl' \prec sl$ [resp. $sl \prec sl'$] (we have assumed an arbitrary total ordering on the set of simple loops). Since $\mathbf{P}_i$ is compatible with the guarded mode $\mathfrak{gmd} = \langle \mathfrak{im}, \mathfrak{md} \rangle$, for every $j \in [1, d]$, we have:

- $\mathfrak{md}(j) = \uparrow$ implies that for all counter values $\boldsymbol{x} \in \mathbb{N}^d$ occurring in the run ρ_i, we get that $\boldsymbol{x}_0(j) \leq \boldsymbol{x}(j) \leq \boldsymbol{x}_\ell(j)$,
- $\mathfrak{md}(j) = \downarrow$ implies that for all counter values $\boldsymbol{x} \in \mathbb{N}^d$ occurring in the run ρ_i, we get that $\boldsymbol{x}_\ell(j) \leq \boldsymbol{x}(j) \leq \boldsymbol{x}_0(j)$.

Moreover, assuming that $\boldsymbol{y} \in \mathbb{N}^d$ is the penultimate vector of counter values in ρ_i, we know that for all counter values $\boldsymbol{x} \in \mathbb{N}^d$ occurring in the run ρ_i until that occurrence of $\boldsymbol{y}$, for every atomic guard $\mathsf{x}_j \sim k$ in AG, we have $\mathfrak{im} \vdash \mathsf{x}_j \sim k$ iff $\boldsymbol{x}(j) \sim k$ iff $\boldsymbol{x}_0(j) \sim k$ iff $\boldsymbol{y}(j) \sim k$ (partly thanks to the property ($\mathcal{P}_2$)).

In order to build a run ρ_{i+1} similar to ρ_i that respects $\mathbf{P}_{i+1}$, that starts by $\langle q_0, \boldsymbol{x}_0 \rangle$ and ends by $\langle q_\ell, \boldsymbol{x}_\ell \rangle$, we decompose the run ρ_i in the following way so that ρ_{i+1} can be then easily defined.

The run ρ_i can be divided as follows. Each subrun $\rho_i^\star$ respects a factor of $\mathbf{P}_i$ (we are a bit liberal here with the notion of respect):

$$\rho_i = \overbrace{\rho_1^\star}^{\pi_0 \ \cdots \ S_{\gamma-1} \ \pi_{\gamma-1} \ S_\gamma^1} \cdot \overbrace{\rho_2^\star}^{S_\gamma^2 \ \pi_\gamma \cdots \pi_{\alpha-1} \ S_\alpha \ \pi} \cdot \overbrace{\rho_3^\star}^{sl} \cdot \overbrace{\rho_4^\star}^{\pi'}$$

For each subrun $\rho_j^\star$, we write $\langle q_0^j, \boldsymbol{x}_0^j \rangle$ [resp. $\langle q_f^j, \boldsymbol{x}_f^j \rangle$] to denote its first [resp. last] configuration. In order to build ρ_i', we introduce two sequences of configurations $\rho_3^{\star\star}$

and $\rho_2^{+\mathfrak{ef}(sl)}$ that will happen to be runs too. $\rho_3^{\star\star}$ is the sequence of configurations obtained from the final configuration $\langle q_f^1, \boldsymbol{x}_f^1 \rangle$ by firing the transitions of the simple loop sl. Similarly, $\rho_2^{+\mathfrak{ef}(sl)}$ is the sequence of configurations obtained from the last configuration of $\rho_3^{\star\star}$ by firing the sequence of transitions used for $\rho_2^{\star}$. Observe that $\rho_2^{\star}$ and $\rho_2^{+\mathfrak{ef}(sl)}$ have the same length and for any configuration $\langle q, \boldsymbol{x} \rangle$ in $\rho_2^{\star}$, say at position h, the configuration at position h in $\rho_2^{+\mathfrak{ef}(sl)}$ is exactly $\langle q, \boldsymbol{x} + \mathfrak{ef}(sl) \rangle$.

Let us consider the sequence of configurations ρ_{i+1} as defined below:

$$\rho_{i+1} = \overbrace{\rho_1^{\star}}^{\pi_0 \ \cdots \ S_{\gamma-1} \ \pi_{\gamma-1} \ S_\gamma^1} \cdot \overbrace{\rho_3^{\star\star}}^{sl} \cdot \overbrace{\rho_2^{+\mathfrak{ef}(sl)}}^{S_\gamma^2 \ \pi_\gamma \cdots \pi_{\alpha-1} \ S_\alpha \ \pi} \cdot \overbrace{\rho_4^{\star}}^{\pi'}$$

Note that the sequence of configurations respects the updates on the transitions. In order to check that ρ_{i+1} is a run, it remains to show that transitions in $\rho_3^{\star\star}$ and in $\rho_2^{+\mathfrak{ef}(sl)}$ can be fired by respecting the guards. Suppose that $\mathfrak{md}(j) = \uparrow$ for some $j \in [1, d]$ (the case $\mathfrak{md}(j) = \downarrow$ admits a similar development). Every tuple $\boldsymbol{y} \in \mathbb{N}^d$ from a configuration in $\rho_3^{\star\star}$ satisfies the following inequations:

$$\boldsymbol{x}_0(j) = \boldsymbol{x}_0^1(j) \leq \boldsymbol{x}_f^1(j) \leq \boldsymbol{y}(j) \leq \boldsymbol{x}_0^4(j) \leq \boldsymbol{x}_f^4(j) = \boldsymbol{x}_\ell(j).$$

By convexity of the atomic guards $\mathsf{x}_j \sim k$ in AG, $\boldsymbol{y}(j) \sim k$ iff $\boldsymbol{y'}(j) \sim k$ where $\boldsymbol{y'}$ is the corresponding vector of counter values in the run $\rho_3^{\star}$ (at the same position). So, $\rho_3^{\star\star}$ is indeed a run of $\mathcal{M}$ respecting sl. Similary, one can show that $\rho_2^{+\mathfrak{ef}(sl)}$ is a run respecting $S_\gamma^2 \pi_\gamma \cdots \pi_{\alpha-1} \ S_\alpha \ \pi$, which concludes the proof. □

Example 2. Let us consider the counter machine described in Example 1. Below, we present a path where, as usual, $(t)^i$ represents i successive copies of t.

$$t_0 \cdot (t_1)^7 \cdot (t_2)^7 (t_1)^8 \cdot t_3 \cdot (t_4)^7 \cdot (t_5)^7 \cdot (t_4)^8.$$

Following the steps in the proof of Lemma 7, we present below the extended path obtained at the 22nd step and the one at the 38th step that corresponds to a small extended path.

- $\mathbf{P}_{22} = t_0 \cdot \{t_1, t_2\} \cdot t_3 \cdot (t_4)^7 \cdot (t_5)^7 \cdot (t_4)^8$.
- $\mathbf{P}_{38} = t_0 \cdot \{t_1, t_2\} \cdot t_3 \cdot \{t_4, t_5\} \cdot (t_4)^6$.

A *small* sequence of extended paths is a sequence of small extended paths $\mathbf{P}_1 \cdots \mathbf{P}_{L'}$ such that $L' \leq ((d \times r) + 1) \times 2Kd$. Not only each extended path of the sequence is compatible with a unique guarded mode but also the extended path is itself small.

Theorem 8. *For any r-reversal-bounded run $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \cdots \langle q_\ell, \boldsymbol{x}_\ell \rangle$, there is an r-reversal-bounded run ρ' between the same configurations that respects a small sequence of extended paths.*

Let $\mathbf{P}$ be a small extended path compatible with the guarded mode $\mathfrak{gmd} = \langle \mathfrak{im}, \mathfrak{md} \rangle$, say $\mathbf{P}$ is of the form below:

$$\pi_0 \ \{sl_1^1, \ldots, sl_1^{n_1}\} \ \pi_1 \ \cdots \ \{sl_\alpha^1, \ldots, sl_\alpha^{n_\alpha}\} \ \pi_\alpha,$$

where q_0 is the first control state in π_0 and q_f is the last control state in π_α.

Lemma 9. *There is a Presburger formula* $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_d, \mathsf{y}_1, \ldots, \mathsf{y}_d)$ *of exponential size in* $\mathrm{size}(\mathcal{M})$ *such that* $[\![\varphi]\!] = \{\langle \boldsymbol{x}_0, \boldsymbol{y}\rangle : \text{ there is a run } \langle q_0, \boldsymbol{x}_0\rangle \xrightarrow{*} \langle q_f, \boldsymbol{y}\rangle \text{ respecting } \mathbf{P}\}$.

The size of $\mathcal{M}$, written $\mathrm{size}(\mathcal{M})$, is defined with an unspecified succinct encoding in which the integers are encoded with a binary representation.

Proof. Let $\pi_\alpha = \pi'_\alpha \cdot t$, so that t is the last transition of π_α. The formula φ states the following properties:

1. the initial counter values belong to the appropriate intervals induced by $\mathfrak{im}$,
2. the counter values for the penultimate configuration $\langle q'_f, \boldsymbol{y}'\rangle$ belong to the right intervals induced by $\mathfrak{im}$,
3. the values for $\bar{\mathsf{y}}$ are obtained from $\bar{\mathsf{x}}$ by considering the effects of the paths π_i plus a finite amount of times the effects of each simple loop occurring in $\mathbf{P}$.

Note that (1) and (2) are sufficient to guarantee that every other configuration $\boldsymbol{x}$ in the run $\langle q_0, \boldsymbol{x}_0\rangle \xrightarrow{*} \langle q_f, \boldsymbol{y}\rangle$, possibly except $\boldsymbol{y}$, belong to the right intervals induced by $\mathfrak{im}$. Indeed, if $\mathfrak{md}(i) = \uparrow$, then $\boldsymbol{x}_0(i) \leq \boldsymbol{x}(i) \leq \boldsymbol{y}'(i)$ and by convexity of the guards in AG, we get that $\boldsymbol{x}(i)$ satisfies the same atomic guards. A similar analysis can be made when $\mathfrak{md}(i) = \downarrow$.

So, the formula φ is of the form below:

$$\exists\ \mathsf{z}_1^1, \ldots, \mathsf{z}_1^{n_1}, \ldots, \mathsf{z}_\alpha^1, \ldots, \mathsf{z}_\alpha^{n_\alpha}$$

$$(\mathsf{z}_1^1 \geq 1) \wedge \cdots \wedge (\mathsf{z}_1^{n_1} \geq 1) \wedge \cdots \wedge (\mathsf{z}_\alpha^1 \geq 1) \wedge \cdots \wedge (\mathsf{z}_\alpha^{n_\alpha} \geq 1) \wedge$$

$$(\bar{\mathsf{y}} = \bar{\mathsf{x}} + \mathfrak{ef}(\pi_0) + \cdots + \mathfrak{ef}(\pi_\alpha) + \sum_{i,j} \mathsf{z}_i^j \mathfrak{ef}(sl_i^j)) \wedge$$

$$(\bigwedge_{\mathfrak{im} \vdash \mathsf{x}_c \sim k} \mathsf{x}_c \sim k) \wedge (\bigwedge_{\text{not } \mathfrak{im} \vdash \mathsf{x}_c \sim k} \neg(\mathsf{x}_c \sim k)) \wedge$$

$$(\bigwedge_{\mathfrak{im} \vdash \mathsf{x}_c \sim k} (\mathsf{x}_c + \mathfrak{ef}(\pi_0)(c) + \cdots + \mathfrak{ef}(\pi_{\alpha-1})(c) + \mathfrak{ef}(\pi'_\alpha)(c) + \sum_{i,j} \mathsf{z}_i^j \mathfrak{ef}(sl_i^j)(c)) \sim k) \wedge$$

$$(\bigwedge_{\text{not } \mathfrak{im} \vdash \mathsf{x}_c \sim k} \neg(\mathsf{x}_c + \mathfrak{ef}(\pi_0)(c) + \cdots + \mathfrak{ef}(\pi_{\alpha-1})(c) + \mathfrak{ef}(\pi'_\alpha)(c) + \sum_{i,j} \mathsf{z}_i^j \mathfrak{ef}(sl_i^j)(c) \sim k)).$$

□

The formula φ in Lemma 9 has size polynomial in $\mathrm{size}(\mathcal{M})$ and in the size of $\mathbf{P}$. The size of $\mathbf{P}$ is itself exponential in $\mathrm{size}(\mathcal{M})$. This is the best we can hope for since the number of simple loops can be obviously exponential in the size of the control graph of $\mathcal{M}$.

Lemma 10. *Let* $\mathbf{P}_1 \cdots \mathbf{P}_{L'}$ *be a sequence of small extended paths. There is a Presburger formula* $\varphi(\bar{\mathsf{x}}, \bar{\mathsf{y}})$ *such that*

$$[\![\varphi]\!] = \{\langle \boldsymbol{x}, \boldsymbol{y}\rangle : \text{ there is a run } \langle q_0, \boldsymbol{x}\rangle \xrightarrow{*} \langle q_f, \boldsymbol{y}\rangle \text{ respecting } \mathbf{P}_1 \cdots \mathbf{P}_{L'}\}.$$

Proof. The proof is by an easy verification by using the formulae from Lemma 9 (typically L' times) and by taking advantage of existential first-order quantifications for $(L' - 1)$ intermediary configurations. Indeed, for each small extended path $\mathbf{P}_i$, let $\varphi_i(\bar{\mathsf{x}}, \bar{\mathsf{y}})$ be the formula constructed in the proof of Lemma 9. The formula φ is then defined as follows:

$$\exists\, \bar{\mathsf{z}}_0, \ldots, \bar{\mathsf{z}}_{L'}\ (\bar{\mathsf{x}} = \bar{\mathsf{z}}_0) \wedge (\bar{\mathsf{y}} = \bar{\mathsf{z}}_{L'}) \wedge \bigwedge_{i=0}^{L'-1} \varphi_{i+1}(\bar{\mathsf{z}}_i, \bar{\mathsf{z}}_{i+1}).$$

□

The formula φ in Lemma 10 is of exponential size in $log(r) + \mathrm{size}(\mathcal{M})$ since counting until $L' \leq ((d \times r) + 1) \times 2Kd$ requires $\mathcal{O}(\log(r) + \mathrm{size}(\mathcal{M})^2)$ bits and each formula φ_j is of exponential size in $\mathrm{size}(\mathcal{M})$. Since the number of small sequences of extended paths is finite and actually double exponential in $log(r) + \mathrm{size}(\mathcal{M})$, we get the following theorem.

Theorem 11. *[Iba78] Let $\langle \mathcal{M}, \langle q_0, \boldsymbol{x} \rangle \rangle$ be an initialised counter machine that is r-reversal-bounded for some $r \geq 0$. For each state $q_f \in Q$, the set $\{\boldsymbol{y} \in \mathbb{N}^d : \langle q_0, \boldsymbol{x} \rangle \xrightarrow{*} \langle q_f, \boldsymbol{y} \rangle\}$ is a computable Presburger set.*

Theorem 11 is clearly a consequence of Theorem 8 and of its corollaries.

Proof. Let us consider the formula $\varphi(\bar{\mathsf{y}})$ below:

$$\exists\, \bar{\mathsf{x}}\ (\bigwedge_{i \in [1,d]} \boldsymbol{x}(i) = \mathsf{x}_i) \wedge \bigvee_{\text{small seq. } \sigma = \mathbf{P}_1 \cdots \mathbf{P}_{L'}} \varphi_\sigma(\bar{\mathsf{x}}, \bar{\mathsf{y}}),$$

where $\varphi_\sigma(\bar{\mathsf{x}}, \bar{\mathsf{y}})$ is the Presburger formula for the small sequence of extended paths $\sigma = \mathbf{P}_1 \cdots \mathbf{P}_{L'}$ obtained from Lemma 10. Moreover, in the disjunction, we assume that $\mathbf{P}_1 \cdots \mathbf{P}_{L'}$ starts by the state q_0 and ends by the state q_f. Moreover, if $q_f = q_0$, we add the disjunct $(\bigwedge_{i \in [1,d]} \boldsymbol{x}(i) = \mathsf{y}_i)$.

Finally, note that the generalised disjunction is finite since the number of small sequences of extended paths is finite and bounded by $2^{2^{p(log(r), \mathrm{size}(\mathcal{M}))}}$ for some polynomial $p(\cdot, \cdot)$. Indeed, a small sequence has length at most $((d \times r) + 1) \times 2Kd$ and the number of small extended paths is double exponential in the size of $\mathcal{M}$. Remember that in an extended path, each set S may contain an exponential amount of simple loops. □

Theorem 12. *Let $\mathcal{M}$ be a counter machine that is uniformly r-reversal-bounded for some $r \geq 0$. For all the states q_0, q_f, one can compute a Presburger formula $\varphi(\bar{\mathsf{x}}, \bar{\mathsf{y}})$ such that $[\![\varphi]\!] = \{\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{N}^{2d} : \langle q_0, \boldsymbol{x} \rangle \xrightarrow{*} \langle q_f, \boldsymbol{y} \rangle\}$.*

Indeed, it is sufficient to consider the formula $\bigvee_{\text{small seq. } \sigma = \mathbf{P}_1 \cdots \mathbf{P}_{L'}} \varphi_\sigma(\bar{\mathsf{x}}, \bar{\mathsf{y}})$ from the proof of Theorem 11.

As a corollary of the previous developments, we can obtain the following classical result. Let $\Sigma = \{a_1, \ldots, a_k\}$ be a finite alphabet equipped with an arbitrary linear ordering on the letters, say $a_1 < \cdots < a_k$. Given a word $w \in \Sigma^*$, its *Parikh image* is defined as a tuple $\mathit{\Pi}(w) \in \mathbb{N}^k$ such that for every $i \in [1,k]$, $\mathit{\Pi}(w)(i)$ is the number of occurrences of the letter a_i in the word w. Naturally, the *Parikh image* of the language $\mathrm{L} \subseteq \Sigma^*$ is the set $\{\mathit{\Pi}(w) \in \mathbb{N}^k : w \in \mathrm{L}\}$. Parikh's remarkable result states that the Parikh image of any context-free language is semilinear [Par66] and that its representation is effectively computable from a pushdown automaton or from a context-free grammar. The above technique provides a proof for this result for the restricted class of regular languages.

Theorem 13. *[BL81, Theorem 2] For every regular language* L*, there is a regular language* L' *defined as a finite union of regular languages obtained from regular expressions of the form* $u_0 v_0^* u_1 v_1^* \cdots u_m v_m^* u_{m+1}$ *(the* u_i*'s and* v_i*'s are finite words) such that* $\mathrm{L}' \subseteq \mathrm{L}$ *and* $\mathit{\Pi}(\mathrm{L}') = \mathit{\Pi}(\mathrm{L})$.

Consequently, $\mathit{\Pi}(\mathrm{L})$ is a Presburger set since it is quite easy to define a Presburger formula defining $\mathit{\Pi}(u_0 v_0^* u_1 v_1^* \cdots u_m v_m^* u_{m+1})$. In order to build L' from L, it is sufficient to take a finite-state automaton $\mathcal{A}$ accepting L, to add a counter by letter and to increment it whenever a letter is read. This provides a 0-reversal-bounded counter machine such that the union of the reachability sets for accepting states from $\mathcal{A}$ provides $\mathit{\Pi}(\mathrm{L})$. Moreover, the finite amount of small extended paths allows to define the finite union of regular expressions of the form $u_0 v_0^* u_1 v_1^* \cdots u_m v_m^* u_{m+1}$, leading to L'.

3.5 Reachability problem with bounded number of reversals

Let us consider the following reachability problem that takes as input an *arbitrary* counter machine and a bound r intended to enforce the maximal number of reversals by counter, which corresponds to a perspective a bit different from what has been done earlier.
REACHABILITY PROBLEM WITH BOUNDED NUMBER OF REVERSALS:

Input: a counter machine $\mathcal{M}$, a bound $r \in \mathbb{N}$, an initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ and a final configuration $\langle q_f, \boldsymbol{x}_f \rangle$,

Question: Is there a finite run of $\mathcal{M}$ with initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ and final configuration $\langle q_f, \boldsymbol{x}_f \rangle$ such that each counter has at most r reversals?

Observe that when $\langle \mathcal{M}, \langle q_0, \boldsymbol{x}_0 \rangle \rangle$ is r'-reversal-bounded for some $r' \leq r$, we get an instance of the reachability problem with initial configuration $\langle q_0, \boldsymbol{x}_{\mathbf{0}} \rangle$.

In the input, we include the configurations $\langle q_0, \boldsymbol{x}_0 \rangle$ and $\langle q_f, \boldsymbol{x}_f \rangle$ since we ask for a reachability question between configurations. Remember that the counter machines do not come with initial/final control states or even initial/final configurations. So, this needs to be specified in the input.

Theorem 14 below is another consequence of Theorem 8. Its proof shows how to transform a counter machine into an r-reversal-bounded counter machine whose runs are exactly the r-reversal-bounded runs of the original counter machine.

Theorem 14. *The reachability problem with bounded number of reversals is decidable.*

Proof. Here is the decidability proof that uses Theorem 11. Let $\mathcal{M} = \langle Q, T, C\rangle$, $r \in \mathbb{N}$, $\langle q_0, \boldsymbol{x}_0\rangle$ and $\langle q_f, \boldsymbol{x}_f\rangle$ be an instance of the problem. First, we build a variant counter machine $\mathcal{M}' = \langle Q', T', C\rangle$ with $Q' = Q \times \{\uparrow, \downarrow\}^d \times [0, r]^d$.

By construction of $\mathcal{M}'$, we guarantee that $\langle \mathcal{M}', \langle\langle q_0, \uparrow, \boldsymbol{0}\rangle, \boldsymbol{x}_0\rangle\rangle$ is r-reversal-bounded ($\uparrow$ stands for the constant mode vector with the value $\uparrow$ only). Indeed, each counter records the number of reversals and by construction of $\mathcal{M}'$ we enforce that it is bounded by r on each run. The set of transitions T' is defined as follows: $\langle q, \mathfrak{md}, \sharp\boldsymbol{alt}\rangle \xrightarrow{\langle \mathfrak{g}, \boldsymbol{a}\rangle} \langle q', \mathfrak{md}', \sharp\boldsymbol{alt}'\rangle \in T' \overset{\text{def}}{\Leftrightarrow} q \xrightarrow{\langle \mathfrak{g}, \boldsymbol{a}\rangle} q' \in T$ and for every $i \in [1, d]$, the relation described by the following table is verified. The values of two first columns induce values for the two last columns (when it is possible, see e.g. the condition $\sharp\boldsymbol{alt}(i) < r$).

$\boldsymbol{a}$	$\mathfrak{md}(i)$	$\mathfrak{md}'(i)$	$\sharp\boldsymbol{alt}'(i)$
$\boldsymbol{a}(i) < 0$	$\downarrow$	$\downarrow$	$\sharp\boldsymbol{alt}(i)$
$\boldsymbol{a}(i) < 0$	$\uparrow$	$\downarrow$	$\sharp\boldsymbol{alt}(i) + 1$ and $\sharp\boldsymbol{alt}(i) < r$
$\boldsymbol{a}(i) > 0$	$\uparrow$	$\uparrow$	$\sharp\boldsymbol{alt}(i)$
$\boldsymbol{a}(i) > 0$	$\downarrow$	$\uparrow$	$\sharp\boldsymbol{alt}(i) + 1$ and $\sharp\boldsymbol{alt}(i) < r$
$\boldsymbol{a}(i) = 0$	$\downarrow$	$\downarrow$	$\sharp\boldsymbol{alt}(i)$
$\boldsymbol{a}(i) = 0$	$\uparrow$	$\uparrow$	$\sharp\boldsymbol{alt}(i)$

By construction, $\mathcal{M}'$ is uniformly r-reversal-bounded and the properties below are equivalent:

1. there is a run of $\mathcal{M}$ with initial configuration $\langle q_0, \boldsymbol{x}_0\rangle$ and final configuration $\langle q_f, \boldsymbol{x}_f\rangle$ such that each counter has at most r reversals,
2. $\langle\langle q_f, \mathfrak{md}, \sharp\boldsymbol{alt}\rangle, \boldsymbol{x}_f\rangle$ is reachable from $\langle\langle q_0, \uparrow, \boldsymbol{0}\rangle, \boldsymbol{x}_{\boldsymbol{0}}\rangle$ in $\mathcal{M}'$ for some $\mathfrak{md}$ and $\sharp\boldsymbol{alt}$.

The number of distinct pairs $\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle$ is bounded by $2^d \times (r+1)^d$ and therefore (1.) is equivalent to the existence of $\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle$ among a finite set such that

3. $\langle\langle q_f, \mathfrak{md}, \sharp\boldsymbol{alt}\rangle, \boldsymbol{x}_f\rangle$ is reachable from $\langle\langle q_0, \uparrow, \boldsymbol{0}\rangle, \boldsymbol{x}_{\boldsymbol{0}}\rangle$ in $\mathcal{M}'$.

By Theorem 11, the set

$$X_{\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle} = \{\boldsymbol{x}' \in \mathbb{N}^d : \langle\langle q_0, \uparrow, \boldsymbol{0}\rangle, \boldsymbol{x}_0\rangle \xrightarrow{*} \langle\langle q_f, \mathfrak{md}, \sharp\boldsymbol{alt}\rangle, \boldsymbol{x}'\rangle\}$$

is a computable Presburger set. This means that one can construct a Presburger formula $\varphi_{\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle}$ such that $[\![\varphi_{\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle}]\!] = X_{\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle}$ and checking whether $\boldsymbol{x} \in X_{\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle}$ amounts to verify the satisfiability of the formula

$$(\bigwedge_{i=1}^{d} \mathsf{x}_i = \boldsymbol{x}(i)) \wedge \varphi_{\langle \mathfrak{md}, \sharp\boldsymbol{alt}\rangle}.$$

Since the satisfiability problem for Presburger arithmetic is decidable, we get an algorithm to solve the reachability problem with a bounded number of reversals. Indeed, it amounts to checking satisfiability of some Presburger formula made of a disjunction with at most $2^d(r+1)^d$ disjuncts. □

The proof of Theorem 14 is interesting but does not help much to understand the computational complexity of the reachability problem with bounded number of reversals. However, the complexity can be nailed down thanks to the following developments. First, let us make use of Lemma 10.

Lemma 15. *If there is a run from $\langle q_0, \boldsymbol{x}_0 \rangle$ to $\langle q_f, \boldsymbol{x}_f \rangle$ such that each counter has at most r reversals, then there is an r-reversal-bounded run between these configurations respecting a small sequence of extended paths such that each simple loop is visited at most a doubly-exponential number of times in $log(r) + \mathrm{size}(\boldsymbol{x}_0) + \mathrm{size}(\boldsymbol{x}_f) + \mathrm{size}(\mathcal{M})$.*

The size of $\boldsymbol{x} \in \mathbb{N}^d$ is defined so that $\mathrm{size}(\boldsymbol{x}) \in \mathcal{O}(d \times log(m))$ where m is the maximal value among the components of $\boldsymbol{x}$. Similarly, the size of a counter machine $\mathcal{M}$ uses a reasonably succinct encoding with integers encoded in binary.

Proof. Let ρ be an r-reversal-bounded run from $\langle q_0, \boldsymbol{x}_0 \rangle$ to $\langle q_f, \boldsymbol{x}_f \rangle$. By Theorem 8, there is an r-reversal-bounded run ρ' between the same configurations that respects a small sequence of extended paths $\mathbf{P}_1 \cdots \mathbf{P}_{L'}$.

Let $\varphi(\bar{\mathsf{x}}, \bar{\mathsf{y}})$ be the Presburger formula for that sequence. The formula $\varphi(\bar{\mathsf{x}}, \bar{\mathsf{y}})$ is equivalent to an existential formula (in prenex normal form, only the existential quantifier occurs) and it is of size exponential in $log(r) + \mathrm{size}(\mathcal{M})$. Note that most of the existentially quantified variables are related to the number of times simple loops are visited. So, the formula

$$(\bigwedge_{j \in [1,d]} (\mathsf{x}_j = \boldsymbol{x}_0(j) \wedge \mathsf{y}_j = \boldsymbol{x}_f(j))) \wedge \varphi(\bar{\mathsf{x}}, \bar{\mathsf{y}})$$

is satisfiable, which is equivalent to the satisfiability of a quantifier-free formula φ' by removing the quantifications. The formula φ' is satisfiable with values at most exponential in its size (see Section 2.2). Consequently, each simple loop is visited at most a doubly-exponential amount of times. □

Since in a small sequence of extended paths, there are at most $((d \times r) + 1) \times 2Kd$ extended paths, and each extended path has at most $\mathrm{card}(T)^{\mathrm{card}(Q)}$ simple loops and at most $\mathrm{card}(Q)^2 + \mathrm{card}(Q) + 2$ transitions, that do not occur in simple loops, if there is an r-reversal-bounded run from $\langle q_0, \boldsymbol{x}_0 \rangle$ to $\langle q_f, \boldsymbol{x}_f \rangle$, then there is such a run of length at most double exponential in $log(r) + \mathrm{size}(\mathcal{M}) + \mathrm{size}(\boldsymbol{x}_0) + \mathrm{size}(\boldsymbol{x}_f)$. This means that a nondeterministic exponential space algorithm can guess such a run and therefore the reachability problem with bounded number of reversals is in EXPSPACE by Savitch's Theorem [Sav70]. This can be improved: the runs are much more structured, which allows us to show NEXPTIME-completeness.

Theorem 16. *The reachability problem with bounded number of reversals is in* NEXPTIME *assuming that all natural numbers are encoded in binary.*

Proof. Let $\mathcal{M}$, r, $\langle q_0, \boldsymbol{x}_0 \rangle$ and $\langle q_f, \boldsymbol{x}_f \rangle$ be an instance of size N for the reachability problem with bounded number of reversals. We have $N \in \mathcal{O}(\mathrm{size}(\mathcal{M}) + log(r) + \mathrm{size}(\boldsymbol{x}_0) + \mathrm{size}(\boldsymbol{x}_f))$. We have seen that there is an r-reversal-bounded run from $\langle q_0, \boldsymbol{x}_0 \rangle$ to $\langle q_f, \boldsymbol{x}_f \rangle$ iff there is an r-reversal-bounded run ρ between these configurations that respects a small sequence of extended paths, that is of length at most double exponential in N and each simple loop on that sequence is taken at most a double exponential number of times.

Such a sequence has at most $((d \times r) + 1) \times 2Kd$ small extended paths. Each extended path is compatible with a guarded mode and it has at most $\mathrm{card}(T)^{\mathrm{card}(Q)}$ simple loops and at most $1 + \mathrm{card}(Q)$ paths of length at most $3 \times \mathrm{card}(Q)$. These are rough bounds that also take into account the degenerated small extended paths.

The NEXPTIME algorithm below guesses on-the-fly the small sequence of extended paths and computes the effect of taking a path of length at most $3 \times \mathrm{card}(Q)$ or a simple loop compatible with a guarded mode a double exponential number of times. Computing the effect of taking such an amount of times a simple loop can be computed in exponential time because the natural numbers are encoded in binary. We do not compute the full run but only the intermediate configurations after firing a path or a simple loop.

The number of paths of length at most $3 \times \mathrm{card}(Q)$ or the number of simple loops visited along the small sequence of extended paths is bounded by:

$$G = ((d \times r) + 1) \times 2Kd \times (\mathrm{card}(T)^{\mathrm{card}(Q)} + \mathrm{card}(Q) + 1).$$

Here is the algorithm:

```
1. $\langle q_{\mathrm{cur}}, \boldsymbol{x}_{\mathrm{cur}} \rangle := \langle q_0, \boldsymbol{x}_0 \rangle$; Guess $\alpha \leq G$; $\beta := 1$;
2. While $\beta \leq \alpha$ do
   (a) Guess either a path $\pi$ of length at most $3 \times \mathrm{card}(Q)$ or, a simple loop $sl$ and a guarded mode $\mathfrak{gmd} = \langle \mathfrak{im}, \mathfrak{md} \rangle$ and $\gamma$ of double exponential value in $N$ such that $sl$ is compatible with $\mathfrak{gmd}$;
   (b) If a simple loop is guessed in (a), then check that $\boldsymbol{x}_{\mathrm{cur}}$ and $\boldsymbol{x}_{\mathrm{cur}} + (\gamma - 1)\mathfrak{ef}(sl) + \mathfrak{ef}(sl^{\backslash \mathrm{last}})$ are in the right intervals: for every $i \in [1, d]$, $\boldsymbol{x}_{\mathrm{cur}}(i)$ and $(\boldsymbol{x}_{\mathrm{cur}} + (\gamma - 1)\mathfrak{ef}(sl) + \mathfrak{ef}(sl^{\backslash \mathrm{last}}))(i)$ belong to $\mathfrak{im}(\mathsf{x}_i)$ where $sl^{\backslash \mathrm{last}}$ equals $sl$ minus its last transition.
   (c) If a path $\pi$ is guessed in (a), then check that the sequence of transitions in $\pi$ can be fired from $\langle q_{\mathrm{cur}}, \boldsymbol{x}_{\mathrm{cur}} \rangle$ and set $\langle q_{\mathrm{cur}}, \boldsymbol{x}_{\mathrm{cur}} \rangle := \langle q_{\mathrm{cur}}, \boldsymbol{x}_{\mathrm{cur}} \rangle + \mathfrak{ef}(\pi)$.
   (d) $\beta := \beta + 1$;
3. Return ($\langle q_{\mathrm{cur}}, \boldsymbol{x}_{\mathrm{cur}} \rangle = \langle q_f, \boldsymbol{x}_f \rangle$).
```

Checking that the algorithm runs in nondeterministic exponential-time is then by an easy verification. What is missing above, is a means to check that the number of reversals is indeed bounded by r and this can be done similarly to what is presented in the proof of Theorem 14. So strictly speaking, the above algorithm should be completed and additional variables should be introduced to count the number of reversals per counter. □

Theorem 17. *The reachability problem with bounded number of reversals is* NEXPTIME-*complete.*

Even though the reachability problem with bounded number of reversals is not explicitly considered in [GI81,HR87], the proof for the complexity lower bound in

Theorem 17 is due to [HR87] whereas the proof for the complexity upper bound in Theorem 17 is due to [GI81] (and it uses small solutions of inequation systems as in [Rac78]). The lower bound is obtained by encoding computations of a Turing machine running in nondeterministic exponential time by a counter machine, by using ideas similar to [Min67]. The reachability problem with bounded number of reversals is in NP when the number of reversals r is encoded in unary [GI81]. As for the reachability problem for integer VASS, see e.g. [HH14], the NP upper bound follows from the fact that satisfiability of existential Presburger formulae can be witnessed by small solutions of polynomial size in the size of the input formula.

3.6 Decidable repeated reachability problems

In this section, we show how to reduce the control state repeated reachability problem to the reachability problem when reversal-bounded counter machines are involved. Let us consider the following problem.

CONTROL STATE REPEATED REACHABILITY PROBLEM WITH BOUNDED NUMBER OF REVERSALS:

Input: a counter machine $\mathcal{M}$, a bound $r \in \mathbb{N}$, an initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ and an accepting control state q_f,

Question: Is there an infinite run of $\mathcal{M}$ with initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ such that each counter has at most r reversals and q_f is repeated infinitely often?

Before explaining how to solve the above problem, let us introduce a simple variant for which we already have decidability.

CONTROL STATE REACHABILITY PROBLEM WITH BOUNDED NUMBER OF REVERSALS:

Input: a counter machine $\mathcal{M}$, a bound $r \in \mathbb{N}$, an initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ and an accepting control state q_f,

Question: Is there a finite run of $\mathcal{M}$ with initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ such that each counter has at most r reversals and the final control state is q_f?

As a consequence of Theorem 11 and Theorem 14, we get the following result.

Lemma 18. *Control state reachability problem with bounded number of reversals is decidable.*

Indeed, reachability sets are computable Presburger sets. Let us take advantage of this to establish Lemma 19.

Lemma 19. *[DIP01] Control state repeated reachability problem with bounded number of reversals is decidable.*

The proof of Lemma 19 reduces repeated reachability to reachability by observing that on an infinite r-reversal-bounded run, after some position ℓ, each counter remains constant or increases. Since the control state q_f is repeated infinitely often, one can find two positions $\ell \leq \ell_1 < \ell_2$ that visit q_f and so the subrun between the positions ℓ_1 and ℓ_2 can be repeated infinitely often, assuming that the counters that increase after the position ℓ have a value greater than any constant occurring in guards in the machine. The details are provided in the proof below.

Proof. Let $\langle \mathcal{M}, \langle q_0, \boldsymbol{x}_0 \rangle \rangle$ be an initialised counter machine, $r \geq 0$, with $\mathcal{M} = \langle Q, T, C \rangle$ and $q_f \in Q$.

We propose an algorithm to answer the following question: is there an infinite r-reversal-bounded run starting at $\langle q_0, \boldsymbol{x}_0 \rangle$ such that the control state q_f is repeated infinitely often? We reduce it to an instance of the control state reachability problem with bounded number of reversals, which is decidable by Lemma 18. Let $k_{max} \in \mathbb{N}$ denote the maximal constant k occurring in an atomic guard of the form $\mathsf{x} \sim k$ in $\mathcal{M}$.

Let ($\star$) be the desired property:

($\star$) There is an r-reversal-bounded infinite run from $\langle q_0, \boldsymbol{x}_0 \rangle$ such that q_f is repeated infinitely often.

Let ($\star\star$) be the property specified below:

($\star\star$) There exist an r-reversal-bounded finite run $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \xrightarrow{t_1} \langle q_1, \boldsymbol{x}_1 \rangle \cdots \xrightarrow{t_\ell} \langle q_l, \boldsymbol{x}_\ell \rangle$, $\ell' \in [0, \ell - 1]$ and $C_= \subseteq C$ such that
(a) $q_\ell = q_{\ell'} = q_f$,
(b) for all $\mathsf{x}_i \in C_=$ and $j \in [\ell' + 1, \ell]$, $\boldsymbol{x}_{j-1}(i) = \boldsymbol{x}_j(i)$,
(c) for all $\mathsf{x}_i \in (C \setminus C_=)$ and $j \in [\ell' + 1, \ell]$, $\boldsymbol{x}_{j-1}(i) \leq \boldsymbol{x}_j(i)$,
(d) for all $\mathsf{x}_i \in (C \setminus C_=)$, we have $k_{max} < \boldsymbol{x}_{\ell'}(i)$,
(e) for all $\mathsf{x}_i \in C_=$, have $\boldsymbol{x}_{\ell'}(i) \leq k_{max}$.

Below, we show that ($\star$) and ($\star\star$) are equivalent, which allows us to reduce control state repeated reachability to control state reachability. Indeed, checking ($\star\star$) amounts to introduce 2^d copies of $\mathcal{M}$ (one for each subset of C).

First, let us show that ($\star$) and ($\star\star$) are equivalent. Suppose ($\star$). There exists an infinite r-reversal-bounded run $\rho = \langle q_0, \boldsymbol{x}_0 \rangle \xrightarrow{t_1} \langle q_1, \boldsymbol{x}_1 \rangle \xrightarrow{t_2} \langle q_2, \boldsymbol{x}_2 \rangle \cdots$ such that q_f is repeated infinitely often. Let $C_=^\rho$ be the subset of C that contains exactly the counters whose values are less or equal to k_{max}, apart from a finite prefix. Since ρ is r-reversal-bounded, there exists $I \geq 0$ such that for some $n \geq I$, no counters in $C \setminus C_=^\rho$ is decremented and their values are strictly greater than k_{max} and all the counters in $C_=^\rho$ have a constant value less or equal to k_{max}. Since q_f is repeated infinitely often, there are $I \leq \ell' < \ell$ such that $q_\ell = q_{\ell'} = q_f$ and (b)-(e) hold.

Now suppose that there exist an r-reversal-bounded finite run

$$\rho = \langle q_0, \boldsymbol{x}_0 \rangle \xrightarrow{t_1} \langle q_1, \boldsymbol{x}_1 \rangle \cdots \xrightarrow{t_\ell} \langle q_\ell, \boldsymbol{x}_\ell \rangle,$$

$\ell' \in [0, \ell - 1]$ and $C_= \subseteq C$ witnessing the satisfaction of ($\star\star$). It is then easy to show that the ω-sequence of transitions $t_1 \cdots t_{\ell'} (t_{\ell'+1} \cdots t_\ell)^\omega$ allows us to define an infinite r-reversal-bounded run ρ' that extends ρ. It is clear that in the run ρ' the control state q_f is repeated infinitely often. Guards on transitions are satisfied by the counter values because of conditions (c), (d) and (e) and values for counters in $(C \setminus C_=)$ are non-negative thanks to (c) and (d).

We construct a reversal-bounded counter machine $\mathcal{M}' = \langle Q', T', C \rangle$ such that ($\star\star$) iff there is a finite r-reversal-bounded run from $\langle q_0, \boldsymbol{x}_0 \rangle$ that reaches the control state q_{new}. It remains to define the counter machine $\mathcal{M}'$. It is made of the original version of $\mathcal{M}$ (called below the *original copy*) augmented with 2^d copies of $\mathcal{M}$; each copy $\mathcal{M}_i$ with $i \in [0, 2^d - 1]$ corresponds to the set of counters $C_=^i \stackrel{\text{def}}{=} \{\mathsf{x}_j \mid j\text{th bit of } i \text{ is } 1\}$ that plays the role of $C_=$ in ($\star\star$). By $\mathcal{M}_i$, we mean the restriction of $\mathcal{M}$ such that:

– no transition modifies a counter from $C^i_=$,
– no transition decrements a counter in $(C \setminus C^i_=)$.

For each $i \in [0, 2^d - 1]$, the control states of $\mathcal{M}_i$ are pairs in $(Q \uplus \{q_f^{\mathrm{in}}\}) \times \{i\}$. The second component simply indicates to which copy belongs the control state.

In order to simulate the subrun $\langle q_{\ell'}, \boldsymbol{x}_{\ell'} \rangle \cdots \langle q_\ell, \boldsymbol{x}_\ell \rangle$ for the satisfaction of $(\star\star)$ in $\mathcal{M}$, we nondeterministically move from the original copy to some copy $\mathcal{M}_i$ in $\mathcal{M}'$ (and therefore we choose which counters remain constant below k_{max} forever). To do so, for every $i \in [0, 2^d - 1]$, we consider in $\mathcal{M}'$ a transition from the control state q_f in the original copy to $\langle q_f^{\mathrm{in}}, i \rangle$ whose task is to check that

1. all counters in $C^i_=$ have values less or equal to k_{max},
2. all counters in $(C \setminus C^i_=)$ have values strictly greater than k_{max} (and the transition has no effect). Of course, the guard of such a transition is the following one:

$$(\bigwedge_{\mathsf{x} \in (C \setminus C^i_=)} \mathsf{x} \geq (k_{max} + 1)) \wedge (\bigwedge_{\mathsf{x} \in C^i_=} \mathsf{x} \leq k_{max}).$$

Such a transition has also no effect on the number of reversals.

As soon as in $\mathcal{M}_i$, we reach again a control state whose first component is q_f, we may jump to the final control state q_{new}. So, in the counter machine $\mathcal{M}_i$, the control state $\langle q_f^{\mathrm{in}}, i \rangle$ has a unique incoming transition (from q_f) and $\langle q_f^{\mathrm{in}}, i \rangle$ and $\langle q_f, i \rangle$ have identical outgoing transitions. Moreover, from the control state $\langle q_f, i \rangle$, there is a final transition to q_{new}.

Note that in $\mathcal{M}'$, it is sufficient to look for a r-reversal-bounded run. Figure 4 illustrates the construction of $\mathcal{M}'$. □

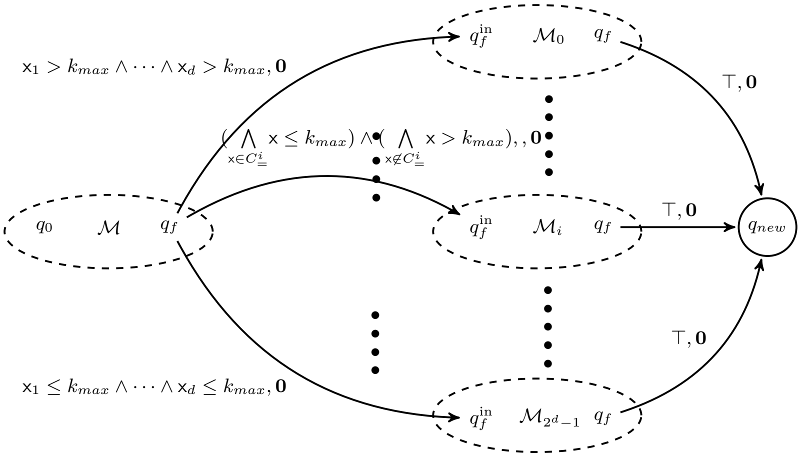
$\mathsf{x}_1 > k_{max} \wedge \cdots \wedge \mathsf{x}_d > k_{max}, \mathbf{0}$
$(\bigwedge_{\mathsf{x} \in C^i_=} \mathsf{x} \leq k_{max}) \wedge (\bigwedge_{\mathsf{x} \notin C^i_=} \mathsf{x} > k_{max}), , \mathbf{0}$
$\mathsf{x}_1 \leq k_{max} \wedge \cdots \wedge \mathsf{x}_d \leq k_{max}, \mathbf{0}$
q_0
$\mathcal{M}$
q_f
q_f^{in}
$\mathcal{M}_0$
q_f
q_f^{in}
$\mathcal{M}_i$
q_f
q_f^{in}
$\mathcal{M}_{2^d-1}$
q_f
$\top, \mathbf{0}$
$\top, \mathbf{0}$
$\top, \mathbf{0}$
q_{new}

Fig. 4. Machine $\mathcal{M}'$ with 2^d copies of $\mathcal{M}$

Theorem 20. *The control state repeated reachability problem with bounded number of reversals is* NEXPTIME*-complete.*

Proof. NEXPTIME-hardness follows from the NEXPTIME-hardness of the reachability problem with bounded number of reversals (Theorem 17). In order to obtain the NEXPTIME upper bound, it is sufficient to consider the proof of Lemma 19. From $\mathcal{M}$, $\langle q_0, \boldsymbol{x_0} \rangle$, q_f and $r \geq 0$ (instance of size N), we construct a counter machine $\mathcal{M}' = \langle Q', T', C \rangle$ such that the control state q_{new} can be reached from $\langle q_0, \boldsymbol{x}_0 \rangle$ for some r-reversal-bounded run iff there is an infinite r-reversal-bounded run from $\langle q_0, \boldsymbol{x}_0 \rangle$ such that q_f is repeated infinitely often. Since $\mathcal{M}'$ essentially restricts the behaviours of $\mathcal{M}$ (by guessing at some stage a set of counters $C_=$), q_{new} can be reached from $\langle q_0, \boldsymbol{x}_0 \rangle$ with an r-reversal-bounded run sharing the structural properties of small runs from the proof of Theorem 17, whence the NEXPTIME upper bound. □

Lemma 19 can be extended so that, instead of repeating infinitely often control states, properties on counters definable in Presburger arithmetic are repeated infinitely often. Let us introduce the following problem.
∃-PRESBURGER INFINITELY OFTEN PROBLEM

Input: Initialised counter machine $\langle \mathcal{M}, \langle q_0, \boldsymbol{x_0} \rangle \rangle$ with d counters, $r \geq 0$ and a Presburger formula $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_d)$.
Question: Is there an infinite r-reversal-bounded run from $\langle q_0, \boldsymbol{x_0} \rangle$ such that infinitely often $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_d)$ holds?

Theorem 21. *[DPK03] The ∃-Presburger infinitely often problem is decidable.*

3.7 A simple undecidable extension

We conclude Section 3 by considering the class of counter machines $\mathcal{C}^=$ that extends the class $\mathcal{C}$ so that the atomic guards of the form $\mathsf{x} \sim k$ ($\mathsf{x} \in C$ and $k \in \mathbb{N}$) are augmented with guards of the form either $\mathsf{x}_i = \mathsf{x}_j$ and $\mathsf{x}_i \neq \mathsf{x}_j$.

Theorem 22. *[ISD$^+$02] The reachability problem with bounded number of reversals for the class $\mathcal{C}^=$ is undecidable.*

Proof. (sketch) Undecidability can be shown even if r is restricted to zero (no reversal) and the only guards in transitions are equalities or inequalities.

To prove this result, we sketch a reduction from the halting problem for Minsky machines (similar to the control state reachability problem with the halting state as the final control state). Indeed, assuming that guards of the form $\mathsf{x}_i = \mathsf{x}_{i'}$ and $\mathsf{x}_i \neq \mathsf{x}_{i'}$ are allowed, each counter x_i from the Minsky machine $\mathcal{M}$ provides two increasing counters x_i^{inc} and x_i^{dec}, that counts the number of increments on x_i and the number of decrements, respectively. Zero-test for counter x_i is then simulated by a test $\mathsf{x}_i^{inc} = \mathsf{x}_i^{dec}$. Similarly, before incrementing x_i^{dec} (simulating a decrement in $\mathcal{M}$), we test whether $\mathsf{x}_i^{inc} \neq \mathsf{x}_i^{dec}$. □

As a conclusion, reversal-bounded counter machines from the class $\mathcal{C}$ admits decidable control state reachability problem as well as control state repeated reachability problem thanks to the fact that reachability sets are effective Presburger sets. By contrast, allowing equality/inequality guards in the counter machines leads to undecidability. In Section 4 below, we show how to extend the set of guards and the set of temporal properties while preserving decidability thanks to a notion of reversal-boundedness that takes into account reversal-boundedness for terms introduced in [BD11].

4 A taste of temporal reasoning

Reachability problems ask for the existence of runs reaching some configuration or reaching some control state in some specific way. Often, it is desirable to check how events are temporally organised along a run and in order to specify such properties temporal logics have been advocated since [Pnu77] (see also the recent book on temporal logics [DGL16]). Furthermore, we wish to include in the logical language the possibility to express directly constraints between variables of the program (typically between counters), whence giving up the standard abstraction made with propositional variables. So, a proposition like "x is greater than the next value of y" can be encoded in such extended temporal logics by $\mathsf{x} > \mathsf{X}\mathsf{y}$ but this time the models are sequences of configurations. This means that each position comes with a control state and a valuation for variables. Hence, the basic idea behind the design of the logic CLTL below is to refine the language of atomic formulae and to allow the possibility to compare counter values at successive positions of the run of the counter machines. Temporal logics with Presburger constraints have been quite widely developped, see e.g. in [Čer94,MP95,BEH95,Dem06,LMP10]. Results about reversal-bounded counter machines and Freeze LTL can be also found in [DS10].

4.1 The temporal logic CLTL

We define below a linear-time temporal logic CLTL in which atomic formulae are either control states or arithmetical constraints about counter values at the current position and next position. *Counter variables* in $\mathrm{VAR} = \{\mathsf{x}_1, \mathsf{x}_2, \ldots\}$ are free variables, only interpreted by the counter values on configurations. As for defining quantifier-free fragment of Presburger arithmetic, *arithmetical terms* are defined by the grammar below:

$$\mathtt{t} ::= a\mathsf{x} \mid a\mathsf{X}\mathsf{x} \mid \mathtt{t} + \mathtt{t},$$

with $\mathsf{x} \in \mathrm{VAR}$ and $a \in \mathbb{Z}$. Intuitively, x refers to the current value for the counter x, $\mathsf{X}\mathsf{x}$ refers to the counter value for x at the next position. Formulae of CLTL are defined as follows:

$$\varphi ::= \top \mid q \mid \mathtt{t} \sim k \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi,$$

with $q \in Q$, $\sim \in \{<, \leq, >, \geq, =\}$ and $k \in \mathbb{Z}$. As usual, we pose $\mathsf{F}\varphi \overset{\text{def}}{=} \top\mathsf{U}\varphi$ and $\mathsf{G}\varphi \overset{\text{def}}{=} \neg\mathsf{F}\neg\varphi$. The formula $\mathsf{GF}(\mathsf{x}_1 - \mathsf{x}_2 = 3)$ states that infinitely often the value for the first counter is equal to the value for the second counter plus 3.

Models for CLTL are intended to be infinite runs of counter machines; hence they are of the form $\rho = \langle q_0, \boldsymbol{x_0} \rangle, \langle q_1, \boldsymbol{x_1} \rangle, \langle q_2, \boldsymbol{x_2} \rangle, \ldots$ with $\rho \in (Q \times \mathbb{N}^d)^\omega$. The satisfaction relation $\models$ is defined as follows:

- $\rho, i \models q \overset{\text{def}}{\Leftrightarrow} q = q_i$,
- In order to deal with arithmetical constraints, we need to introduce a few notations. Given a term $\mathtt{t}$ from CLTL, we write $\mathtt{t}'$ to denote the term in quantifier-free Presburger arithmetic obtained from $\mathtt{t}$ by replacing $\mathsf{X}\mathsf{x}_i$ by x'_i. Then $\rho, i \models \mathtt{t} \sim k \overset{\text{def}}{\Leftrightarrow} \mathfrak{v} \models \mathtt{t}' \sim k$ where for every $j \in [1, n]$, $\mathfrak{v}(\mathsf{x}_j) \overset{\text{def}}{=} \boldsymbol{x_i}(j)$ and $\mathfrak{v}(\mathsf{x}'_j) \overset{\text{def}}{=} \boldsymbol{x_{i+1}}(j)$.
- $\rho, i \models \varphi \wedge \varphi' \overset{\text{def}}{\Leftrightarrow} \rho, i \models \varphi$ and $\rho, i \models \varphi'$,
- $\rho, i \models \neg\varphi \overset{\text{def}}{\Leftrightarrow} \rho, i \not\models \varphi$,
- $\rho, i \models \mathsf{X}\varphi \overset{\text{def}}{\Leftrightarrow} \rho, i+1 \models \varphi$,
- $\rho, i \models \varphi \mathsf{U} \varphi' \overset{\text{def}}{\Leftrightarrow}$ there is $j \geq i$ such that $\rho, j \models \varphi'$ and for all $h \in [i, j-1]$, we have $\rho, h \models \varphi$.

Semantics with finite runs instead of infinite runs can be defined similarly. Let us turn to the definition of EXISTENTIAL MODEL-CHECKING PROBLEM for CLTL:

Input: A counter machine $\mathcal{M} = \langle Q, T, C \rangle$, an initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$ and a CLTL formula φ.
Question: Is there an infinite run ρ starting at $\langle q_0, \boldsymbol{x}_0 \rangle$ such that $\rho, 0 \models \varphi$?

Temporal logics with Presburger constraints have been widely developed, for instance in the works [BEH95,CC00,BDR03]. Some of them have quite expressive decidable fragments. Undecidability of the existential model-checking problem for CLTL can be shown using the undecidability of the halting problem for Minsky machines, see e.g., [CC00]. In the sequel, we provide an example of restriction in order to regain decidability.

4.2 Bounding the number of reversals for terms

In this section, we consider counter machines for which the atomic formulae in guards are of the form $\mathtt{t} \leq k$ or $\mathtt{t} \geq k$ with $k \in \mathbb{Z}$ and $\mathtt{t}$ is of the form $\sum_i a_i \mathsf{x}_i$ with the a_i's in $\mathbb{Z}$. This defines the class of counter machines $\mathcal{C}^+$, strictly extending the class $\mathcal{C}$ introduced in Section 2.1. We have seen that a reversal for a counter occurs in the run of some counter machine when there is an alternation from a nonincreasing mode to a nondecreasing mode and vice-versa. Below, we propose a slight generalisation from [BD11] that captures the notion of reversal-boundedness from [Iba78]; reversal-boundedness applies to counters but also to terms occurring in guards.

Let $\mathcal{M} = \langle Q, T, C \rangle$ be a counter machine in $\mathcal{C}^+$ and $\mathtt{T}$ be a finite set of terms including $\{\mathsf{x}_1, \ldots, \mathsf{x}_d\}$. From a run $\rho = \langle q_0, \boldsymbol{x_0} \rangle, \langle q_1, \boldsymbol{x_1} \rangle, \ldots$ of $\mathcal{M}$, in order to describe the behavior of counters and terms varying along ρ, we define a sequence of *mode vectors* $\mathfrak{md}_0, \mathfrak{md}_1, \ldots$ such that each $\mathfrak{md}_i$ has profile $\mathtt{T} \to \{\uparrow, \downarrow\}$. Intuitively, each value in a mode vector records whether a term is currently in an increasing phase or in a decreasing phase (this includes the counters themselves as in standard reversal-boundedness). Given $\mathtt{t} = \sum_i a_i \mathsf{x}_i$ and a counter vector $\boldsymbol{x} \in \mathbb{N}^d$, we write $\boldsymbol{x}(\mathtt{t})$ to denote the integer $\sum a_i \boldsymbol{x}(i)$.

We are now ready to define the sequence $\mathfrak{md}_0, \mathfrak{md}_1, \ldots$ By convention, $\mathfrak{md}_0$ is the constant map $\uparrow$. For every $j \geq 0$ and $\mathtt{t} \in \mathtt{T}$, we have

- $\mathfrak{md}_{j+1}(\mathtt{t}) \overset{\text{def}}{=} \mathfrak{md}_j(\mathtt{t})$ when $\boldsymbol{x}_j(\mathtt{t}) = \boldsymbol{x}_{j+1}(\mathtt{t})$,
- $\mathfrak{md}_{j+1}(\mathtt{t}) \overset{\text{def}}{=} \uparrow$ when $\boldsymbol{x}_{j+1}(\mathtt{t}) - \boldsymbol{x}_j(\mathtt{t}) > 0$ and,
- $\mathfrak{md}_{j+1}(\mathtt{t}) \overset{\text{def}}{=} \downarrow$ when $\boldsymbol{x}_{j+1}(\mathtt{t}) - \boldsymbol{x}_j(\mathtt{t}) < 0$.

Let $Rev_{\mathtt{t}} = \{j \in [0, \mid \rho \mid -1] : \mathfrak{md}_j(\mathtt{t}) \neq \mathfrak{md}_{j+1}(\mathtt{t})\}$. We say that ρ is r-$\mathtt{T}$-*reversal-bounded* for some $r \geq 0 \overset{\text{def}}{\Leftrightarrow}$ for all $\mathtt{t} \in \mathtt{T}$, $\mathrm{card}(Rev_{\mathtt{t}}) \leq r$. Given a counter machine $\mathcal{M}$, we write $\mathtt{T}_{\mathcal{M}}$ to denote the set of terms $\mathtt{t}$ occurring in atomic formulae of the form $\mathtt{t} \sim k$ with $\sim \in \{\leq, \geq\}$ augmented with the counters in $\{\mathsf{x}_1, \ldots, \mathsf{x}_d\}$. An initialised counter machine $\langle \mathcal{M}, \langle q, \boldsymbol{x} \rangle \rangle$ is *reversal-bounded* $\overset{\text{def}}{\Leftrightarrow}$ there is $r \geq 0$ such that every run from $\langle q, \boldsymbol{x} \rangle$ is r-$\mathtt{T}_{\mathcal{M}}$-reversal-bounded. When $\mathtt{T}$ is reduced to $\{\mathsf{x}_1, \ldots, \mathsf{x}_d\}$, $\mathtt{T}$-reversal-boundedness is equivalent to reversal-boundedness from [Iba78].

Theorem 23. *[Iba78,BD11] Given a counter machine $\mathcal{M}$ in $\mathcal{C}^+$, $r \geq 0$ and control states q, q', one can effectively compute a Presburger formula*

$$\varphi_{q,q'}(\mathsf{x}_1, \ldots, \mathsf{x}_d, \mathsf{y}_1, \ldots, \mathsf{y}_d)$$

such that for all valuations $\mathfrak{v}$, we have $\mathfrak{v} \models \varphi$ iff there is an r-$\mathtt{T}_{\mathcal{M}}$-reversal-bounded run from $\langle q, \langle \mathfrak{v}(\mathsf{x}_1), \ldots, \mathfrak{v}(\mathsf{x}_d) \rangle \rangle$ to $\langle q', \langle \mathfrak{v}(\mathsf{y}_1), \ldots, \mathfrak{v}(\mathsf{y}_d) \rangle \rangle$.

Results about model-checking reversal-bounded counter machines with CLTL can be found in [BD11,HL11]. Below, we recall the definition for the reversal-bounded model-checking problem (RBMC). Its peculiarity is that the input initialised counter machines are not necessarily reversal-bounded but the input contains an explicit bound r about the maximal number of reversals within a run. Moreover, given a CLTL formula φ, we write $\mathtt{T}_{\varphi}$ to denote the finite set of terms of the form $\sum_k (a_k + b_k)\mathsf{x}_k$ when $\mathtt{t} = (\sum_k a_k \mathsf{X}\mathsf{x}_k) + (\sum_k b_k \mathsf{x}_k)$ is a term occurring in φ (modulo AC for the operator $+$) in an atomic formula of the form $\mathtt{t} \sim k$ with $\sim \in \{\leq, \geq, <, >, =\}$ and $k \in \mathbb{Z}$. Since the next value of the counter x_k (denoted by $\mathsf{X}\mathsf{x}_k$) is equal to the current value of the counter plus some $b \in \mathbb{Z}$ (depending on the update vectors of the transitions), the value of the term $(\sum_k a_k \mathsf{X}\mathsf{x}_k) + (\sum_k b_k \mathsf{x}_k)$ is equal to the current value of $\sum_k (a_k + b_k)\mathsf{x}_k$ plus some constant depending on the next transition. This explains the current definition of $\mathtt{T}_{\varphi}$. The problem RBMC is defined as follows:

Input: a counter machine $\mathcal{M}$, an initial configuration $\langle q_0, \boldsymbol{x}_0 \rangle$, a CLTL formula φ, a bound $r \in \mathbb{N}$ (in binary),

Question: Is there an infinite run ρ from $\langle q, \boldsymbol{x} \rangle$ such that $\rho, 0 \models \varphi$ and ρ is r-$\mathtt{T}$-reversal-bounded with $\mathtt{T} = \mathtt{T}_{\mathcal{M}} \cup \mathtt{T}_{\varphi}$?

The computational complexity for RBMC can be precisely characterised; the upper bound can be obtained by a refined analysis on runs, see e.g. [GI81,BD11].

Theorem 24. *[BD11,HL11]*RBMC *is* NEXPTIME*-complete.*

Hence, the NEXPTIME upper bound established for the reachability problem with bounded number of reversals (Theorem 17) can be extended to richer classes of counter machines and to richer specification languages such as CLTL [BD11,HL11]. Note that the decidability with both extensions is possible thanks to the introduction of a new concept for reversal-boundedness that makes explicit the role of arithmetical terms [BD11] and it captures previous notions on reversal-boundedness. In [HL11], operational models extending pushdown systems with counters and clocks are also considered; a version of reversal-bounded LTL model-checking is shown to be co-NEXPTIME [HL11, Theorem 2]. A prototypical implementation and experimental results are also presented in [HL11]. Finally, in [KWT10, Theorem 22], EXPTIME upper bound for LTL model-checking over reversal-bounded counter automata is shown but the logical language has no arithmetical constraint and the number of reversals r is encoded in unary (see also [WT10]).

5 Conclusion

In the paper, we have provided several developments about the verification of reversal-bounded counter machines. All the decision procedures have been designed by translation into Presburger arithmetic, witnessing once more, the translation approach to solve verification problems (more details can be found in [BD11,HL11] for instance). The idea of having bounds in runs is quite rich and bounding the number of reversals is not the only option. Moreover, multi-pushdown systems generalise the counter machines by having different stacks instead of different counters (while both models are Turing-complete). Bounding the number of switches [resp. phases] have been studied in [QR05] [resp. in [TMP07]]. A switch is an update from one stack to another whereas a phase is understood as a sequence of steps in which pop actions are performed on one stack only whereas push actions can occur on any stack. Other restrictions on multi-pushdown systems can be found in [Ati10], followed by many other studies on richer models in which runs are bounded in a way or another.

Acknowledgment. I would like to thank the anonymous reviewers for their suggestions and remarks that help me to improve the quality of the document.

References

[Ati10] M.F. Atig. Global model checking of ordered multi-pushdown systems. In *FST&TCS'10*, pages 216–227. LIPICS, 2010.

[BB74] B. Baker and R. Book. Reversal-bounded multipushdown machines. *Journal of Computer and System Sciences*, 8:315–332, 1974.

[BCC$^+$03] A. Biere, A. Cimatti, E. Clarke, O. Strichman, and Y. Zhu. Bounded model checking. *Advances in Computers*, 58:118–149, 2003.

[BD11] M. Bersani and S. Demri. The complexity of reversal-bounded model-checking. In *FROCOS'11*, volume 6989 of *LNAI*, pages 71–86. Springer, 2011.

[BDR03] V. Bruyère, E. Dall'Olio, and J.F. Raskin. Durations, parametric model-checking in timed automata with Presburger arithmetic. In *STACS'03*, volume 2607 of *LNCS*, pages 687–698. Springer, 2003.

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

[Ber80] L. Berman. The complexity of logical theories. *Theoretical Computer Science*, 11:71–78, 1980.

[BGO06] D. Bresolin, J. Golińska-Pilarek, and E. Orłowska. Relational dual tableaux for interval temporal logics. *Journal of Applied Non-Classical Logics*, 16(3–4):251–278, 2006.

[BIK09] M. Bozga, R. Iosif, and F. Konecný. Fast acceleration of ultimately periodic relations. In *CAV'10*, volume 6174 of *LNCS*, pages 227–242. Springer, 2009.

[BL81] M. Blattner and M. Latteux. Parikh-Bounded Languages. In *ICALP'81*, volume 115 of *LNCS*, pages 316–323. Springer, 1981.

[BL10] M. Bojańczyk and S. Lasota. An extension of data automata that captures XPath. In *LICS'10*, pages 243–252. IEEE, 2010.

[BO99] Ph. Balbiani and E. Orłowska. A hierarchy of modal logics with relative accessibility relations. *Journal of Applied Non-Classical Logics*, 9:303–328, 1999.

[Boi99] B. Boigelot. *Symbolic methods for exploring infinite state spaces*. PhD thesis, Université de Liège, 1999.

[BT76] I. Borosh and L. Treybig. Bounds on positive integral solutions of linear Diophantine equations. *American Mathematical Society*, 55:299–304, 1976.

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

[Čer94] K. Čer$\bar{\text{a}}$ns. Deciding properties of integral relational automata. In *ICALP'94*, volume 820 of *LNCS*, pages 35–46. Springer, 1994.

[CFM11] M. Cadilhac, A. Finkel, and P. McKenzie. On the expressiveness of Parikh automata and related models. In *Proceedings of the 3rd Workshop on Non-Classical Models of Automata and Applications (NCMA'11)*, volume 282 of *books@ocg.at*, pages 103–119. Austrian Computer Society, 2011.

[Cha81] T. Chan. Reversal complexity of counter machines. In *STOC'81*, pages 146–157. ACM, 1981.

[CHL13] W. Czerwiński, P. Hofman, and S. Lasota. Reachability problem for weak multi-pushdown automata. *Logical Methods in Computer Science*, 9(3), 2013.

[CJ98] H. Comon and Y. Jurski. Multiple counter automata, safety analysis and Presburger Arithmetic. In *CAV'98*, volume 1427 of *LNCS*, pages 268–279. Springer, 1998.

[DDS13] S. Demri, A.K. Dhar, and A. Sangnier. On the complexity of verifying regular properties on flat counter systems. In *ICALP'13*, volume 7966 of *LNCS*, pages 162–173. Springer, 2013.

[DDS15] S. Demri, A.K. Dhar, and A. Sangnier. Taming past LTL and flat counter systems. *Information & Computation*, 242:306–339, 2015.

[Dem06] S. Demri. Linear-Time Temporal Logics with Presburger Constraints: An Overview. *Journal of Applied Non-Classical Logics*, 16(3–4):311–347, 2006.

[DFP16] S. Demri, D. Figueira, and M. Praveen. Reasoning about data repetitions with counter systems. *Logical Methods in Computer Science*, 12, 2016.

[DGL16] S. Demri, V. Goranko, and M. Lange. *Temporal Logics in Computer Science*. Cambridge University Press, 2016.

[DIB^{+}00] Z. Dang, O.H. Ibarra, T. Bultan, R.A. Kemmerer, and J. Su. Binary reachability analysis of discrete pushdown timed automata. In *CAV'00*, volume 1855 of *LNCS*, pages 69–84. Springer, 2000.

[DIP01] Z. Dang, O. Ibarra, and P. San Pietro. Liveness verification of reversal-bounded multicounter machines with a free counter. In *FST&TCS'01*, volume 2245 of *LNCS*, pages 132–143. Springer, 2001.

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

[DS10] S. Demri and A. Sangnier. When model checking freeze LTL over counter machines becomes decidable. In *FOSSACS'10*, volume 6014 of *LNCS*, pages 176–190. Springer, 2010.

[EG11] J. Esparza and P. Ganty. Complexity of pattern-based verification for multithreaded programs. In *POPL'11*, pages 499–510. ACM, 2011.

[Esp94] J. Esparza. On the decidability of model checking for several μ-calculi and Petri nets. In *ICALP'94*, volume 787 of *LNCS*, pages 115–129. Springer, 1994.

[Esp98] J. Esparza. Decidability and complexity of Petri net problems — an introduction. In *Advances in PN'98*, volume 1491 of *LNCS*, pages 374–428. Springer, 1998.

[FdCO85] L. Fariñas del Cerro and E. Orłowska. DAL – A logic for data analysis. *Theoretical Computer Science*, 36:251–264, 1985.

[FIS03] A. Finkel, S. Purushothaman Iyer, and G. Sutre. Well-abstracted transition systems: Applications to FIFO automata. *Information & Computation*, 181(1):1–31, 2003.

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

[FO97] L. Fribourg and H. Olsén. Proving safety properties of infinite state systems by compilation into Presburger arithmetic. In *CONCUR'97*, volume 1243 of *LNCS*, pages 213–227. Springer, 1997.

[Fri00] L. Fribourg. Petri nets, flat languages and linear arithmetic. In *9th Workshop on Functional and Logic Programming (WFLP)*, pages 344–365, 2000.

[FS08] A. Finkel and A. Sangnier. Reversal-bounded counter machines revisited. In *MFCS'08*, volume 5162 of *LNCS*, pages 323–334. Springer, 2008.

[GI81] E. Gurari and O. Ibarra. The complexity of decision problems for finite-turn multicounter machines. In *ICALP'81*, volume 115 of *LNCS*, pages 495–505. Springer, 1981.

[GS66] S. Ginsburg and E. Spanier. Semigroups, Presburger formulas and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.

[Haa14] Ch. Haase. Subclasses of Presburger arithmetic and the weak EXP hierarchy. In *LICS'14*. ACM Press, 2014.

[Hab97] P. Habermehl. On the complexity of the linear-time mu-calculus for Petri nets. In *ICATPN'97*, volume 1248 of *LNCS*, pages 102–116. Springer, 1997.

[HH14] Ch. Haase and S. Halfon. Integer vector addition systems with states. In *RP'14*, volume 8762 of *LNCS*, pages 112–124. Springer, 2014.

[HL11] M. Hague and A. W. Lin. Model checking recursive programs numeric data types. In *CAV'11*, volume 6806 of *LNCS*, pages 743–759. Springer, 2011.

[HR87] R. Howell and L. Rosier. An analysis of the nonemptiness problem for classes of reversal-bounded multicounter machines. *Journal of Computer and System Sciences*, 34(1):55–74, 1987.

[HR89] R. Howell and L. Rosier. Problems concerning fairness and temporal logic for conflict-free petri nets. *Theoretical Computer Science*, 64:305–329, 1989.

[Iba74] O. Ibarra. A note on semilinear sets and bounded-reversal multihead pushdown automata. *Information Processing Letters*, 3(1):25–28, 1974.

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

[ISD$^+$02] O. Ibarra, J. Su, Z. Dang, T. Bultan, and R. Kemmerer. Counter machines and verification problems. *Theoretical Computer Science*, 289(1):165–189, 2002.

[Jan90] P. Jančar. Decidability of a temporal logic problem for Petri nets. *Theoretical Computer Science*, 74(1):71–93, 1990.

[KM69] R. Karp and R. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.

[Kos82] R. Kosaraju. Decidability of reachability in vector addition systems. In *STOC'82*, pages 267–281, 1982.

[KR02] F. Klaedtke and H. Ruess. Parikh automata and monadic second-order logics with linear cardinality constraints. Technical Report 177, Institute of Computer Science at Freiburg University, 2002.

[KWT10] E. Kopczynski and A. Widjaja To. Parikh images of grammars: Complexity and applications. In *LICS'10*, pages 80–89. IEEE, 2010.

[Ler09] J. Leroux. The general vector addition system reachability problem by Presburger inductive invariants. In *LICS'09*, pages 4–13. IEEE, 2009.

[Lip76] R.J. Lipton. The reachability problem requires exponential space. Technical Report 62, Department of Computer Science, Yale University, 1976.

[LMP10] F. Laroussinie, A. Meyer, and E. Petonnet. Counting LTL. In *TIME'10*, pages 51–58. IEEE, 2010.

[LS15] J. Leroux and S. Schmitz. Demystifying reachability in vector addition systems. In *LICS'15*, pages 56–67. IEEE, 2015.

[May84] E. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal of Computing*, 13(3):441–460, 1984.

[Min61] M. Minsky. Recursive unsolvability of Post's problems of 'tag' and other topics in theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, 1961.

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

[MP95] Z. Manna and A. Pnueli. *Temporal verification of reative systems: safety*. Springer, 1995.

[Orło73] E. Orłowska. *Theorem-proving systems*. PhD thesis, Mathematical Department of the Warsaw University, 1973. Published in the series Dissertationes Mathematicae from the Polska Akademia Nauk, Instytut Matematyczny.

[Orło85] E. Orłowska. Logic of nondeterministic information. *Studia Logica*, 44:93–102, 1985.

[Orło88] E. Orłowska. Relational interpretation of modal logics. In H. Andréka, D. Monk, and I. Németi, editors, *Algebraic logic. Colloquia Mathematica Societatis Janos Bolyai 54*, pages 443–471. North-Holland, Amsterdam, 1988.

[Orło89] E. Orłowska. Logic for reasoning about knowledge. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 35:559–568, 1989.

[Orło93] E. Orłowska. Dynamic logic with program specifications and its relational proof system. *Journal of Applied Non-Classical Logics*, 3(2):147–171, 1993.

[Orło95] E Orłowska. Temporal logics in a relational framework. In L. Bolc and A. Szalas, editors, *Time and Logic- A Computational Approach*, pages 249–277. UCL Press, 1995.

[Pap81] Chr. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981.

[Par66] R. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

[Pnu77] A. Pnueli. The temporal logic of programs. In *FOCS'77*, pages 46–57. IEEE, 1977.

[Pre29] M. Presburger. 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, Warszawa*, pages 92–101, 1929.

[QR05] S. Qaader and J. Rehof. Context-bounded model checking of concurrent software. In *TACAS'05*, volume 3440 of *LNCS*, pages 93–107. Springer, 2005.

[Rac78] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978.

[Reu90] C. Reutenauer. *The mathematics of Petri nets*. Masson and Prentice, 1990.

[San08] A. Sangnier. *Vérification de systèmes avec compteurs et pointeurs*. Thèse de doctorat, LSV, ENS Cachan, France, 2008.

[Sav70] W.J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1986.

[Sch17] S. Schmitz. *Algorithmic Complexity of Well-Quasi-Orders*. Mémoire d'habilitation, École Normale Supérieure Paris-Saclay, France, 2017.

[SJ80] N. Suzuki and D. Jefferson. Verification Decidability of Presburger Array Programs. *Journal of the ACM*, 27(1):191–205, 1980.

[TMP07] S. La Torre, P. Madhusudan, and G. Parlato. A robust class of context-sensitive languages. In *LICS'07*, pages 161–170. IEEE, 2007.

[WT10] A. Widjaja To. *Model Checking Infinite-State Systems: Generic and Specific Approaches*. PhD thesis, School of Informatics, University of Edinburgh, 2010.