Information and Computation 242 (2015) 306–339

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Taming past LTL and flat counter systems ☆

Stéphane Demri [a], Amit Kumar Dhar [b,*], Arnaud Sangnier [b]

[a] LSV, CNRS, France
[b] LIAFA, Univ. Paris Diderot, Sorbonne Paris Cité, CNRS, France

ARTICLE INFO

Article history:
Received 20 July 2012
Received in revised form 4 February 2015
Available online 24 March 2015

Keywords:
Linear-time temporal logic
Stuttering
Model-checking
Counter system
Flatness
Complexity
System of equations
Small solution
Presburger arithmetic

ABSTRACT

Reachability and LTL model-checking problems for flat counter systems are known to be decidable but whereas the reachability problem can be shown in NP, the best known complexity upper bound for the latter problem is made of a tower of several exponentials. Herein, we show that this problem is only NP-complete even if LTL admits past-time operators and arithmetical constraints on counters. As far as past-time operators are concerned, their addition to LTL immediately leads to complications and hence an NP upper bound cannot be deduced by translating formulae into LTL and studying the problem only for this latter logic. We also provide other complexity results obtained by restricting further the class of flat counter systems.

1. Introduction

Flat counter systems Counter systems are finite-state automata equipped with program variables (counters) interpreted over non-negative integers. They are used in many places like, broadcast protocols [17] and programs with pointers [20] to quote a few examples. But, along with their large scope of usability, many problems on general counter systems are known to be undecidable [34]. Indeed, this computational model can simulate Turing machines. This is not the end of the story since decidability of reachability problems or model-checking problems based on temporal logics can be regained by considering subclasses of counter systems (this includes restrictions on the instructions, on the control graphs or on more semantical properties). An important and natural class of counter systems, in which various practical cases of infinite-state systems (e.g. broadcast protocols [19]) can be modelled, are those with a *flat* control graph, i.e., those where no control state occurs in more than one simple cycle, see e.g. [2,9,19,31,6]. Decidability results on verifying safety and reachability properties on flat counter systems have been obtained in [9,19,4]. However, so far, such properties have been rarely considered in the framework of any formal specification language (see an exception in [8]). In [14], a class of Presburger counter systems is identified for which the local model checking problem for Presburger-CTL* is shown decidable. These are Presburger counter systems defined over flat control graphs with transitions labelled by adequate Presburger formulae (representing

☆ Work partially supported by ANR project REACHARD ANR-11-BS02-001 and by the EU Seventh Framework Programme (under grant PIOF-GA-2011-301166, DATAVERIF). This is a completed and revised version of [11].

* Corresponding author.
E-mail addresses: demri@lsv.ens-cachan.fr (S. Demri), dhar@liafa.univ-paris-diderot.fr (A.K. Dhar), sangnier@liafa.univ-paris-diderot.fr (A. Sangnier).

http://dx.doi.org/10.1016/j.ic.2015.03.007
0890-5401/

guards). Even though flatness is clearly a substantial restriction, it is shown in [31] that many classes of counter systems with computable Presburger-definable reachability sets are *flattable*, i.e. there exists a flat unfolding of the counter system with identical reachability sets. Hence, the possibility of flattening a counter system is strongly related to semilinearity of its reachability set. Moreover, in [8] model-checking relational counter systems over LTL formulae is shown decidable when restricted to flat formulae (their translation into automata leads to flat structures). Flat counter systems can be also seen as a way to under-approximate the behavior of general counter systems. In fact, by unfolding nested loops of the original system with different strategies, one can produce an enumeration of flat systems which characterize some behaviors of the system. Note that such an approximation allows to describe more behaviors than the approach that consists in looking at bounded behaviors (where the values of the counters do not overpass a certain given bound).

Towards the complexity of temporal model-checking flat counter systems Our goal is to revisit here standard decidability results for subclasses of counter systems obtained by translation into Presburger arithmetic in order to obtain optimal complexity upper bounds. Indeed, effectively composing the translation of a verification problem into Presburger arithmetic (PrA) and then using a solver for (PrA) is not necessarily optimal computationally. We focus in this work on the model-checking of flat counter systems taking, as specification language, linear-time temporal logic with past and extended with counter constraints.

Linear temporal logic (LTL) was first introduced as a specification language for verification in [36]. The model-checking problem for LTL and its subclasses over finite structures like Kripke structures has been extensively studied and its exact complexity characterization is well-known (see [7]). Moreover, it is known that, even though LTL with past-time operators is expressively similar to LTL with only future temporal operators [21], LTL with past is known to be more succinct than LTL with only future operator [29]. We plan to investigate the model checking problem for LTL with past operators and with counter constraints as atomic formulae over flat counter system. Even if it is known that such a problem is decidable for flat counter systems [14], no work has so far provided tight complexity bounds. Due to the popularity of LTL with past in the verification community and to the strong expressive power of counter systems, we believe that providing optimal algorithms for this model-checking problem is of great interest and could as well be useful to the field of verification of infinite state-systems.

Related works Many papers study reachability problems for flat control structures manipulating integer variables or counters. The main differences between these different works lie in the type of guards and updates over the counters that are allowed in the system. In [9], Comon and Jurski study reachability problems for a class of counter systems with non-deterministic updates where each transition of the system is labelled by a difference bound matrix (DBM) characterizing the difference between the actual and the successive values of the counters; they show that, for such a model, reachability is decidable. Latter on, in [5], this latter problem is proved to be NP-complete. In [19], Finkel and Leroux study another class of counter systems, where the guards are given by Presburger arithmetic formulae and the updates on the counters are performed thanks to linear functions, and they show that it is possible to compute the reachability set of such systems which is expressible in Presburger arithmetic. Note that this latter work extends a previous similar result proposed by Boigelot in [2]. In this work, we focus on the class of counter systems whose guards are Boolean combinations of linear constraints and whose updates are translations, hence this class of systems is incomparable with the class of counter systems labelled with difference bound matrices (we are more expressive in the guards but less powerful in the updates) and it consists in a subclass of the systems studied by Finkel and Leroux.

In [14], it is proved that CTL* model-checking over the class of so-called *admissible* counter systems is decidable by reduction into the satisfiability problem for Presburger arithmetic, the decidable first-order theory of natural numbers with addition. Note that the flat counter systems considered by Finkel and Leroux, and hence the one we propose to study, are admissible. Even though this latter paper gives the decidability status of the model-checking of Past LTL over flat counter systems (CTL* being strictly more expressive that LTL), the introduced decision procedure provides a very rough complexity upper bound in 4ExpTime, whereas, as we shall see, this problem is NP-complete.

Our contributions In this paper, we establish several computational complexity characterizations of model-checking problems restricted to flat counter systems in the presence of a rich LTL-like specification language with arithmetical constraints and past-time operators. Not only we provide an optimal complexity but also, we believe that our techniques could be reused for further extensions (see the recent work [12] about regular specification languages). Indeed, we combine three proof techniques: the general stuttering theorem [28], the property of small integer solutions of equation systems [3] (this latter technique is used since [37,23]) and the elimination of disjunctions in guards (see Section 7). Let us be a bit more precise.

We extend the general stuttering principle established in [28] for LTL (without past-time operators) to Past LTL. However, since this principle will be applied to path schemas, a fundamental structure in flat counter systems, we do not aim at being optimal; what matters in fact for our main result is an NP upper bound. A path schema is simply a finite alternation of path segments and simple loops (no repetition of edges) and the principle states that satisfaction of an LTL formula requires only to take loops a number of times that is linear in the temporal depth of the formula. This principle has been already used to establish that LTL model-checking over *weak* Kripke structures is in NP [27] (weakness corresponds to flatness). It is worth noting that another way to show a similar result would be to eliminate past-time operators thanks

to Gabbay's Separation Theorem [21] (preserving initial equivalence) but the temporal depth of formulae might increase at least exponentially, which is a crucial parameter in our complexity analysis. We show that the model-checking problem restricted to flat counter systems in the presence of LTL with past-time operators is in NP (Theorem 8.4) by combining the above-mentioned proof techniques (we call this problem MC(PLTL[$\mathtt{C}$], $\mathtt{FlatCS}$)). Apart from the use of the general stuttering theorem (Theorem 4.1), we take advantage of the other properties stated for instance in Lemma 6.1 (characterization of runs by quantifier-free Presburger formulae) and Theorem 7.11 (elimination of disjunctions in guards preserving flatness). Note that the loops in runs are visited a number of times that can be exponential in the worst case, but this does not prevent us from establishing the NP upper bound. We also take advantage of the fact that model-checking ultimately periodic models with Past LTL is in PTime [29]. We also point out the fact that our main decision procedure is not automata-based, unlike the approach from [39]. In this paper, complexity results for fragments/subproblems are also considered. For instance, we get a sharp lower bound since we establish that the model-checking problem on path schemas with only 2 loops is already NP-hard (see Lemma 5.6). A summary table of results can be found in Section 9.

The present paper is an extended and completed version of [11]. It includes all the proofs and their explanation in details and furthermore we have added a few more results (see e.g. Section 8.2).

2. Flat counter systems and its LTL dialect

We write $\mathbb{N}$ [resp. $\mathbb{Z}$] to denote the set of natural numbers [resp. integers] and $[i, j]$ to denote the set $\{k \in \mathbb{Z} : i \leq k \text{ and } k \leq j\}$. For every $\mathbf{v} \in \mathbb{Z}^n$, $\mathbf{v}[i]$ denotes the ith element of $\mathbf{v}$ for every $i \in [1, n]$. For some n-ary tuple t, we also write $\pi_j(t)$ to denote the jth element of t ($j \leq n$). In the sequel, integers are encoded with a binary representation. 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 ω-words over Σ. For a finite word $w = a_1 \dots a_k$ over Σ, we write $\mathrm{len}(w)$ to denote its *length* k. For $0 \leq i < \mathrm{len}(w)$, $w(i)$ represents the $(i+1)$-th letter of the word, here a_{i+1}.

2.1. Counter systems

Counter constraints are defined below as a subclass of Presburger formulae whose free variables are understood as counters. Such constraints are used to define guards in counter systems but also to define arithmetical constraints in temporal formulae.

Let $\mathtt{C} = \{\mathtt{x}_1, \mathtt{x}_2, \dots\}$ be a countably infinite set of *counters* (variables interpreted over non-negative integers) and $\mathrm{AT} = \{\mathtt{p}_1, \mathtt{p}_2, \dots\}$ be a countable infinite set of propositional variables (abstract properties about program points). We write $\mathtt{C}_n$ to denote the restriction of $\mathtt{C}$ to $\{\mathtt{x}_1, \mathtt{x}_2, \dots, \mathtt{x}_n\}$.

Definition 2.1 *(Guards).* The set $\mathtt{G}(\mathtt{C}_n)$ of *guards* (arithmetical constraints on counters in $\mathtt{C}_n$) is defined inductively as follows:

$$\mathtt{t} \quad ::= \quad a.\mathtt{x} \mid \mathtt{t} + \mathtt{t}$$

$$\mathtt{g} \quad ::= \quad \mathtt{t} \sim b \mid \mathtt{g} \wedge \mathtt{g} \mid \mathtt{g} \vee \mathtt{g}$$

where $\mathtt{x} \in \mathtt{C}_n$, $a \in \mathbb{Z}$, $b \in \mathbb{N}$ and $\sim \in \{=, \leq, \geq, <, >\}$.

Note that such guards are closed under negations (but negation is not a logical connective) and the truth constants $\top$ and $\bot$ can be easily defined too.

Given $\mathtt{g} \in \mathtt{G}(\mathtt{C}_n)$ and a vector $\mathbf{v} \in \mathbb{N}^n$, we say that $\mathbf{v}$ satisfies $\mathtt{g}$, written $\mathbf{v} \models \mathtt{g}$, if the formula obtained by replacing each $\mathtt{x}_i$ by $\mathbf{v}[i]$ holds.

Definition 2.2 *(Counter system).* For a natural number $n \geq 1$, an n-dim counter system (shortly a counter system) S is a tuple $\langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$ where:

- Q is a finite set of *control states*.
- $\mathbf{l} : Q \to 2^{\mathrm{AT}}$ is a *labelling function*.
- $\Delta \subseteq Q \times \mathtt{G}(\mathtt{C}_n) \times \mathbb{Z}^n \times Q$ is a finite set of edges labeled by guards and updates of the counter values (*transitions*).

For every transition $\delta = \langle q, \mathtt{g}, \mathbf{u}, q' \rangle$ in Δ, we use the following notations:

- $\mathit{source}(\delta) = q$; $\mathit{target}(\delta) = q'$,
- $\mathit{guard}(\delta) = \mathtt{g}$; $\mathit{update}(\delta) = \mathbf{u}$.

As usual, to a counter system $S = \langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$, we associate a labeled transition system $\mathfrak{T}(S) = \langle C, \to \rangle$ where $C = Q \times \mathbb{N}^n$ is the set of *configurations* and $\to \subseteq C \times \Delta \times C$ is the *transition relation* defined by: $\langle \langle q, \mathbf{v} \rangle, \delta, \langle q', \mathbf{v}' \rangle \rangle \in \to$ (also written $\langle q, \mathbf{v} \rangle \xrightarrow{\delta} \langle q', \mathbf{v}' \rangle$) iff the conditions below are satisfied:

- $q = source(\delta)$ and $q' = target(\delta)$,
- $\mathbf{v} \models guard(\delta)$ and $\mathbf{v}' = \mathbf{v} + update(\delta)$.

Note that in such a transition system, the counter values are non-negative since $C = Q \times \mathbb{N}^n$. We extend the transition relation $\rightarrow$ to finite words of transitions in Δ^+ as follows. For each $w = \delta_0\delta_1 \ldots \delta_\alpha \in \Delta^+$, we have $\langle q, \mathbf{v}\rangle \xrightarrow{w} \langle q', \mathbf{v}'\rangle$ if there are $c_0, c_1, \ldots, c_{\alpha+1} \in C$ such that $c_i \xrightarrow{\delta_i} c_{i+1}$ for all $i \in [0, \alpha]$, $c_0 = \langle q, \mathbf{v}\rangle$ and $c_{\alpha+1} = \langle q', \mathbf{v}'\rangle$. We say that an ω-word $w \in \Delta^\omega$ is *fireable* in S from a configuration $c_0 \in Q \times \mathbb{N}^n$ if for all finite prefixes w' of w there exists a configuration $c \in Q \times \mathbb{N}^n$ such that $c_0 \xrightarrow{w'} c$. We write $lab(c_0)$ to denote the set of ω-words which are fireable from c_0 in S. Note that from a given configuration there could be multiple transitions that are fireable and hence counter systems are inherently non-deterministic.

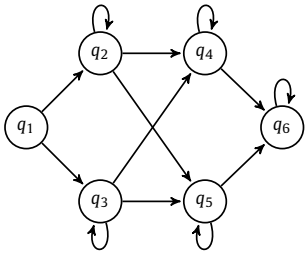
Given an initial configuration $c_0 \in Q \times \mathbb{N}^n$, a *run* ρ starting from c_0 in S is an infinite path in the associated transition system $\mathfrak{T}(S)$ denoted as:

$$\rho := c_0 \xrightarrow{\delta_0} \cdots \xrightarrow{\delta_{\alpha-1}} c_\alpha \xrightarrow{\delta_\alpha} \cdots$$

where $c_i \in Q \times \mathbb{N}^n$ and $\delta_i \in \Delta$ for all $i \in \mathbb{N}$. Let $lab(\rho)$ be the ω-word $\delta_0\delta_1 \ldots$ associated to the run ρ. Note that by definition we have $lab(\rho) \in lab(c_0)$. When E is an ω-regular expression over the finite alphabet Δ and c_0 is an initial configuration, $lab(E, c_0)$ is defined as the set of labels of infinite runs ρ starting at c_0 such that $lab(\rho)$ belongs to the language defined by E. So $lab(E, c_0) \subseteq lab(c_0)$.

We say that a counter system is *flat* if every node in the underlying graph belongs to at most one simple cycle (a cycle being simple if no edge is repeated twice in it) [9]. In a flat counter system, simple cycles can be organized as a DAG where two simple cycles are in the relation whenever there is path between a node of the first cycle and a node of the second cycle. We write `FlatCS` to denote the class of flat counter systems.

Below, we present the control graph of a flat counter system (guards and updates are omitted).



As mentioned in Section 1, one can define other types of counter systems as it is done for instance in [9,2,19,5] by using different types of guards and updates.

A *Kripke structure* S is a tuple $\langle Q, \Delta, \mathbf{l}\rangle$ where $\Delta \subseteq Q \times Q$ and $\mathbf{l} : Q \rightarrow 2^{\mathsf{AT}}$ is the labelling function. When Q is finite, it can be viewed as a degenerate form of counter systems without counters (in the sequel, we take the freedom to see them as counter systems). All standard notions on counter systems naturally apply to (finite) Kripke structures too (configuration, run, flatness, etc.). In the sequel, we shall also investigate the complexity of model-checking problems on flat Kripke structures (such a class is denoted by `FlatKS`).

2.2. Linear-time temporal logic with past and arithmetical constraints

Model-checking problem for Past LTL over finite-state systems is known to be PSPACE-complete [38]. In spite of this nice feature, a propositional variable `p` only represents an abstract property about the current configuration of the system. A more satisfactory solution is to include in the logical language the possibility to express directly constraints between variables of the program, and doing so refining the standard abstraction made with propositional variables. When the variables are typed, they may be interpreted in some specific domain like integers, strings and so on; reasoning in such theories can be performed thanks to satisfiability modulo theories proof techniques, see e.g., [22] in which SMT solvers are used for model-checking infinite-state systems. Hence, the basic idea behind the design of the logic PLTL[C] is to refine the language of atomic formulae and to allow comparisons of counter values. Similar motivations can be found in the introduction of concrete domains in description logics, that are logic-based formalisms for knowledge representation [1,32]. We define below a version of Linear-time Temporal Logic (LTL), dedicated to counter systems in which the atomic formulae are linear constraints and the temporal operators are those of LTL. Note that capacity constraints from [16] are arithmetical constraints different from those defined below.

The formulae of the logic PLTL[C] are defined as follows:

$$\phi \quad ::= \quad \mathtt{p} \mid \mathtt{g} \mid \neg\phi \mid \phi \vee \phi \mid \mathtt{X}\phi \mid \phi \mathtt{U} \phi \mid \mathtt{X}^{-1}\phi \mid \phi \mathtt{S} \phi$$

where $\mathtt{p} \in \mathsf{AT}$ and $\mathtt{g} \in \mathtt{G}(\mathtt{C}_n)$ for some n. We may use the standard abbreviations $\mathtt{F}$, $\mathtt{G}$, $\mathtt{G}^{-1}$ etc. For instance, the formula $\mathtt{GF}(\mathtt{x}_1 + 2 \geq \mathtt{x}_2)$ states that infinitely often the value of counter 1 plus 2 is greater than the value of counter 2. The past-time operators $\mathtt{S}$ and $\mathtt{X}^{-1}$ do not add expressive power to the logic itself [21], but it is known that it helps a lot to express properties succinctly, see e.g. [30,29]. The temporal depth of ϕ, written $td(\phi)$, is defined as the maximal number of imbrications of temporal operators in ϕ. Restriction of PLTL[C] to atomic formulae from AT only is written PLTL[$\emptyset$], it corresponds to the standard version of LTL with past-time operators. Models of PLTL[C] are essentially abstractions of runs from counter systems, i.e. ω-sequences $\sigma : \mathbb{N} \to 2^{\mathsf{AT}} \times \mathbb{N}^{\mathtt{C}}$. Given a model σ and a position $i \in \mathbb{N}$, the satisfaction relation $\models$ for PLTL[C] is defined as follows (Boolean clauses are omitted):

$$
\begin{array}{rcl}
\sigma, i \models \mathtt{p} & \stackrel{\text{def}}{\Leftrightarrow} & \mathtt{p} \in \pi_1(\sigma(i)) \\
\sigma, i \models \mathtt{g} & \stackrel{\text{def}}{\Leftrightarrow} & \mathbf{v}_i \models \mathtt{g} \text{ where } \mathbf{v}_i(\mathtt{x}_j) = \pi_2(\sigma(i))(\mathtt{x}_j) \\
\sigma, i \models \mathtt{X}\phi & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, i+1 \models \phi \\
\sigma, i \models \phi_1 \mathtt{U} \phi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, j \models \phi_2 \text{ for some } i \leq j \\
 & & \text{such that } \sigma, k \models \phi_1 \text{ for all } i \leq k < j \\
\sigma, i \models \mathtt{X}^{-1}\phi & \stackrel{\text{def}}{\Leftrightarrow} & i > 0 \text{ and } \sigma, i-1 \models \phi \\
\sigma, i \models \phi_1 \mathtt{S} \phi_2 & \stackrel{\text{def}}{\Leftrightarrow} & \sigma, j \models \phi_2 \text{ for some } 0 \leq j \leq i \\
 & & \text{such that } \sigma, k \models \phi_1 \text{ for all } j < k \leq i
\end{array}
$$

Given a counter system $\langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$ and a run $\rho := \langle q_0, \mathbf{v}_0 \rangle \xrightarrow{\delta_0} \cdots \xrightarrow{\delta_{p-1}} \langle q_p, \mathbf{v}_p \rangle \xrightarrow{\delta_p} \cdots$, we consider the model $\sigma_\rho : \mathbb{N} \to 2^{\mathsf{AT}} \times \mathbb{N}^{\mathtt{C}}$ such that $\pi_1(\sigma_\rho(i)) \stackrel{\text{def}}{=} \mathbf{l}(q_i)$ and $\pi_2(\sigma_\rho(i))(\mathtt{x}_j) \stackrel{\text{def}}{=} \mathbf{v}_i[j]$ for all $j \in [1, n]$ and for all $i \in \mathbb{N}$. Note that $\pi_2(\sigma_\rho(i))(\mathtt{x}_j)$ is arbitrary for all $j \notin [1, n]$. We extend the satisfaction relation to runs so that $\rho, i \models \phi \stackrel{\text{def}}{\Leftrightarrow} \sigma_\rho, i \models \phi$ whenever ϕ is built from counters in $\mathtt{C}_n$.

The verification problem we are interested in is the model-checking problem for PLTL[C] over counter systems, written MC(L, $\mathcal{C}$), where L is a fragment of PLTL[C] and $\mathcal{C}$ is a class of counter systems. MC(L, $\mathcal{C}$) is defined as follows:

Input: A counter system $S \in \mathcal{C}$, a configuration c_0 and a formula $\phi \in \mathrm{L}$;
Output: Is there a run ρ starting from c_0 in S such that $\rho, 0 \models \phi$?

If the answer is positive, we write $S, c_0 \models \phi$. It is known that for the full class of counter systems, the model-checking problem is undecidable; this is due to the fact that reachability of a control state is undecidable for counter systems manipulating at least two counters [34]. On the other hand, some restrictions can lead to decidability of this problem. This is the case for flat counter systems, for whom it is proved in [14] that the model-checking problem of some temporal logic more expressive than PLTL[C] is decidable. Unfortunately the decision procedure proposed in [14] involves an exponential reduction to the satisfiability problem for some formulae of the Presburger arithmetic and as a consequence has a high complexity.

Theorem 2.3. *(See [14,27].)* MC(PLTL[C], FlatCS) *can be solved in* 4ExpTime.
MC(PLTL[$\emptyset$], FlatKS) *restricted to formulae with temporal operators* U,X *is NP-complete.*

The main goal of this work is to show that a much better upper bound for MC(PLTL[C], FlatCS) is possible and to provide the precise complexity of this problem and of its related fragments.

3. Fundamental structures: minimal path schemas

In this section, we introduce a fundamental notion for flat counter systems, namely the path schemas. Indeed, every flat counter system can be decomposed into a finite set of (minimal) path schemas and there are only an exponential number of them. In the forthcoming nondeterministic algorithms to solve model-checking problems on flat counter systems, the first step consists in guessing a minimal path schema and then computations are performed on such a structure. This explains why path schemas are a central notion in our work.

Let $S = \langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$ be a flat counter system. A *path segment* p of S is a finite word (or sequence) of transitions from Δ such that $\mathit{target}(p(i)) = \mathit{source}(p(i+1))$ for all $0 \leq i < \mathrm{len}(p) - 1$. We write $\mathit{first}(p)$ [resp. $\mathit{last}(p)$] to denote the first [resp. last] control state of a path segment, in other words $\mathit{first}(p) = \mathit{source}(p(0))$ and $\mathit{last}(p) = \mathit{target}(p(\mathrm{len}(p) - 1))$. We also write

$$\mathit{effect}(p) \stackrel{\text{def}}{=} \sum_{0 \leq i < \mathrm{len}(p)} \mathit{update}(p(i))$$

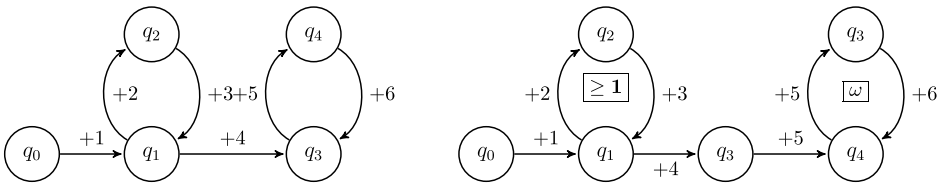


Fig. 1. A flat counter system and one of its minimal path schemas.

representing the total effect of the updates along the path segment. A path segment p is said to be *simple* if $\mathrm{len}(p) > 0$ and for all $0 \leq i, j < \mathrm{len}(p)$, $p(i) = p(j)$ implies $i = j$ (no repetition of transitions). A *loop* is a simple path segment p such that $first(p) = last(p)$. If a path segment is not a loop it is called a *non-loop segment*. A *path schema* P is an ω-regular expression built over the alphabet of transitions such that its language represents an overapproximation of the set of labels obtained from infinite runs following the transitions of P.

Definition 3.1 *(Path schema).* A path schema P is of the form $p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ verifying the following conditions:

1. $l_1, \dots, l_k$ are loops,
2. $p_1 l_1 p_2 l_2 \dots p_k l_k$ is a path segment.

We denote by CPS [resp. KPS] the class of path schemas of counter systems [resp. the class of path schemas of finite Kripke structures]. We write $\mathrm{len}(P)$ for $\mathrm{len}(p_1 l_1 p_2 l_2 \dots p_k l_k)$ and we denote by $\mathrm{nbloops}(P)$ the number of loops in P (i.e. $\mathrm{nbloops}(P) = k$). Let $\mathcal{L}(P)$ stand for the set of infinite words in Δ^ω that belong to the language defined by P. Note that some elements of $\mathcal{L}(P)$ may not correspond to any run whenever constraints on counter values are not satisfied. Given $w \in \mathcal{L}(P)$, we write $iter_P(w)$ to denote the unique tuple in $(\mathbb{N} \setminus \{0\})^{k-1}$ such that $w = p_1 l_1^{iter_P(w)[1]} p_2 l_2^{iter_P(w)[2]} \dots p_k l_k^\omega$. So $iter_P(w)[i]$ is the number of times the loop l_i is taken, for every $i \in [1, k-1]$. Then, given a configuration c_0, the set $iter_P(c_0)$ is defined as the set of vectors

$$iter_P(c_0) \stackrel{\text{def}}{=} \{iter_P(w) \in (\mathbb{N} \setminus \{0\})^{k-1} \mid w \in lab(P, c_0)\}$$

We recall that $lab(P, c_0)$ denotes the set of labels of infinite runs ρ starting at c_0 such that $lab(\rho)$ belongs to $\mathcal{L}(P)$.

Finally, we say that a run ρ starting in a configuration c_0 *respects* a path schema P if $lab(\rho) \in lab(P, c_0)$ and for such a run, we write $iter_P(\rho)$ to denote $iter_P(lab(\rho))$. By definition, if ρ respects P, then each loop l_i is visited at least once, and the last one infinitely.

So far, a flat counter system may have an infinite set of path schemas. To see this, it is sufficient to unroll loops and consider the unrolling as an alternating sequence of path and loop segments. However, we can impose minimality conditions on path schemas without sacrificing completeness. A path schema $p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ is *minimal* whenever

1. $p_1 \cdots p_k$ is either the empty word or a simple non-loop segment,
2. $l_1, \dots, l_k$ are loops with disjoint sets of transitions.

We can then deduce the following lemma which is a simple consequence of the fact that in a minimal path schema, each transition occurs at most twice.

Lemma 3.2. *Given a flat counter system $S = \langle Q, C_n, \Delta, \mathbf{l} \rangle$, the total number of minimal path schemas of S is finite and is smaller than* $\mathrm{card}(\Delta)^{(2 \times \mathrm{card}(\Delta))}$ *and the length of a minimal path schema of S is bounded by* $2 \times \mathrm{card}(\Delta)$.

In Fig. 1, we present a flat counter system S with a unique counter and one of its minimal path schemas. Each transition δ_i labelled by $+i$ corresponds to a transition with the guard $\top$ and the update value $+i$. The minimal path schema shown in Fig. 1 corresponds to the ω-regular expression $\delta_1(\delta_2\delta_3)^+\delta_4\delta_5(\delta_6\delta_5)^\omega$. In order to avoid confusions between path schemas and flat counter systems that look like path schemas, simple loops in the representation are labelled by $\boxed{\omega}$ or $\boxed{\geq 1}$ depending whether the simple loop is the last one or not. Note that in the representation of path schemas, a state may occur several times, as it is the case for q_3 (this cannot happen in the representation of counter systems).

Minimal path schemas play a crucial role in the sequel, mainly because of the following properties.

Lemma 3.3. *Let S be a flat counter system and P be one of its path schemas. There is a minimal path schema P' such that every run respecting P respects P' as well.*

Proof. Consider any non-minimal path schema $P = p_1(l_1)^+ \cdots p_k(l_k)^\omega$. P is non-minimal because some edge δ occurs strictly more than twice in P. Consider the segments from P containing three consecutive occurrences of δ. Let w denote the subword of $p_1 l_1 \cdots p_k l_k$ formed by taking the segments together. We can write the word w as $w_1 \delta w_2 \delta w_3 \delta w_4$

and thus $source(\delta) = target(w_2) = target(w_3)$. Similarly, it is clear that $target(\delta) = source(w_2) = source(w_3)$. Since P is obtained from a flat counter system, δ belongs to at most one simple cycle. In this case, clearly there are two loops δw_2 and δw_3 starting and ending at $source(\delta)$ and hence both δw_2 and δw_3 are iterations of the same loop. Thus there exists a loop segment l' such that $\delta w_2 = l'^{\alpha}$ and $\delta w_3 = l'^{\beta}$ for some $\alpha, \beta \in \mathbb{N}$ and $target(l') = source(l') = source(\delta)$. Let P_1 be the path schema obtained by replacing $\sigma_i \dots \sigma_j \dots \sigma_k$ with $w_1 (l')^+ \delta w_4$ in P. Clearly, P_1 is a path schema in due form and $\mathcal{L}(P_1) = \mathcal{L}(P)$. Note that the above transformation reduces the number of times δ appear by at least one. Performing the above construction repeatedly for any transition appearing strictly more than twice in P, we get a minimal path schema P'. Since, the set of accepted words is preserved in the transformation, $\mathcal{L}(P) = \mathcal{L}(P')$ and hence for every run respecting P there is a run respecting P' (and the other way around). □

Finally, the conditions imposed on the structure of path schemas imply that each run in a flat counter system starting at configuration c_0 respects a path schema (see similar statements in [31]). Lemmas 3.2 and 3.3 entail the following result.

Corollary 3.4. *For a flat counter system S and a configuration c_0, there is a finite set of minimal path schemas X of cardinality at most* $\mathrm{card}(\Delta)^{(2\times \mathrm{card}(\Delta))}$ *such that* $lab(c_0) = lab(\bigcup_{P\in X} P, c_0)$.

4. Stuttering theorem for PLTL[Ø]

Stuttering of finite words or single letters has been instrumental to show several results about the expressive power of PLTL[Ø] fragments, see e.g. [35,28]; for instance, PLTL[Ø] restricted to the temporal operator U characterizes exactly the class of formulae defining classes of models invariant under stuttering. This is refined in [28] for PLTL[Ø] restricted to U and X, by taking into account not only the U-depth but also the X-depth of formulae and by introducing a principle of stuttering that involves both letter stuttering and word stuttering. In this section, we establish another substantial generalization that involves the full logic PLTL[Ø] (with its past-time temporal operators). Roughly speaking, we show that if $\sigma_1 \mathbf{s}^M \sigma_2, 0 \models \phi$ where $\sigma_1 \mathbf{s}^M \sigma_2$ is a PLTL[Ø] model ($\sigma_1, \mathbf{s}$ being finite words), ϕ is PLTL[Ø] formula verifying $td(\phi) \leq N$ and $M > 2N$, then $\sigma_1 \mathbf{s}^{2N+1} \sigma_2, 0 \models \phi$ (and other related properties). This allows us to conclude that if there is a run

(a) satisfying a path schema P (see Section 3) and,
(b) verifying a PLTL[Ø] formula ϕ,

then there is a run satisfying (a), (b) and each loop is visited at most $2 \times td(\phi) + 5$ times, leading to an NP upper bound (see Proposition 5.1). This extends a result without past-time operators [27]. Moreover, this turns out to be a key property (Theorem 4.1) to establish the NP upper bound even in the presence of counters (but additional work needs to be done, see Section 6). It is worth observing that Theorem 4.1 below is interesting for its own sake, independently of our investigation on flat counter systems.

Given $M, M', N \in \mathbb{N}$, we write $M \approx_N M'$ iff $\min(M, N) = \min(M', N)$. Given $w = w_1 u^M w_2$, $w' = w_1 u^{M'} w_2 \in \Sigma^\omega$ and $i, i' \in \mathbb{N}$, we define an equivalence relation $\langle w, i\rangle \approx_N \langle w', i'\rangle$ (implicitly parameterized by w_1, w_2 and u) such that $\langle w, i\rangle \approx_N \langle w', i'\rangle$ means that the number of copies of u before position i and the number of copies of u before position i' are related by $\approx_N$ and the same applies for the number of copies after the positions. Moreover, if i and i' occur in the part where u is repeated, then they correspond to identical positions in u. More formally, $\langle w, i\rangle \approx_N \langle w', i'\rangle \stackrel{\text{def}}{\Leftrightarrow} M \approx_{2N} M'$ and either, $M, M' \leq 2N$ and $i = i'$, or one of the following conditions holds true:

1. $i, i' < \mathrm{len}(w_1) + N \cdot \mathrm{len}(u)$ and $i = i'$.
2. $i \geq \mathrm{len}(w_1) + (M - N) \cdot \mathrm{len}(u)$ and $i' \geq \mathrm{len}(w_1) + (M' - N) \cdot \mathrm{len}(u)$ and $(i - i') = (M - M') \cdot \mathrm{len}(u)$.
3. $\mathrm{len}(w_1) + N \cdot \mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M - N) \cdot \mathrm{len}(u)$ and $\mathrm{len}(w_1) + N \cdot \mathrm{len}(u) \leq i' < \mathrm{len}(w_1) + (M' - N) \cdot \mathrm{len}(u)$ and $|i - i'| = 0 \bmod \mathrm{len}(u)$.

The conditions merely state that if $M, M' \leq 2N$ then the positions in the respective words have to be same. Since $M \approx_{2N} M'$, we have $M, M' \leq 2N$ implies that $M = M'$, that is $w = w'$. Otherwise, when $M, M' > 2N$, the indices i and i' should point to the corresponding equivalent positions in the words w and w' when i, i' are located upto the first N iterations of u or from the last N iterations of u. Finally, if the positions i and i' belong to any iteration of u between the block of the first N iterations of u and the block of the last N iterations of u, then i and i' have the same relative position with respect to u.

Fig. 2 presents two words w and w' over the alphabet $\Sigma = \{\square, \blacksquare\}$ such that w is of the form $w_1 (\square\blacksquare)^7 w_2$ and w' is of the form $w_1 (\square\blacksquare)^8 w_2$. The relation $\approx_3$ is represented by edges between positions: each edge from positions i of w to positions i' of w' represents the fact that $\langle w, i\rangle \approx_3 \langle w', i'\rangle$.

In order to prove our stuttering theorem for PLTL[Ø], we need to express some properties concerning the relation $\approx$ whose proofs can be found in Appendix A. Let $w = w_1 u^M w_2$, $w' = w_1 u^{M'} w_2 \in \Sigma^\omega$, $i, i' \in \mathbb{N}$ and $N \geq 2$ such that $M, M' > 2N$ and $\langle w, i\rangle \approx_N \langle w', i'\rangle$. We can show the following properties:

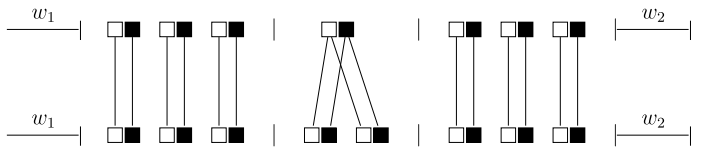


Fig. 2. Two words w, w' with $u = \square\blacksquare$ and the relation $\approx_3$.

(Claim 1) $\langle w, i\rangle \approx_{N-1} \langle w', i'\rangle$ and $w(i) = w'(i')$.
(Claim 2) $i, i' > 0$ implies $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$.
(Claim 3) $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$
(Claim 4) For all $j \geq i$, there is $j' \geq i'$ such that $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$ and for all $k' \in [i', j'-1]$, there is $k \in [i, j-1]$ such that $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
(Claim 5) For all $j \leq i$, there is $j' \leq i'$ such that $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$ and for all $k' \in [j'-1, i']$, there is $k \in [j-1, i]$ such that $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.

We now state our stuttering theorem for PLTL[Ø] that is tailored for our future needs.

Theorem 4.1 *(Stuttering). Let $N \geq 2$ and $M, M' > 2N$ and $\sigma = \sigma_1 \mathbf{s}^M \sigma_2$, $\sigma' = \sigma_1 \mathbf{s}^{M'} \sigma_2 \in (2^{\mathrm{AT}})^\omega$. For all $i, i' \in \mathbb{N}$ such that $\langle \sigma, i\rangle \approx_N \langle \sigma', i'\rangle$ and for all* PLTL[Ø] *formulae ϕ with $td(\phi) \leq N$, we have $\sigma, i \models \phi$ iff $\sigma', i' \models \phi$.*

This theorem basically states that no PLTL[Ø] formula ϕ with $td(\phi) \leq N$ is able to distinguish the infinite words $\sigma = \sigma_1 \mathbf{s}^M \sigma_2$ and $\sigma' = \sigma_1 \mathbf{s}^{M'} \sigma_2$ when $M, M' > 2N$. Before proving the theorem we assign a tuple of naturals for each PLTL[Ø] formula ϕ denoted as $\mathtt{rank}(\phi) \in \mathbb{N} \times \mathbb{N}$ and defined as $\mathtt{rank}(\phi) = \langle td(\phi), \mathrm{size}(\phi)\rangle$ where $\mathrm{size}(\phi)$ denotes the size of the formula ϕ which is defined as usual. For instance, $\mathrm{size}(\mathtt{p}) = 1$ and $\mathrm{size}(\phi_1 \wedge \phi_2) = \mathrm{size}(\phi_1 \mathtt{U} \phi_2) = \mathrm{size}(\phi_1) + \mathrm{size}(\phi_2) + 1$. The tuple $\mathtt{rank}(\phi)$ is called the rank of the formula ϕ and we use the lexicographic order $<$ over the ranks of formulae. This order is defined as follows $\langle m_1, n_1\rangle < \langle m_2, n_2\rangle$ iff either $m_1 < m_2$ or ($m_1 = m_2$ and $n_1 < n_2$).

Proof (Stuttering Theorem). Let $N \geq 2$ and $M, M' > 2N$ and $\sigma = \sigma_1 \mathbf{s}^M \sigma_2$, $\sigma' = \sigma_1 \mathbf{s}^{M'} \sigma_2 \in (2^{\mathrm{AT}})^\omega$. We consider as well $i, i' \in \mathbb{N}$ such that $\langle \sigma, i\rangle \approx_N \langle \sigma', i'\rangle$. We will show that for all PLTL[Ø] formula ϕ with $td(\phi) \leq N$, we have $\sigma, i \models \phi$ iff $\sigma', i' \models \phi$. The proof is by induction on the formula rank. We assume furthermore that (Claim 1)–(Claim 5) are true (the proofs of these claims being provided in Appendix A).

- **Base case:** As base case we consider a PLTL[Ø] formula ϕ with $\mathtt{rank}(\phi) = \langle 0, 1\rangle$ (i.e. formulae which are just atomic propositions). Thus we have $\phi = \mathtt{p}$ for some $\mathtt{p} \in \mathrm{AT}$. Since, $\langle \sigma, i\rangle \approx_N \langle \sigma', i'\rangle$ for some $N \geq 2$ and by (Claim 1), $w(i) = w(i')$, either $\mathtt{p} \in w(i)$ or $\mathtt{p} \notin w(i)$. In any case, $\sigma, i \models \mathtt{p}$ iff $\sigma', i \models \mathtt{p}$.
- **Induction step:** The induction hypothesis is the following one: for all formulae ψ with $\mathtt{rank}(\psi) < \mathtt{rank}(\phi)$ and for all j, j' such that $\langle \sigma, j\rangle \approx_{td(\psi)} \langle \sigma', j'\rangle$, we have $\sigma, j \models \psi$ iff $\sigma', j' \models \psi$. To prove that the desired property holds, we proceed by the following case analysis on the shape of the formula ϕ.
  - $\phi = \mathtt{X}\phi'$: Since $td(\phi') < td(\phi)$, we have $\mathtt{rank}(\phi') < \mathtt{rank}(\phi)$. By (Claim 3), $\langle \sigma, i+1\rangle \approx_{N-1} \langle \sigma', i'+1\rangle$ and using (Claim 1) repeatedly $\langle \sigma, i+1\rangle \approx_{td(\phi')} \langle \sigma', i'+1\rangle$. By induction hypothesis, we have $\sigma, i+1 \models \phi'$ iff $\sigma', i'+1 \models \phi'$. Thus, $\sigma, i \models \mathtt{X}\phi'$ iff $\sigma', i' \models \mathtt{X}\phi'$.
  - $\phi = \mathtt{X}^{-1}\phi'$: First if $i = 0$ or $i' = 0$, since $\langle \sigma, i\rangle \approx_N \langle \sigma', i'\rangle$, we have $i = i' = 0$ and in that case $\sigma, i \not\models \mathtt{X}^{-1}\phi'$ and $\sigma', i' \not\models \mathtt{X}^{-1}\phi'$. Assume now $i > 0$ and $i' > 0$. Since $td(\phi') < td(\phi)$, we have $\mathtt{rank}(\phi') < \mathtt{rank}(\phi)$. By (Claim 2), $i, i' > 0$ implies $\langle \sigma, i-1\rangle \approx_{N-1} \langle \sigma', i'-1\rangle$ and consequently, thanks to (Claim 1), we deduce $\langle \sigma, i-1\rangle \approx_{td(\phi')} \langle \sigma', i'-1\rangle$. Using induction hypothesis, $\sigma, i-1 \models \phi'$ iff $\sigma', i'-1 \models \phi'$. This allows us to deduce that $\sigma, i \models \mathtt{X}^{-1}\phi'$ iff $\sigma', i' \models \mathtt{X}^{-1}\phi'$.
  - $\phi = \phi_1 \mathtt{U} \phi_2$: First, we suppose that $\sigma, i \models \phi_1 \mathtt{U} \phi_2$. There is $j \geq i$ such that $\sigma, j \models \phi_2$ and $\sigma, k \models \phi_1$ for every $k \in [i, j-1]$. We have that $td(\phi_1) \leq N-1$ and $td(\phi_2) \leq N-1$, and consequently $\mathtt{rank}(\phi_1) < \mathtt{rank}(\phi)$ and $\mathtt{rank}(\phi_2) < \mathtt{rank}(\phi)$. Using (Claim 4), there is $j' \geq i'$ such that $\langle \sigma, j\rangle \approx_{N-1} \langle \sigma', j'\rangle$ and for all $k' \in [i', j'-1]$ there is $k \in [i, j-1]$ such that $\langle \sigma, k\rangle \approx_{N-1} \langle \sigma', k'\rangle$. Using (Claim 1) and the induction hypothesis, we deduce that $\sigma', j' \models \phi_2$ and $\sigma', k' \models \phi_1$ for every $k' \in [i', j'-1]$. Thus $\sigma', i' \models \phi_1 \mathtt{U} \phi_2$. Following the same reasoning, we can prove that if $\sigma', i' \models \phi_1 \mathtt{U} \phi_2$, we have $\sigma, i \models \phi_1 \mathtt{U} \phi_2$.
  - $\phi = \phi_1 \mathtt{S} \phi_2$: We first suppose that $\sigma, i \models \phi_1 \mathtt{S} \phi_2$; consequently there is $0 \leq j \leq i$ such that $\sigma, j \models \phi_2$ and $\sigma, k \models \phi_1$ for every $k \in [j+1, i]$. We have that $td(\phi_1) \leq N-1$ and $td(\phi_2) \leq N-1$, and consequently $\mathtt{rank}(\phi_1) < \mathtt{rank}(\phi)$ and $\mathtt{rank}(\phi_2) < \mathtt{rank}(\phi)$. Using (Claim 5), there is $0 \leq j' \leq i'$ such that $\langle \sigma, j\rangle \approx_{N-1} \langle \sigma', j'\rangle$ and for all $k' \in [j'+1, i']$ there is $k \in [j+1, i]$ such that $\langle \sigma, k\rangle \approx_{N-1} \langle \sigma', k'\rangle$. Using (Claim 1) and the induction hypothesis, we deduce that $\sigma', j' \models \phi_2$ and $\sigma', k' \models \phi_1$ for every $k' \in [j'+1, i']$. Thus $\sigma', i' \models \phi_1 \mathtt{S} \phi_2$. Following the same reasoning, we can prove that if $\sigma', i' \models \phi_1 \mathtt{S} \phi_2$, we have $\sigma, i \models \phi_1 \mathtt{S} \phi_2$.

- $\phi = \neg\phi'$: In that case $td(\phi') = td(\phi)$ and $\mathrm{size}(\phi') = \mathrm{size}(\phi) - 1$, hence $\mathtt{rank}(\phi') < \mathtt{rank}(\phi)$. Since $\langle \sigma, i\rangle \approx_N \langle \sigma', i'\rangle$, using (Claim 1), we have $\langle \sigma, i\rangle \approx_{td(\phi')} \langle \sigma', i'\rangle$ (remember that $td(\phi') \leq N$) and by induction hypothesis, we deduce that $\sigma, i \not\models \phi'$ iff $\sigma', i' \not\models \phi'$. Thus, $\sigma, i \models \neg\phi'$ iff $\sigma', i' \models \neg\phi'$.
- $\phi = \phi_1 \vee \phi_2$: For such a formula, we consider the two following subcases:
  (a) ($td(\phi_1) \leq td(\phi) - 1$ and $td(\phi_2) = td(\phi)$) or ($td(\phi_2) \leq td(\phi) - 1$ and $td(\phi_1) = td(\phi)$). Without loss of generality, let us assume that $td(\phi_1) \leq td(\phi) - 1$ and $td(\phi_2) = td(\phi)$. We have $\mathrm{size}(\phi_2) \leq \mathrm{size}(\phi) - 1$. This implies that $\mathtt{rank}(\phi_1) < \mathtt{rank}(\phi)$ and $\mathtt{rank}(\phi_2) < \mathtt{rank}(\phi)$. Since $\langle \sigma, i\rangle \approx_N \langle \sigma', i'\rangle$, using (Claim 1) repeatedly, we have $\langle \sigma, i\rangle \approx_{td(\phi_1)} \langle \sigma', i'\rangle$ and $\langle \sigma, i\rangle \approx_{td(\phi_2)} \langle \sigma', i'\rangle$ (remember that $td(\phi_1) \leq N$ and that $td(\phi_2) \leq N$). Thus, by induction hypothesis, $\sigma, i \models \phi_1$ iff $\sigma', i' \models \phi_1$ and $\sigma, i \models \phi_2$ iff $\sigma', i' \models \phi_2$. Hence, $\sigma, i \models \phi_1 \vee \phi_2$ iff $\sigma', i' \models \phi_1 \vee \phi_2$.
  (b) $td(\phi_1) = td(\phi_2) = td(\phi)$. In that case, we have $\mathrm{size}(\phi_1) \leq \mathrm{size}(\phi) - 1$ and $\mathrm{size}(\phi_2) \leq \mathrm{size}(\phi) - 1$. Consequently $\mathtt{rank}(\phi_1) < \mathtt{rank}(\phi)$ and $\mathtt{rank}(\phi_2) < \mathtt{rank}(\phi)$. As previously, this allows us to deduce that $\sigma, i \models \phi_1 \vee \phi_2$ iff $\sigma', i' \models \phi_1 \vee \phi_2$. □

A similar proof can be found in [28] and our proof provides a generalization, in some respect, by dealing with past-time operators. On the other hand, for the purpose of model-checking flat counter systems, we need a simpler property than what is established in [28]. The proof of Theorem 4.1 essentially amounts to design a suitable strategy in Ehrenfeucht–Fraïssé (EF) games [18]. Consequently, an alternative proof is possible by using EF games [18]; this does not necessarily provide a shorter proof and it requires to use properties of the game, as done in [18]. In particular even though the above proof and the proof of [18, Theorem 4.4] are also similar in nature, we nevertheless provide the proof which is tailored to our specific need and moreover this allows us to be self-contained. Note that to obtain the same result thanks to EF games, it is enough to observe that the relation $\langle w, i\rangle \approx_N \langle w', i'\rangle$ leads exactly to a winning strategy for the Duplicator in the Ehrenfeucht–Fraïssé game for Past LTL over the words w and w'. Another reason for being self-contained is that the properties shown in both proofs are slightly different; typically, for our model-checking procedure, we need a polynomial bound.

Finally, from Theorem 4.1, we conclude that given a formula ϕ from PLTL[Ø], a word σ with an infix **s** repeated more than $\max(2.td(\phi)+1, 5)$ times satisfies ϕ iff the word σ' in which **s** is repeated exactly $\max(2.td(\phi)+1, 5)$ times satisfies ϕ.

5. Model-checking path schemas

Path schemas in flat counter systems and Kripke structures are defined as ω-regular expressions over the alphabet of transitions. Nevertheless, they can also be viewed as restricted flat systems, i.e. as restricted flat counter systems or Kripke structures, with the proviso that each loop is visited at least once and the structure is more constrained since it follows the definition of path schemas. For instance, in a path schema, it is not possible to have two transitions from the same "state" leading to two different loops. In this section, we consider the model-checking problem with input path schemas (instead of counter systems). Studying such restricted flat systems is indeed important to understand where stands the frontier for the lower and upper bounds of our complexity results.

5.1. Complexity results for path schemas of Kripke structures

We begin by looking at the model-checking problem for PLTL[Ø] over path schemas of a flat Kripke structure. We write MC(PLTL[Ø], KPS) to denote the problem defined below:

Input: A finite flat Kripke structure S, a path schema P of S, a configuration c_0 and a formula ϕ of PLTL[Ø];
Output: Is there a run ρ starting at c_0, respecting P, and such that $\rho, 0 \models \phi$?

In case of positive answer, we write $P, c_0 \models \phi$. Let ρ and ρ' be runs respecting P. For $\alpha \geq 0$, we write $\rho \equiv_\alpha \rho' \stackrel{\text{def}}{\Leftrightarrow}$ for every $i \in [1, \mathrm{nbloops}(P) - 1]$, we have $\min(iter_P(\rho)[i], \alpha) = \min(iter_P(\rho')[i], \alpha)$. We state below a result concerning the runs of flat counter systems (including finite flat Kripke structures) when respecting the same path schema.

Proposition 5.1. *Let S be a flat counter system, P be a path schema, and $\phi \in$ PLTL[Ø]. For all runs ρ and ρ' respecting P such that $\rho \equiv_{2td(\phi)+5} \rho'$, we have $\rho, 0 \models \phi$ iff $\rho', 0 \models \phi$.*

This property can be proved by applying the Stuttering Theorem (Theorem 4.1) repeatedly in order to get rid of the unwanted iterations of the loops.

Our algorithm for MC(PLTL[Ø], KPS) takes advantage of a result from [29] for model-checking ultimately periodic models with formulae from Past LTL. An *ultimately periodic path* is an infinite word in Δ^ω of the form uv^ω where uv is a path segment and $first(v) = last(v)$. More generally, an *ultimately periodic* word over the alphabet Σ is an ω-word in Σ^ω that can be written as uv^ω where u is the *prefix* and v is the *loop*. According to [29], given an ultimately periodic path w, and a formula $\phi \in$ PLTL[Ø], the problem of checking whether there exists a run ρ such that $lab(\rho) = w$ and $\rho, 0 \models \phi$ is in PTime. Using Theorem 4.1, we can bound the maximal number of iterations of v we need to check for a given Past LTL formula ϕ.

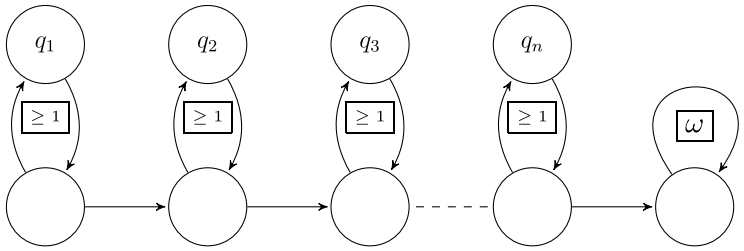


Fig. 3. A simple path schema P.

Lemma 5.2. MC(PLTL[Ø], KPS) *is in NP.*

Proof. This result is a consequence of Proposition 5.1 and [29]. Indeed, given $\phi \in \text{PLTL}[\emptyset]$ and $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$, our algorithm first guesses $\mathbf{y} \in [1, 2td(\phi)+5]^{k-1}$ and check whether $\rho, 0 \models \phi$ where ρ is the obvious ultimately periodic word such that $lab(\rho) = p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots p_k l_k^\omega$. Since $\mathbf{y}$ is of polynomial size and since $\rho, 0 \models \phi$ can be checked in polynomial time by [29, Theorem 5.1], we get the NP upper bound. □

Furthermore, from [27], we have the lower bound for MC(PLTL[Ø], KPS).

Lemma 5.3. *[27]* MC(PLTL[Ø], KPS) *is NP-hard even if the temporal operators in the considered* PLTL[Ø] *formulae are restricted to* X *and* F.

For the sake of completeness, we provide the proof presented in [27] adapted to our context.

Proof. The proof is by reduction from the SAT problem. Let Φ be a Boolean formula built over the propositional variables $PV = \{\mathtt{p}_1, \cdots, \mathtt{p}_n\}$. We build a path schema P and a formula ψ such that Φ is satisfiable iff there is a run respecting P and satisfying ψ. The path schema P is the one described in Fig. 3 so that the truth of the propositional variable $\mathtt{p}_i$ is encoded by the fact that the loop containing q_i is visited twice, otherwise it is visited once. The formula ψ is defined as a conjunction $\psi_{1\vee 2} \wedge \psi_{truth}$ where $\psi_{1\vee 2}$ states that each loop is visited at most twice and ψ_{truth} establishes the correspondence between the truth of $\mathtt{p}_i$ and the number of times the loop containing q_i is visited. Formula $\psi_{1\vee 2}$ is equal to $[\bigwedge_i (\mathtt{G}(q_i \wedge \mathtt{XX}q_i \Rightarrow \mathtt{XXXG}\neg q_i))]$ whereas ψ_{truth} is defined from Φ by replacing each occurrence of $\mathtt{p}_i$ by $\mathtt{F}(q_i \wedge \mathtt{XX}q_i)$.

Let us check the correctness of the reduction. Let $v : PV \to \{\top, \bot\}$ be a valuation satisfying Φ. Let us consider the run ρ respecting P such that $iter_P(\rho)[i] \stackrel{\text{def}}{=} 2$ if $v(\mathtt{p}_i) = \top$, otherwise $iter_P(\rho)[i] \stackrel{\text{def}}{=} 1$ for all $i \in [1, n]$. It is easy to check that $\rho, 0 \models \psi$. Conversely, if there is a run ρ respecting P such that $\rho, 0 \models \psi$, the valuation v satisfies Φ where for all $i \in [1, n]$, we have $v(\mathtt{p}_i) = \top \stackrel{\text{def}}{\Leftrightarrow} iter_P(\rho)[i] = 2$. □

Then, the NP-completeness of MC(PLTL[Ø], KPS) follows from the two previous lemmas. Let us also consider the case where we restrict the class of path schemas by bounding the number of loops. Hence, for a fixed $k \in \mathbb{N}$, we write MC(PLTL[Ø], KPS(k)) to denote the restriction of MC(PLTL[Ø], KPS) to path schemas with at most k loops. When k is fixed, the number of ultimately periodic paths w in $\mathcal{L}(P)$ such that each loop (except the last one) is visited at most $2td(\phi)+5$ times is bounded by $(2td(\phi)+5)^k$, which is polynomial in the size of the input (because k is fixed). From these considerations, we deduce the following result.

Theorem 5.4. MC(PLTL[Ø], KPS) *is NP-complete.*
Given a fixed $k \in \mathbb{N}$, MC(PLTL[Ø], KPS(k)) *is in* PTime.

It can be shown that MC(PLTL[Ø], KPS(k)) is in NC, hence giving a tighter upper bound for the problem. This can be obtained by observing that we can run the NC algorithm from [26] for model checking PLTL[Ø] over ultimately periodic paths in parallel on $(2td(\phi)+5)^k$ (polynomially many) different paths.

5.2. Result for flat Kripke structures

Now, we show how to solve MC(PLTL[Ø], FlatKS) by using Lemma 5.2. From Lemma 3.2, the number of minimal path schemas in a flat Kripke structure $S = \langle Q, \Delta, \mathbf{l} \rangle$ is finite and the length of a minimal path schema is at most $2 \times \text{card}(\Delta)$. Hence, for solving the model-checking problem for an initial state q_0 and a PLTL[Ø] formula ϕ, a possible algorithm consists in choosing non-deterministically a minimal path schema P starting at q_0 and then apply the algorithm used to establish Lemma 5.2. This new algorithm would be in NP. Furthermore, thanks to Corollary 3.4, we know that if there exists a run

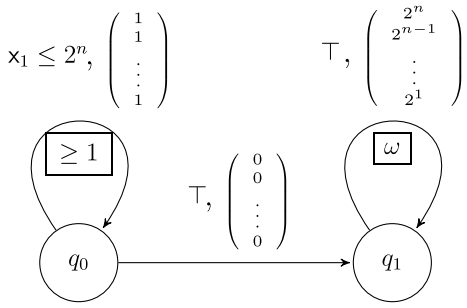


Fig. 4. Path schema P.

ρ of S such that $\rho, 0 \models \phi$ then there exists a minimal path schema P such that ρ respects P. Consequently there is an algorithm in NP to solve MC(PLTL[$\emptyset$], FlatKS) and NP-hardness can be established as a variant of the proof of Lemma 5.3.

Theorem 5.5. MC(PLTL[$\emptyset$], FlatKS) *is NP-complete.*

5.3. Some lower bounds in the presence of counters

We will now provide some complexity lower bounds when considering path schemas over counter systems. As for path schemas from Kripke structures, we use CPS(k) to denote the class of path schemas obtained from flat counter systems with number of loops bounded by k. Surprisingly, in the presence of counters, bounding the number of loops preserves NP-hardness as soon as there are at least two loops. The case with a single loop is dealt with in Section 8.2.

Lemma 5.6. *For $k \geq 2$,* MC(PLTL[C], CPS(k)) *is NP-hard.*

The proof is by reduction from SAT and it is less straightforward than the proof for Lemma 5.3 or the reduction presented in [27] when path schemas are involved. Indeed, we cannot encode the nondeterminism in the structure itself and the structure has only a constant number of loops. Actually, we cannot use a separate loop for each counter; the reduction is done by encoding the nondeterminism in the (possibly exponential) number of times a single loop is taken, and then using its binary encoding as an assignment for the propositional variables.

Proof. The proof is by reduction from the problem SAT. Let Φ be a Boolean formula built over the propositional variables in $\{\mathtt{p}_1, \ldots, \mathtt{p}_n\}$. We build a path schema $P \in \mathtt{CPS}(2)$, an initial configuration (in which all counters will be equal to zero) and a formula ψ such that Φ is satisfiable iff there is a run respecting P and starting at the initial configuration such that it satisfies ψ. The path schema P is the one described in Fig. 4; it has one internal loop and a second loop that is visited infinitely. The guard $\mathsf{x}_1 \leq 2^n$ enforces that the first loop is visited α times with $\alpha \in [1, 2^n]$, which corresponds to guess a propositional valuation such that the truth value of the propositional variable $\mathtt{p}_i$ is $\top$ whenever the ith bit of $\alpha - 1$ is equal to 1. When $\alpha - 1$ is encoded in binary with n bits, we assume the first bit is the most significant one. Note that the internal loop has to be visited at least once since P is a path schema.

Since the logical language does not allow to access to the ith bit of a counter value, we simulate the test by arithmetical constraints in the formula when the second loop of the path schema is visited. For every $\alpha \in [1, 2^n]$ and every $i \in [1, n]$, we write α^i_u to denote the value in $[0, 2^{i-1} - 1]$ corresponding to the $i - 1$ first bits of $\alpha - 1$. When $i = 1$, by convention $\alpha^i_u = 0$. Similarly, we write α^i_d to denote the value in $[0, 2^{n+1-i} - 1]$ corresponding to the $(n + 1 - i)$ last bits of $\alpha - 1$. Observe that $\alpha - 1 = \alpha^i_u \times 2^{n-i+1} + \alpha^i_d$. One can show that the propositions below are equivalent:

1. ith bit of $\alpha - 1$ is 1,
2. there is some $k \geq 0$ such that $k \times 2^{n+1-i} + (\alpha - 1) \in [2^n + 2^{n-i}, 2^n + 2^{n+1-i} - 1]$.

Actually, we shall show that k is unique and the only possible value is $2^{i-1} - \alpha^i_u$. Before showing the equivalence between (1.) and (2.), we can observe that condition (2.) can be expressed by the formula

$$\mathtt{F}(q_1 \wedge ((\mathsf{x}_i - 1) \geq 2^n + 2^{n-i}) \wedge ((\mathsf{x}_i - 1) \leq 2^n + 2^{n-i+1} - 1))$$

First, note that $[2^n + 2^{n-i}, 2^n + 2^{n+1-i} - 1]$ contains 2^{n-i} distinct values and therefore satisfaction of (2.) implies unicity of k since $2^{n+1-i} > 2^{n-i}$. Second, the ith bit of $\alpha - 1$ is equal to 1 iff $\alpha^i_d \in [2^{n-i}, 2^{n+1-i} - 1]$. Now, observe that $(2^{i-1} - \alpha^i_u)2^{n+1-i} + (\alpha - 1) = 2^n + \alpha^i_d$. So, if (1.), then $\alpha^i_d \in [2^{n-i}, 2^{n+1-i} - 1]$ and consequently $2^n + \alpha^i_d \in [2^n + 2^{n-i}, 2^n + 2^{n+1-i} - 1]$. So, there is some $k \geq 0$ such that $k \times 2^{n+1-i} + (\alpha - 1) \in [2^n + 2^{n-i}, 2^n + 2^{n+1-i} - 1]$ (take $k = 2^{i-1} - \alpha^i_u$). Now, suppose that (2.) holds true. There is $k \geq 0$ such that $k \times 2^{n+1-i} + (\alpha - 1) \in [2^n + 2^{n-i}, 2^n + 2^{n+1-i} - 1]$. So, $k \times 2^{n+1-i} + (\alpha - 1) - 2^n \in$

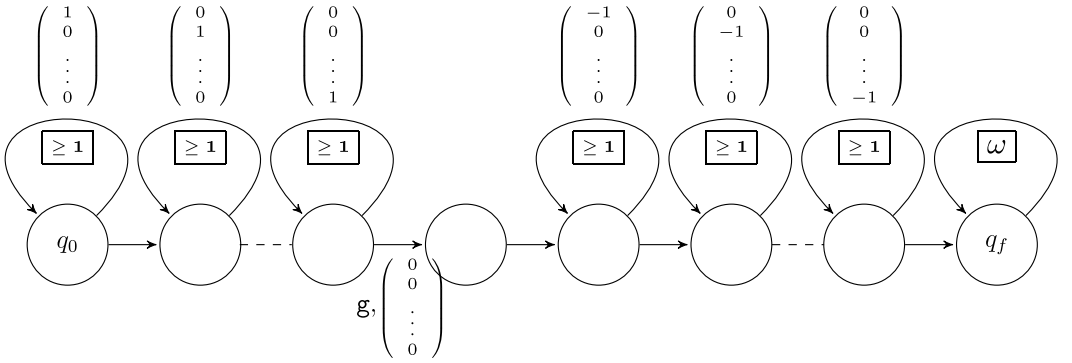


Fig. 5. A simple path schema.

$[2^{n-i}, 2^{n+1-i}-1]$ and therefore $k \times 2^{n+1-i} + \alpha_d^i - (2^{i-1} - \alpha_u^i) \times 2^{n+1-i} \in [2^{n-i}, 2^{n+1-i}-1]$. Since the expression denotes a non-negative value, we have $k \geq (2^{i-1} - \alpha_u^i)$ (indeed $\alpha_d^i < 2^{n+1-i}$) and since it denotes a value less or equal to $2^{n+1-i}-1$, we have $k \leq (2^{i-1} - \alpha_u^i)$. Consequently, $k = 2^{i-1} - \alpha_u^i$ and therefore $\alpha_d^i \in [2^{n-i}, 2^{n+1-i}-1]$, which is precisely equivalent to the fact that the ith bit of $\alpha - 1$ is equal to 1.

The formula ψ is defined from Φ by replacing each occurrence of $\mathtt{p}_i$ by $\mathtt{F}(q_1 \wedge ((\mathtt{x}_i - 1) \geq 2^n + 2^{n-i}) \wedge ((\mathtt{x}_i - 1) \leq 2^n + 2^{n-i+1} - 1))$. Intuitively, P contains one counter by propositional variable and all the counters hold the same value after the first loop. Next, in the second loop, we check that the ith bit of $\alpha - 1$ is one by incrementing $\mathtt{x}_i$ by 2^{n+1-i}. We had to consider n counters since the increments differ. In order to check whether the ith bit of counter $\mathtt{x}_i$ is one, we add repeatedly 2^{n+1-i} to the counter. Note that this ensures that the bits at positions i to n remains the same for the counter whereas the counter is incremented till its value is greater or equal to 2^n. Eventually, we may deduce that the counter value will belong to $[2^n + 2^{n-i}, 2^n + 2^{n-i+1} - 1]$.

Let us check the correctness of the reduction. Let $v : \{\mathtt{p}_1, \ldots, \mathtt{p}_n\} \to \{\top, \bot\}$ be a valuation satisfying Φ. We consider the run ρ respecting P such that the first loop is taken $\alpha = (v(\mathtt{p}_1)v(\mathtt{p}_2)\cdots v(\mathtt{p}_n))_2 + 1$ times and the initial counter values are all equal to zero. $\top$ is read as 1, $\bot$ as 0 and $(v(\mathtt{p}_1)v(\mathtt{p}_2)\cdots v(\mathtt{p}_n))_2$ denotes the value of the natural number made of n bits in binary encoding. Hence, for every $i \in [1, n]$, the counter $\mathtt{x}_i$ contains the value α after the first loop. As noted earlier, $v(\mathtt{p}_i) = \top$ implies that adding 2^{n-i+1} repeatedly to $\mathtt{x}_i$ in the last loop, we will hit $[2^n + 2^{n-i}, 2^n + 2^{n-i+1} - 1]$. Hence, the formula $\mathtt{F}(q_1 \wedge ((\mathtt{x}_i - 1) \geq 2^n + 2^{n-i}) \wedge ((\mathtt{x}_i - 1) \leq 2^n + 2^{n-i+1} - 1))$ will be satisfied by ρ iff $v(\mathtt{p}_i) = \top$. It is easy to check thus, that $\rho, 0 \models \psi$. Conversely, if there is a run ρ respecting P such that $\rho, 0 \models \psi$ and the initial counter values are all equal to zero, the valuation v satisfies Φ where for all $i \in [1, n]$, we have $v(\mathtt{p}_i)$ iff the ith bit in the binary encoding of $\alpha - 1$ is 1, where α is the number of times the first loop is taken. □

Now, for the sake of completeness, we provide a simple proof that the reachability problem in flat counter systems is NP-hard too. As explained earlier, a path schema in CPS can be seen as a flat counter system with the proviso that each internal loop is visited at least once and the control structure has a limited amount of nondeterminism. For any state q, we write $conf_0(q)$ to denote the configuration $\langle q, \langle 0, \cdots, 0\rangle\rangle$ (all counter values are equal to zero). The reachability problem REACH($\mathcal{C}$) for a class of counter system $\mathcal{C}$ is defined as:

Input: A counter system $S \in \mathcal{C}$ and two states q_0 and q_f;
Output: Is there a run from $conf_0(q_0)$ to $conf_0(q_f)$?

We have then the following result concerning the lower bound of reachability in flat counter systems and path schemas from flat counter systems.

Lemma 5.7. REACH(CPS) *and* REACH(FlatCS) *are NP-hard.*

Proof. The proofs are by reduction from the SAT problem. Using the fact that CPS is a special and constrained FlatCS, we will only prove NP-hardness of REACH(CPS). As a corollary, we obtain the result for REACH(FlatCS). Let Φ be a Boolean formula built over the propositional variables $PV = \{\mathtt{p}_1, \cdots, \mathtt{p}_n\}$. We build a path schema P such that Φ is satisfiable iff there is a run respecting P starting with the configuration $conf_0(q_0)$ and visiting the configuration $conf_0(q_f)$. The path schema P is the one described in Fig. 5 so that the truth of the propositional variable $\mathtt{p}_i$ is encoded by the fact that the loop incrementing $\mathtt{x}_i$ is visited at least twice. The guard g is defined as a formula that establishes the correspondence between the truth value of $\mathtt{p}_i$ and the number of times the loop incrementing $\mathtt{x}_i$ is visited. It is defined from Φ by replacing each occurrence of $\mathtt{p}_i$ by $\mathtt{x}_i \geq 2$. Since the ith and $(n+i)$th loops perform the complementary operation on the same counters, both of the loops can be taken equal number of times (so that q_f is reached with all the counters equal to zero).

Let us check the correctness of the reduction. Let $v : PV \to \{\top, \bot\}$ be a valuation satisfying Φ. We consider the run ρ respecting P such that $iter_P(\rho)[i] = k$ and $iter_P(\rho)[n+i] = k$ for some $k \geq 2$, if $v(\mathfrak{p}_i) = \top$, otherwise $iter_P(\rho)[i] = 1$ and $iter_P(\rho)[n+i] = 1$ for all $i \in [1, n]$. It is easy to check that the guard $\mathfrak{g}$ is satisfied by the run and taking ith loop and $(n+i)$th loop equal number times ensures resetting the counter values to zero. Hence, the configuration $conf_0(q_f)$ is reachable. Conversely, if there is a run ρ respecting P and starting with configuration $conf_0(q_0)$ such that the configuration $conf_0(q_f)$ is reachable, then the guard $\mathfrak{g}$ ensures that the valuation v satisfies Φ where for all $i \in [1, n]$, we have $v(\mathfrak{p}_i) = \top \stackrel{\text{def}}{\Leftrightarrow} iter_P(\rho)[i] \geq 2$. □

6. Characterizing infinite runs by Presburger formulae

In this section, we show how to build a quantifier-free Presburger formula (also called a constraint system herein) from a path schema P and a configuration c_0 such that it encodes the set of all runs respecting P from c_0. This can be done for path schemas without disjunctions in guards that satisfy an additional *infiniteness* property. A path schema $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ satisfies the *infiniteness* property whenever it satisfies the conditions below:

1. $effect(l_k) \geq \mathbf{0}$,
2. all the guards in transitions of l_k are conjunctions of atomic guards, and for each atomic guard occurring in the loop l_k of the form $\sum_i a_i \mathsf{x}_i \sim b$ we have
   - $\sum_i a_i \times effect(l_k)[i] \leq 0$ if $\sim \in \{\leq, <\}$,
   - $\sum_i a_i \times effect(l_k)[i] = 0$ if $\sim \in \{=\}$,
   - $\sum_i a_i \times effect(l_k)[i] \geq 0$ if $\sim \in \{\geq, >\}$.

It is easy to check that these conditions are necessary to visit the last loop l_k infinitely. More specifically, if a path schema does not satisfy the infiniteness property, then no infinite run can respect it (assuming that no disjunction occurs in guards). Moreover, given a path schema, one can decide in polynomial time whether it satisfies the infiniteness property.

Now, let us consider a path schema $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ $(k \geq 1)$ obtained from an n-dim flat counter system S such that it satisfies the infiniteness property and all the guards on transitions are conjunctions of atomic guards of the form $\sum_i a_i \mathsf{x}_i \sim b$ where $a_i \in \mathbb{Z}$, $b \in \mathbb{Z}$ and $\sim \in \{=, \leq, \geq, <, >\}$. Hence, disjunctions are *disallowed* in guards. The goal of this section (see Lemma 6.1 below) is to characterize the set $iter_P(c_0) \subseteq (\mathbb{N} \setminus \{0\})^{k-1}$ for some initial configuration c_0 as the set of solutions of a constraint system. For each internal loop l_i of the path schema P, we introduce a variable y_i. The number of variables in the systems/formulae is hence precisely $k-1$. A *constraint system* $\mathcal{E}$ over the set of variables $\{\mathsf{y}_1, \dots, \mathsf{y}_{k-1}\}$ is a quantifier-free Presburger formula built over $\{\mathsf{y}_1, \dots, \mathsf{y}_{k-1}\}$ as a conjunction of atomic constraints of the form $\sum_i a_i \mathsf{y}_i \sim b$ where $a_i, b \in \mathbb{Z}$ and $\sim \in \{=, \leq, \geq, <, >\}$. Conjunctions of atomic counter constraints and constraint systems are essentially the same objects but the distinction in this place allows to emphasize the different purposes: guard on counters in operational models and symbolic representation of sets of tuples.

Now, we explain how to build from the path schema P and from an initial configuration $c_0 = \langle q_0, \mathbf{v_0} \rangle$, a constraint system $\mathcal{E}$ over the set of variables $\{\mathsf{y}_1, \dots, \mathsf{y}_{k-1}\}$ that characterizes the set $iter_P(c_0) \subseteq (\mathbb{N} \setminus \{0\})^{k-1}$. The intuition behind this construction being that each variable y_i represents the number of times the internal loop l_i is taken.

For all $\alpha \in [1, k]$ and for all $i \in [1, n]$, we write $effect^<(l_\alpha)[i]$ to denote the term below:

$$\mathbf{v_0}[i] + (effect(p_1) + \dots + effect(p_\alpha))[i] + effect(l_1)[i]\mathsf{y}_1 + \dots + effect(l_{\alpha-1})[i]\mathsf{y}_{\alpha-1}$$

It corresponds to the value of the counter x_i just before entering in the loop l_α. Similarly, for all $\alpha \in [1, k]$ and for all $i \in [1, n]$, we write $effect^<(p_\alpha)[i]$ to denote:

$$\mathbf{v_0}[i] + (effect(p_1) + \dots + effect(p_{\alpha-1}))[i] + effect(l_1)[i]\mathsf{y}_1 + \dots + effect(l_{\alpha-1})[i]\mathsf{y}_{\alpha-1}$$

It corresponds to the value of the counter x_i just before entering in the segment p_α. In this way, for each segment p in P (either a loop or a non-loop segment) and for each $\beta \in [0, \mathrm{len}(p) - 1]$ the term below refers to the value of counter x_i just before entering for the first time in the $(\beta+1)$th transition of p:

$$effect^<(p)[i] + effect(p(0) \cdots p(\beta - 1))[i]$$

Similarly, the value of counter x_i just before entering for the last time in the $(\beta+1)$th transition of l_α is represented by the term below:

$$effect^<(p)[i] + effect(l_\alpha)[i](\mathsf{y}_\alpha - 1) + effect(l_\alpha(0) \cdots l_\alpha(\beta - 1))[i]$$

The set of conjuncts in $\mathcal{E}$ is defined as follows. Each conjunct corresponds to a specific requirement in runs respecting P.

$\mathcal{E}_1$: Each loop is visited at least once:

$$\mathsf{y}_1 \geq 1 \wedge \dots \wedge \mathsf{y}_{k-1} \geq 1$$

$\mathcal{E}_2$: Counter values are non-negative. We consider the following constraints:

- For each segment p and each $\beta \in [0, \mathrm{len}(p)-1]$, the value of counter x_i just before entering for the first time in the $(\beta+1)$th transition of p is non-negative:

$$\mathit{effect}^{<}(p)[i] + \mathit{effect}(p(0)\cdots p(\beta-1))[i] \geq 0$$

  The segment p can be either a loop or a non-loop segment.
- For each $\alpha \in [1, k-1]$ and for each $\beta \in [0, \mathrm{len}(l_\alpha)-1]$, the value of counter x_i just before entering for the last time in the $(\beta+1)$th transition of l_α is non-negative:

$$\mathit{effect}^{<}(l_\alpha)[i] + \mathit{effect}(l_\alpha)[i](\mathsf{y}_\alpha - 1) + \mathit{effect}(l_\alpha(0)\cdots l_\alpha(\beta-1))[i] \geq 0$$

We point out that it is sufficient for preserving non-negativity to check the guards the first time and the last time the run enters in a loop.

$\mathcal{E}_3$: Counter values should satisfy the guards the first time when a transition is visited. For each segment p in P, for each $\beta \in [0, \mathrm{len}(p)-1]$ and for each atomic guard $\sum_i a_i \mathsf{x}_i \sim b$ occurring in $\mathit{guard}(p(\beta))$, we add the atomic constraint:

$$\sum_i a_i(\mathit{effect}^{<}(p)[i] + \mathit{effect}(p(0)\cdots p(\beta-1))[i]) \sim b$$

$\mathcal{E}_4$: Counter values should satisfy the guards the last time when a transition is visited. This applies to loops only. For each $\alpha \in [1, k-1]$, for each $\beta \in [0, \mathrm{len}(l_\alpha)-1]$ and for each atomic guard $\sum_i a_i \mathsf{x}_i \sim b$ occurring in $\mathit{guard}(l_\alpha(\beta))$, we add the atomic constraint:

$$\sum_i a_i(\mathit{effect}^{<}(l_\alpha)[i] + \mathit{effect}(l_\alpha)[i](\mathsf{y}_\alpha - 1) + \mathit{effect}(l_\alpha(0)\cdots l_\alpha(\beta-1))[i]) \sim b$$

No condition is needed for the last loop since the path schema P satisfies the infiniteness property. Furthermore, the guards are conjunctions of atomic guards (linear constraints), whence they describe a convex set. So, if along a run, the guard is satisfied the first time the loop is visited and the last time the loop is visited, then the guard is satisfied for all the intermediate visits.

Now, let us bound the number of equalities or inequalities above. To do so, we write N_1 to denote the number of atomic guards in P.

- The number of conjuncts in $\mathcal{E}_1$ is less than k.
- The number of conjuncts in $\mathcal{E}_2$ is bounded by

$$\mathrm{len}(P) \times n + \mathrm{len}(P) \times n = 2n \times \mathrm{len}(P).$$

- The number of conjuncts in $\mathcal{E}_3$ [resp. $\mathcal{E}_4$] is bounded by $\mathrm{len}(P) \times N_1$.

So, the number of conjuncts in $\mathcal{E}$ is bounded by $2 \times \mathrm{len}(P) \times (1 + n + N_1) \leq 2 \times \mathrm{len}(P) \times n(1 + N_1)$. Since $n \leq \mathrm{size}(P)$ and $1 + N_1 \leq \mathrm{size}(P)$, we get that this number is bounded by $\mathrm{len}(P) \times 2 \times \mathrm{size}(P)^2$.

Let K be the maximal absolute value of constants occurring either in P or in $\mathbf{v_0}$. Now, we explain how it is possible to bound the maximal absolute value of constants in $\mathcal{E}$. To do so, we start by a few observations.

- A path segment p has at most $\mathrm{len}(P)$ transitions and therefore the maximal absolute value occurring in $\mathit{effect}(p)$ is at most $K \times \mathrm{len}(P)$.
- The maximal absolute value occurring in $\mathit{effect}^{<}(p)$ is at most $K + K \times \mathrm{len}(P) = K(1 + \mathrm{len}(P))$. The first occurrence of K comes from the counter values in the initial configuration.

Consequently, we can make the following conclusions.

- The maximal absolute values of constants in $\mathcal{E}_1$ is 1.
- The maximal absolute values of constants in the first part of $\mathcal{E}_2$ is bounded by $K(1 + \mathrm{len}(P)) + K\mathrm{len}(P) \leq (K + 1)(\mathrm{len}(P) + 1)$.
- The maximal absolute values of constants in the second part of $\mathcal{E}_2$ is bounded by $K(1 + \mathrm{len}(P)) + K\mathrm{len}(P) + K\mathrm{len}(P) \leq 2(K + 1)(\mathrm{len}(P) + 1)$. So, the maximal absolute values of constants in $\mathcal{E}_2$ is bounded by $2(K + 1)(\mathrm{len}(P) + 1)$.
- The maximal absolute values of constants in $\mathcal{E}_3$ or $\mathcal{E}_4$ is bounded by $n \times K \times 2(K+1)(\mathrm{len}(P)+1)+K$. The last occurrence of K is due to the constant b in the atomic constraint.

Consequently, the maximal absolute value of constants in $\mathcal{E}$ is bounded by $2n \times K(K + 2) \times (\mathrm{len}(P) + 1)$. When P is a minimal path schema, note that $\mathrm{len}(P) \leq 2 \times \mathrm{card}(\Delta) \leq 2 \times \mathrm{size}(S)$ and $k \leq \mathrm{card}(Q) \leq \mathrm{size}(S)$.

Lemma 6.1. *Let $S = \langle Q, C_n, \Delta, \mathbf{l}\rangle$ be a flat counter system without disjunctions in guards, $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ be one of its path schemas satisfying the infiniteness property and c_0 be a configuration. One can compute a constraint system $\mathcal{E}$ such that*

- *the set of solutions of $\mathcal{E}$ is equal to $iter_P(c_0)$,*
- *$\mathcal{E}$ has $k-1$ variables,*
- *$\mathcal{E}$ has at most $\mathrm{len}(P) \times 2 \times \mathrm{size}(P)^2$ conjuncts,*
- *the greatest absolute value from constants in $\mathcal{E}$ is bounded by $2n \times K(K+2) \times (\mathrm{len}(P)+1)$.*

Proof. The constraint system $\mathcal{E}$ is the one built above.

($\star$) Let $\rho = \langle q_0, \mathbf{v_0}\rangle\langle q_1, \mathbf{v_1}\rangle\langle q_2, \mathbf{v_2}\rangle \cdots$ be an infinite run respecting the path schema P with $c_0 = \langle q_0, \mathbf{v_0}\rangle$. We write $V : \{\mathsf{y}_1, \dots, \mathsf{y}_{k-1}\} \to \mathbb{N}$ to denote the valuation such that for every $\alpha \in [1, k-1]$, we have $V(\mathsf{y}_\alpha) = iter_P(\rho)[\alpha]$. V is extended naturally to terms built over variables in $\{\mathsf{y}_1, \dots, \mathsf{y}_{k-1}\}$, the range becoming $\mathbb{Z}$. Let us check that $V \models \mathcal{E}$.

1. Since ρ respects P, each loop l_i is visited at least once and therefore $V \models \mathcal{E}_1$.
2. We have seen that the value below

   $$V(\mathit{effect}^<(p)[i] + \mathit{effect}(p(0)\cdots p(\beta-1))[i])$$

   is equal to the value of counter x_i just before entering for the first time in the $(\beta+1)$th transition of p. Similarly, the value below

   $$V(\mathit{effect}^<(l_\alpha)[i] + \mathit{effect}(l_\alpha)[i](\mathsf{y}_\alpha - 1) + \mathit{effect}(l_\alpha[0]\cdots l_\alpha[\beta-1])[i])$$

   is equal to the value of counter x_i before entering for the last time in the $(\beta+1)$th transition of l_α. Since ρ is a run, these values are non-negative, whence $V \models \mathcal{E}_2$.
3. Since ρ is a run, whenever a transition is fired, all its guards are satisfied. Hence, for each segment p in P, for each $\beta \in [0, \mathrm{len}(p)-1]$ and for each atomic guard $\sum_i a_i \mathsf{x}_i \sim b$ in $\mathit{guard}(p(\beta))$, we have

   $$\sum_i a_i V(\mathit{effect}^<(p)[i] + \mathit{effect}(p(0)\cdots p(\beta-1))[i]) \sim b$$

   Similarly, for each $\alpha \in [1, k-1]$, for each $\beta \in [0, \mathrm{len}(l_\alpha)-1]$ and for each atomic guard $\sum_i a_i \mathsf{x}_i \sim b \in \mathit{guard}(l_\alpha(\beta))$, we have

   $$\sum_i a_i V(\mathit{effect}^<(l_\alpha)[i] + \mathit{effect}(l_\alpha)[i](\mathsf{y}_\alpha - 1) + \mathit{effect}(l_\alpha(0)\cdots l_\alpha(\beta-1))[i]) \sim b$$

   Consequently, $V \models \mathcal{E}_3 \wedge \mathcal{E}_4$.

($\star\star$) It remains to show the property in the other direction.

Let $V : \{\mathsf{y}_1, \dots, \mathsf{y}_{k-1}\} \to \mathbb{N}$ be a solution of $\mathcal{E}$. Let

$$w = p_1 l_1^{V(\mathsf{y}_1)} \cdots p_{k-1} l_1^{V(\mathsf{y}_{k-1})} p_k l_k^\omega \in \Delta^\omega$$

and let us build an ω-sequence $\rho' = \langle q_0, \mathbf{v'_0}\rangle\langle q_1, \mathbf{v'_1}\rangle\langle q_2, \mathbf{v'_2}\rangle \cdots \in (Q \times \mathbb{Z}^n)^\omega$, that will be later shown to be an infinite run respecting the path schema P with $c_0 = \langle q_0, \mathbf{v_0}\rangle$. Here is how ρ' is defined (note that the definition does not assume that ρ' needs to be a run):

- For every $i \geq 0$, $q_i \stackrel{\text{def}}{=} \mathit{source}(w(i))$,
- $\mathbf{v'_0} \stackrel{\text{def}}{=} \mathbf{v_0}$ and for every $i \geq 1$, we have $\mathbf{v'_i} \stackrel{\text{def}}{=} \mathbf{v'_{i-1}} + \mathit{update}(w(i))$.

In order to show that ρ' is an infinite run respecting P, we have to check three main properties.

1. Since $V \models \mathcal{E}_2$, for each segment p in P and for each $\beta \in [0, \mathrm{len}(p)-1]$, counter values just before entering for the first time in the $(\beta+1)$th transition of p are non-negative. Moreover, for each $\alpha \in [1, k-1]$ and for each $\beta \in [0, \mathrm{len}(l_\alpha)-1]$, counter values just before entering for the last time in the $(\beta+1)$th transition of l_α are non-negative too. We have also to guarantee that for $j \in [2, V(\mathsf{y}_\alpha)-1]$, counter values just before entering for the jth time in the $(\beta+1)$th transition of l_α are non-negative. This is a consequence of the fact that if $\mathbf{v} \geq 0$ and $\mathbf{v} + V(\mathsf{y}_\alpha)\mathit{effect}(l_\alpha) \geq 0$, then for $j \in [2, V(\mathsf{y}_\alpha)-1]$, we have $\mathbf{v} + j \times \mathit{effect}(l_\alpha) \geq 0$ (convexity). Consequently we have $\mathbf{v'_i} \geq \mathbf{0}$ for all $i \geq 0$,.
2. Similarly, counter values should satisfy the guards for each fired transition. Since $V \models \mathcal{E}_3$, for each segment p in P, for each $\beta \in [0, \mathrm{len}(p)-1]$ and for each atomic guard $\sum_i a_i \mathsf{x}_i \sim b$ in $\mathit{guard}(p(\beta))$, counter values satisfy it the first time the transition is visited. Moreover, since $V \models \mathcal{E}_3$, for each $\alpha \in [1, k-1]$, for each $\beta \in [0, \mathrm{len}(l_\alpha)-1]$ and for each atomic guard $\sum_i a_i \mathsf{x}_i \sim b$ in $\mathit{guard}(l_\alpha(\beta))$ occurs, counter values satisfy it the first time the transition is visited.

However, we have also to guarantee that for $j \in [2, V(\mathsf{y}_\alpha) - 1]$, counter values just before entering for the jth time in the $(\beta+1)$th transition of l_α, all the guards are satisfied. This is a consequence of the fact that if $\sum_i a_i \mathbf{v}[i] \sim b$ and $\sum_i a_i(\mathbf{v} + V(\mathsf{y}_\alpha)\mathit{effect}(l_\alpha))[i] \sim b$, then for $j \in [2, V(\mathsf{y}_\alpha) - 1]$, we have $\sum_i a_i(\mathbf{v} + j \times \mathit{effect}(l_\alpha))[i] \sim b$ (convexity). Hence, ρ' is a run starting at c_0.

3. It remains to show that ρ' respects P. Since ρ' is a run (see (1) and (2) above), by construction of ρ', it respects P thanks to $V \models \mathcal{E}_1$. This last condition is indeed needed since by definition, to respect a path schema each loop has to be visited at least once. □

7. From one minimal schema to several schemas

Section 6 is restricted to path schemas with no disjunction in guards. However, having disjunctions in guards is not a real problem as soon as we allow quantifiers in the constructed formula. But, on the one hand, in full generality we allow disjunction in guards by definition and, on the other hand, we would like to generate only quantifier-free formulae from Presburger arithmetic for its computational properties. This leaves us with two ways to encode the runs using quantifier-free Presburger formulae.

1. To encode the runs in a path schema (with disjunction in guards) using quantified Presburger formulae and to perform quantifier-elimination procedure to obtain equivalent quantifier-free formulae,
2. To transform the path schema (with disjunction in guards) into path schemas with no disjunction in guards and then to encode the runs respecting such path schemas by using quantifier-free Presburger formulae from Section 6.

The quantifier-elimination procedure is known to be computationally expensive, see e.g. [10], whereas the second method, as shown in the sequel, allows to obtain a polynomial-size formula thanks to non-deterministic guesses. In order to get an optimal complexity bound for the model-checking procedure, we will in fact follow the second method. Given a flat counter system $S = \langle Q, \mathsf{C}_n, \Delta, \mathbf{l}\rangle$, a configuration $c_0 = \langle q_0, \mathbf{v_0}\rangle$ and a minimal path schema P starting from the configuration c_0, we build a finite set Y of path schemas such that:

1. each path schema in Y has transitions without disjunctions in guards,
2. the existence of a run respecting P is equivalent to the existence of a path schema in Y having the run respecting it,
3. each path schema in Y is obtained from P by unfolding loops so that the terms in each loop satisfy the same atomic guards.

Moreover, we shall see how the cardinality of Y is at most exponential in the size of P. Each path schema in Y comes with an implicit counter system (typically containing exactly the states and transitions occurring in the path schema). We explain below how we can get rid of disjunctions. By contrast, disjunctions can be easily eliminated at the cost of adding new transitions between states and using disjunctive normal form (DNF). This type of transformation easily breaks flatness and may cause an exponential blow-up because of the DNF. That is why we shall follow a different approach.

7.1. Term maps

Before defining the set of path schemas with no disjunctions, let us introduce a few definitions. Let B be a finite non-empty set of integers containing all the constants b occurring in guards of S of the form $\mathtt{t} \sim b$ and T be a finite set of terms containing all the terms $\mathtt{t}$ occurring in guards of S of the form $\mathtt{t} \sim b$. Assuming that $B = \{b_1, \ldots, b_m\}$ with $b_1 < \cdots < b_m$, we write I to denote the finite set of (non-empty) intervals

$$I \stackrel{\text{def}}{=} \{(-\infty, b_1 - 1], [b_1, b_1], [b_1 + 1, b_2 - 1], [b_2, b_2], \cdots, [b_m, b_m], [b_m + 1, \infty)\} \setminus \{\emptyset\}$$

Note that we may have $[b_j + 1, b_{j+1} - 1] = \emptyset$ if $b_{j+1} = b_j + 1$. The set I contains at most $2m+1$ intervals and at least $m+2$ intervals. We consider the natural linear ordering $\leq$ on intervals in I that respects the standard relation $\leq$ on integers. In other words,

$$(-\infty, b_1 - 1] \leq [b_1, b_1] \leq [b_1 + 1, b_2 - 1] \leq [b_2, b_2] \leq \cdots \leq [b_m, b_m] \leq [b_m + 1, \infty)$$

A *term map* $\mathbf{m}$ is a map $\mathbf{m} : T \to I$ that abstracts term values by associating to them an interval.

Definition 7.1. Given a loop effect $\mathbf{u} \in \mathbb{Z}^n$, we define the relation $\preceq_{\mathbf{u}}$ such that $\mathbf{m} \preceq_{\mathbf{u}} \mathbf{m}' \stackrel{\text{def}}{\Leftrightarrow}$ for every term $\mathtt{t} = \sum_i a_i \mathbf{x}_i \in T$, we have

- $\mathbf{m}(\mathtt{t}) \leq \mathbf{m}'(\mathtt{t})$ if $\sum_i a_i \mathbf{u}[i] \geq 0$,
- $\mathbf{m}(\mathtt{t}) \geq \mathbf{m}'(\mathtt{t})$ if $\sum_i a_i \mathbf{u}[i] \leq 0$,
- $\mathbf{m}(\mathtt{t}) = \mathbf{m}'(\mathtt{t})$ if $\sum_i a_i \mathbf{u}[i] = 0$.

We write $\mathbf{m} \prec_{\mathbf{u}} \mathbf{m}'$ whenever $\mathbf{m} \preceq_{\mathbf{u}} \mathbf{m}'$ and $\mathbf{m} \neq \mathbf{m}'$.

Sequences of strictly increasing term maps have bounded length.

Lemma 7.2. *Let* $\mathbf{u} \in \mathbb{Z}^n$ *and* $\mathbf{m}_1 \prec_{\mathbf{u}} \mathbf{m}_2 \prec_{\mathbf{u}} \cdots \prec_{\mathbf{u}} \mathbf{m}_L$. *Then,* $L \leq \mathrm{card}(I) \times \mathrm{card}(T) \leq 2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)$.

Proof. Given a term map $\mathbf{m}$ and a term $\mathtt{t}$, $\mathbf{m}(\mathtt{t})$ can obviously take one of the $\mathrm{card}(I)$ values from I. Due to monotonicity, for each term $\mathtt{t}$,

(increasing) either $\mathbf{m}_1(\mathtt{t}) \leq \cdots \leq \mathbf{m}_L(\mathtt{t})$
(decreasing) or $\mathbf{m}_L(\mathtt{t}) \leq \cdots \leq \mathbf{m}_1(\mathtt{t})$.

Also, there are $\mathrm{card}(T)$ distinct terms in T (obvious). Hence, the number of different maps that are either decreasing or increasing can be $\mathrm{card}(T) \times \mathrm{card}(I)$. Again, we know that $\mathrm{card}(I) \leq 2 \times \mathrm{card}(B) + 1$. Hence, L, the number of different term maps in a sequence which is either increasing or decreasing, can be at most $\mathrm{card}(I) \times \mathrm{card}(T) \leq 2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)$. □

Given a guard $\mathtt{g}$ using the syntactic resources from T and B, and a term map $\mathbf{m}$, we write $\mathbf{m} \vdash \mathtt{g}$ with the following inductive definition:

- $\mathbf{m} \vdash \mathtt{t} = b \stackrel{\text{def}}{\Leftrightarrow} \mathbf{m}(\mathtt{t}) = [b, b]$,
- $\mathbf{m} \vdash \mathtt{t} \leq b \stackrel{\text{def}}{\Leftrightarrow} \mathbf{m}(\mathtt{t}) \subseteq (-\infty, b]$,
- $\mathbf{m} \vdash \mathtt{t} \geq b \stackrel{\text{def}}{\Leftrightarrow} \mathbf{m}(\mathtt{t}) \subseteq [b, +\infty)$,
- $\mathbf{m} \vdash \mathtt{t} < b \stackrel{\text{def}}{\Leftrightarrow} \mathbf{m}(\mathtt{t}) \subseteq (-\infty, b)$,
- $\mathbf{m} \vdash \mathtt{t} > b \stackrel{\text{def}}{\Leftrightarrow} \mathbf{m}(\mathtt{t}) \subseteq (b, +\infty)$,
- $\mathbf{m} \vdash \mathtt{g}_1 \wedge \mathtt{g}_2 \stackrel{\text{def}}{\Leftrightarrow} \mathbf{m} \vdash \mathtt{g}_1$ and $\mathbf{m} \vdash \mathtt{g}_2$,
- $\mathbf{m} \vdash \mathtt{g}_1 \vee \mathtt{g}_2 \stackrel{\text{def}}{\Leftrightarrow} \mathbf{m} \vdash \mathtt{g}_1$ or $\mathbf{m} \vdash \mathtt{g}_2$.

The relation $\vdash$ is simply the symbolic satisfaction relation between term values and guards. Since term maps and guards are built over the same sets of terms and constants, completeness is obtained as stated in Lemma 7.3(II) below. Furthermore, Lemma 7.3(I) states that the relation $\vdash$ is easy to check.

Lemma 7.3.

(I) $\mathbf{m} \vdash \mathtt{g}$ *can be checked in* PTIME *in* $\mathrm{size}(\mathbf{m}) + \mathrm{size}(\mathtt{g})$.
(II) $\mathbf{m} \vdash \mathtt{g}$ *iff for all* $v : \{\mathtt{x}_1, \mathtt{x}_2, \cdots, \mathtt{x}_n\} \to \mathbb{N}$ *and for all* $\mathtt{t} \in T$, $v(\mathtt{t}) \in \mathbf{m}(\mathtt{t})$ *implies* $v \models \mathtt{g}$.

It is worth noting that $\mathrm{size}(\mathbf{m})$ is in $\mathcal{O}(\mathrm{size}(I) \times \mathrm{size}(T))$. The values $\mathrm{size}(I)$ and $\mathrm{size}(T)$ have not been formally defined but we assume a reasonably succinct encoding using a binary representation for integers.

Proof.

(I) For the polynomial-time algorithm we follow the following steps. First, for each constraint $\mathtt{t} \sim b$ appearing in $\mathtt{g}$, we replace it either by $\top$ (true) or by $\bot$ (false) depending whether $\mathbf{m} \vdash \mathtt{t} \sim b$ or not. After replacing all constraints, we are left with a positive Boolean formula whose atomic formulae are either $\top$ or $\bot$. It can be evaluated in logarithmic space in the size of the resulting formula (less than $\mathrm{size}(\mathtt{g})$), see e.g. [33].
Given a term map $\mathbf{m}$ and a constraint $\mathtt{t} \sim b$, checking $\mathbf{m} \vdash \mathtt{t} \sim b$ amounts to check the containment of interval $\mathbf{m}(\mathtt{t})$ in a specified interval depending on $\sim$. This can be achieved by comparing the end-points of the intervals, which can be done in polynomial time in $\mathrm{size}(\mathtt{t}) + \mathrm{size}(\mathbf{m})$. As the number of constraints is also bounded by $\mathrm{size}(\mathtt{g})$, the replacement of atomic constraints can be performed in polynomial time in $\mathrm{size}(\mathbf{m}) + \mathrm{size}(\mathtt{g})$. Thus, the procedure runs in polynomial time in $\mathrm{size}(\mathbf{m}) + \mathrm{size}(\mathtt{g})$.

(II) Consider that $\mathbf{m} \vdash \mathtt{g}$ and some $v : \{\mathtt{x}_1, \mathtt{x}_2, \cdots, \mathtt{x}_n\} \to \mathbb{N}$ such that $v(\mathtt{t})$ lies in the interval $\mathbf{m}(\mathtt{t})$ for each term $\mathtt{t} \in T$. Now we prove inductively on the structure of $\mathtt{g}$ that $v \models \mathtt{g}$.
- As base case we have arithmetical constraints of the guard. Consider a constraint of the form $\mathtt{t} \leq b$. Since $\mathbf{m} \vdash \mathtt{g}$, we have that $\mathbf{m}(\mathtt{t}) \subseteq (-\infty, b]$. Since, $v(\mathtt{t})$ lies in the interval $\mathbf{m}(\mathtt{t})$, $v(\mathtt{t}) \in (-\infty, b]$. In this case $v \models \mathtt{t} \leq b$. Similarly, for other constraints $\mathtt{t} \sim b$, observe that if $v(\mathtt{t}) \in \mathbf{m}(\mathtt{t})$ then $v(\mathtt{t})$ lies in the interval specified in the definition of $\vdash$ and thus, $v \models \mathtt{t} \sim b$.
- The induction step for $\wedge$ and $\vee$, follows easily.

On the other hand, assume that for all valuations v and for all $\mathtt{t} \in T$, we have $v(\mathtt{t}) \in \mathbf{m}(\mathtt{t})$ implies $v \models \mathtt{g}$. Similar to above, we will use inductive argument to show that $\mathbf{m} \vdash \mathtt{g}$

- **Base Case:** Let us specifically consider the atomic guard $\mathtt{t} \geq b$ and v be a valuation such that $v(\mathtt{t}) \in \mathbf{m}(\mathtt{t})$. We have $v \models \mathtt{t} \geq b$, and consequently $v(\mathtt{t}) \in [b, +\infty)$. Since, $v(\mathtt{t}) \in \mathbf{m}(\mathtt{t})$, we have that, $\mathbf{m}(\mathtt{t}) \subseteq [b, +\infty)$. Hence, $\mathbf{m} \vdash \mathtt{t} \geq b$. Similarly, for constraints of other forms $\mathtt{t} \sim b$, $v(\mathtt{t})$ lies in the interval exactly specified in the definition of $\vdash$. Thus, $\mathbf{m} \vdash \mathtt{t} \sim b$.
- **Inductive step:** Again, the induction step for $\wedge$ and $\vee$ follows easily. $\square$

Finally, for a loop segment l, the tuple of term maps $\langle \mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_{\mathrm{len}(l)-1} \rangle$ is said to be *final* iff for every term $\mathtt{t} = \sum_j a_j \mathtt{x}_j \in T$ and for all $i \in [0, \mathrm{len}(l) - 1]$,

- $\sum_j a_j \mathit{effect}(l)[j] > 0$ implies $\mathbf{m}_i(\mathtt{t})$ is maximal in I.
- $\sum_j a_j \mathit{effect}(l)[j] < 0$ implies $\mathbf{m}_i(\mathtt{t})$ is minimal in I.

where $\mathit{effect}(l)$ is defined in Section 3.

7.2. Resources and footprints

A *resource* R is a triple $\langle X, T, B \rangle$ such that X is a finite set of propositional variables, T is a finite set of terms and B is a finite set of integers. Without any loss of generality, we can assume that all these sets are non-empty (to avoid treatments of –easy– degenerated cases). A formula $\phi \in \mathrm{PLTL}[\mathtt{C}]$ is *built over* R whenever the atomic formulae are of the form either $\mathtt{p} \in X$ or $\mathtt{t} \sim b$ with $\mathtt{t} \in T$ and $b \in B$. A *footprint* is an abstraction of a model for $\mathrm{PLTL}[\mathtt{C}]$ restricted to elements from the resource R. More precisely, a footprint ft is of the form $\mathsf{ft} : \mathbb{N} \to 2^X \times I^T$ where I is the set of intervals built from B, whence the first element of $\mathsf{ft}(i)$ is a propositional valuation and the second one is a term map. The satisfiability relation $\models$ involving models or runs can be adapted to footprints as follows where formulae and footprints are obtained from the same resource R and $\models_{\mathsf{symb}}$ is the new *symbolic* satisfaction relation:

- $\mathsf{ft}, i \models_{\mathsf{symb}} \mathtt{p} \stackrel{\text{def}}{\Leftrightarrow} \mathtt{p} \in \pi_1(\mathsf{ft}(i))$,
- $\mathsf{ft}, i \models_{\mathsf{symb}} \mathtt{t} \geq b \stackrel{\text{def}}{\Leftrightarrow} \pi_2(\mathsf{ft}(i))(\mathtt{t}) \subseteq [b, +\infty)$,
- $\mathsf{ft}, i \models_{\mathsf{symb}} \mathtt{t} \leq b \stackrel{\text{def}}{\Leftrightarrow} \pi_2(\mathsf{ft}(i))(\mathtt{t}) \subseteq (-\infty, b]$,
- $\mathsf{ft}, i \models_{\mathsf{symb}} \mathtt{X}\phi \stackrel{\text{def}}{\Leftrightarrow} \mathsf{ft}, i+1 \models_{\mathsf{symb}} \phi$,
- $\mathsf{ft}, i \models_{\mathsf{symb}} \phi \mathtt{U} \psi \stackrel{\text{def}}{\Leftrightarrow}$ there is $j \geq i$ such that $\mathsf{ft}, j \models_{\mathsf{symb}} \psi$ and for every $j' \in [i, j-1]$, we have $\mathsf{ft}, j' \models_{\mathsf{symb}} \phi$.

We omit the clauses for Boolean connectives, past-time operators and other arithmetical constraints since their definitions are as expected. Actually, $\models_{\mathsf{symb}}$ is exactly the satisfaction relation for plain Past LTL when arithmetical constraints are understood as abstract propositions.

Definition 7.4. Let $\mathsf{R} = \langle X, T, B \rangle$ be a resource and $\rho = \langle q_0, \mathbf{v_0} \rangle, \langle q_1, \mathbf{v_1} \rangle \cdots$ be an infinite run of S. The *footprint* of ρ with respect to R is the footprint ft_ρ such that for every $i \geq 0$, we have $\mathsf{ft}_\rho(i) \stackrel{\text{def}}{=} \langle \mathbf{l}(q_i) \cap X, \mathbf{m}_i \rangle$ where for every term $\mathtt{t} = \sum_j a_j \mathtt{x}_j \in T$, we have $\sum_j a_j \mathbf{v_i}[j] \in \mathbf{m}_i(\mathtt{t})$.

Note that the value $\sum_j a_j \mathbf{v_i}[j]$ belongs to a unique element of I since I is a partition of $\mathbb{Z}$, hence Definition 7.4 makes sense. Lemma 7.6 below roughly states that satisfaction of a formula on a run can be checked symbolically from the footprint (this turns out to be useful for the correctness of forthcoming Algorithm 1).

Lemma 7.5. *Let $\mathsf{R} = \langle X, T, B \rangle$ be a resource, $\rho = \langle q_0, \mathbf{v_0} \rangle, \langle q_1, \mathbf{v_1} \rangle \cdots$ be an infinite run, $i \geq 0$ be a position and ϕ be a formula in* $\mathrm{PLTL}[\mathtt{C}]$ *built over* R. *Then $\rho, i \models \phi$ iff $\mathsf{ft}_\rho, i \models_{\mathsf{symb}} \phi$.*

Proof. The proof is by structural induction.

- **Base Case 1** ($\mathtt{p} \in X$). The propositions below are equivalent:
  - $\rho, i \models \mathtt{p}$,
  - $\mathtt{p} \in \mathbf{l}(q_i)$ (by definition of $\models$),
  - $\mathtt{p} \in \pi_1(\mathsf{ft}_\rho(i))$ (by definition of ft_ρ),
  - $\mathsf{ft}_\rho, i \models_{\mathsf{symb}} \mathtt{p}$ (by definition of $\models_{\mathsf{symb}}$).
- **Base Case 2** ($\sum_j a_j \mathtt{x}_j \leq b$ with $\sum_j a_j \mathtt{x}_j \in T$ and $b \in B$). The propositions below are equivalent:
  - $\rho, i \models \sum_j a_j \mathtt{x}_j \leq b$,
  - $\sum_j a_j \mathbf{v_i}[j] \leq b$ (by definition of $\models$),

- $\pi_2(\mathsf{ft}_\rho(i))(\sum_j a_j \mathtt{x}_j) \subseteq (-\infty, b]$ (because, by definition of ft_ρ, we have $\sum_j a_j \mathbf{v_i}[j] \in \pi_2(\mathsf{ft}_\rho(i))(\sum_j a_j \mathtt{x}_j)$),
- $\mathsf{ft}_\rho, i \models_{\mathsf{symb}} \sum_j a_j \mathtt{x}_j \leq b$ (by definition of $\models_{\mathsf{symb}}$).

The base cases for the other arithmetical constraints can be shown similarly.

- For the induction step, by way of example we deal with the case $\phi = \mathtt{X}\psi$ (the cases for the Boolean operators or for the other temporal operators are analogous). We have the following equivalences:
  - $\rho, i \models \mathtt{X}\psi$,
  - $\rho, i+1 \models \psi$ (by definition of $\models$),
  - $\mathsf{ft}_\rho, i+1 \models_{\mathsf{symb}} \psi$ (by induction hypothesis),
  - $\mathsf{ft}_\rho, i \models_{\mathsf{symb}} \mathtt{X}\psi$ (by definition $\models_{\mathsf{symb}}$). □

As a corollary, we obtain the following lemma.

Lemma 7.6. *Let $\mathsf{R} = \langle X, T, B\rangle$ be a resource and ρ and ρ' be two infinite runs with identical footprints with respect to R. For all formulae ϕ built over R and all positions $i \geq 0$, we have $\rho, i \models \phi$ iff $\rho', i \models \phi$.*

Given a minimal path schema $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ and a run ρ respecting P, the footprint ft_ρ (with respect to a resource $\mathsf{R} = \langle X, T, B\rangle$) is an ultimately periodic word that can be written of the form $w \cdot u^\omega$ where $\mathrm{len}(u) = \mathrm{len}(l_k)$.

7.3. Unfolding

Let $\mathsf{R} = \langle X, T, B\rangle$ be a resource, $c_0 = \langle q_0, \mathbf{v_0}\rangle$ be an initial configuration and $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ be a minimal path schema such that $\mathit{first}(p_1) = q_0$ (disjunctions in guards are allowed). In the sequel, we assume that T contains the terms present in P and, B contains the constants present in P. In order to define the set of path schemas Y_{R,P,c_0} parameterized by R, P and c_0 and with no disjunction in the guards, we need to define intermediate objects. We list some of them below:

- a set of guards $\mathtt{G}^\star(T, B, U)$ (parameterized by the finite set U of updates from P),
- a set of (structured) control states $Q' = Q \times I^T$,
- a set of transitions Δ' defined from Q', $\mathtt{G}^\star(T, B, U)$ and U.

Let Δ_P be the set of transitions occurring in P. Given a term $\mathtt{t} = \sum_j a_j \mathtt{x}_j \in T$, an update $\mathbf{u} \in \mathbb{Z}^n$ and a term map $\mathbf{m}$, we write $\psi(\mathtt{t}, \mathbf{u}, \mathbf{m}(\mathtt{t}))$ to denote the formula below (where $b, b' \in B$):

- $\psi(\mathtt{t}, \mathbf{u}, (-\infty, b]) \stackrel{\text{def}}{=} \sum_j a_j(\mathtt{x}_j + \mathbf{u}[j]) \leq b$,
- $\psi(\mathtt{t}, \mathbf{u}, [b, +\infty)) \stackrel{\text{def}}{=} \sum_j a_j(\mathtt{x}_j + \mathbf{u}[j]) \geq b$,
- $\psi(\mathtt{t}, \mathbf{u}, [b, b']) \stackrel{\text{def}}{=} ((\sum_j a_j(\mathtt{x}_j + \mathbf{u}[j]) \leq b') \wedge ((\sum_j a_j(\mathtt{x}_j + \mathbf{u}[j]) \geq b)$.

The formulae of the form $\psi(\mathtt{t}, \mathbf{u}, \mathit{int})$ ($\mathit{int} \in I$) have been designed to satisfy the property below.

Lemma 7.7. *Let $v, v' : \{\mathtt{x}_1, \dots, \mathtt{x}_n\} \to \mathbb{N}$ be such that for every $i \in [1, n]$, $v'(\mathtt{x}_i) = v(\mathtt{x}_i) + \mathbf{u}[i]$. For every interval $\mathit{int} \in I$, for every term $\mathtt{t} \in T$, $v \models \psi(\mathtt{t}, \mathbf{u}, \mathit{int})$ iff $v'(\mathtt{t}) \in \mathit{int}$.*

The proof is by an easy verification.

We write $\mathtt{G}^\star(T, B, U)$ to denote the set of guards of the form

$$\bigwedge_{\mathtt{t} \in T} \psi(\mathtt{t}, \mathbf{u}, \mathbf{m}(\mathtt{t}))$$

where $\mathbf{u} \in U$ and $\mathbf{m} : T \to I$. Even though the cardinal of $\mathtt{G}^\star(T, B, U)$ is exponential, the new path schemas defined below remain of polynomial size, partly because each guard in $\mathtt{G}^\star(T, B, U)$ is of polynomial size in the size of P.

We define Δ' as a finite subset of $Q' \times \Delta_P \times \mathtt{G}^\star(T, B, U) \times U \times Q'$ such that for every $\langle q, \mathbf{m}\rangle \xrightarrow{\delta, \langle \mathtt{g}_{\mathbf{m}'}, \mathbf{u}\rangle} \langle q', \mathbf{m}'\rangle \in \Delta'$, the conditions below are satisfied:

- $q = \mathit{source}(\delta)$ and $q' = \mathit{target}(\delta)$,
- $\mathtt{g}_{\mathbf{m}'}$ is a guard that states that after the update $\mathbf{u}$, for each $\mathtt{t} \in T$, its value belongs to $\mathbf{m}'(\mathtt{t})$. Precisely, $\mathtt{g}_{\mathbf{m}'}$ is equal to $\bigwedge_{\mathtt{t} \in T} \psi(\mathtt{t}, \mathbf{u}, \mathbf{m}'(\mathtt{t}))$
- Term values belong to intervals that make true $\mathit{guard}(\delta)$, i.e. $\mathbf{m} \vdash \mathit{guard}(\delta)$.
- $\mathbf{u} = \mathit{update}(\delta)$.

We extend the definition of $source(\delta)/target(\delta)$ to $\delta' = \langle q, \mathbf{m}\rangle \xrightarrow{\delta, \langle \mathfrak{g}_{\mathbf{m}'}, \mathbf{u}\rangle} \langle q', \mathbf{m}'\rangle \in \Delta'$ with $source(\delta') \stackrel{\text{def}}{=} \langle q, \mathbf{m}\rangle$ and $target(\delta') \stackrel{\text{def}}{=} \langle q', \mathbf{m}'\rangle$. Similarly, for a finite word $w \in (\Delta')^+$, we define $source(w) \stackrel{\text{def}}{=} source(w(0))$ and $target(w) \stackrel{\text{def}}{=} target(w(\text{len}(w)-1))$.

Below, we define skeletons as slight variants of path schemas in Y_{R,P,c_0}. The slight differences are explained a bit later. A *skeleton* $\mathtt{sk}$ (compatible with R, P and $c_0 = \langle q_0, \mathbf{v_0}\rangle$) is a finite word over Δ', written

$$\langle q_1, \mathbf{m}_1\rangle \xrightarrow{\delta_1, \langle \mathfrak{g}^1_{\mathbf{m}'}, \mathbf{u}_1\rangle} \langle q_2, \mathbf{m}_2\rangle \xrightarrow{\delta_2, \langle \mathfrak{g}^2_{\mathbf{m}'}, \mathbf{u}_2\rangle} \langle q_3, \mathbf{m}_3\rangle \cdots \xrightarrow{\delta_K, \langle \mathfrak{g}^K_{\mathbf{m}'}, \mathbf{u}_K\rangle} \langle q_{K+1}, \mathbf{m}_{K+1}\rangle,$$

verifying the following conditions:

(init) For every term $\mathtt{t} = \sum_j a_j \mathtt{x}_j \in T$, we have $\sum_j a_j \mathbf{v_0}[j] \in \mathbf{m}_1(\mathtt{t})$ where $\mathbf{v_0}$ is the initial vector. If $\mathbf{v_0}$ is not fixed, typically to solve the global model-checking problem, there is no need to require any condition on $\mathbf{m}_1$.

(schema) Let $f : (\Delta')^* \to \Delta^*$ be the map such that $f(\varepsilon) = \varepsilon$, $f(w \cdot w') = f(w) \cdot f(w')$ and $f(\langle q, \mathbf{m}\rangle \xrightarrow{\delta, \langle \mathfrak{g}_{\mathbf{m}'}, \mathbf{u}\rangle} \langle q', \mathbf{m}'\rangle) = \delta$. We require that $f(\mathtt{sk}) \in p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^+$. So, a skeleton can be seen as a finite sequence of transitions that is the prefix of a word in the language generated from the path schema P. Moreover, it is decorated by additional pieces of information.

(minimality) For every factor $w = \langle q_H, \mathbf{m}_H\rangle \xrightarrow{\delta_H, \langle \mathfrak{g}^H_{\mathbf{m}'}, \mathbf{u}_H\rangle} \langle q_{H+1}, \mathbf{m}_{H+1}\rangle \cdots \xrightarrow{\delta_{J-1}, \langle \mathfrak{g}^{J-1}_{\mathbf{m}'}, \mathbf{u}_{J-1}\rangle} \langle q_J, \mathbf{m}_J\rangle$ of $\mathtt{sk}$ such that $f(w) = (l)^3$ for some loop l of P (therefore $J = H + 3 \times \text{len}(l)$), there is $\alpha \in [1, \text{len}(l)]$ such that $\mathbf{m}_{H+\alpha} \prec_{effect(l)} \mathbf{m}_{H+\alpha+2\times\text{len}(l)}$. We disallow three consecutive sequences of transitions related to the loop l without any progress on the term maps. Indeed, if this occurs, then this would be more adequate to capture some of these sequences by a loop in the new path schema.

(last-loop) For the unique suffix $w = \langle q_H, \mathbf{m}_H\rangle \to \cdots \to \langle q_{H+\text{len}(l_k)}, \mathbf{m}_{H+\text{len}(l_k)}\rangle$ of $\mathtt{sk}$ of length $\text{len}(l_k)$ (so $f(w) = l_k$), we have $source(w) = target(w)$ (i.e. $\langle q_H, \mathbf{m}_H\rangle = \langle q_{H+\text{len}(l_k)}, \mathbf{m}_{H+\text{len}(l_k)}\rangle$) and $\langle \mathbf{m}_H, \dots, \mathbf{m}_{H+(\text{len}(l_k)-1)}\rangle$ is *final* for the loop l_k.

Lemma 7.8. *For a skeleton* $\mathtt{sk}$, $\text{len}(\mathtt{sk}) \leq (\text{len}(p_1) + \cdots + \text{len}(p_k)) + 2 \times (2 \times \text{card}(T) \times \text{card}(B) + \text{card}(T)) \times (\text{len}(l_1) + \cdots + \text{len}(l_k))$.

Proof. Since $f(\mathtt{sk}) \in p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^+$, let $f(\mathtt{sk}) = p_1 l_1^{n_1} p_2 l_2^{n_2} \dots p_k l_k^{n_k}$ for some $n_1, \dots, n_k \geq 1$. We have $\text{len}(\mathtt{sk}) \leq (\text{len}(p_1) + \cdots + \text{len}(p_k)) + \max(n_i) \times (\text{len}(l_1) + \cdots + \text{len}(l_k))$. It remains to bound the values $n_1, \dots, n_k$. For each factor w of $\mathtt{sk}$ such that $f(w) = (l_i)^{n_i}$ with $i \in [1, k]$, by the **(minimality)** condition and Lemma 7.2, we conclude that $n_i \leq 2 \times (2 \times \text{card}(T) \times \text{card}(B) + \text{card}(T))$. Consequently, $\text{len}(\mathtt{sk}) \leq (\text{len}(p_1) + \cdots + \text{len}(p_k)) + 2 \times (2 \times \text{card}(T) \times \text{card}(B) + \text{card}(T)) \times (\text{len}(l_1) + \cdots + \text{len}(l_k))$. □

From skeletons, we shall define unfolded path schemas built over the alphabet $Q' \times \mathbb{G}^\star(T, B, U) \times U \times Q'$ (transitions are not anymore formally labelled by elements in Δ_P; sometimes we keep these labels for convenience). As for the definition of f, let $\tilde{\Delta}$ be a finite subset of $(Q' \times \mathbb{G}^\star(T, B, U) \times U \times Q')$ and let $h : (\Delta')^* \to (\tilde{\Delta})^*$ be the map such that $h(\varepsilon) = \varepsilon$, $h(w \cdot w') = h(w) \cdot h(w')$ and $h(\langle q, \mathbf{m}\rangle \xrightarrow{\delta, \langle \mathfrak{g}_{\mathbf{m}'}, \mathbf{u}\rangle} \langle q', \mathbf{m}'\rangle) = \langle q, \mathbf{m}\rangle \xrightarrow{\langle \mathfrak{g}_{\mathbf{m}'}, \mathbf{u}\rangle} \langle q', \mathbf{m}'\rangle$. This time, elements of Δ_P are removed instead of being kept as for f. Given a skeleton $\mathtt{sk}$, we shall define a path schema $P_{\mathtt{sk}} = p'_1 (l'_1)^+ p'_2 (l'_2)^+ \dots p'_{k'} (l'_{k'})^\omega$ such that $h(\mathtt{sk}) = p'_1 l'_1 p'_2 l'_2 \dots p'_{k'} l'_{k'}$. Hence, skeletons slightly differ from the path schemas. It remains to specify how the loops in $P_{\mathtt{sk}}$ are identified. Every factor $w = \langle q_H, \mathbf{m}_H\rangle \xrightarrow{\delta_H, \langle \mathfrak{g}^H_{\mathbf{m}'}, \mathbf{u}_H\rangle} \langle q_{H+1}, \mathbf{m}_{H+1}\rangle \cdots \xrightarrow{\delta_{J-1}, \langle \mathfrak{g}^{J-1}_{\mathbf{m}'}, \mathbf{u}_{J-1}\rangle} \langle q_J, \mathbf{m}_J\rangle$ of $\mathtt{sk}$ such that

1. $f(w) = l$ for some loop l of P,
2. w is not the suffix of $\mathtt{sk}$ of length $\text{len}(l_k)$,
3. the sequence of the $\text{len}(l)$ next elements after w is also equal to w,

is replaced by $(h(w))^+$. Finally, $l'_{k'}$ is equal to $h(w)$ where w is the unique suffix of $\mathtt{sk}$ of length $\text{len}(l_k)$. The path schema $P_{\mathtt{sk}}$ is unique by the condition (**minimality**). Indeed, there is no factor of $\mathtt{sk}$ of the form w^3 such that $f(w) = l$ for some loop l of P. As far as the labeling function is concerned, the labels of q and $\langle q, \mathbf{m}\rangle$ are identical with respect to the set X, i.e. $\mathbf{l}'(\langle q, \mathbf{m}\rangle) \stackrel{\text{def}}{=} \mathbf{l}(q) \cap X$. Hence,

1. $k' \leq k \times (2 \times \text{card}(T) \times \text{card}(B) + \text{card}(T))$,
2. $\text{len}(P_{\mathtt{sk}}) \leq (\text{len}(p_1) + \cdots + \text{len}(p_k)) + 2 \times (2 \times \text{card}(T) \times \text{card}(B) + \text{card}(T)) \times (\text{len}(l_1) + \cdots + \text{len}(l_k))$,
3. $P_{\mathtt{sk}}$ has no guards with disjunctions.

The construction of a path schema from a skeleton cannot be done by simply taking the path segments as before and the copies of the loop segments as alternating path and loop segments in the new path schema. For example, consider this system with one counter $\mathtt{x}$, $I = \{(-\infty, -1], [0, 0], [1, 1], [2, 2], [3, \infty)\}$ and $T = \{\mathtt{x} + 1\}$, the representation of a path schema P and of two unfolded path schemas represented in Fig. 6. Thanks to Fig. 6, we notice that

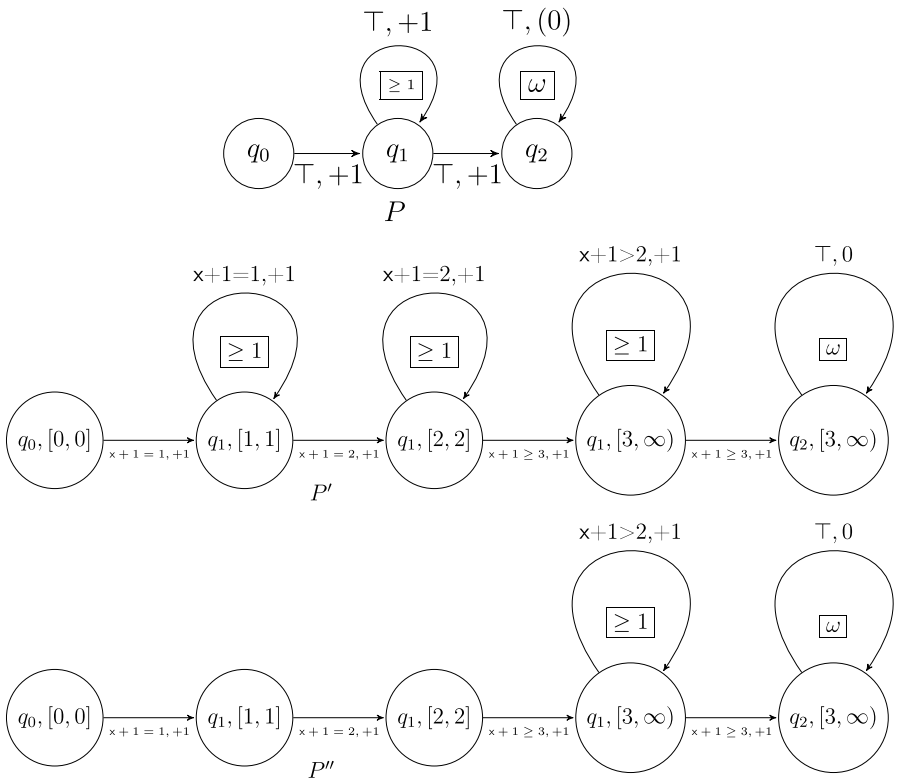


Fig. 6. A path P and two unfolded path schemas.

- $p_1(l_1^1)^+l_1^2(l_1^3)^+l_1^4(l_1^5)^+p_2(l_2)^\omega$ does not have any run respecting it as a path schema, as the loops l_1^1, l_1^3 cannot be taken even once in any run.
- $p_1l_1^2l_1^4(l_1^5)^+p_2(l_2)^\omega$ has a run respecting it as a path schema. But, here all the unfoldings of the loop l_1 are taken as path segments.

We write Y_{R,P,c_0} to denote the set of unfolded path schemas $P_{\mathtt{sk}}$ obtained from skeletons $\mathtt{sk}$ compatible with the resource R, the minimal path schema P and the initial configuration c_0.

Lemma 7.9. *Checking whether a word* $w \in (Q' \times \Delta \times \mathbb{G}^\star(T,B,U) \times U \times Q')^*$ *is a skeleton compatible with* R, P *and* $\langle q_0, \mathbf{v_0}\rangle$ *assuming that* $\mathrm{len}(w) \leq (\mathrm{len}(p_1) + \cdots + \mathrm{len}(p_k)) + 2(2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)) \times (\mathrm{len}(l_1) + \cdots + \mathrm{len}(l_k))$ *can be done in polynomial time in the size of* $\langle q_0, \mathbf{v_0}\rangle$, $\mathrm{size}(P)$, $\mathrm{size}(T)$ *and* $\mathrm{size}(B)$.

Observe that in the statement of Lemma 7.9, the length of w depends on the length or cardinal of several objects whereas the checking procedure depends on their respective size, partly because we need to take into account the binary encoding of the integers. As previously, the sizes $\mathrm{size}(P)$, $\mathrm{size}(T)$ and $\mathrm{size}(B)$ have not been formally defined but we assume a reasonably succinct encoding using a binary representation for integers.

Proof. Let w be a word over $Q' \times \Delta_P \times \mathbb{G}^\star(T,B,U) \times U \times Q'$ whose length is bounded by $(\mathrm{len}(p_1) + \cdots + \mathrm{len}(p_k)) + 2(2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)) \times (\mathrm{len}(l_1) + \cdots + \mathrm{len}(l_k))$. Let N be the sum of size of $\langle q_0, \mathbf{v_0}\rangle$, $\mathrm{size}(P)$, $\mathrm{size}(T)$ and $\mathrm{size}(B)$. Since the length of w is bounded, its size is also polynomial in N.

Checking whether an element in $Q' \times \Delta_P \times \mathbb{G}^\star(T,B,U) \times U \times Q'$ belongs to Δ' can be done in polynomial time in N thanks to Lemma 7.3(I). Hence, checking whether w belongs to $(\Delta')^*$ can be done in polynomial time in N too since its length is also polynomial in N. It remains to check the conditions for skeletons.

- Condition **(schema)** can be checked by building first $f(w)$ (this requires linear time in N) and then by checking whether it belongs to $p_1l_1^+p_2l_2^+ \ldots p_kl_k^+$ (requires also linear time in N).
- Condition **(last-loop)** can be checked by extracting the suffix of w of length $\mathrm{len}(l_k)$.
- Condition **(minimality)** can be checked by considering all the factors w' of w (there are less than $\mathrm{len}(w)^2$ of them) and whenever $f(w') = l^3$ for some loop l, we verify that the condition is satisfied. All these operations can be done in polynomial time in N.

- Finally, condition **(init)** is also easy to check in polynomial time in N. □

The main properties about the set of unfolded path schemas Y_{R,P,c_0} are stated below.

Proposition 7.10.

(I) *Let ρ be an infinite run respecting P and starting at $c_0 = \langle q_0, \mathbf{v_0}\rangle$. Then, there is a path schema P' in Y_{R,P,c_0} and an infinite run ρ' starting at $\langle\langle q_0, \mathbf{m}_0\rangle, \mathbf{v_0}\rangle$ respecting P' and such that $\mathsf{ft}_\rho = \mathsf{ft}_{\rho'}$.*

(II) *Let ρ be an infinite run starting at $\langle\langle q_0, \mathbf{m}_0\rangle, \mathbf{v_0}\rangle$ and respecting P' for some $P' \in Y_{\mathsf{R},P,c_0}$. Then, there is an infinite run ρ' starting at $c_0 = \langle q_0, \mathbf{v_0}\rangle$ and respecting P such that $\mathsf{ft}_\rho = \mathsf{ft}_{\rho'}$.*

Proof. (I) Let $\rho = \langle q_0, \mathbf{v_0}\rangle \xrightarrow{\delta_0} \langle q_1, \mathbf{v_1}\rangle \xrightarrow{\delta_1} \cdots$ be an infinite run respecting P with footprint $\mathsf{ft}_\rho : \mathbb{N} \to 2^{AT} \times I^T$. We write $\langle Z_i, \mathbf{m}_i\rangle$ to denote $\mathsf{ft}_\rho(i)$. In order to build ρ' and P', first we enrich the structure ρ and then we define a skeleton from the enriched structure that allows us to define P'. The run ρ' is then defined from ρ so that the sequences of counter values are identical. From ρ, we consider the infinite sequence below:

$$w = \langle q_0, \mathbf{m}_0\rangle \xrightarrow{\delta_0, \langle \mathfrak{g}_{\mathbf{m}_1}, update(\delta_0)\rangle} \langle q_1, \mathbf{m}_1\rangle \xrightarrow{\delta_1, \langle \mathfrak{g}_{\mathbf{m}_2}, update(\delta_1)\rangle} \cdots$$

It is easy to check that w can be viewed as an element of $(\Delta')^\omega$ where Δ' is defined as a finite subset of $Q' \times \Delta_P \times \mathbb{G}^\star(T, B, U) \times U \times Q'$ where U is the finite set of updates from $P = p_1(l_1)^+ p_2(l_2)^+ \cdots (l_{k-1})^+ p_k(l_k)^\omega$. Moreover, we have $f(w) \in \mathcal{L}(P)$, that is $f(w) = p_1(l_1)^{n_1} p_2(l_2)^{n_2} \cdots (l_{k-1})^{n_{k-1}} p_k(l_k)^\omega$ for some $n_1, \ldots, n_{k-1} \geq 1$. From w, one can build a skeleton $\mathtt{sk}$ compatible with P and $\langle q_0, \mathbf{v_0}\rangle$. $\mathtt{sk}$ is formally a subword of w such that

$$f(\mathtt{sk}) = p_1(l_1)^{n'_1} p_2(l_2)^{n'_2} \cdots (l_{k-1})^{n'_{k-1}} p_k(l_k)^{n'_k}$$

with $1 \leq n'_i \leq \min(n_i, 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T))$ for every $i \in [1, k-1]$ and $1 \leq n'_k \leq 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T))$. We have $w = w' \cdot w_0 \cdot w_0 \cdot (w_0)^\omega$ with $f(w_0) = l_k$ for some w_0. The skeleton $\mathtt{sk}$ is obtained from $w' \cdot w_0 \cdot w_0$ by deleting copies of loops as soon as two copies are consecutive. More precisely, every maximal factor of $w' \cdot w_0 \cdot w_0$ of the form $(w^\star)^N$ with $N > 2$ such that $f(w^\star) = l_i$ for some loop l_i of P, is replaced by $(w^\star)^2$. This type of replacement can be done at most $k \times (2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)))$ times. One can check that $\mathtt{sk}$ is indeed a skeleton compatible with R, P and $\langle q_0, \mathbf{v_0}\rangle$. Considering the path schema $P_{\mathtt{sk}}$ built from $\mathtt{sk}$, one can show that the sequence ρ' below is an infinite run respecting $P_{\mathtt{sk}}$:

$$\langle\langle q_0, \mathbf{m}_0\rangle, \mathbf{v_0}\rangle \xrightarrow{\langle \mathfrak{g}_{\mathbf{m}_1}, update(\delta_0)\rangle} \langle\langle q_1, \mathbf{m}_1\rangle, \mathbf{v_1}\rangle \xrightarrow{\langle \mathfrak{g}_{\mathbf{m}_2}, update(\delta_1)\rangle} \langle\langle q_2, \mathbf{m}_2\rangle, \mathbf{v_2}\rangle \cdots$$

so that $\mathsf{ft}_\rho = \mathsf{ft}_{\rho'}$. When entering in the last loop of $P_{\mathtt{sk}}$, counter values still evolve but the sequence of control states forms a periodic word made of the last $\mathrm{len}(l_k)$ control states of $\mathtt{sk}$. By construction of $\mathtt{sk}$ and $P_{\mathtt{sk}}$, it is clear that ρ and ρ' have the same sequences of counter values (they have actually the same sequences of updates) and by definition of the labels, they have also the same sequences of sets of atomic propositions. It remains to check that ρ' is indeed a run, which amounts to verify that guards are satisfied. This is guaranteed by the way guards are defined, by the completeness result in Lemma 7.3(II) and by the result of Lemma 7.7.

(II) Let ρ be some run respecting some $P' \in Y_{\mathsf{R},P,c_0}$ of the following form:

$$\langle\langle q_0, \mathbf{m}_0\rangle, \mathbf{v_0}\rangle \xrightarrow{\delta_0, \langle \mathfrak{g}_{\mathbf{m}_1}, update(\delta_0)\rangle} \langle\langle q_1, \mathbf{m}_1\rangle, \mathbf{v_1}\rangle \xrightarrow{\delta_1, \langle \mathfrak{g}_{\mathbf{m}_2}, update(\delta_1)\rangle} \langle\langle q_2, \mathbf{m}_2\rangle, \mathbf{v_2}\rangle \cdots$$

In the above run, we have decorated the steps by transitions from P as P' is defined from a skeleton in which transitions are decorated by such transitions. After a tedious (but not difficult) verification, one can show that

$$\rho' = \langle q_0, \mathbf{v_0}\rangle \xrightarrow{\delta_0} \langle q_1, \mathbf{v_1}\rangle \xrightarrow{\delta_1} \cdots$$

is a run respecting P such that $\mathsf{ft}_\rho = \mathsf{ft}_{\rho'}$. Satisfaction of guards is guaranteed by the way Δ' is defined. The fact that ρ' respects P is even easier to justify since all the path schemas in Y_{R,P,c_0} can be viewed as specific instances of P that differ in the way the term maps evolve (see the condition **(schema)**). Details are omitted. □

Let $P' = p'_1(l'_1)^+ p'_2(l'_2)^+ \ldots p'_{k'}(l'_{k'})^\omega$ be a path schema in Y_{R,P,c_0} and ρ be a run $\langle\langle q_0, \mathbf{m}_0\rangle, \mathbf{v_0}\rangle \xrightarrow{\langle \mathfrak{g}_{\mathbf{m}_1}, update(\delta_0)\rangle} \langle\langle q_1, \mathbf{m}_1\rangle, \mathbf{v_1}\rangle \xrightarrow{\langle \mathfrak{g}_{\mathbf{m}_2}, update(\delta_1)\rangle} \langle\langle q_2, \mathbf{m}_2\rangle, \mathbf{v_2}\rangle \cdots$ respecting P'. It is easy to show that for $i \geq 0$, we have $\pi_2(\mathsf{ft}_\rho(i)) = \mathbf{m}_i$ and ft_ρ is an ultimately periodic word of the form $w \cdot u^\omega$ where $\mathrm{len}(u) = \mathrm{len}(l'_{k'}) = \mathrm{len}(l_k)$ and $\mathrm{len}(w) = (\mathrm{len}(p'_1) + \cdots + \mathrm{len}(p'_{k'})) + (iter_{P'}(\rho)[1] \times \mathrm{len}(l'_1) + \cdots + iter_{P'}(\rho)[k'-1] \times \mathrm{len}(l'_{k'-1}))$. As seen previously, we have $\rho, 0 \models \phi$ iff $\mathsf{ft}_\rho, 0 \models_{\mathsf{symb}} \phi$.

Let us also define the function proj which associates to $w \in \tilde{\Delta}^\omega$ the ω-sequence $\mathrm{proj}(w) : \mathbb{N} \to 2^X \times I^T$ such that for all $i \in \mathbb{N}$, if $w(i) = \langle\langle q, \mathbf{m}\rangle, \mathfrak{g}, \mathbf{u}, \langle q', \mathbf{m}'\rangle\rangle$ and $\mathbf{l}(q) \cap X = L$ then $\mathrm{proj}(w)(i) \stackrel{\text{def}}{=} \langle L, \mathbf{m}\rangle$. Now, we can state the main theorem about removing disjunction in the guards by unfolding of loops. It entails the main properties we expect from Y_{R,P,c_0}.

Theorem 7.11. *Given a flat counter system S, a minimal path schema P, a resource $\mathsf{R} = \langle X, T, B\rangle$ such that the set of terms T includes those in P, the set of constants B includes those in P, and an initial configuration $c_0 = \langle q_0, \mathbf{v_0}\rangle$, there is a finite set of path schemas Y_{R,P,c_0}, such that:*

1. *No path schema in Y_{R,P,c_0} contains disjunctions occurring in guards.*
2. *For every $P' \in Y_{\mathsf{R},P,c_0}$, $\mathrm{len}(P')$ is polynomial in $\mathrm{len}(P) + \mathrm{card}(T) + \mathrm{card}(B)$.*
3. *Checking whether P' belongs to Y_{R,P,c_0} can be done in polynomial time in $\mathrm{size}(P) + \mathrm{size}(T) + \mathrm{size}(B)$.*
4. *For every run ρ respecting P and starting at $\langle q_0, \mathbf{v_0}\rangle$, we can find a run ρ' respecting some $P' \in Y_{\mathsf{R},P,c_0}$ such that $\rho \models \phi$ iff $\rho' \models \phi$ for every ϕ built over R.*
5. *For every run ρ' respecting some $P' \in Y_{\mathsf{R},P,c_0}$ with initial counter values $\mathbf{v_0}$, we can find a run ρ respecting P such that $\rho \models \phi$ iff $\rho' \models \phi$ for every ϕ built over R.*
6. *For every $P' \in Y_{\mathsf{R},P,c_0}$, for every ultimately periodic word $w \cdot u^\omega \in \mathcal{L}(P')$, for every ϕ built over R checking whether $\mathsf{proj}(w \cdot u^\omega), 0 \models_{\mathsf{symb}} \phi$ can be done in polynomial time in the size of $w \cdot u$ and in the size of ϕ.*

Proof. Let Y_{R,P,c_0} be the set of path schemas defined from the resource R, the minimal path schema P and the initial configuration c_0.

1. For every path schema in Y_{R,P,c_0}, the guards on transitions are of the form $\bigwedge_{\mathtt{t}\in T} \psi(\mathtt{t}, \mathbf{u}, \mathbf{m}(\mathtt{t}))$ and each guard $\psi(\mathtt{t}, \mathbf{u}, \mathbf{m}(\mathtt{t}))$ is itself an atomic guard or a conjunction of two atomic guards. Hence, no path schema in Y_P contains any disjunction in some guard.
2. By Lemma 7.8, every skeleton defining a path schema in Y_{R,P,c_0} has polynomial length in $\mathrm{len}(P) + \mathrm{card}(T) + \mathrm{card}(B)$. Each path schema in Y_{R,P,c_0} has a linear length in the length of its corresponding skeleton. Consequently, for every $P' \in Y_{\mathsf{R},P,c_0}$, its length $\mathrm{len}(P')$ is polynomial in $\mathrm{len}(P) + \mathrm{card}(T) + \mathrm{card}(B)$.
3. Given a path schema P' in Y_{R,P,c_0}, one can easily identify its underlying skeleton $\mathtt{sk}$ by removing iteration operators such as $^+$ and $^\omega$ (easy at the cost of keeping track of transitions from Δ_P). By Lemma 7.9, checking whether $\mathtt{sk}$ is compatible with R, P and $\langle q_0, \mathbf{v_0}\rangle$ can be done in polynomial time in $\mathrm{size}(P) + \mathrm{size}(T) + \mathrm{size}(B)$. In particular, if $\mathtt{sk}$ is too long, this can be checked in polynomial time too.
4. By Proposition 7.10(I), for every run ρ respecting P and starting at $\langle q_0, \mathbf{v_0}\rangle$, there are $P' \in Y_{\mathsf{R},P,c_0}$ and a run ρ' respecting P' such that $\mathsf{ft}_\rho = \mathsf{ft}_{\rho'}$. By Lemma 7.6, $\rho \models \phi$ iff $\rho' \models \phi$.
5. Similar to (4.) by using Proposition 7.10(II).
6. Let us consider an ultimately periodic word $w \cdot u^\omega \in \mathcal{L}(P')$. We can build in linear time the ultimately periodic word $w' \cdot u'^\omega = \mathsf{proj}(w \cdot u^\omega)$ over the alphabet $2^X \times I^T$ and the size of the word w' [resp. u'] is linear in the size of the word w [resp. w']. By [29, Theorem 5.1], we know that $w' \cdot u'^\omega, 0 \models_{\mathsf{symb}} \phi$ can be checked in time $\mathcal{O}(\mathrm{size}(\phi)^2 \times \mathrm{len}(w' \cdot u'))$. Indeed, $\models_{\mathsf{symb}}$ is analogous to the satisfiability relation for plain Past LTL. □

We write $Y_{\mathsf{R},P}$ to denote the set of path schemas defined as for those in Y_{R,P,c_0} except that there is no constraint on the first term map (this amount to ignore the condition **(init)**). This set will be useful to solve the global model-checking problem.

8. Model-checking PLTL[C] over flat counter systems

8.1. A decision procedure in NP

We provide here a nondeterministic polynomial-time algorithm to solve MC(PLTL[C], FlatCS). To do so, we combine the properties of the general stuttering theorem for LTL with past-time operators (see Theorem 4.1) with small solutions of constraint systems. In Algorithm 1 below, nondeterministic steps (guesses) are performed only at the beginning of the algorithm. Note that a polynomial $p^\star(\cdot)$ is used and its existence follows from forthcoming Theorem 8.1. Similarly, the polynomial $q^\star(\cdot)$ follows from Theorem 7.11.

Now, we explain how $p^\star(\cdot)$ is defined. Let S be a flat counter system, $c_0 = \langle q_0, \mathbf{v_0}\rangle$ be an initial configuration and $\phi \in \mathrm{PLTL}[\mathtt{C}]$. Let $N = \mathrm{size}(S) + \mathrm{size}(\langle q_0, \mathbf{v_0}\rangle) + \mathrm{size}(\phi)$. Let P be a minimal path schema of S. We have:

- $\mathrm{len}(P) \leq 2 \times \mathrm{card}(\Delta) \leq 2N$,
- $\mathrm{nbloops}(P) \leq \mathrm{card}(Q) \leq N$.

Let T be the set of terms $\mathtt{t}$ occurring in S and ϕ in guards of the form $\mathtt{t} \sim b$. We have $\mathrm{card}(T) \leq \mathrm{size}(S) + \mathrm{size}(\phi) \leq N$. Let B be the set of constants b occurring in either S or ϕ in guards of the form $\mathtt{t} \sim b$. We have $\mathrm{card}(B) \leq \mathrm{size}(S) + \mathrm{size}(\phi) \leq N$. Let $\mathsf{R} = \langle X, T, B\rangle$ be the resource such that X is the finite set of propositional variables occurring in ϕ. Such a resource is said to be coherent with S and ϕ.

Let $\mathtt{M}$ be the maximal absolute value of a constant occurring in S, ϕ, $\mathbf{v_0}$ (either as an element of B or as a coefficient in front of a counter or as a component in $\mathbf{v_0}$). We have $\mathtt{M} \leq 2^N$.

Algorithm 1 The main algorithm in NP with inputs S, $c_0 = \langle q, \mathbf{v_0}\rangle$, ϕ.

1: guess a minimal path schema P of S satisfying the infiniteness property
2: build a resource $\mathsf{R} = \langle X, T, B\rangle$ coherent with S and ϕ
3: guess a path schema $P' = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ such that $\mathrm{len}(P') \leq q^\star(\mathrm{len}(P) + \mathrm{card}(T) + \mathrm{card}(B))$
4: guess $\mathbf{y} \in [1, 2td(\phi)+5]^{k-1}$
5: guess $\mathbf{y}' \in [1, 2^{p^\star(\mathrm{size}(S)+\mathrm{size}(c_0)+\mathrm{size}(\phi))}]^{k-1}$
6: check that P' belongs to Y_{R,P,c_0}
7: check that $\mathrm{proj}(p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$
8: build the constraint system $\mathcal{E}$ over the variables $\mathsf{y}_1, \dots, \mathsf{y}_{k-1}$ for P' with initial counter values $\mathbf{v_0}$ (obtained from Lemma 6.1)
9: **for** $i = 1 \to k-1$ **do**
10: **if** $\mathbf{y}[i] = 2td(\phi)+5$ **then**
11: $\psi_i \leftarrow$ "$\mathsf{y}_i \geq 2td(\phi)+5$"
12: **else**
13: $\psi_i \leftarrow$ "$\mathsf{y}_i = \mathbf{y}[i]$"
14: **end if**
15: **end for**
16: check that $\mathbf{y}' \models \mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$

Now, let P' be a path schema in Y_{R,P,c_0} with $P' = p_1(l_1)^+ p_2(l_2)^+ \cdots p_k(l_k)^\omega$. Since $\mathrm{len}(P') \leq (\mathrm{len}(p_1) + \dots + \mathrm{len}(p_k)) + 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)) \times (\mathrm{len}(l_1) + \dots + \mathrm{len}(l_k))$, we have $\mathrm{len}(P') \leq 5 \times \mathrm{card}(T) \times \mathrm{card}(B) \times \mathrm{len}(P) \leq 5N^3$. Similarly, $\mathrm{nbloops}(P') \leq 5N^3$. The number of guards occurring in P' is bounded by $\mathrm{len}(P') \times 2 \times \mathrm{card}(T) \leq 10 \times N^4$. The maximal constant $\mathtt{M}'$ occurring in P' is bounded by $\mathtt{M} + n \times \mathtt{M}^2$ which is bounded by $N \times 2^{2\times N}$. Let $\mathcal{E}$ be the constraint system defined from P' (see Lemma 6.1).

- The number of variables is equal to $\mathrm{nbloops}(P')$ which is bounded by $5N^3$.
- The number of conjuncts is bounded by $2 \times \mathrm{len}(P') \times n \times (1 + N_1)$ where N_1 is the number of atomic guards in P'. Hence, this number is bounded by $2 \times 5N^3 \times N \times (1 + 10 \times N^4) \leq 110N^8$.
- The greatest absolute value from constants in $\mathcal{E}$ is bounded by $n \times \mathrm{nbloops}(P') \times (\mathtt{M}')^4 \times \mathrm{len}(P')^3$, which is bounded by $N(5N^3)(N \times 2^{2\times N})^4 \times 5^3 N^9 \leq 625 \times N^{17} \times 2^{8\times N}$.

We will see in the sequel how these bounds allows us to show that $\mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$ admits a small solution using the theorem below for any $\psi_1 \wedge \dots \wedge \psi_{k-1}$ built from Algorithm 1.

Theorem 8.1. *[3] Let $\mathcal{M} \in [-M, M]^{a\times b}$ and $\mathbf{b} \in [-M, M]^a$, where $a, b, M \in \mathbb{N}$. If there is $\mathbf{x} \in \mathbb{N}^b$ such that $\mathcal{M}\mathbf{x} \geq \mathbf{b}$, then there is $\mathbf{y} \in [0, (\max\{b, M\})^{\mathsf{C}a}]^b$ such that $\mathcal{M}\mathbf{y} \geq \mathbf{b}$, where C is some constant.*

By Theorem 8.1, $\mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$ has a solution iff $\mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$ has a solution whose counter values are bounded by

$$(625 \times N^{17} \times 2^{8\times N})^{\mathsf{C}\times 2\times(110\times N^8 + 5\times N^3)}$$

which can be easily shown to be bounded by $2^{p^\star(N)}$ for some polynomial $p^\star(\cdot)$ (of degree 9). This is precisely, the polynomial $p^\star(\cdot)$ that is used in Algorithm 1 (for obvious reasons). In order to justify the coefficient 2 before 110, note that any constraint of the form $\sum_i a_i \mathsf{y}_i \sim b$ with $\sim \in \{=, \leq, \geq, <, >\}$ can be equivalently replaced by 1 or 2 atomic constraints of the form $\sum_i a_i \mathsf{y}_i \geq b$.

Algorithm 1 starts by guessing a path schema P (line 1) and an unfolded path schema $P' = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ (line 3) and check whether P' belongs to Y_{R,P,c_0} (line 5). It remains to check whether there is a run ρ respecting P' such that $\rho \models \phi$. Suppose there is such a run ρ; let $\mathbf{y}$ be the unique tuple in $[1, 2td(\phi)+5]^{k-1}$ such that $\mathbf{y} \approx_{2td(\phi)+5} iter_{P'}(\rho)$. By Proposition 5.1, we have $\mathrm{proj}(p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$. Since the set of tuples of the form $iter_{P'}(\rho)$ is characterized by a quantifier-free Presburger formula, by the existence of small solutions from [3], we can assume that $iter_{P'}(\rho)$ contains only small values. Hence line 4 guesses $\mathbf{y}$ and $\mathbf{y}'$ (corresponding to $iter_{P'}(\rho)$ with small values). Line 6 precisely checks $\mathrm{proj}(p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$ whereas line 11 checks whether $\mathbf{y}'$ encodes a run respecting P' with $\mathbf{y}' \approx_{2td(\phi)+5} \mathbf{y}$.

Lemma 8.2. *Algorithm 1 runs in nondeterministic polynomial time.*

Proof. We first check that all the guesses can be done in polynomial time.

- A minimal path schema P of S is of polynomial size with respect to the size of S.
- The path schema P' is of polynomial size with respect to the size of P, ϕ and c_0 (Theorem 7.11(2)).
- $\mathbf{y}$ and $\mathbf{y}'$ are obviously of polynomial size since their components have values bounded by some exponential expression only (values in $\mathbf{y}$ can be much smaller than the values in $\mathbf{y}'$).

Now, we verify that all the checks can be in done in polynomial time too.

- Both P and P' are in polynomial size with respect to the size of the inputs and checking compatibility amounts to verify that P' is an unfolding of P, which can be done in polynomial time (see Theorem 7.11(3)).
- Checking whether $\mathsf{proj}(p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^{\omega}), 0 \models_{\mathsf{symb}} \phi$ can be done in polynomial time using Theorem 7.11(6) since $p_1 l_1^{\mathbf{y}(1)} p_2 l_2^{\mathbf{y}(2)} \dots l_{k-1}^{\mathbf{y}(k-1)} p_k l_k$ is of polynomial size with respect to the size of P' and ϕ.
- Building $\mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$ can be done in polynomial time since $\mathcal{E}$ can be built in polynomial time with respect to the size of P' (see Section 6) and $\psi_1 \wedge \dots \wedge \psi_{k-1}$ can be built in polynomial time with respect to the size of ϕ ($td(\phi) \leq \mathsf{size}(\phi)$).
- $\mathbf{y}' \models \mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$ can be finally checked in polynomial time since the values in $\mathbf{y}'$ are of exponential magnitude and the combined constraint system is of polynomial size. □

It remains to verify that Algorithm 1 is correct, which is stated below.

Lemma 8.3. *$S, c_0 \models \phi$ iff Algorithm 1 on inputs S, c_0, ϕ has an accepting run.*

In the proof of Lemma 8.3, we take advantage of all our preliminary results.

Proof. First, let us show that if Algorithm 1 on inputs S, $c_0 = \langle q_0, \mathbf{v_0} \rangle$, ϕ has an accepting computation, then S, $c_0 \models \phi$. This means that there are P, P', $\mathbf{y}$, $\mathbf{y}'$ that satisfy all the checks. Let $w = p_1 l_1^{\mathbf{y}'[1]} \cdots p_{k-1} l_{k-1}^{\mathbf{y}'[k-1]} p_k l_k^{\omega}$ and $\rho = \langle\langle q_0, \mathbf{m}_0 \rangle, \mathbf{v'_0}\rangle\langle\langle q_1, \mathbf{m}_1 \rangle, \mathbf{v'_1}\rangle\langle\langle q_2, \mathbf{m}_2 \rangle, \mathbf{v'_2}\rangle \cdots \in (Q' \times \mathbb{Z}^n)^{\omega}$ be defined as follows:

- For every $i \geq 0$, $q_i \stackrel{\text{def}}{=} \pi_1(source(w(i)))$,
- $\mathbf{v'_0} \stackrel{\text{def}}{=} \mathbf{v_0}$ and $\mathbf{v'_i} \stackrel{\text{def}}{=} \mathbf{v'_{i-1}} + update(w(i))$ for every $i \geq 1$.

By Lemma 6.1, since $\mathbf{y}' \models \mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$, we deduce that ρ is a run respecting P' starting at the configuration $\langle\langle q_0, \mathbf{m}_0 \rangle, \mathbf{v_0}\rangle$. Since $\mathbf{y}' \models \psi_1 \wedge \dots \wedge \psi_{k-1}$ and $\mathbf{y} \models \psi_1 \wedge \dots \wedge \psi_{k-1}$, by Proposition 5.1, the statements below are equivalent:

($\maltese$) $\mathsf{proj}(p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^{\omega}), 0 \models_{\mathsf{symb}} \phi$,
($\maltese\maltese$) $\mathsf{proj}(p_1 l_1^{\mathbf{y}'[1]} p_2 l_2^{\mathbf{y}'[2]} \dots l_{k-1}^{\mathbf{y}'[k-1]} p_k l_k^{\omega}), 0 \models_{\mathsf{symb}} \phi$.

Line 6 from Algorithm 1 guarantees that $\mathsf{proj}(p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^{\omega}), 0 \models_{\mathsf{symb}} \phi$, whence we have ($\maltese\maltese$). Since $\mathsf{proj}(p_1 l_1^{\mathbf{y}'[1]} p_2 l_2^{\mathbf{y}'[2]} \dots l_{k-1}^{\mathbf{y}'[k-1]} p_k l_k^{\omega}) = \mathsf{ft}_{\rho}$, by Lemma 7.5, we deduce that $\rho, 0 \models \phi$. By Theorem 7.11(5), there is an infinite run ρ', starting at the configuration $\langle q_0, \mathbf{v_0} \rangle$ and respecting P, such that $\rho', 0 \models \phi$.

Now, suppose that $S, c_0 \models \phi$. We show that there exist P, P', $\mathbf{y}$, $\mathbf{y}'$ that allow to build an accepting computation of Algorithm 1. There is a run ρ starting at c_0 such that $\rho, 0 \models \phi$. By Corollary 3.4, ρ respects some minimal path schema of S, say P. By Theorem 7.11(4), there is a path schema $P' = p_1 l_1^{+} p_2 l_2^{+} \dots p_k l_k^{\omega}$ in Y_{R,P,c_0} for which there is a run ρ' satisfying ϕ. Furthermore, since $P' \in Y_{\mathsf{R},P,c_0}$, we know that $\mathsf{len}(P') \leq q^{\star}(\mathsf{len}(P) + \mathsf{card}(T) + \mathsf{card}(B))$ for some polynomial $q^{\star}(\cdot)$. From $iter_{P'}(\rho') \in (\mathbb{N} \setminus \{0\})^{k-1}$, for every $i \in [1, k-1]$, we consider ψ_i such that ψ_i is equal to $\mathsf{y}_i = iter_{P'}(\rho')[i]$ if $iter_{P'}(\rho')[i] \leq 2td(\phi) + 5$, otherwise ψ_i is equal to $\mathsf{y}_i \geq 2td(\phi) + 5$. Since P' admits at least one infinite run ρ' such that $iter_{P'}(\rho')$ satisfies $\psi_1 \wedge \dots \wedge \psi_{k-1}$, the constraint system $\mathcal{E}$ obtained from P' (thanks to Lemma 6.1) but augmented with $\psi_1 \wedge \dots \wedge \psi_{k-1}$ admits at least one solution. Let us define $\mathbf{y}' \in [1, 2^{p^{\star}(\mathsf{size}(S)+\mathsf{size}(c_0)+\mathsf{size}(\phi))}]^{k-1}$ as a small solution of $\mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$ and $\mathbf{y} \in [1, 2td(\phi) + 5]^{k-1}$ be defined such that $\mathbf{y}[i] = \max(\mathbf{y}'[i], 2td(\phi) + 5)$ for $i \in [1, k-1]$. As shown previously, the bound $2^{p^{\star}(\mathsf{size}(S)+\mathsf{size}(c_0)+\mathsf{size}(\phi))}$ is sufficient if there is a solution. Clearly, $\mathbf{y}' \models \mathcal{E} \wedge \psi_1 \wedge \dots \wedge \psi_{k-1}$. So $p_1 l_1^{\mathbf{y}'[1]} p_2 l_2^{\mathbf{y}'[2]} \dots l_{k-1}^{\mathbf{y}'[k-1]} p_k l_k^{\omega}$ generates a genuine run. Since $\mathsf{ft}_{\rho'} = \mathsf{proj}(p_1 l_1^{\mathbf{y}'[1]} p_2 l_2^{\mathbf{y}'[2]} \dots l_{k-1}^{\mathbf{y}'[k-1]} p_k l_k^{\omega})$ (see Lemma 7.7) and since by Lemma 7.5, we have $\mathsf{ft}_{\rho'} \models_{\mathsf{symb}} \phi$, we get that $\mathsf{proj}(p_1 l_1^{\mathbf{y}'[1]} p_2 l_2^{\mathbf{y}'[2]} \dots l_{k-1}^{\mathbf{y}'[k-1]} p_k l_k^{\omega}), 0 \models_{\mathsf{symb}} \phi$. This also implies that P' satisfies the infiniteness property. Hence

$$\mathsf{proj}(p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \dots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^{\omega}), 0 \models_{\mathsf{symb}} \phi$$

thanks to Proposition 5.1. Consequently, we have all the ingredients to build safely an accepting run for Algorithm 1 on inputs S, c_0, ϕ. □

The two previous lemmas allow us to state the main result of this paper.

Theorem 8.4. MC(PLTL[C], `FlatCS`) *is NP-complete.*

As a corollary, we can also solve the global model-checking problem with existential Presburger formulae (we knew that Presburger formulae exist for global model-checking [14] but we can conclude that they are structurally simple and we provide an alternative proof).

Corollary 8.5. *Given a flat counter system S, a control state q_0 and a formula $\phi \in \mathrm{PLTL}[\mathtt{C}]$ one can effectively build an existential Presburger formula ϕ that represents the initial counter values $\mathbf{v_0}$ such that there is an infinite run ρ starting at $\langle q_0, \mathbf{v_0}\rangle$ such that $\rho, 0 \models \phi$.*

It is sufficient to consider the formula below:

$$\bigvee_{\text{minimal path schema } P} \ \bigvee_{P'=p_1 l_1^+ p_2 l_2^+ \ldots l_{k-1}^+ p_k l_k^\omega \in Y_{\mathsf{R},P}} \ (\bigwedge_{\mathtt{t}\in T} \varphi(\mathtt{t}, \mathbf{m}^{\text{first}}_{P'}(\mathtt{t}))) \wedge$$

$$\bigvee_{\mathbf{y}\ s.t.\ \mathsf{ft}_{p_1 l_1^{\mathbf{y}[1]} p_2 l_2^{\mathbf{y}[2]} \ldots l_{k-1}^{\mathbf{y}[k-1]} p_k l_k^\omega}, 0 \models_{\mathsf{symb}} \phi} \ \exists\, \mathsf{y}_1 \cdots \mathsf{y}_{k-1}\ \mathcal{E}'_{P'} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$$

where

- the first generalized disjunction deals with minimal path schemas starting on q_0,
- the second disjunction enumerates the unfolded path schema in $Y_{\mathsf{R},P}$ with no constraint on the first term map, say $\mathbf{m}^{\text{first}}_{P'}$,
- given a term $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$ and a term map $\mathbf{m}$, we write $\varphi(\mathtt{t}, \mathbf{m}(\mathtt{t}))$ to denote the formula below (where $b, b' \in B$):
  - $\varphi(\mathtt{t}, (-\infty, b]) \stackrel{\text{def}}{=} \sum_j a_j \mathsf{x}_j \leq b$,
  - $\varphi(\mathtt{t}, [b, +\infty)) \stackrel{\text{def}}{=} \sum_j a_j \mathsf{x}_j \geq b$,
  - $\varphi(\mathtt{t}, [b, b']) \stackrel{\text{def}}{=} ((\sum_j a_j \mathsf{x}_j \leq b') \wedge ((\sum_j a_j \mathsf{x}_j \geq b)$.
- the third generalized disjunction deals with $\mathbf{y} \in [1, 2td(\phi) + 5]^{k-1}$,
- the constraint system $\mathcal{E}'_{P'}$ is obtained from $\mathcal{E}_{P'}$ by replacing initial counter values by free variables, i.e. by replacing occurrences of $\mathbf{v_0}[i]$ by x_i in expressions built in Section 6.

8.2. The special case of path schemas with a single loop

Thanks to Lemma 5.6, we have seen that MC(PLTL[C], CPS(k)) is NP-hard as soon as $k \geq 2$. By contrast, we prove that MC(PLTL[C], CPS(1)) is in PTime by using the previous proof techniques.

Consider a path schema $P = p \cdot l^\omega$ in a counter system with only one loop l. Due to the structure of P there exists at most one run ρ respecting P and starting from a given initial configuration c_0. The footprint ft_ρ of this run (see Section 7) is of the form $u.v^\omega$, which is an ultimately periodic word. Since the only loop l is to be taken an infinite number of times, we have $\mathrm{len}(v) = \mathrm{len}(l)$, which is polynomial in the input size, but $\mathrm{len}(u)$ can be exponential. In fact, in the word $lab(\rho(0)\rho(1)\cdots\rho(\mathrm{len}(u))) \in p \cdot l^+$, the number of repetitions that can be attached to the loop l may be exponential. The algorithm computes the number of different possible sets of term maps (defined in Section 7), that can be generated while visiting the loop l. There are at most a polynomial number of such term maps due to Lemma 7.2. Next, for each such assignment of term maps to the nodes of l, the algorithm calculates the number of iterations β of l, for which the terms remain in their respective term map. However, each of these β_i can be exponentially large. Now, the formula is symbolically verified over the ultimately periodic path where the nodes of the path schema are augmented with the term maps.

Before providing the algorithm formally, we need to introduce auxiliary notions. For a path segment $p = \delta_0\delta_1\cdots\delta_{\mathrm{len}(p)-1}$, we define $p[i, j] = \delta_i\delta_{i+1}\cdots\delta_j$ with $0 \leq i \leq j \leq \mathrm{len}(p) - 1$ and $p[i, j] = \varepsilon$ (the empty word) if $j < i$. We use furthermore the convention that $\mathit{effect}(\varepsilon) = 0$.

Since the unique run respecting P must contain p and copies of l, we can specify the term maps for $w = p \cdot l$. Consider the function $f_{\text{init}} : [0, \mathrm{len}(w)] \to I^T$ for a given configuration $c_0 = \langle q_0, \mathbf{v}_0\rangle$, defined as follows:

- $f_{\text{init}}(0) \stackrel{\text{def}}{=} \mathbf{m}_0$ iff for each term $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$, we have $\sum_j a_j.\mathbf{v}_0[j] \in \mathbf{m}_0(\mathtt{t})$ and $\mathbf{m}_0 \vdash \mathit{guard}(w(0))$.
- for every $i \in [1, \mathrm{len}(w)]$, $f_{\text{init}}(i) \stackrel{\text{def}}{=} \mathbf{m}_i$ iff, for each term $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$, we have $\sum_j a_j.(\mathit{effect}(w[0, i-1])[j] + \mathbf{v}_0[j]) \in \mathbf{m}_i(\mathtt{t})$ and if $i < \mathrm{len}(w)$, then $\mathbf{m}_i \vdash \mathit{guard}(w(i))$.
- Otherwise, if the term maps do not satisfy the guards, then there does not exist any run and hence $f_{\text{init}}(i)$ is undefined.

In the algorithm, we also consider a map $\mathit{curr} : T \to \mathbb{Z}$ that provides the value of the terms at specific positions of the run. From this function, we define the function $\mathit{val}_{\mathit{curr}} : \Delta^+ \to I^T$ as follows, $\mathit{val}_{\mathit{curr}}(w) = \mathbf{m}$ where $\mathbf{m}$ verifies: $\mathit{curr}(\mathtt{t}) + \sum_j a_j.(\mathit{effect}(w)[j]) \in \mathbf{m}(\mathtt{t})$ for all $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$. Finally, given a path segment $p = \delta_0\delta_1\cdots\delta_{\mathrm{len}(p)-1}$ with

Algorithm 2 The PTIME algorithm with inputs $P = p \cdot l^{\omega}$, $c_0 = \langle q_0, \mathbf{v}_0 \rangle$, ϕ.

1: Build a resource $\mathsf{R} = \langle X, T, B \rangle$ and a set of intervals I coherent with P and ϕ.
2: Compute $f_{\mathsf{init}}(i)$ for all $i \in [0, \mathsf{len}(p.l)]$.
3: **if** for some $i \in [0, \mathsf{len}(p \cdot l)]$, $f_{\mathsf{init}}(i)$ is undefined **then abort**
4: $h := 1$;
5: $A[h] := \langle f_{\mathsf{init}}(\mathsf{len}(p)), f_{\mathsf{init}}(\mathsf{len}(p)+1), \cdots, f_{\mathsf{init}}(\mathsf{len}(p.l)) \rangle$
6: For each term $\mathtt{t} \in T$, $curr(\mathtt{t}) := A[h][\mathsf{len}(l)+1](\mathtt{t})$
7: **while** $A[h]$ is not final **do**
8: Compute, $\beta_h = \min\{\beta \in \mathbb{N} | i \in [0, \mathsf{len}(l)-1], \mathtt{t} \in T, val_{curr}(l^{\beta} \cdot l[0, i-1])(\mathtt{t}) \neq A[h][i](\mathtt{t})\}$.
9: $h := h+1$
10: $A[h] := \langle \mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_{\mathsf{len}(l)} \rangle$, such that $val_{curr}(l^{\beta_h} \cdot l[0, i-1]) = \mathbf{m}_i$ for all $i \in [0, \mathsf{len}(l)]$.
11: For each term $\mathtt{t} \in T$, $curr(\mathtt{t}) := A[h][\mathsf{len}(l)+1](\mathtt{t})$.
12: **if** there is $i \in [1, \mathsf{len}(l)]$ such that $A[h][i] \not\vdash guard(l(i-1))$ **then abort**
13: **end while**
14: For $j \in [1, h-1]$, $TR[j] := \min(\beta_j + 1, 2td(\phi) + 5)$
15: Check that $\mathsf{proj}((p \otimes \langle f_{\mathsf{init}}(0), \ldots, f_{\mathsf{init}}(\mathsf{len}(p)) \rangle \cdot (l \otimes A[1])^{TR[1]} . (l \otimes A[2])^{TR[2]} \ldots (l \otimes A[h-1])^{TR[h-1]} (l \otimes A[h])^{\omega}), 0 \models_{\mathsf{symb}} \phi$

$\delta_i = \langle q_i, \mathsf{g}_i, \mathbf{u}_i, q_{i+1} \rangle \in \Delta$ for $i \in [0, \mathsf{len}(p)-1]$ and a given tuple of term maps $\mathbf{M} = \langle \mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_{\mathsf{len}(p)} \rangle$, we define $p \otimes \mathbf{M} \stackrel{\text{def}}{=} \delta'_0 \delta'_1 \cdots \delta'_{\mathsf{len}(p)-1}$ where $\delta'_i \stackrel{\text{def}}{=} \langle \langle q_i, \mathbf{m}_i \rangle, \mathsf{g}_i, \mathbf{u}_i, \langle q_{i+1}, \mathbf{m}_{i+1} \rangle \rangle$.

Now, we describe our algorithm that solves MC(PLTL[C], CPS(1)) in polynomial time. Given an initial configuration c_0, it begins by computing the term maps for each position of p and the first iteration of l, using f_{init}. Subsequently, it computes new tuples of term maps $\langle \mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_{\mathsf{len}(l)} \rangle$ for l and the number of iterations $\beta + 1$ of l for which the terms remain in their respective term map from the tuple. We store those tuples made of term maps in an array A and the integer variable $\beta_i + 1$ is used to store the number of iterations corresponding to tuple $A[i]$. If for some position the algorithm reaches some term map that does not satisfy some guard, the procedure aborts (this means that there is no infinite run).

There are only polynomially many entries in A but each number of loop iterations $\beta_i + 1$ can possibly be exponential. The algorithm performs then symbolic model checking over a path schema augmented with the calculated term maps. The augmented path schema is obtained by performing $l \otimes A[i]$ for each i. Finally since the number of times $l \otimes A[i]$ is repeated can be exponential, when checking for the satisfaction of the formula, instead of taking $\beta_i + 1$ times the loop $l \otimes A[i]$, we consider the same run where the loop is taken $\min(\beta_i + 1, 2td(\phi) + 5)$ times. By Theorem 4.1, we have that the two path schemas are equivalent in terms of satisfiability of ϕ. The polynomial-time algorithm is described in Algorithm 2.

It now remains to prove that the algorithm runs in PTIME and is correct.

Lemma 8.6. *Algorithm 2 terminates in polynomial time in the input size.*

Proof. We check that each step of the algorithm can be performed in polynomial time.

- Building a resource and a set of intervals can be done by scanning the input once.
- Since the updates of P is part of the input, we can compute f_{init} for all positions in $p \cdot l$ in polynomial time.
- Computation of *curr* depends on the previous value of *curr* and the coefficients appearing in the guards of P. Hence, it involves addition and multiplication of at most a polynomial number of bits. Thus, this can be performed in polynomial time.
- The maximal possible value for h is bounded by a polynomial given by Lemma 7.8. Indeed, the process described in the while loop is the same as the creation of the set of unfolded path schemas. There is just a major difference: there exists at most one run and hence at most one unfolded path schema.
- Calculation of each value β_h requires computing val_{curr}, which again involves arithmetical operations on polynomially many bits. Thus, this requires polynomial time only.
- Checking $(p \otimes \langle f_{\mathsf{init}}(0), \ldots, f_{\mathsf{init}}(\mathsf{len}(p)) \rangle \cdot (l \otimes A[1])^{TR[1]} (l \otimes A[2])^{TR[2]} \ldots (l \otimes A[h-1])^{TR[h-1]} (l \otimes A[h])^{\omega}, 0 \models_{\mathsf{symb}} \phi$ can be done in polynomial time for the following reasons.
  - By definition of $TR[h]$, size of $(p \otimes \langle f_{\mathsf{init}}(0), \ldots, f_{\mathsf{init}}(\mathsf{len}(p)) \rangle \cdot (l \otimes A[1])^{TR[1]} (l \otimes A[2])^{TR[2]} \ldots (l \otimes A[h-1])^{TR[h-1]} (l \otimes A[h])^{\omega}$ is polynomial in the input size.
  - By [29, Theorem 5.1], $(p \otimes \langle f_{\mathsf{init}}(0), \ldots, f_{\mathsf{init}}(\mathsf{len}(p)) \rangle \cdot (l \otimes A[1])^{TR[1]} (l \otimes A[2])^{TR[2]} \ldots (l \otimes A[h-1])^{TR[h-1]} (l \otimes A[h])^{\omega}, 0 \models_{\mathsf{symb}} \phi$ can be checked in time $\mathcal{O}(\mathsf{size}(\phi)^2 \otimes \mathsf{len}(p \cdot l^{TR[1]} l^{TR[2]} \ldots l^{TR[h-1]} l))$. Indeed, $\models_{\mathsf{symb}}$ is analogous to the satisfaction relation for plain Past LTL. □

Lemma 8.7. *$P, c_0 \models \phi$ iff Algorithm 2 on inputs P, c_0, ϕ has an accepting run.*

Proof. First, we assume that $P, c_0 \models \phi$. We show that there exists a vector of positive integers $\langle \beta_1, \beta_2, \ldots, \beta_h \rangle$ for some $h \in \mathbb{N}$ such that Algorithm 2 has an accepting run. Clearly, the word of transitions taken by a run ρ respecting P and satisfying ϕ is of the form pl^{ω}. Hence, it can be decomposed as the sequence $pl^{\beta_1+1} l^{\beta_2+1} \ldots l^{\beta_h+1} l^{\omega}$, depending on the portion of P that is visited, such that for each consecutive copy of l, the term maps associated with the nodes change. It is easy to see that this decomposition is the same as the one computed by the algorithm. Now, each β_i can be exponential. But due to Lemma 7.5

Classes of Systems	PLTL[∅]	PLTL[C]	Reachability
KPS	NP-complete	–	PTIME
CPS	NP-complete	NP-complete (Theorem 8.4)	NP-complete
KPS(n)	PTIME (Theorem 5.4)	–	PTIME
CPS(n), $n > 1$	open	NP-complete (Lemma 5.6)	open
CPS(1)	PTIME	PTIME	PTIME
FlatKS	NP-complete	–	PTIME
FlatCS	NP-complete	**NP-complete** (Theorem 8.4)	NP-complete

Fig. 7. Summary: computational complexity of the problems MC(L, $\mathcal{C}$).

and Theorem 4.1, we know that $(p \otimes \langle f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p))\rangle \cdot (l \otimes A[1])^{\beta_1+1}(l \otimes A[2])^{\beta_2+1} \ldots (l \otimes A[h-1])^{\beta_{h-1}+1}(l \otimes A[h])^{\omega}, 0 \models_{\text{symb}} \phi$ iff $(p \otimes \langle f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p))\rangle \cdot (l \otimes A[1])^{TR[1]}(l \otimes A[2])^{TR[2]} \ldots (l \otimes A[h-1])^{TR[h-1]}(l \otimes A[h])^{\omega}, 0 \models_{\text{symb}} \phi$. Hence, the algorithm has an accepting run.

Now, suppose that the algorithm has an accepting run on inputs P, c_0 and ϕ. We prove that $P, c_0 \models \phi$. Since the algorithm has an accepting run, let $\beta_1, \beta_2, \cdots, \beta_h$ be the integers it successively computes. Let $w = pl^{\beta_1+1}l^{\beta_2+1} \ldots l^{\beta_h+1}l^{\omega}$ and $\rho = \langle\langle q_0, \mathbf{m}_0\rangle, \mathbf{v}'_{\mathbf{0}}\rangle\langle\langle q_1, \mathbf{m}_1\rangle, \mathbf{v}'_{\mathbf{1}}\rangle\langle\langle q_2, \mathbf{m}_2\rangle, \mathbf{v}'_{\mathbf{2}}\rangle \cdots \in (Q' \times \mathbb{Z}^n)^{\omega}$ be defined as follows:

- for every $i \geq 0$, $q_i \stackrel{\text{def}}{=} \pi_1(source(w(i)))$,
- $\mathbf{v}'_{\mathbf{0}} \stackrel{\text{def}}{=} \mathbf{v_0}$ and,
- for every $i \geq 1$, we have $\mathbf{v}'_{\mathbf{i}} \stackrel{\text{def}}{=} \mathbf{v}'_{\mathbf{i-1}} + update(w(i))$.

By the calculation of β_j for $j \in [1, h]$ in the algorithm, it is easy to check that $\langle q_0, \mathbf{v}'_{\mathbf{0}}\rangle\langle q_1, \mathbf{v}'_{\mathbf{1}}\rangle\langle q_2, \mathbf{v}'_{\mathbf{2}}\rangle \cdots \in (Q \times \mathbb{Z}^n)^{\omega}$ is a run respecting P. Algorithm 2 guarantees that $(p \otimes \langle f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p))\rangle \cdot (l \otimes A[1])^{TR[1]}(l \otimes A[2])^{TR[2]} \ldots (l \otimes A[h-1])^{TR[h-1]}(l \otimes A[h])^{\omega}, 0 \models_{\text{symb}} \phi$. Thus, by Lemma 7.5 and Theorem 4.1, we deduce $\langle q_0, \mathbf{v}'_{\mathbf{0}}\rangle\langle q_1, \mathbf{v}'_{\mathbf{1}}\rangle\langle q_2, \mathbf{v}'_{\mathbf{2}}\rangle \cdots, 0 \models \phi$. □

Consequently, we get the PTIME upper bound.

Proposition 8.8. MC(PLTL[C], CPS(1)) *is in* PTIME.

9. Conclusion

We have investigated the computational complexity of the model-checking problem for flat counter systems with formulae from an enriched version of LTL (with past-time operators and arithmetical constraints on the counters). Our main result is the NP-completeness of the problem MC(PLTL[C], FlatCS), significantly improving the complexity upper bound from [14]. This also improves the results about the effective semilinearity of the reachability relations for such flat counter systems from [9,19]; indeed, our logical dialects allow to specify whether a configuration is reachable. Fig. 7 presents our main results and compares them with the complexity of the reachability problem. Furthermore, our results extend the recent result on the NP-completeness of model-checking flat Kripke structures with LTL from [27] (see also [26]) by adding counters and past-time operators. As far as the proof technique is concerned, the NP upper bound is obtained as a combination of a general stuttering property for LTL with past-time operators (a result extending what is done in [28] with past-time operators) and the use of small integer solutions for quantifier-free Presburger formulae [3]. This latter technique is nowadays widely used to obtain optimal complexity upper bounds for verification problems, see e.g. [25,24]. Herein, our main originality rests on its intricate combination with a very general stuttering principle. There are several related problems which are not addressed in the paper. For instance, the extension of the model-checking problem to full CTL* is known to be decidable [14] and the complexity characterization has been recently solved in [13]. Similarly, the model-checking problem can be extended by considering successively more powerful guards and updates (see the very recent work [5], see also [15]). In this respect, extending our model by allowing quantified Presburger arithmetic formulae immediately results in a reduction from the satisfiability problem for Presburger arithmetic. Extending our model by incorporating updates with finite monoid property in the sense of [19] is another option to consider. In fact, though model-checking is known to be decidable [14], the exact complexity is possibly higher than NP. Another direction for extensions would be to consider richer update functions or guards and to analyze how much our combined proof technique is robust in those cases, for instance by allowing transfer updates.

Appendix A. Proofs of the claims used by the stuttering theorem

A.1. A zone classification for proving (Claim 1)–(Claim 5)

For the proofs of (Claim 1)–(Claim 5), the positions of each word w of the form $w = w_1 u^M w_2 \in \Sigma^{\omega}$ ($\Sigma = 2^{\text{AT}}$, $w_1 \in \Sigma^*$, $u \in \Sigma^+$ and $w_2 \in \Sigma^{\omega}$) with $M > 2N$ are partitioned into five zones (A, B, C, D and E). We also assume that $N \geq 2$. Indeed,

Fig. A.8. The five zones for $w_1(\square\blacksquare)^8 w_2$ with $N=3$ and $u=\square\blacksquare$.

given that $\langle w,i\rangle \approx_N \langle w',i'\rangle$, we shall proceed by a case analysis on the positions i and i' depending on which zones i and i' belong to (assuming that $w' = w_1 u^{M'} w_2 \in \Sigma^\omega$ with $M' > 2N$). The definition of zones is illustrated on Fig. A.8 and here is the formal characterization:

- Zone A corresponds to the set of positions $i \in \mathbb{N}$ such that $0 \leq i < \mathrm{len}(w_1) + (N-1)\cdot \mathrm{len}(u)$.
- Zone B corresponds to the set of positions $i \in \mathbb{N}$ such that $\mathrm{len}(w_1) + (N-1)\cdot \mathrm{len}(u) \leq i < \mathrm{len}(w_1) + N \cdot \mathrm{len}(u)$.
- Zone C corresponds to the set of positions $i \in \mathbb{N}$ such that $\mathrm{len}(w_1) + N\cdot \mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-N)\cdot \mathrm{len}(u)$.
- Zone D corresponds to the set of positions $i \in \mathbb{N}$ such that $\mathrm{len}(w_1) + (M-N)\cdot \mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-(N-1))\cdot \mathrm{len}(u)$.
- Zone E corresponds to the set of positions $i \in \mathbb{N}$ such that $\mathrm{len}(w_1) + (M-(N-1))\cdot \mathrm{len}(u) \leq i$.

Note that the definition of zones depends on the value N (taken from $\approx_N$) and also on u, w_1 and w_2. In the sequel, we may index the zones by N (providing A_N, B_N, etc.) when it is useful to make explicit from which relation $\approx_N$ the definition of zones is made. Moreover, we may use a prime symbol (providing A'_N, B'_N, etc.) to refer to zones for the infinite word w'. So, the relation $\approx_N$ can be redefined as follows when $M, M' > 2N$: $\langle w,i\rangle \approx_N \langle w',i'\rangle \stackrel{\text{def}}{\Leftrightarrow} (M \approx_{2N} M')$ and one of the conditions holds true:

1. $i = i'$ and either ($i \in \mathrm{A}_N$ and $i' \in \mathrm{A}'_N$) or ($i \in \mathrm{B}_N$ and $i' \in \mathrm{B}'_N$).
2. $(i - i') = (M - M')\mathrm{len}(u)$ and either ($i \in \mathrm{D}_N$ and $i' \in \mathrm{D}'_N$) or ($i \in \mathrm{E}_N$ and $i' \in \mathrm{E}'_N$).
3. $i \in \mathrm{C}_N$, $i' \in \mathrm{C}'_N$ and $|i - i'| = 0 \bmod \mathrm{len}(u)$.

A.2. Proof of (Claim 1)

Let us recall what is (Claim 1). Let $w = w_1 u^M w_2$, $w' = w_1 u^{M'} w_2 \in \Sigma^\omega$, $i, i' \in \mathbb{N}$ and $N \geq 2$ such that $M, M' \geq 2N+1$ and $\langle w,i\rangle \approx_N \langle w',i'\rangle$.

(Claim 1) $\langle w,i\rangle \approx_{N-1} \langle w',i'\rangle$; $w(i) = w'(i')$.

Proof. We first prove that $\langle w,i\rangle \approx_{N-1} \langle w',i'\rangle$. Without any loss of generality, we can assume that $M \geq M'$. Since $N > N-1$, it is obvious that $M \approx_{2(N-1)} M'$.

- If $i < \mathrm{len}(w_1) + (N-1)\cdot \mathrm{len}(u)$ **[i is in Zone $\mathbf{A}_N$]**, then $i = i'$. Hence either ($i \in \mathrm{A}_{N-1}$, $i' \in \mathrm{A}'_{N-1}$ and $i = i'$) or ($i \in \mathrm{B}_{N-1}$, $i' \in \mathrm{B}'_{N-1}$ and $i = i'$). Hence, $\langle w,i\rangle \approx_{N-1} \langle w',i'\rangle$.
- If $i \geq \mathrm{len}(w_1) + (M-(N-1))\cdot \mathrm{len}(u)$ **[i is in zone $\mathbf{E}_N$]** then $i = i' + (M-M')\cdot \mathrm{len}(u)$ and $i' \geq \mathrm{len}(w_1) + (M'-(N-1))\cdot \mathrm{len}(u)$ **[i' is in zone $\mathbf{E}'_N$]**. So, either (i is in zone E_{N-1} and i' is in zone E'_{N-1}) or (i is in zone D_{N-1} and i' is in zone D'_{N-1}). Since $i = i' + (M-M')\cdot \mathrm{len}(u)$, we conclude that $\langle w,i\rangle \approx_{N-1} \langle w',i'\rangle$.
- If $\mathrm{len}(w_1) + (N-1)\cdot \mathrm{len}(u) \leq i < \mathrm{len}(w_1) + N\cdot \mathrm{len}(u)$ **[i is in Zone $\mathbf{B}_N$]** then $i = i'$. Hence, $i \in \mathrm{C}_{N-1}$, $i' \in \mathrm{C}'_{N-1}$ and $|i - i'| = 0 \bmod \mathrm{len}(u)$. Hence, $\langle w,i\rangle \approx_{N-1} \langle w',i'\rangle$.
- If $\mathrm{len}(w_1) + N\cdot \mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-N)\cdot \mathrm{len}(u)$ **[i in Zone $\mathbf{C}_N$]**, then $\mathrm{len}(w_1) + N\cdot \mathrm{len}(u) \leq i' < \mathrm{len}(w_1) + (M'-N)\cdot \mathrm{len}(u)$ **[i' is in Zone $\mathbf{C}'_N$]** and $|i - i'| = 0 \bmod \mathrm{len}(u)$. Consequently, i is in Zone C_{N-1}, i' is in Zone C'_{N-1} and $|i - i'| = 0 \bmod \mathrm{len}(u)$. This entails that $\langle w,i\rangle \approx_{N-1} \langle w',i'\rangle$.
- If $\mathrm{len}(w_1) + (M-N)\cdot \mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-(N-1))\cdot \mathrm{len}(u)$ **[i in Zone $\mathbf{D}_N$]**, then i' is in Zone D'_N and $i = i' + (M-M')\cdot \mathrm{len}(u)$. Consequently, i is in Zone C_{N-1}, i' is in Zone C'_{N-1} and $|i - i'| = 0 \bmod \mathrm{len}(u)$. This also entails that $\langle w,i\rangle \approx_{N-1} \langle w',i'\rangle$.

As far as the second property is concerned, it is easy to conclude that $w(i) = w'(i')$, because either i and i' correspond to the same position relatively to the words w_1 or w_2, or they are respectively in the Zone C and they correspond to the same position of u. Indeed, in this latter case, their difference is such that $|i - i'| = 0 \bmod \mathrm{len}(u)$. □

A.3. Proof of (Claim 2)

Let us recall what is (Claim 2). Let $w = w_1 u^M w_2$, $w' = w_1 u^{M'} w_2 \in \Sigma^\omega$, $i, i' \in \mathbb{N}$ and $N \geq 2$ such that $M, M' \geq 2N+1$ and $\langle w,i\rangle \approx_N \langle w',i'\rangle$.

(Claim 2) $i, i' > 0$ implies $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$.

Proof. Without any loss of generality, we can assume that $M \geq M'$. Since $N > N-1$, it is obvious that $M \approx_{2(N-1)} M'$.

- If $i < \mathrm{len}(w_1) + (N-1)\cdot\mathrm{len}(u)$ **[i is in Zone $\mathbf{A}_N$]**, then $i = i'$. Hence, $i-1 \in \mathrm{A}_{N-1}$, $i'-1 \in \mathrm{A}'_{N-1}$ and $i-1 = i'-1$. So, $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$.
- If $i \geq \mathrm{len}(w_1) + (M-(N-1))\cdot\mathrm{len}(u)$ **[i is in zone $\mathbf{E}_N$]** then $i = i' + (M-M')\cdot\mathrm{len}(u)$ and $i' \geq \mathrm{len}(w_1) + (M'-(N-1))\cdot\mathrm{len}(u)$ **[i' is in zone $\mathbf{E}'_N$]**. So, either ($i-1$ is in zone E_{N-1}, $i'-1$ is in zone E'_{N-1} and $i-1 = i'-1+(M-M')\cdot\mathrm{len}(u)$) or ($i-1$ is in zone D_{N-1} and $i'-1$ is in zone D'_{N-1} and $i-1 = i'-1+(M-M')\cdot\mathrm{len}(u)$) or ($i-1$ is in zone C_{N-1} and $i'-1$ is in zone C'_{N-1} and $|(i-1)-(i'-1)| = 0 \bmod \mathrm{len}(u)$). We conclude that $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$.
- If $\mathrm{len}(w_1)+(N-1)\cdot\mathrm{len}(u) \leq i < \mathrm{len}(w_1)+N\cdot\mathrm{len}(u)$ **[i is in Zone $\mathbf{B}_N$]** then $i = i'$. Hence, either ($i-1 \in \mathrm{C}_{N-1}$, $i'-1 \in \mathrm{C}'_{N-1}$ and $|(i-1)-(i'-1)| = 0 \bmod \mathrm{len}(u)$) or ($i-1 \in \mathrm{B}_{N-1}$, $i'-1 \in \mathrm{B}'_{N-1}$ and $i-1 = i'-1$). Hence, $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$.
- If $\mathrm{len}(w_1) + N\cdot\mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-N)\cdot\mathrm{len}(u)$ **[i in Zone $\mathbf{C}_N$]**, then $\mathrm{len}(w_1) + N\cdot\mathrm{len}(u) \leq i' < \mathrm{len}(w_1) + (M'-N)\cdot\mathrm{len}(u)$ **[i' is in Zone $\mathbf{C}'_N$]** and $|i-i'| = 0 \bmod \mathrm{len}(u)$. Consequently, $i-1$ is in Zone C_{N-1}, $i'-1$ is in Zone C'_{N-1} and $|(i-1)-(i'-1)| = 0 \bmod \mathrm{len}(u)$. This entails that $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$.
- If $\mathrm{len}(w_1) + (M-N)\cdot\mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-(N-1))\cdot\mathrm{len}(u)$ **[i in Zone $\mathbf{D}_N$]**, then i' is in Zone D'_N and $i = i' + (M-M')\cdot\mathrm{len}(u)$. Consequently, $i-1$ is in Zone C_{N-1}, $i'-1$ is in Zone C'_{N-1} and $|(i-1)-(i'-1)| = 0 \bmod \mathrm{len}(u)$. This entails that $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$. □

A.4. Proof of (Claim 3)

Let us recall what is (Claim 3). Let $w = w_1 u^M w_2, w' = w_1 u^{M'} w_2 \in \Sigma^\omega$, $i, i' \in \mathbb{N}$ and $N \geq 2$ such that $M, M' \geq 2N+1$ and $\langle w, i\rangle \approx_N \langle w', i'\rangle$.

(Claim 3) $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$.

Proof. The proof is similar to the proof for (Claim 2). Nevertheless, full proof is provided below for the sake of completeness. Without any loss of generality, we can assume that $M \geq M'$. Since $N > N-1$, it is again obvious that $M \approx_{2(N-1)} M'$.

- If $i < \mathrm{len}(w_1) + (N-1)\cdot\mathrm{len}(u)$ **[i is Zone $\mathbf{A}_N$]**, then $i = i'$. Hence either ($i+1 \in \mathrm{A}_{N-1}$, $i'+1 \in \mathrm{A}'_{N-1}$ and $i+1 = i'+1$) or ($i+1 \in \mathrm{B}_{N-1}$, $i'+1 \in \mathrm{B}'_{N-1}$ and $i+1 = i'+1$) or ($i+1 \in \mathrm{C}_{N-1}$, $i'+1 \in \mathrm{C}'_{N-1}$ and $i-i' = 0$). Hence, $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$.
- If $i \geq \mathrm{len}(w_1) + (M-(N-1))\cdot\mathrm{len}(u)$ **[i is in zone $\mathbf{E}_N$]** then $i = i' + (M-M')\cdot\mathrm{len}(u)$ and $i' \geq \mathrm{len}(w_1) + (M'-(N-1))\cdot\mathrm{len}(u)$ **[i' is in zone $\mathbf{E}'_N$]**. So, either ($i+1$ is in zone E_{N-1} and $i'+1$ is in zone E'_{N-1}) or ($i+1$ is in zone D_{N-1} and $i'+1$ is in zone D'_{N-1}). Since $i+1 = i'+1+(M-M')\cdot\mathrm{len}(u)$, we conclude that $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$.
- If $\mathrm{len}(w_1) + (N-1)\cdot\mathrm{len}(u) \leq i < \mathrm{len}(w_1) + N\cdot\mathrm{len}(u)$ **[i is in Zone $\mathbf{B}_N$]** then $i = i'$. Hence, $i+1 \in \mathrm{C}_{N-1}$, $i'+1 \in \mathrm{C}'_{N-1}$ and $|(i+1)-(i'+1)| = 0 \bmod \mathrm{len}(u)$. Hence, $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$.
- If $\mathrm{len}(w_1) + N\cdot\mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-N)\cdot\mathrm{len}(u)$ **[i in Zone $\mathbf{C}_N$]**, then $\mathrm{len}(w_1) + N\cdot\mathrm{len}(u) \leq i' < \mathrm{len}(w_1) + (M'-N)\cdot\mathrm{len}(u)$ **[i' is in Zone $\mathbf{C}'_N$]** and $|i-i'| = 0 \bmod \mathrm{len}(u)$. Consequently, $i+1$ is in Zone C_{N-1}, $i'+1$ is in Zone C'_{N-1} and $|(i+1)-(i'+1)| = 0 \bmod \mathrm{len}(u)$. This entails that $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$.
- If $\mathrm{len}(w_1) + (M-N)\cdot\mathrm{len}(u) \leq i < \mathrm{len}(w_1) + (M-(N-1))\cdot\mathrm{len}(u)$ **[i in Zone $\mathbf{D}_N$]**, then i' is in Zone D'_N and $i = i' + (M-M')\cdot\mathrm{len}(u)$. Consequently, either ($i+1$ is in Zone C_{N-1}, $i'+1$ is in Zone C'_{N-1} and $|(i+1)-(i'+1)| = 0 \bmod \mathrm{len}(u)$) or ($i+1$ is in Zone D_{N-1}, $i'+1$ is in Zone D'_{N-1} and $i+1 = i'+1+(M-M')\cdot\mathrm{len}(u)$). This also entails that $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$. □

A.5. Proof of (Claim 4)

Before providing the detailed proof, we give a concrete example on Fig. A.9. On this example, we assume that the top word w and the bottom word w' and their respective positions i and i' are such that $\langle w, i\rangle \approx_3 \langle w', i'\rangle$. We want to illustrate (Claim 4) and for this matter, we choose a position j in w. Now observe that according to the zone classification, j is in the Zone C of the word w and furthermore it is not possible to find a $j' > i'$ in the Zone C of the word w' such that j and j' points on the same position of the word u. That is why we need to consider at this stage not the relation $\approx_3$ but instead $\approx_2$. In fact, as shown at the bottom of Fig. A.9, we can find for j, a position j' in w' such that $\langle w, j\rangle \approx_2 \langle w', j'\rangle$ (take $j = j'$) and this figure also shows that for all $i' \leq k \leq j'$, $\langle w, k\rangle \approx_2 \langle w', k\rangle$.

Let us recall what is (Claim 4). Let $w = w_1 u^M w_2, w' = w_1 u^{M'} w_2 \in \Sigma^\omega$, $i, i' \in \mathbb{N}$ and $N \geq 2$ such that $M, M' \geq 2N+1$ and $\langle w, i\rangle \approx_N \langle w', i'\rangle$.

(Claim 4) For all $j \geq i$, there is $j' \geq i'$ such that $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$ and for all $k' \in [i', j'-1]$, there is $k \in [i, j-1]$ such that $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.

Proof. We proceed by a case analysis on the positions i and j:

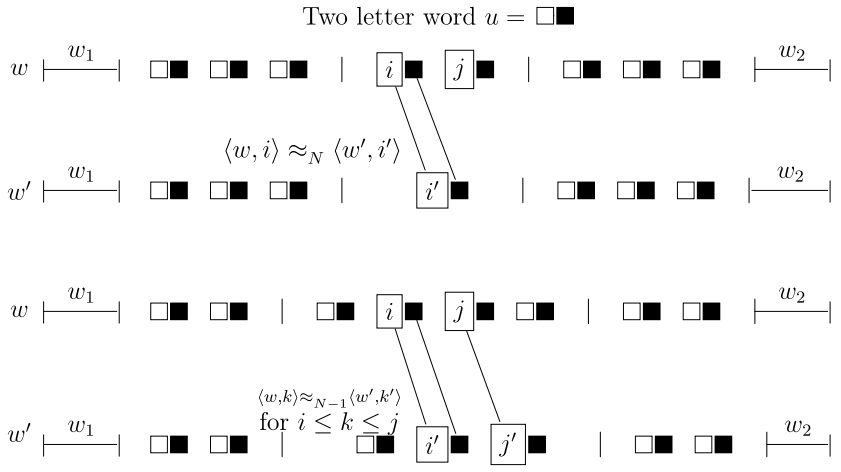


Fig. A.9. Relation between $\approx_N$ and $\approx_{N-1}$.

- If $i \geq \text{len}(w_1) + (M-N)\text{len}(u)$ **[i is in Zone $\mathbf{D}_N$ or $\mathbf{E}_N$]** then $j \geq \text{len}(w_1) + (M-N)\text{len}(u)$ **[j is in Zone $\mathbf{D}_N$ or $\mathbf{E}_N$]** and $i' \geq \text{len}(w_1) + (M'-N)\text{len}(u)$ **[i' is in Zone D_N or E_N]** and $i = i' + (M-M')\text{len}(u)$. We define $j' = j - (M-M')\text{len}(u)$. Then it is clear that $j' \geq i'$ and $\langle w, j\rangle \approx_N \langle w', j'\rangle$. By (Claim 1), we get $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Let $k' \in [i', j'-1]$ and let $k = k' + (M-M')\text{len}(u)$, then we have that $k \in [i, j-1]$ and also $\langle w, k\rangle \approx_N \langle w', k'\rangle$, hence by (Claim 1), $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
- If $i < \text{len}(w_1) + N\text{len}(u)$ **[i is in Zone $\mathbf{A}_N$ or $\mathbf{B}_N$]** then $i' < \text{len}(w_1) + N\text{len}(u)$ **[i' is in Zone $\mathbf{A}_N$ or $\mathbf{B}_N$]**, $i = i'$ and we have the following possibilities for the position $j \geq i$:
  - If $j < \text{len}(w_1) + N\text{len}(u)$ **[j is in Zone $\mathbf{A}_N$ or $\mathbf{B}_N$]**, then let $j' = j$. Consequently we have $\langle w, j\rangle \approx_N \langle w', j'\rangle$ and by (Claim 1) we get $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Let $k' \in [i', j'-1]$ and $k = k'$. Then we have that $k \in [i, j-1]$ and also $\langle w, k\rangle \approx_N \langle w', k'\rangle$ and by (Claim 1), $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
  - If $\text{len}(w_1) + N\text{len}(u) \leq j < \text{len}(w_1) + (M-N)\text{len}(u)$ **[j is in Zone $\mathbf{C}_N$]**, then let $\ell = (j - (\text{len}(w_1) + N\ \text{len}(u))) \bmod \text{len}(u)$ (ℓ is the relative position of j in the word u it belongs to). Consequently $0 \leq \ell < \text{len}(u)$. Let $j' = \text{len}(w_1) + N\text{len}(u) + \ell$ (we choose j' at the same relative position of j in the first word u of the Zone C'_N). Then $\text{len}(w_1) + N\text{len}(u) \leq j' < \text{len}(w_1) + (M'-N)\text{len}(u)$ **[j' is in Zone $\mathbf{C}'_N$]** (because $(M'-N) > 0$) and $|j - j'| = 0 \bmod \text{len}(u)$. We deduce that $\langle w, j\rangle \approx_N \langle w', j'\rangle$ and by (Claim 1) we get $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Then let $k' \in [i', j'-1]$ and let $k = k'$. Then we have that $k \in [i, j-1]$. Furthermore, if $k' < \text{len}(w_1) + N\text{len}(u)$ **[k' is in Zone $\mathbf{A}'_N$ or $\mathbf{B}'_N$]** we obtain $\langle w, k\rangle \approx_N \langle w', k'\rangle$ and by (Claim 1), $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. Moreover, if $\text{len}(w_1) + N\text{len}(u) \leq k'$ **[k' is in Zone $\mathbf{C}'_N$]** then k is in Zone C_N and $|k - k'| = 0 \bmod \text{len}(u)$ since $k = k'$. So, $\langle w, k\rangle \approx_N \langle w', k'\rangle$ and by (Claim 1), $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
  - If $\text{len}(w_1) + (M-N)\text{len}(u) \leq j$ **[j is in Zone $\mathbf{E}_N$ or $\mathbf{D}_N$]**, let $j' = j - (M-M')\text{len}(u)$. Then, we have $\text{len}(w_1) + (M'-N)\text{len}(u) \leq j'$ **[j' is in Zone $\mathbf{D}'_N$ or $\mathbf{E}'_N$]** and we deduce that $\langle w, j\rangle \approx_N \langle w', j'\rangle$ and by (Claim 1) we get $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Then let $k' \in [i', j'-1]$. If $k' < \text{len}(w_1) + N\text{len}(u)$ **[k' is in Zone $\mathbf{A}'_N$ or $\mathbf{B}'_N$]**, for $k = k'$, we obtain $\langle w, k\rangle \approx_N \langle w', k'\rangle$ and by (Claim 1), $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. For the case when $k' \geq \text{len}(w_1) + (M'-N)\text{len}(u)$ **[k' is in Zone $\mathbf{D}_N$ or $\mathbf{E}_N$]**, we choose $k = k' + (M-M')\text{len}(u)$ and here also we deduce $\langle w, k\rangle \approx_N \langle w', k'\rangle$ and by (Claim 1), $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. If on the other hand, $w_1 + N\text{len}(u) \leq k' < \text{len}(w_1) + (M'-N)\text{len}(u)$ **[k' is in Zone $\mathbf{C}'_N$]**, let $\ell = (k' - (\text{len}(w_1) + N\text{len}(u))) \bmod \text{len}(u)$ (ℓ is the relative position of k' in the word u it belongs to) and let $k = \text{len}(w_1) + N\text{len}(u) + \ell$ (k is placed at the same relative position of k' in the first word u of the Zone C_N). Then we have $w_1 + N\text{len}(u) \leq k < \text{len}(w_1) + (M-N)\text{len}(u)$ and $|k - k'| = 0 \bmod \text{len}(u)$ which allows to deduce that $\langle w, k\rangle \approx_N \langle w', k'\rangle$ and by (Claim 1), $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
- If $\text{len}(w_1) + N\text{len}(u) \leq i < \text{len}(w_1) + (M-N)\text{len}(u)$ **[i is in Zone $\mathbf{C}_N$]** then $\text{len}(w_1) + N\text{len}(u) \leq i' < \text{len}(w_1) + (M'-N)\text{len}(u)$ **[i' is in Zone $\mathbf{C}'_N$]** and $|i - i'| = 0 \bmod \text{len}(u)$. Let $\ell = (i - (\text{len}(w_1) + N\text{len}(u))) \bmod \text{len}(u)$ (the relative position of i in the word u). We have the following possibilities for the position $j \geq i$:
  - If $j - i < \text{len}(u) - \ell + \text{len}(u)$ (j is either in the same word u as i or in the next word u), then $j < \text{len}(w_1) + (M - (N-1))\text{len}(u)$ **[j is in Zone $\mathbf{C}_N$ or $\mathbf{D}_N$]**. We define $j' = i' + (j - i)$ and we have that $\text{len}(w_1) + N\text{len}(u) \leq j' < \text{len}(w_1) + (M' - (N-1))\text{len}(u)$ **[j' is in Zone $\mathbf{C}'_N$ or $\mathbf{D}'_N$]** and since $|i - i'| = 0 \bmod \text{len}(u)$, we deduce $|j - j'| = 0 \bmod \text{len}(u)$. From this, we obtain $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Let $k' \in [i', j'-1]$ and $k = i + k' - i'$. We have then that $k \in [i, j-1]$ and $\text{len}(w_1) + N\text{len}(u) \leq k' < \text{len}(w_1) + (M' - (N-1))\text{len}(u)$ and $\text{len}(w_1) + N\text{len}(u) \leq k < \text{len}(w_1) + (M - (N-1))\text{len}(u)$. Since $|i - i'| = 0 \bmod \text{len}(u)$, we also have $|k - k'| = 0 \bmod \text{len}(u)$. Hence, $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
  - If $j - i \geq \text{len}(u) - \ell + \text{len}(u)$ (j is neither in the same word u as i nor in the next word u) and $j \geq \text{len}(w_1) + (M-N)\text{len}(u)$ **[j is in Zone $\mathbf{E}_N$ or $\mathbf{D}_N$]**. Let $j' = j - (M-M')\text{len}(u)$ then $j' \geq \text{len}(w_1) + (M'-N)\text{len}(u)$ **[j' is in Zone E'_N or D'_N]** and consequently $\langle w, j\rangle \approx_N \langle w', j'\rangle$ and by (Claim 1) we get $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Then let $k' \in [i', j'-1]$. If $k' \geq \text{len}(w_1) + (M'-N)\text{len}(u)$ **[k' is in Zone $\mathbf{D}'_N$ or $\mathbf{E}'_N$]**, then let $k = k' + (M-M')\text{len}(u)$; we have in this case that $k \geq \text{len}(w_1) + (M-N)\text{len}(u)$ and this allows us to deduce that $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. Now assume $k' < \text{len}(w_1) +$

$(M'-N)\mathrm{len}(u)$ **[k' is in Zone C'_N]** and $k'-i' < \mathrm{len}(u)-\ell$ (k' and i' are in the same word u), then let $k = i+k'-i'$. In this case we have $k < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[k is in Zone C_N]** and since $|i-i'| = 0 \bmod \mathrm{len}(u)$, we also have $|k-k'| = 0 \bmod \mathrm{len}(u)$, whence $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$. Now assume $k' < \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)$ **[k' is in Zone C'_N]** and $k'-i' \geq \mathrm{len}(u)-\ell$ (k' and i' are not in the same word u). We denote by $\ell' = (k'-(\mathrm{len}(w_1)+N\mathrm{len}(u))) \bmod \mathrm{len}(u)$ the relative position of k' in u and let $k = i+(\mathrm{len}(u)-\ell)+\ell'$ (k and k' occur in the same position in u but k occurs in the word u just after the word u in which i belongs to) Then $k \in [i, j-1]$ (because $\ell' < \mathrm{len}(u)$ and $j-i \geq \mathrm{len}(u)-\ell+\mathrm{len}(u)$) and $k < \mathrm{len}(w_1)+(M-(N-1))\mathrm{len}(u)$ (because $i+(\mathrm{len}(u)-\ell) < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ and $\ell' < \mathrm{len}(u)$) and $|k-k'| = 0 \bmod \mathrm{len}(u)$ (k and k' are both pointing on the ℓ'-th position in word u). This allows us to deduce that $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$.

- If $j-i \geq \mathrm{len}(u)-\ell+\mathrm{len}(u)$ (j is neither in the same word u as i nor in the next word u) and $j < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[j is in Zone C_N]**. Then let $\ell' = (j-(\mathrm{len}(w_1)+N\mathrm{len}(u))) \bmod \mathrm{len}(u)$ the relative position of j in u. We choose $j' = i'+(\mathrm{len}(u)-\ell)+\ell'$ (j and j' occur in the same position in u but j' occurs in the word u just after the word u in which i' belongs to) We have then that $j' < \mathrm{len}(w_1)+(M'-(N-1))\mathrm{len}(u)$ **[j' is in Zone C'_N or D'_N]** (because $i'+(\mathrm{len}(u)-\ell) < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ and $\ell' < \mathrm{len}(u)$) and $|j-j'| = 0 \bmod \mathrm{len}(u)$ (j and j' are both pointing on the ℓ'-th position in word u), hence $\langle w,j\rangle \approx_{N-1} \langle w',j'\rangle$. Let $k' \in [i', j'-1]$. If $k'-i' < \mathrm{len}(u)-\ell$ (k' and i' are in the same word u), then let $k = i+k'-i'$. In this case we have $k < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[k is in Zone C_N]** and since $|i-i'| = 0 \bmod \mathrm{len}(u)$, we also have $|k-k'| = 0 \bmod \mathrm{len}(u)$, hence $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$. If $k'-i' \geq \mathrm{len}(u)-\ell$ (k' and i' are not in the same word u), then $j'-k' < \ell'$ and let $k = j-j'-k'$. In this case we have $k < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[k is in Zone C_N]** and since $|j-j'| = 0 \bmod \mathrm{len}(u)$, we also have $|k-k'| = 0 \bmod \mathrm{len}(u)$, hence $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$. □

A.6. Proof of (Claim 5)

Let us recall what is (Claim 5). Let $w = w_1u^Mw_2$, $w' = w_1u^{M'}w_2 \in \Sigma^\omega$, $i, i' \in \mathbb{N}$ and $N \geq 2$ such that $M, M' \geq 2N+1$ and $\langle w,i\rangle \approx_N \langle w',i'\rangle$.

(Claim 5) for all $j \leq i$, there is $j' \leq i'$ such that $\langle w,j\rangle \approx_{N-1} \langle w',j'\rangle$ and for all $k' \in [j'-1, i']$, there is $k \in [j-1, i]$ such that $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$.

Proof. The proof is similar to the proof for (Claim 4) by looking backward instead of looking forward (still there are slight differences because past is finite). Nevertheless, full proof is provided below for the sake of completeness. We proceed by a case analysis on the positions i and j.

- If $i < \mathrm{len}(w_1)+N\mathrm{len}(u)$ **[i is in Zone A_N or B_N]** then $j < \mathrm{len}(w_1)+N\mathrm{len}(u)$ **[j is in Zone A_N or B_N]**, $i' < \mathrm{len}(w_1)+N\mathrm{len}(u)$ **[i' is in Zone A'_N or B'_N]** and $i = i'$. We define $j' = j$. Then it is clear that $j' < i'$ and $\langle w,j\rangle \approx_N \langle w',j'\rangle$. By (Claim 1), we get $\langle w,j\rangle \approx_{N-1} \langle w',j'\rangle$. Let $k' \in [j'-1, i']$ and let $k = k'$, then we have that $k \in [j-1, i]$ and also $\langle w,k\rangle \approx_N \langle w',k'\rangle$, hence by (Claim 1), $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$.
- If $i \geq \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[i is Zone D_N or E_N]** then $i' \geq \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)$ **[i' is in Zone D'_N or E'_N]** and $i = i'+(M-M')\mathrm{len}(u)$ and we have the following possibilities for the position $j \leq i$:
  - If $j \geq \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[j is in Zone D_N or E_N]**, then let $j' = j-(M-M')\mathrm{len}(u)$. Consequently, we have $\langle w,j\rangle \approx_N \langle w',j'\rangle$ and by (Claim 1) we get $\langle w,j\rangle \approx_{N-1} \langle w',j'\rangle$. Let $k' \in [j'-1, i']$ and $k = k'+(M-M')\mathrm{len}(u)$. Then we have that $k \in [j-1, i]$ and also $\langle w,k\rangle \approx_N \langle w',k'\rangle$. By (Claim 1), $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$.
  - If $\mathrm{len}(w_1)+N\mathrm{len}(u) \leq j < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[j is in Zone C_N]**, then let $\ell = (j-(\mathrm{len}(w_1)+N\ \mathrm{len}(u))) \bmod \mathrm{len}(u)$ (ℓ is the relative position of j in the word u it belongs to). Consequently $0 \leq \ell < \mathrm{len}(u)$. Let $j' = \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)-(\mathrm{len}(u)-\ell)$ (j' is at the same position as j in the last word u of the Zone C'_N). Then $\mathrm{len}(w_1)+N\mathrm{len}(u) \leq j' < \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)$ **[j' is in Zone C'_N]** (because $(M' \geq 2N+1)$ and $|j-j'| = 0 \bmod \mathrm{len}(u)$, they are at the same position in the word u). We deduce that $\langle w,j\rangle \approx_N \langle w',j'\rangle$ and by (Claim 1) we get $\langle w,j\rangle \approx_{N-1} \langle w',j'\rangle$. Then let $k' \in [j'-1, i']$ and let $k = k'+(M-M')\mathrm{len}(u)$. Then we have that $k \in [j-1, i]$. Furthermore, if $k' \geq \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)$ **[k' is in Zone D'_N or E'_N]** then $k \geq \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[k is in Zone D_N or E_N]** and we obtain $\langle w,k\rangle \approx_N \langle w',k'\rangle$ and by (Claim 1), $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$. Moreover, if $k' < \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)$ then necessarily $\mathrm{len}(w_1)+N\mathrm{len}(u) \leq k'$ **[k' is in Zone C'_N]** (because $j' < k'$) and $|k-k'| = 0 \bmod \mathrm{len}(u)$ (because $k = k'+(M-M')\mathrm{len}(u)$). Whence, k is in Zone C_N and $\langle w,k\rangle \approx_N \langle w',k'\rangle$. By (Claim 1), we obtain $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$.
  - If $j < \mathrm{len}(w_1)+N\mathrm{len}(u)$ **[j is in Zone A_N or B_N]**, let $j' = j$. We have then $j' < \mathrm{len}(w_1)+N\mathrm{len}(u)$ **[j' is in Zone A'_N or B'_N]**. We deduce that $\langle w,j\rangle \approx_N \langle w',j'\rangle$ and by (Claim 1) we get $\langle w,j\rangle \approx_{N-1} \langle w',j'\rangle$. Then let $k' \in [j'-1, i']$. If $k' < \mathrm{len}(w_1)+N\mathrm{len}(u)$ **[k' is in Zone A'_N]**, for $k = k'$, we obtain $\langle w,k\rangle \approx_N \langle w',k'\rangle$ and by (Claim 1), $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$. If $k' \geq \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)$ **[k' is in Zone D'_N or E'_N]**, we choose $k = k'+(M-M')\mathrm{len}(u)$ and here also we deduce $\langle w,k\rangle \approx_N \langle w',k'\rangle$ and by (Claim 1), $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$. If $w_1+N\mathrm{len}(u) \leq k' < \mathrm{len}(w_1)+(M'-N)\mathrm{len}(u)$ **[k' is in Zone C'_N]**, let $\ell = (k'-(\mathrm{len}(w_1)+N\mathrm{len}(u))) \bmod \mathrm{len}(u)$ (ℓ is the relative position of k' in the word u it belongs to) and let $k = \mathrm{len}(w_1)+N\mathrm{len}(u)+\ell$ (k is at the same position of k' in the first word of the Zone C_N). Then we have $w_1+N\mathrm{len}(u) \leq k < \mathrm{len}(w_1)+(M-N)\mathrm{len}(u)$ **[k is in the Zone C_N]** and $|k-k'| = 0 \bmod \mathrm{len}(u)$ which allows to deduce that $\langle w,k\rangle \approx_N \langle w',k'\rangle$ and by (Claim 1), $\langle w,k\rangle \approx_{N-1} \langle w',k'\rangle$.

- If $\text{len}(w_1) + N\text{len}(u) \leq i < \text{len}(w_1) + (M-N)\text{len}(u)$ **[i is in Zone $\mathbf{C}_N$]** then $\text{len}(w_1) + N\text{len}(u) \leq i' < \text{len}(w_1) + (M'-N)\text{len}(u)$ **[i' in Zone $\mathbf{C}'_N$]** and $|i-i'| = 0 \bmod \text{len}(u)$. Let $\ell = (i - (\text{len}(w_1) + N\text{len}(u))) \bmod \text{len}(u)$ (ℓ is the relative position of i in the word u it belongs to). We have the following possibilities for the position $j \leq i$:
  - If $i - j < \ell + \text{len}(u)$ (j is in the same word u as i or in the previous word u) then $j \geq \text{len}(w_1) + (N-1)\text{len}(u)$ **[j is in Zone $\mathbf{B}_N$ or $\mathbf{C}_N$]**. We define $j' = i' - (i-j)$ and we have that $\text{len}(w_1) + (N-1)\text{len}(u) \leq j' < \text{len}(w_1) + (M'-N)\text{len}(u)$ **[j' is in Zone $\mathbf{B}'_N$ or $\mathbf{C}'_N$]** and since $|i-i'| = 0 \bmod \text{len}(u)$, we deduce $|j-j'| = 0 \bmod \text{len}(u)$. From this, we obtain $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Let $k' \in [j'-1, i']$ and $k = i - (i'-k')$. We have then that $k \in [j-1, i]$ and $\text{len}(w_1) + (N-1)\text{len}(u) \leq k' < \text{len}(w_1) + (M'-N)\text{len}(u)$ **[k' is in Zone $\mathbf{B}'_N$ or $\mathbf{C}'_N$]** and $\text{len}(w_1) + (N-1)\text{len}(u) \leq k < \text{len}(w_1) + (M-N)\text{len}(u)$ **[k is in Zone $\mathbf{B}_N$ or $\mathbf{C}_N$]** and since $|i-i'| = 0 \bmod \text{len}(u)$, we also have $|k-k'| = 0 \bmod \text{len}(u)$. Consequently $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
  - If $i - j \geq \ell + \text{len}(u)$ (j is neither in the same word u as i nor in the previous word u) and $j < \text{len}(w_1) + N\text{len}(u)$ **[j is in zone $\mathbf{A}_N$ or $\mathbf{B}_N$]**. Let $j' = j$. So, $j' < \text{len}(w_1) + N\text{len}(u)$ and $\langle w, j\rangle \approx_N \langle w', j'\rangle$. By using (Claim 1) we get $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Then let $k' \in [j'-1, i']$. If $k' < \text{len}(w_1) + N\text{len}(u)$ **[k' is in Zone $\mathbf{A}'_N$ or $\mathbf{B}'_N$]**, then let $k = k'$; we have in this case that $k < \text{len}(w_1) + N\text{len}(u)$ and this allows us to deduce that $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. Now assume $k' \geq \text{len}(w_1) + N\text{len}(u)$ **[k' is in Zone $\mathbf{C}'_N$]** and $i' - k' \leq \ell$ (k' and i' are in the same word u), then let $k = i - (i'-k')$. In this case we have $k \geq \text{len}(w_1) + N\text{len}(u)$ **[k is in Zone $\mathbf{C}_N$]** and since $|i-i'| = 0 \bmod \text{len}(u)$, we also have $|k-k'| = 0 \bmod \text{len}(u)$, hence $(w, k) \approx_{N-1} (w', k')$. Now assume $k' \geq \text{len}(w_1) + N\text{len}(u)$ **[k' is in Zone $\mathbf{C}'_N$]** and $i' - k' > \ell$ (k' and i' are not in the same word u). We denote by $\ell' = (k' - (\text{len}(w_1) + N\text{len}(u))) \bmod \text{len}(u)$ the relation position of k' in u and let $k = i - \ell - (\text{len}(u) - \ell')$ (k is at the same position as k' of k in the word u preceding the word u i belongs to). Then $k \in [j-1, i]$ (because $\text{len}(u) - \ell' < \text{len}(u)$ and $i - j \geq \ell + \text{len}(u)$) and $k \geq \text{len}(w_1) + (N-1)\text{len}(u)$ (because $i + (\text{len}(u) - \ell) \geq \text{len}(w_1) + (M-N)\text{len}(u)$ and $\text{len}(u) - \ell < \text{len}(u)$) and $|k-k'| = 0 \bmod \text{len}(u)$ (k and k' are both pointing on the ℓ'-th position in word u). This allows us to deduce that $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
  - If $j - i \geq \ell + \text{len}(u)$ (j is neither in the same word u as i nor in the previous word u) and $j \geq \text{len}(w_1) + N\text{len}(u)$ **[j is in zone $\mathbf{C}_N$]**. Then let $\ell' = (j - (\text{len}(w_1) + N\text{len}(u))) \bmod \text{len}(u)$ the relative position of $j \in u$. We choose $j' = i' - \ell - (\text{len}(u) - \ell')$ (j' and j are on the same position of u but in the word u precedent the one to which i belongs to). We have then that $j' \geq \text{len}(w_1) + (N-1)\text{len}(u)$ **[j' is zone $\mathbf{B}'_N$ or $\mathbf{C}'_N$]** (because $i' - \ell \geq \text{len}(w_1) + N)\text{len}(u)$ and $\text{len}(u) - \ell' \leq \text{len}(u)$) and $|j-j'| = 0 \bmod \text{len}(u)$ (j and j' are both pointing on the ℓ'-th position in word u), hence $\langle w, j\rangle \approx_{N-1} \langle w', j'\rangle$. Let $k' \in [j'-1, i']$. If $i' - k' \leq \ell$ (k' and i' are in the same word u), then let $k = i - (i'-k')$. In this case we have $k \geq \text{len}(w_1) + N\text{len}(u)$ **[k is in Zone $\mathbf{C}_N$]** and since $|i-i'| = 0 \bmod \text{len}(u)$, we also have $|k-k'| = 0 \bmod \text{len}(u)$, hence $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. If $i' - k' > \ell$ (k' and i' are not in the same word u), then $k' - j' < \text{len}(u) - \ell'$ and let $k = j + k' - j'$. In this case we have $\text{len}(w_1) + N\text{len}(u) \leq k < \text{len}(w_1) + (M-N)\text{len}(u)$ **[k is in Zone $\mathbf{C}_N$]** and since $|j-j'| = 0 \bmod \text{len}(u)$, we also have $|k-k'| = 0 \bmod \text{len}(u)$, hence $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. □

References

[1] F. Baader, P. Hanschke, A scheme for integrating concrete domains into concept languages, in: IJCAI'91, 1991, pp. 452–457.
[2] B. Boigelot, Symbolic methods for exploring infinite state spaces, Ph.D. thesis, Université de Liège, 1998.
[3] I. Borosh, L. Treybig, Bounds on positive integral solutions of linear Diophantine equations, Am. Math. Soc. 55 (1976) 299–304.
[4] M. Bozga, R. Iosif, F. Konecný, Fast acceleration of ultimately periodic relations, in: CAV'10, in: LNCS, vol. 6174, Springer, 2009, pp. 227–242.
[5] M. Bozga, R. Iosif, F. Konecný, Safety problems are NP-complete for flat integer programs with octagonal loops, in: VMCAI'14, in: LNCS, vol. 8318, Springer, 2014, pp. 242–261.
[6] M. Bozga, R. Iosif, Y. Lakhnech, Flat parametric counter automata, Fundam. Inform. 91 (2) (2009) 275–303.
[7] E.M. Clarke Jr., O. Grumberg, D.A. Peled, Model Checking, MIT Press, Cambridge, MA, USA, 1999.
[8] H. Comon, V. Cortier, Flatness is not a weakness, in: CSL'00, in: LNCS, vol. 1862, Springer, 2000, pp. 262–276.
[9] H. Comon, Y. Jurski, Multiple counter automata, safety analysis and PA, in: CAV'98, in: LNCS, vol. 1427, Springer, 1998, pp. 268–279.
[10] D. Cooper, Theorem proving in arithmetic without multiplication, Mach. Learn. 7 (1972) 91–99.
[11] S. Demri, A. Dhar, A. Sangnier, Taming past LTL and flat counter systems, in: IJCAR'12, in: LNAI, vol. 7364, Springer, 2012, pp. 179–193.
[12] S. Demri, A. Dhar, A. Sangnier, On the complexity of verifying regular properties on flat counter systems, in: ICALP'13, in: LNCS, vol. 7966, Springer, 2013, pp. 162–173.
[13] S. Demri, A. Dhar, A. Sangnier, Equivalence between model-checking flat counter systems and Presburger arithmetic, in: RP'14, in: LNCS, vol. 8762, Springer, 2014, pp. 85–97.
[14] S. Demri, A. Finkel, V. Goranko, G. van Drimmelen, Model-checking CTL* over flat Presburger counter systems, J. Appl. Non-Class. Log. 20 (4) (2010) 313–344.
[15] A. Dhar, Algorithms for model-checking flat counter systems, Ph.D. thesis, Université Paris Diderot, 2014.
[16] C. Dixon, M. Fisher, B. Konev, Temporal logic with capacity constraints, in: FROCOS'07, in: LNCS, vol. 4720, Springer, 2007, pp. 163–177.
[17] J. Esparza, A. Finkel, R. Mayr, On the verification of broadcast protocols, in: LICS'99, 1999, pp. 352–359.
[18] K. Etessami, T. Wilke, An until hierarchy and other applications of an Ehrenfeucht–Fraïssé game for temporal logic, Inf. Comput. 160 (1–2) (2000) 88–108.
[19] A. Finkel, J. Leroux, How to compose Presburger accelerations: applications to broadcast protocols, in: FST&TCS'02, in: LNCS, vol. 2256, Springer, 2002, pp. 145–156.
[20] A. Finkel, E. Lozes, A. Sangnier, Towards model-checking programs with lists, in: Infinity in Logic & Computation, in: LNAI, vol. 5489, Springer, 2009, pp. 56–82.
[21] D. Gabbay, The declarative past and imperative future, in: Temporal Logic in Specification, in: LNCS, vol. 398, Springer, Altrincham, UK, 1987, pp. 409–448.
[22] S. Ghilardi, E. Nicolini, S. Ranise, D. Zucchelli, Towards SMT model checking of array-based systems, in: IJCAR'08, in: LNCS, vol. 5195, Springer, 2008, pp. 67–82.

[23] E. Gurari, O. Ibarra, The complexity of decision problems for finite-turn multicounter machines, in: ICALP'81, in: LNCS, vol. 115, Springer, 1981, pp. 495–505.
[24] C. Haase, S. Halfon, Integer vector addition systems with states, in: RP'14, in: LNCS, vol. 8762, Springer, 2014, pp. 112–124.
[25] C. Haase, S. Kreutzer, J. Ouaknine, J. Worrell, Reachability in succinct and parametric one-counter automata, in: CONCUR'09, in: LNCS, vol. 5710, Springer, 2009, pp. 369–383.
[26] L. Kuhtz, Model checking finite paths and trees, Ph.D. thesis, Universität des Saarlandes, 2010.
[27] L. Kuhtz, B. Finkbeiner, Weak Kripke structures and LTL, in: CONCUR'11, in: LNCS, vol. 6901, Springer, 2011, pp. 419–433.
[28] A. Kučera, J. Strejček, The stuttering principle revisited, Acta Inform. 41 (7–8) (2005) 415–434.
[29] F. Laroussinie, N. Markey, P. Schnoebelen, Temporal logic with forgettable past, in: LICS'02, IEEE, 2002, pp. 383–392.
[30] F. Laroussinie, P. Schnoebelen, Specification in CTL + past for verification in CTL, Inf. Comput. 156 (2000) 236–263.
[31] J. Leroux, G. Sutre, Flat counter systems are everywhere!, in: ATVA'05, in: LNCS, vol. 3707, Springer, 2005, pp. 489–503.
[32] C. Lutz, NEXPTIME-complete description logics with concrete domains, in: IJCAR'01, in: LNCS, vol. 2083, Springer, 2001, pp. 46–60.
[33] N. Lynch, Log space recognition and translation of parenthesis languages, J. ACM 24 (4) (1977) 583–590.
[34] M. Minsky, Computation, Finite and Infinite Machines, Prentice Hall, 1967.
[35] D. Peled, T. Wilke, Stutter-invariant temporal properties are expressible without the next-time operator, Inf. Process. Lett. 63 (1997) 243–246.
[36] A. Pnueli, The temporal logic of programs, in: FOCS'77, IEEE Computer Society Press, 1977, pp. 46–57.
[37] C. Rackoff, The covering and boundedness problems for vector addition systems, Theor. Comput. Sci. 6 (2) (1978) 223–231.
[38] A. Sistla, E. Clarke, The complexity of propositional linear temporal logic, J. ACM 32 (3) (1985) 733–749.
[39] M. Vardi, Alternating automata: unifying truth and validity checking for temporal logics, in: CADE'97, in: LNCS, vol. 1249, Springer, 1997, pp. 191–206.