Taming Past LTL and Flat Counter Systems*

Stéphane Demri[1], Amit Kumar Dhar[2], and Arnaud Sangnier[2]

[1] LSV, CNRS, ENS Cachan, INRIA, France
[2] LIAFA, Univ Paris Diderot, Sorbonne Paris Cité, CNRS, France

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 the problem is only NP-complete even if LTL admits past-time operators and arithmetical constraints on counters. Actually, the NP upper bound is shown by adequately combining a new stuttering theorem for Past LTL and the property of small integer solutions for quantifier-free Presburger formulae. Other complexity results are proved, for instance for restricted classes 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 [8] and programs with pointers [11] to quote a few examples. But, alongwith their large scope of usability, many problems on general counter systems are known to be undecidable. Indeed, this computational model can simulate Turing machines. Decidability of reachability problems or model-checking problems based on temporal logics, can be regained by considering subclasses of counter systems, see e.g. [13]. An important and natural class of counter systems, in which various practical cases of infinite-state systems (e.g. broadcast protocols [10]) 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. [1,5,10,19]. Decidability results on verifying safety and reachability properties on flat counter systems have been obtained in [5,10,3]. However, so far, such properties have been rarely considered in the framework of any formal specification language (see an exception in [4]). In [6], 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 arcs labelled by adequate Presburger formulae. Even though flatness is clearly a substantial restriction, it is shown in [19] 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

* Supported by ANR project REACHARD ANR-11-BS02-001.

related to semilinearity of its reachability set. Moreover, in [4] model-checking relational counter systems over LTL formulae is shown decidable when restricted to flat formulae (their translation into automata leads to flat structures).

Towards the complexity of temporal model-checking flat counter systems. In [6], it is shown 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. Obviously CTL* properties are more expressive than reachability properties but this has a cost. However, for the class of counter systems considered in this paper, this provides a very rough complexity upper bound in 4ExpTime. Herein, our goal is to revisit standard decidability results for subclasses of counter systems obtained by translation into Presburger arithmetic in order to obtain optimal complexity upper bounds.

Our contributions. In the 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 proof technique could be reused for further extensions. Indeed, we combine three proof techniques: the general stuttering theorem [16], the property of small integer solutions of equation systems [2] (this latter technique is used since [23]) and the elimination of disjunctions in guards (see Section 5.2). Let us be a bit more precise.

We extend the stuttering principle established in [16] for LTL (without past-time operators) to Past LTL. The stuttering theorem from [16] for LTL without past-time operators has been used to show that LTL model-checking over *weak* Kripke structures is in NP [15] (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 [12] (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 17) by combining the above-mentioned proof techniques. Apart from the use of the general stuttering theorem (Theorem 3), we take advantage of the other properties stated for instance in Lemma 12 (characterization of runs by quantifier-free Presburger formulae) and Theorem 14 (elimination of disjunctions in guards preserving flatness). In the 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 (a fundamental structure in flat counter systems) with only 2 loops is already NP-hard (see Lemma 11). A summary table can be found in Section 6.

Omitted proofs can be found in the technical appendix.

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 $\{k \in \mathbb{Z} : i \leq k \text{ and } k \leq j\}$. For $\boldsymbol{v} \in \mathbb{Z}^n$, $\boldsymbol{v}[i]$ denotes the i^{th} element of $\boldsymbol{v}$ for every $i \in [1, n]$. For some n-ary tuple t, we write $\pi_j(t)$ to denote the j^{th} 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 $w = a_1 \ldots a_k$ 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

Let $\mathtt{C} = \{\mathtt{x}_1, \mathtt{x}_2, \ldots\}$ be a countably infinite set of *counters* (variables interpreted over non-negative integers) and $\mathrm{AT} = \{p_1, p_2, \ldots\}$ be a countable infinite set of propositional variables (abstract properties about program points). We write $\mathtt{C}_n$ to denote $\{\mathtt{x}_1, \mathtt{x}_2, \ldots, \mathtt{x}_n\}$. The set $\mathtt{G}(\mathtt{C}_n)$ of *guards* (arithmetical constraints on counters in $\mathtt{C}_n$) is defined inductively as follows: $\mathtt{t} ::= a.\mathtt{x} \mid \mathtt{t} + \mathtt{t}$ and $\mathtt{g} ::= \mathtt{t} \sim b \mid \mathtt{g} \wedge \mathtt{g} \mid \mathtt{g} \vee \mathtt{g}$, where $\mathtt{x} \in \mathtt{C}_n$, $a \in \mathbb{Z}$, $b \in \mathbb{Z}$ and $\sim \in \{=, \leq, \geq, <, >\}$. Such guards are closed under negations (but negation is not part of the logical connectives) 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 $\boldsymbol{v}[i]$ holds.

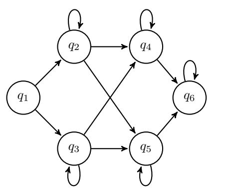
Definition 1 (Counter system). *For $n \geq 1$, a counter system S is a tuple $\langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$ where Q is a finite set of* control states*, $\mathbf{l} : Q \to 2^{\mathrm{AT}}$ is a* labelling function *and $\Delta \subseteq Q \times \mathtt{G}(\mathtt{C}_n) \times \mathbb{Z}^n \times Q$ is a finite set of edges labeled by guards and updates of the counter values (*transitions*).*

For $\delta = (q, \mathtt{g}, \mathbf{u}, q')$ in Δ, we use the following notations $\mathit{source}(\delta) = q$, $\mathit{target}(\delta) = q'$, $\mathit{guard}(\delta) = \mathtt{g}$ and $\mathit{update}(\delta) = \mathbf{u}$. As usual, to a counter system $S = \langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$, we associate a labeled transition system $TS(S) = \langle C, \to \rangle$ 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 $q = \mathit{source}(\delta)$, $q' = \mathit{target}(\delta)$, $\mathbf{v} \models \mathit{guard}(\delta)$ and $\mathbf{v}' = \mathbf{v} + \mathit{update}(\delta)$. In such a transition system, the counter values are non-negative since $C = Q \times \mathbb{N}^n$. We extend the transition relation $\to$ to finite words of transitions in Δ^+ as follows. For each $w = \delta_1 \delta_2 \ldots \delta_\alpha \in \Delta^+$, 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 $\mathit{lab}(c_0)$ to denote the set of ω-words (*labels*) which are fireable from c_0 in S.

Given a 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 $TS(S)$ 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) [5]. 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 denote by $\mathcal{CFS}$ the class of flat counter systems.



On the left, we present the control graph of a flat counter system (guards and updates are omitted). A *Kripke structure* S is a tuple $\langle Q, \Delta, \mathbf{l}\rangle$ where $\Delta \subseteq Q \times Q$ and $\mathbf{l}$ is labelling. 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 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 $\mathcal{KFS}$).

2.2 Linear Temporal Logic with Past and Arithmetical Constraints

Model-checking problem for Past LTL over finite state systems is known to be PSPACE-complete. 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, whence giving up the standard abstraction made with propositional variables. We define below a version of LTL dedicated to counter systems in which the atomic formulae are linear constraints; this is analogous to the use of concrete domains in description logics [20]. Note that capacity constraints from [7] are arithmetical constraints different from those defined below. Formulae of PLTL[$\mathtt{C}$] are defined from $\phi ::= p \mid \mathtt{g} \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathtt{X}\phi \mid \phi\mathtt{U}\phi \mid \mathtt{X}^{-1}\phi \mid \phi\mathtt{S}\phi$ where $p \in \mathrm{AT}$ 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 $\mathtt{GF}(\mathsf{x}_1 + 2 \geq \mathsf{x}_2)$ 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, but it is known that it helps a lot to express properties succinctly, see e.g. [18,17]. The temporal depth of ϕ, written $td(\phi)$, is defined as the maximal number of imbrications of temporal operators in ϕ. Restriction of PLTL[$\mathtt{C}$] to atomic formulae from AT only is written PLTL[$\emptyset$], standard version of LTL with past-time operators. Models of PLTL[$\mathtt{C}$] are essentially abstractions of runs from counter systems, i.e. ω-sequences $\sigma : \mathbb{N} \to 2^{\mathrm{AT}} \times \mathbb{N}^{\mathtt{C}}$. Given a model σ and a position $i \in \mathbb{N}$, the satisfaction relation $\models$ for PLTL[C] is defined as follows (other cases can be defined similarly, see e.g. [17]):

- $\sigma, i \models\ p \overset{\mathsf{def}}{\Leftrightarrow} p \in \pi_1(\sigma(i))$, $\sigma, i \models\ \mathtt{g} \overset{\mathsf{def}}{\Leftrightarrow} \mathbf{v}_i \models \mathtt{g}$ where $\mathbf{v}_i[j] \overset{\text{def}}{=} \pi_2(\sigma(i))(\mathsf{x}_j)$,
- $\sigma, i \models\ \mathtt{X}\phi \overset{\mathsf{def}}{\Leftrightarrow} \sigma, i+1 \models\ \phi$,
- $\sigma, i \models\ \phi_1 \mathtt{S} \phi_2 \overset{\mathsf{def}}{\Leftrightarrow} \sigma, j \models\ \phi_2$ for some $0 \leq j \leq i$ s.t. $\sigma, k \models\ \phi_1, \forall j < k \leq i$.

Given $\langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$ 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 $\sigma_\rho : \mathbb{N} \to 2^{\mathrm{AT}} \times \mathbb{N}^{\mathtt{C}}$ such that $\pi_1(\sigma_\rho(i)) \overset{\text{def}}{=} \mathbf{l}(q_i)$ and $\pi_2(\sigma_\rho(i))(\mathsf{x}_j) \overset{\text{def}}{=} \mathbf{v}_i[j]$ for all $j \in [1, n]$ and all $i \in \mathbb{N}$. Note that $\pi_2(\sigma_\rho(i))(\mathsf{x}_j)$ is arbitrary for $j \notin [1, n]$. As expected, we extend the satisfaction relation to runs so that $\rho, i \models \phi \overset{\text{def}}{\Leftrightarrow} \sigma_\rho, i \models \phi$ whenever ϕ is built from counters in $\mathtt{C}_n$.

Given a fragment L of PLTL[C] and a class $\mathcal{C}$ of counter systems, we write MC(L, $\mathcal{C}$) to denote the existential model checking problem: given $S \in \mathcal{C}$, a configuration c_0 and $\phi \in$ L, does there exist ρ starting from c_0 such that $\rho, 0 \models \phi$? In that case, we write $S, c_0 \models \phi$. It is known that for the full class of counter systems, the model-checking problem is undecidable, see e.g. [21]. Some restrictions, such as flatness, can lead to decidability as shown in [6] but the decision procedure there involves an exponential reduction to Presburger Arithmetic, whence the high complexity.

Theorem 2. *[6,15]* MC(PLTL[C], $\mathcal{CFS}$) *can be solved in* 4ExpTime.
MC(PLTL[$\emptyset$], $\mathcal{KFS}$) *restricted to formulae with temporal operators* U,X *is* NP-*complete.*

Our main goal is to characterize the complexity of MC(PLTL[C], $\mathcal{CFS}$).

3 Stuttering Theorem for PLTL[$\emptyset$]

Stuttering of finite words or single letters has been instrumental to show results about the expressive power of PLTL[$\emptyset$] fragments, see e.g. [22,16]; for instance, PLTL[$\emptyset$] restricted to the temporal operator U characterizes the class of formulae defining classes of models invariant under stuttering. This is refined in [16] for PLTL[$\emptyset$] 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 PLTL[$\emptyset$] with 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[$\emptyset$] model ($\sigma_1, \mathbf{s}$ being finite words), $\phi \in$ PLTL[$\emptyset$], $td(\phi) \leq N$ and $M \geq 2N+1$, then $\sigma_1 \mathbf{s}^{2N+1} \sigma_2, 0 \models \phi$ (and other related properties). This extends a result without past-time operators [15]. Moreover, this turns out to be a key property (Theorem 3) to establish the NP upper bound even in the presence of counters. Note that Theorem 3 below is interesting for its own sake, independently of our investigation on flat counter systems. By lack of space, we state below the main definitions and result.

Given $M, M', N \in \mathbb{N}$, we write $M \approx_N M'$ iff $\texttt{Min}(M,N) = \texttt{Min}(M',N)$. 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, $\langle w, i \rangle \approx_N \langle w', i' \rangle \overset{\text{def}}{\Leftrightarrow} M \approx_{2N} M'$ and one of the following conditions holds true: (1) $i, i' < \text{len}(w_1) + N \cdot \text{len}(u)$ and $i = i'$, (2) $i \geq \text{len}(w_1) + (M - N) \cdot \text{len}(u)$, $i' \geq \text{len}(w_1) + (M' - N) \cdot \text{len}(u)$ and $(i - i') = (M - M') \cdot \text{len}(u)$, (3) $\text{len}(w_1) + N \cdot \text{len}(u) \leq i < \text{len}(w_1) + (M - N) \cdot \text{len}(u)$, $\text{len}(w_1) + N \cdot \text{len}(u) \leq i' < \text{len}(w_1) + (M' - N) \cdot \text{len}(u)$ and $|i - i'| = 0 \mod \text{len}(u)$. We state our stuttering theorem for PLTL[$\emptyset$] that is tailored for our future needs.

Theorem 3 (Stuttering). *Let $\sigma = \sigma_1 \mathbf{s}^M \sigma_2, \sigma' = \sigma_1 \mathbf{s}^{M'} \sigma_2 \in (2^{\text{AT}})^\omega$ and $i, i' \in \mathbb{N}$ such that $N \geq 2$, $M, M' \geq 2N + 1$ and $\langle \sigma, i \rangle \approx_N \langle \sigma', i' \rangle$. Then, for every* PLTL[$\emptyset$] *formula ϕ with $td(\phi) \leq N$, we have $\sigma, i \models \phi$ iff $\sigma', i \models \phi$.*

Proof. (sketch) The proof is by structural induction on the formula but first we need to establish properties 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 $M, M' \geq 2N + 1$ and $\langle w, i \rangle \approx_N \langle w', i' \rangle$. We can show the following properties:

(Claim 1) $\langle w, i \rangle \approx_{N-1} \langle w', i' \rangle$ and $w(i) = w'(i')$.
(Claim 2) $\langle w, i+1 \rangle \approx_{N-1} \langle w', i'+1 \rangle$ and $i, i' > 0$ implies $\langle w, i-1 \rangle \approx_{N-1} \langle w', i'-1 \rangle$.
(Claim 3) 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 4) 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 $\langle w, k \rangle \approx_{N-1} \langle w', k' \rangle$.

By way of example, let us present the induction step for subformulae of the form $\psi_1 \texttt{U} \psi_2$. We show that $\sigma, i \models \psi_1 \texttt{U} \psi_2$ implies $\sigma', i' \models \psi_1 \texttt{U} \psi_2$. Suppose there is $j \geq i$ such that $\sigma, j \models \psi_2$ and for every $k \in [i, j-1]$, we have $\sigma, k \models \psi_1$. There is $j' \geq i'$ satisfying (Claim 3). Since $td(\psi_1), td(\psi_2) \leq N - 1$, by (IH), we have $\sigma', j' \models \psi_2$. Moreover, for every $k' \in [i', j'-1]$, there is $k \in [i, j-1]$ such that $\langle w, k \rangle \approx_{N-1} \langle w', k' \rangle$ and by (IH), we have $\sigma', k' \models \psi_1$ for every $k' \in [i', j'-1]$. Hence, $\sigma', i' \models \psi_1 \texttt{U} \psi_2$. □

An alternative proof consists in using Ehrenfeucht-Fraïssé games [9].

4 Fundamental Structures: Minimal Path Schemas

In this section, we introduce the notion of a fundamental structure for flat counter systems, namely a path schema. 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. So, all our nondeterministic algorithms on flat counter systems have a preliminary step that first guesses a minimal path schema.

4.1 Minimal Path Schemas

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 sequence of transitions from Δ such that $target(p(i)) = source(p(i+1))$ for all $0 \leq i < \mathrm{len}(p) - 1$. We write $first(p)$ [resp. $last(p)$] to denote the first [resp. last] control state of a path segment, in other words $first(p) = source(p(0))$ and $last(p) = target(p(\mathrm{len}(p) - 1))$. We also write $effect(p)$ to denote the sum vector $\sum_{0 \leq i < \mathrm{len}(p)} update(p(i))$ 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 $first(p) = last(p)$. A *path schema* P is an ω-regular expression built over Δ such that its language represents an overapproximation of the set of labels obtained from infinite runs following the transitions of P. A path schema P is of the form $p_1 l_1^+ p_2 l_2^+ \ldots p_k l_k^\omega$ where (1) l_1, $\ldots$, l_k are loops and (2) $p_1 l_1 p_2 l_2 \ldots p_k l_k$ is a path segment.

We write $\mathrm{len}(P)$ to denote $\mathrm{len}(p_1 l_1 p_2 l_2 \ldots p_k l_k)$ and $\mathrm{nbloops}(P)$ as its number k of loops. Let $\mathcal{L}(P)$ denote the set of infinite words in Δ^ω which belong to the language defined by P. Note that some elements of $\mathcal{L}(P)$ may not correspond to any run because of constraints on counter values. 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 $w = p_1 l_1^{iter_P(w)[1]} p_2 l_2^{iter_P(w)[2]} \ldots p_k l_k^\omega$. So, for every $i \in [1, k-1]$, $iter_P(w)[i]$ is the number of times the loop l_i is taken. Then, for a configuration c_0, the set $iter_P(c_0)$ is the set of vectors $\{iter_P(w) \in (\mathbb{N} \setminus \{0\})^{k-1} \mid w \in lab(P, c_0)\}$. 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 $iter_P(lab(\rho))$. Note that 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. However, we can impose minimality conditions on path schemas without sacrifying completeness. A path schema $p_1 l_1^+ p_2 l_2^+ \ldots p_k l_k^\omega$ is *minimal* whenever $p_1 \cdots p_k$ is either the empty word or a simple non-loop segment, and l_1, $\ldots$, l_k are loops with disjoint sets of transitions.

Lemma 4. *Given a flat counter system* $S = \langle Q, \mathtt{C}_n, \Delta, \mathbf{l}\rangle$*, the total number of minimal path schemas of* S *is finite and is smaller than* $\mathrm{card}(\Delta)^{(2 \times \mathrm{card}(\Delta))}$.

This is a simple consequence of the fact that in a minimal path schema, each transition occurs at most twice. In Figure 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 Figure 1 corresponds to the ω-regular expression $\delta_1(\delta_2\delta_3)^+\delta_4\delta_5(\delta_6\delta_5)^\omega$. Note that in the representation of path schemas, a state may occur several times, as it is the case for q_3 (this cannot occur in the representation of counter systems). Minimal path schemas play a crucial role in the sequel. Indeed, given a path schema P, there is a minimal path schema P' such that every run respecting P respects P' too. This can be easily

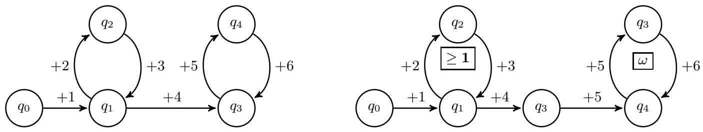


Fig. 1. A flat counter system and one of its minimal path schemas

shown since whenever a maximal number of copies of a simple loop is identified as a factor of $p_1 l_1 \cdots p_k l_k$, this factor is replaced by the simple loop unless it is already present in the path schema.

Finally, the conditions imposed on the structure of path schemas implies the following corollary which states that the number of minimal path schemas for a given flat counter system is at most exponential in the size of the system (see similar statements in [19]).

Corollary 5. *Given 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* $lab(c_0) = lab(\bigcup_{P\in X} P, c_0)$.

4.2 Complexity Results

We write $\mathcal{CPS}$ [resp. $\mathcal{KPS}$] to denote the class of path schemas from counter systems [resp. the class of path schemas from Kripke structures]. As a preliminary step, we consider the problem MC(PLTL[$\emptyset$], $\mathcal{KPS}$) that takes as inputs a path schema P in $\mathcal{KPS}$, and $\phi \in$ PLTL[$\emptyset$] and asks whether there is a run respecting P that satisfies ϕ. 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 $\mathtt{Min}(iter_P(\rho)[i], \alpha) = \mathtt{Min}(iter_P(\rho')[i], \alpha)$. We state below a result concerning the runs of flat counter systems when respecting the same path schema.

Proposition 6. *Let S be a flat counter system, P be a path schema, and $\phi \in$* PLTL[$\emptyset$]. *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 Theorem 3 repeatedly in order to get rid of the unwanted iterations of the loops. Our algorithm for MC(PLTL[$\emptyset$], $\mathcal{KPS}$) takes advantage of a result from [17] 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 were uv is a path segment and consequently $first(v) = last(v)$. According to [17], given an ultimately periodic path w, and a formula $\phi \in$ PLTL[$\emptyset$], the problem of checking whether there exists a run ρ such that $lab(\rho) = w$ and $\rho, 0 \models \phi$ is in PTime (a tighter bound of NC can be obtained by combining results from [14] and Theorem 3).

Lemma 7. MC(PLTL[$\emptyset$], $\mathcal{KPS}$) *is in* NP.

The proof is a consequence of Proposition 6 and [17]. Indeed, given $\phi \in$ PLTL[$\emptyset$] and $P = p_1 l_1^+ p_2 l_2^+ \ldots p_k l_k^\omega$, first guess $\mathbf{m} \in [1, 2td(\phi)+5]^{k-1}$ and check whether $\rho, 0 \models \phi$ where ρ is the obvious ultimately periodic word such that $lab(\rho) = p_1 l_1^{\mathbf{m}[1]} p_2 l_2^{\mathbf{m}[2]} \ldots p_k l_k^\omega$. Since $\mathbf{m}$ is of polynomial size and $\rho, 0 \models \phi$ can be checked in polynomial time by [17], we get the NP upper bound.

From [15], we have the lower bound for MC(PLTL[$\emptyset$], $\mathcal{KPS}$).

Lemma 8. *[15]* MC(PLTL[$\emptyset$], $\mathcal{KPS}$) *is* NP*-hard even if restricted to* `X` *and* `F`*.*

For a fixed $n \in \mathbb{N}$, we write MC(PLTL[$\emptyset$], $\mathcal{KPS}(n)$) to denote the restriction of MC(PLTL[$\emptyset$], $\mathcal{KPS}$) to path schemas with at most n loops. When n is fixed, the number of ultimately periodic paths w in $\mathcal{L}(P)$ such that each loop (except the last one) is visited is at most $2td(\phi)+5$ times is bounded by $(2td(\phi)+5)^n$, which is polynomial in the size of the input (because n is fixed).

Theorem 9. MC(PLTL[$\emptyset$], $\mathcal{KPS}$) *is* NP*-complete.*
Given a fixed $n \in \mathbb{N}$*,* MC(PLTL[$\emptyset$], $\mathcal{KPS}(n)$) *is in* PTime*.*

Note that it can be proved that MC(PLTL[$\emptyset$], $\mathcal{KPS}(n)$) 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 for model checking PLTL[$\emptyset$] over ultimately periodic paths parallelly on $(2td(\phi)+5)^n$ (polynomially many) different paths.

Now, we present how to solve MC(PLTL[$\emptyset$], $\mathcal{KFS}$) using Lemma 7. From Lemma 4, we know that 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 a state q and a PLTL[$\emptyset$] formula ϕ, a possible algorithm consists in choosing non-deterministically a minimal path schema P starting at q and then apply the algorithm used to establish Lemma 7. This new algorithm would be in NP. Furthermore, thanks to Corollary 5, we know that if there exists a run 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[$\emptyset$], $\mathcal{KFS}$).

Theorem 10. MC(PLTL[$\emptyset$], $\mathcal{KFS}$) *is* NP*-complete.*

NP-hardness can be established as a variant of the proof of Lemma 8.

Similarly, $\mathcal{CPS}(k)$ denotes the class of path schemas obtained from flat counter systems with number of loops bounded by k.

Lemma 11. *For* $k \geq 2$*,* MC(PLTL[`C`], $\mathcal{CPS}(k)$) *is* NP*-hard.*

The proof by reduction from SAT and it is less straightforward than the proof for Lemma 8 or the reduction presented in [15] 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 (see C for details). Hence, the reduction uses in an essential way the counter values and the arithmetical constraints in the formula. By contrast, MC(PLTL[`C`], $\mathcal{CPS}(1)$) can be shown in PTime(see Appendix H).

5 Model-checking PLTL[$\mathtt{C}$] over Flat Counter Systems

In this section, we provide a nondeterministic polynomial-time algorithm to solve MC(PLTL[$\mathtt{C}$],$\mathcal{CFS}$) (see Algorithm 1). To do so, we combine Theorem 3 with small solutions of constraint systems.

5.1 Characterizing Runs by System of Equations

In this section, we show how to build a system of equations from a path schema P and a configuration c_0 such that the system of equations 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 *validity* property. A path schema $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ is *valid* whenever *effect*$(l_k)[i] \geq 0$ for every $i \in [1, n]$ (see Section 4 for the definition of *effect*(l_k)) and if all the guards in transitions in l_k are conjunctions of atomic guards, then for each guard occurring in the loop l_k of the form $\sum_i a_i \mathsf{x}_i \sim b$ with $\sim \in \{\leq, <\}$ [resp. with $\sim \in \{=\}$, with $\sim \in \{\geq, >\}$] , we have $\sum_i a_i \times$ *effect*$(l_k)[i] \leq 0$ [resp. $\sum_i a_i \times$ *effect*$(l_k)[i] = 0$, $\sum_i a_i \times$ *effect*$(l_k)[i] \geq 0$]. It is easy to check that these conditions are necessary to visit the last loop l_k infinitely. More specifically, if a path schema is not valid, then no infinite run can respect it. Moreover, given a path schema, one can decide in polynomial time whether it is valid.

Now, let us consider a (not necessarily minimal) valid path schema $P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$ $(k \geq 1)$ obtained from a flat counter system S such that 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 12 below) is to characterize the set $iter_P(c_0) \subseteq \mathbb{N}^{k-1}$ for some configuration c_0 as the set of solutions of a constraint system. For each loop l_i, we introduce a variable y_i, whence the number of variables of the system/formula is precisely $k - 1$. A *constraint system* $\mathcal{E}$ over the set of variables $\{\mathsf{y}_1, \dots, \mathsf{y}_n\}$ is a quantifier-free Presburger formula built over $\{\mathsf{y}_1, \dots, \mathsf{y}_n\}$ 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 allows to emphasize the different purposes: guard on counters in operational models and symbolic representation of sets of tuples.

Lemma 12. *Let $S = \langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$ be a flat counter system without disjunctions in guards, P be a valid path schema and c_0 be a configuration. One can compute in polynomial time a constraint system $\mathcal{E}$ such that the set of solutions of $\mathcal{E}$ is equal to $iter_P(c_0)$, $\mathcal{E}$ has* $\mathrm{nbloops}(P) - 1$ *variables, $\mathcal{E}$ has at most* $\mathrm{len}(P) \times 2 \times \mathrm{size}(S)^2$ *conjuncts and the greatest absolute value from constants in $\mathcal{E}$ is bounded by* $n \times \mathrm{nbloops}(P) \times K^4 \times \mathrm{len}(P)^3$ *where K is the greatest absolute value for constants occurring in S.*

5.2 Elimination of Arithmetical Constraints and Disjunctions

As stated in Lemma 12, the procedure for characterizing infinite runs in a counter system by a system of equations works only for a flat counter system with no disjunction in guards (convexity of guards is essential). In this section, we show how to obtain such a system from a general flat counter system. Given a flat counter system $S = \langle Q, \mathtt{C}_n, \Delta, \mathbf{l} \rangle$, a configuration $c_0 = \langle q_0, \boldsymbol{v_0} \rangle$ and a minimal path schema P starting from the configuration c_0, we show that it is possible to build a finite set Y_P of path schemas such that (1) each path schema in Y_P has transitions without disjunctions in guards, (2) existence of a run ρ respecting P is equivalent to the existence of a path schema in Y_P having a run similar to ρ respecting it and (3) each path schema in Y_P is obtained from P by unfolding loops so that the terms in each loop satisfy the same atomic guards. Note that disjunctions could be easily eliminated at the cost of adding new transitions between states but this type of transformation may easily destroy flatness. Hence, the necessity to present a more sophisticated elimination procedure

We first introduce a few definitions. A (syntactic) *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 $\mathtt{t}$ appearing in some guards of the form $\mathtt{t} \sim b$ (with $b \in \mathbb{Z}$) and B is a finite set of integers. We say that a resource $\mathsf{R} = \langle X, T, B \rangle$ is *coherent* with a counter system S [resp. with a path schema P] if B contains all the constants b occurring in guards of S [resp. of P] of the form $\mathtt{t} \sim b$ and T contains all the terms $\mathtt{t}$ occurring in guards of S [resp. of P] of the form $\mathtt{t} \sim b$. The resource R is coherent with a formula $\phi \in$ PLTL[$\mathtt{C}$], whenever the atomic formulae of ϕ are either of the form $p \in X$ or $\mathtt{t} \sim b$ with $\mathtt{t} \in T$ and $b \in B$. In the sequel, we assume that the considered resource is always coherent with S.

Assuming that $B = \{b_1, \ldots, b_m\}$ with $b_1 < \cdots < b_m$, we write I to denote the finite set of intervals $I = \{(-\infty, b_1 - 1], [b_1, b_1], [b_1 + 1, b_2 - 1], [b_2, b_2], \cdots, [b_m, b_m], [b_m + 1, \infty)\}$. Note that I contains exactly $2m + 1$ intervals. A *term map* $\boldsymbol{m}$ is a map $\boldsymbol{m} : T \to I$ that abstracts term values. A *footprint* is an abstraction of a model for PLTL[$\mathtt{C}$] restricted to elements from the resource R: it is of the form $\mathsf{ft} : \mathbb{N} \to 2^X \times I^T$ where I is the set of intervals built from B. The satisfaction relation $\models$ involving models or runs can be adapted to footprints as follows (formulae and footprints are from the same resource):

- $\mathsf{ft}, i \models_{\mathsf{symb}} p \overset{\text{def}}{\Leftrightarrow} p \in \pi_1(\mathsf{ft}(i))$; $\mathsf{ft}, i \models_{\mathsf{symb}} \mathtt{t} \geq b \overset{\text{def}}{\Leftrightarrow} \pi_2(\mathsf{ft}(i))(\mathtt{t}) \subseteq [b, +\infty)$,
- $\mathsf{ft}, i \models_{\mathsf{symb}} \mathtt{t} \leq b \overset{\text{def}}{\Leftrightarrow} \pi_2(\mathsf{ft}(i))(\mathtt{t}) \subseteq (-\infty, +b]$,
- $\mathsf{ft}, i \models_{\mathsf{symb}} \mathtt{X}\phi \overset{\text{def}}{\Leftrightarrow} \mathsf{ft}, i + 1 \models_{\mathsf{symb}} \phi$,
- $\mathsf{ft}, i \models_{\mathsf{symb}} \phi \mathtt{U} \psi \overset{\text{def}}{\Leftrightarrow} \exists j \geq i$ s.t. $\mathsf{ft}, j \models_{\mathsf{symb}} \psi$ and $\forall j' \in [i, j-1]$, $\mathsf{ft}, j' \models_{\mathsf{symb}} \phi$.

We omit the other obvious clauses. $\models_{\mathsf{symb}}$ is the satisfaction relation for Past LTL when arithmetical constraints are understood as abstract propositions. Let $\mathsf{R} = \langle X, T, B \rangle$ be a resource and $\rho = \langle q_0, \boldsymbol{v_0} \rangle, \langle q_1, \boldsymbol{v_1} \rangle \cdots$ be an infinite run of S. The *footprint* of ρ with respect to R is the footprint $\mathsf{ft}(\rho)$ such that for $i \geq 0$, we have $\mathsf{ft}(\rho)(i) \overset{\text{def}}{=} \langle \mathbf{l}(q_i) \cap X, \boldsymbol{m}_i \rangle$ where for every term $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$, we have $\sum_j a_j \boldsymbol{v_i}[j] \in \boldsymbol{m}_i(\mathtt{t})$. Note that $\sum_j a_j \boldsymbol{v_i}[j]$ belongs to a unique element of I since I is a partition of $\mathbb{Z}$. Hence, this definition makes sense. Lemma 13 below roughly states that satisfaction of a formula on a run can be checked symbolically from the footprint (this is useful for the correctness of forthcoming Algorithm 1).

Lemma 13. *Let ϕ be in* PLTL[C]*,* $\mathsf{R} = \langle X, T, B\rangle$ *be coherent with ϕ,* $\rho = \langle q_0, \boldsymbol{v_0}\rangle, \langle q_1, \boldsymbol{v_1}\rangle \cdots$ *be an infinite run and $i \geq 0$. (I) Then $\rho, i \models \phi$ iff $\mathsf{ft}(\rho), i \models_{\mathsf{symb}} \phi$. (II) If ρ' is an infinite run s.t. $\mathsf{ft}(\rho) = \mathsf{ft}(\rho')$, then $\rho, i \models \phi$ iff $\rho', i \models \phi$.*

In Appendix D, we explain in details how to build a set Y_P of path schemas without disjunctions from a minimal path schema P, an initial configuration $\langle q_0, \boldsymbol{v_0}\rangle$ and a resource R. The main idea of this construction consists in adding to the control states of path schemas some information on the intervals to which belongs each term of T. In fact, in the transitions appearing in path schemas of Y_P the states belong to $Q' = Q \times I^T$. Before stating the properties of Y_P, we introduce some notations. Given $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$, $\mathbf{u} \in \mathbb{Z}^n$ and a term map $\boldsymbol{m}$, we write $\psi(\mathtt{t}, \mathbf{u}, \boldsymbol{m}(\mathtt{t}))$ to denote the formula below ($b, b' \in B$): $\psi(\mathtt{t}, \mathbf{u}, (-\infty, b]) \stackrel{\text{def}}{=} \sum_j a_j(\mathsf{x}_j + \mathbf{u}(j)) \leq b$; $\psi(\mathtt{t}, \mathbf{u}, [b, +\infty)) \stackrel{\text{def}}{=} \sum_j a_j(\mathsf{x}_j + \mathbf{u}(j)) \geq b$ and $\psi(\mathtt{t}, \mathbf{u}, [b, b']) = ((\sum_j a_j(\mathsf{x}_j + \mathbf{u}(j)) \leq b') \wedge ((\sum_j a_j(\mathsf{x}_j + \mathbf{u}(j)) \geq b)$. We write $\mathtt{G}^\star(T, B, U)$ to denote the set of guards of the form $\psi(\mathtt{t}, \mathbf{u}, \boldsymbol{m}(\mathtt{t}))$ where $\mathtt{t} \in T$, U is the finite set of updates from P and $\boldsymbol{m} : T \to I$. Each guard in $\mathtt{G}^\star(T, B, U)$ is of linear size in the size of P. We denote $\tilde{\Delta}$ the set of transitions $Q' \times \mathtt{G}^\star(T, B, U) \times U \times Q'$. Note that the transitions in $\tilde{\Delta}$ do not contain guards with disjunctions and $\tilde{\Delta}$ is finite. We also define a function proj which associates to $w \in \tilde{\Delta}^\omega$ the ω-sequence $\mathsf{proj}(w) : \mathbb{N} \to 2^X \times I^T$ such that for all $i \in \mathbb{N}$, if $w(i) = \langle\langle q, \boldsymbol{m}\rangle, \mathsf{g}, \mathbf{u}, \langle q', \boldsymbol{m'}\rangle\rangle$ and $\mathbf{l}(q) \cap X = L$ then $\mathsf{proj}(w)(i) \stackrel{\text{def}}{=} \langle L, \boldsymbol{m}\rangle$.

We show that it is possible to build a finite set Y_P of path schemas over $\tilde{\Delta}$ such that if $P' = p'_1(l'_1)^+ p'_2(l'_2)^+ \ldots p'_{k'}(l'_{k'})^\omega$ is a path schema in Y_P and ρ is a run $\langle\langle q_0, \boldsymbol{m}_0\rangle, \boldsymbol{v_0}\rangle \to \langle\langle q_1, \boldsymbol{m}_1\rangle, \boldsymbol{v_1}\rangle \to \langle\langle q_2, \boldsymbol{m}_2\rangle, \boldsymbol{v_2}\rangle \cdots$ respecting P' we have that $\mathsf{proj}(lab(\rho)) = \mathsf{ft}(\rho)$. This point will be useful for Algorithm 1. The following theorem lists the main properties of the set Y_P.

Theorem 14. *Given a flat counter system S, a minimal path schema P, a resource $\mathsf{R} = \langle X, T, B\rangle$ coherent with P and a configuration $\langle q_0, \boldsymbol{v_0}\rangle$, there is a finite set of path schemas Y_P over $\tilde{\Delta}$ satisfying (1)–(6) below.*

1. *No path schema in Y_P contains guards with disjunctions in it.*
2. *There exists a polynomial $q^\star(\cdot)$ such that for every $P' \in Y_P$, $\mathrm{len}(P') \leq q^\star(\mathrm{len}(P) + \mathrm{card}(T) + \mathrm{card}(B))$.*
3. *Checking whether a path schema P' over $\tilde{\Delta}$ belongs to Y_P can be done in polynomial time in $\mathrm{size}(P) + \mathrm{card}(T) + \mathrm{card}(B)$.*
4. *For every run ρ respecting P and starting at $\langle q_0, \boldsymbol{v_0}\rangle$, we can find a run ρ' respecting some $P' \in Y_P$ such that $\rho \models \phi$ iff $\rho' \models \phi$ for every ϕ built over* R*.*
5. *For every run ρ' respecting some $P' \in Y_P$ with initial values $\boldsymbol{v_0}$, 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 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.*

5.3 Main Algorithm

In Algorithm 1 below, a polynomial $p^\star(\cdot)$ is used. In Appendix F, we explain how $p^\star(\cdot)$ is defined (this is the place where Lemma 12 and small solutions for constraint systems [2] are used). Note that $\boldsymbol{y'}$ is a refinement of $\boldsymbol{y}$ (for all i, we have $\boldsymbol{y'}[i] \approx_{2td(\phi)+5} \boldsymbol{y}[i]$) in which counter values are taken into account.

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

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

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

Lemma 15. *Algorithm 1 runs in nondeterministic polynomial time.*

It remains to check that Algorithm 1 is correct.

Lemma 16. $S, c_0 \models \phi$ *iff Algorithm 1 on inputs* S, c_0, ϕ *has an accepting run.*

In the proof of Lemma 16, we take advantage of all our preliminary results.

Proof. By way of example, we show that if Algorithm 1 on inputs S, $c_0 = \langle q_0, \boldsymbol{v_0} \rangle$, ϕ has an accepting computation, then $S, c_0 \models \phi$. This means that there are P, P', $\boldsymbol{y}$, $\boldsymbol{y'}$ that satisfy all the checks. Let $w = p_1 l_1^{\boldsymbol{y'}[1]} \cdots p_{k-1} l_{k-1}^{\boldsymbol{y'}[k-1]} p_k l_k^\omega$ and $\rho = \langle \langle q_0, \boldsymbol{m}_0 \rangle, \boldsymbol{v_0} \rangle \langle \langle q_1, \boldsymbol{m}_1 \rangle, \boldsymbol{x_1} \rangle \langle \langle q_2, \boldsymbol{m}_2 \rangle, \boldsymbol{x_2} \rangle \cdots \in (Q' \times \mathbb{Z}^n)^\omega$ be defined as follows: for every $i \geq 0$, $q_i \stackrel{\text{def}}{=} \pi_1(source(w(i)))$, and for every $i \geq 1$, we have $\boldsymbol{x_i} \stackrel{\text{def}}{=} \boldsymbol{x_{i-1}} + update(w(i))$. By Lemma 12, since $\boldsymbol{y'} \models \mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$, ρ is a run respecting P' starting at the configuration $\langle \langle q_0, \boldsymbol{m}_0 \rangle, \boldsymbol{v_0} \rangle$. Since $\boldsymbol{y'} \models \psi_1 \wedge \cdots \wedge \psi_{k-1}$ and $\boldsymbol{y} \models \psi_1 \wedge \cdots \wedge \psi_{k-1}$, by Proposition 6, (✠) $\mathsf{proj}(p_1 l_1^{\boldsymbol{y}[1]} p_2 l_2^{\boldsymbol{y}[2]} \ldots l_{k-1}^{\boldsymbol{y}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$, iff (✠✠) $\mathsf{proj}(p_1 l_1^{\boldsymbol{y'}[1]} p_2 l_2^{\boldsymbol{y'}[2]} \ldots l_{k-1}^{\boldsymbol{y'}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$. Algorithm 1 guarantees that $\mathsf{proj}(p_1 l_1^{\boldsymbol{y}[1]} p_2 l_2^{\boldsymbol{y}[2]} \ldots l_{k-1}^{\boldsymbol{y}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$, whence we have (✠✠). Since $\mathsf{proj}(p_1 l_1^{\boldsymbol{y'}[1]} p_2 l_2^{\boldsymbol{y'}[2]} \ldots l_{k-1}^{\boldsymbol{y'}[k-1]} p_k l_k^\omega) = \mathsf{ft}(\rho)$, by Lemma 13, we deduce that $\rho, 0 \models$

ϕ. By Theorem 14(5), there is an infinite run ρ', starting at the configuration $\langle q_0, \boldsymbol{v_0} \rangle$ and respecting P, such that $\rho', 0 \models \phi$.

Now, suppose that $S, c_0 \models \phi$. We shall show that there exist P, P', $\boldsymbol{y}$, $\boldsymbol{y'}$ 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 5, ρ respects some minimal path schema of S, say P. By Theorem 14(4), there is a path schema $P' = p_1 l_1^+ p_2 l_2^+ \ldots p_k l_k^\omega$ in Y_P for which there is a run ρ' satisfying ϕ. Furthermore, since $P' \in Y_P$, $\mathrm{len}(P') \leq q^\star(\mathrm{len}(P) + \mathrm{card}(T) + \mathrm{card}(B))$. 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 $\psi_1 \wedge \cdots \wedge \psi_{k-1}$, the constraint system $\mathcal{E}$ obtained from P' (thanks to Lemma 12) but augmented with $\psi_1 \wedge \cdots \wedge \psi_{k-1}$ admits at least one solution. Let us define $\boldsymbol{y'} \in [1, 2^{p^\star(\mathrm{size}(S)+\mathrm{size}(c_0)+\mathrm{size}(\phi))}]^{k-1}$ as a small solution of $\mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$ and $\boldsymbol{y} \in [1, 2td(\phi)+5]^{k-1}$ be defined such that for $i \in [1, k-1]$, $\boldsymbol{y}[i] = max(\boldsymbol{y'}[i], 2td(\phi) + 5)$. As shown in Appendix F, $2^{p^\star(\mathrm{size}(S)+\mathrm{size}(c_0)+\mathrm{size}(\phi))}$ is sufficient if there is a solution. Clearly, $\boldsymbol{y'} \models \mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$. So $p_1 l_1^{\boldsymbol{y'}[1]} p_2 l_2^{\boldsymbol{y'}[2]} \ldots l_{k-1}^{\boldsymbol{y'}[k-1]} p_k l_k^\omega$ generates a genuine run. Since $\mathsf{ft}(\rho') = \mathsf{proj}(p_1 l_1^{\boldsymbol{y'}[1]} p_2 l_2^{\boldsymbol{y'}[2]} \ldots l_{k-1}^{\boldsymbol{y'}[k-1]} p_k l_k^\omega)$ and since by Lemma 13, we have $\mathsf{ft}(\rho') \models_{\mathsf{symb}} \phi$, we get that $\mathsf{proj}(p_1 l_1^{\boldsymbol{y'}[1]} p_2 l_2^{\boldsymbol{y'}[2]} \ldots l_{k-1}^{\boldsymbol{y'}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$. This also implies that P' is valid. Hence $\mathsf{proj}(p_1 l_1^{\boldsymbol{y}[1]} p_2 l_2^{\boldsymbol{y}[2]} \ldots l_{k-1}^{\boldsymbol{y}[k-1]} p_k l_k^\omega), 0 \models_{\mathsf{symb}} \phi$ thanks to Proposition 6. Consequently, we have everything to build an accepting computation for Algorithm 1 on inputs S, c_0, ϕ. ☐

As a corollary, we can state the main result of the paper.

Theorem 17. *MC(PLTL[*$\mathtt{C}$*],*$\mathcal{CFS}$*) is* NP*-complete.*

6 Conclusion

Classes of Systems	PLTL[$\emptyset$]	PLTL[$\mathtt{C}$]	Reachability
$\mathcal{KPS}$	NP-complete See [15] for $\mathtt{X}$ and $\mathtt{U}$	—	PTime
$\mathcal{CPS}$	NP-complete	NP-complete (Theo. 17)	NP-complete
$\mathcal{KPS}(n)$	PTime (Theo. 9)	—	PTime
$\mathcal{CPS}(n)$, $n > 1$	??	NP-complete (Lem. 11)	??
$\mathcal{CPS}(1)$	PTime	PTime	PTime
$\mathcal{KFS}$	NP-complete See [15] for $\mathtt{X}$ and $\mathtt{U}$	—	PTime
$\mathcal{CFS}$	NP-complete	**NP-complete** (Theo. 17)	NP-complete

We have investigated the computational complexity of the model-checking problem for flat counter systems with formulae from an enriched version of LTL. Our main result is the NP-completeness of MC(PLTL[$\mathtt{C}$],$\mathcal{CFS}$), significantly improving the complexity upper bound from [6]. This also improves the results about the effective semilinearity of the reachability relations for such flat counter systems from [5,10] and it extends the recent result on the NP-completeness of model-checking flat Kripke structures with LTL from [15] by adding counters and past-time operators. Our main results are presented above and compared to the reachability problem (complementary proofs can be found in Appendix I). 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 [16] with past-time operators) and the use of small integer solutions for quantifier-free Presburger formulae [2]. 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 [6] but the characterization of its exact complexity is open.

References

1. B. Boigelot. *Symbolic methods for exploring infinite state spaces.* PhD thesis, Université de Liège, 1998.
2. I. Borosh and L. Treybig. Bounds on positive integral solutions of linear Diophantine equations. *American Mathematical Society*, 55:299–304, 1976.
3. M. Bozga, R. Iosif, and F. Konecný. Fast acceleration of ultimately periodic relations. In *CAV'10*, volume 6174 of *LNCS*, pages 227–242. Springer, 2009.
4. H. Comon and V. Cortier. Flatness is not a weakness. In *CSL'00*, volume 1862 of *LNCS*, pages 262–276. Springer, 2000.
5. H. Comon and Y. Jurski. Multiple counter automata, safety analysis and PA. In *CAV'98*, volume 1427 of *LNCS*, pages 268–279. Springer, 1998.
6. S. Demri, A. Finkel, V. Goranko, and G. van Drimmelen. Model-checking CTL^* over flat Presburger counter systems. *JANCL*, 20(4):313–344, 2010.
7. C. Dixon, M. Fisher, and B. Konev. Temporal logic with capacity constraints. In *FROCOS'07*, volume 4720 of *LNCS*, pages 163–177. Springer, 2007.
8. J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *LICS'99*, pages 352–359, 1999.
9. K. Etessami and T. Wilke. An until hierarchy and other applications of an Ehrenfeucht-Fraïssé game for temporal logic. *I&C*, 160(1–2):88–108, 2000.
10. A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS'02*, volume 2256 of *LNCS*, pages 145–156. Springer, 2002.
11. A. Finkel, E. Lozes, and A. Sangnier. Towards model-checking programs with lists. In *Infinity in Logic & Computation*, volume 5489 of *LNAI*, pages 56–82. Springer, 2009.
12. D. Gabbay. The declarative past and imperative future. In *Temporal Logic in Specification, Altrincham, UK*, volume 398 of *LNCS*, pages 409–448. Springer, 1987.
13. C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *CONCUR'09*, volume 5710 of *LNCS*, pages 369–383. Springer, 2009.
14. L. Kuhtz. *Model Checking Finite Paths and Trees.* PhD thesis, Universität des Saarlandes, 2010.
15. L. Kuhtz and B. Finkbeiner. Weak Kripke structures and LTL. In *CONCUR'11*, volume 6901 of *LNCS*, pages 419–433. Springer, 2011.
16. A. Kučera and J. Strejček. The stuttering principle revisited. *Acta Informatica*, 41(7–8):415–434, 2005.

17. F. Laroussinie, N. Markey, and P. Schnoebelen. Temporal logic with forgettable past. In *LICS'02*, pages 383–392. IEEE, 2002.
18. F. Laroussinie and P. Schnoebelen. Specification in CTL + past for verification in CTL. *I&C*, 156:236–263, 2000.
19. J. Leroux and G. Sutre. Flat counter systems are everywhere! In *ATVA'05*, volume 3707 of *LNCS*, pages 489–503. Springer, 2005.
20. C. Lutz. NEXPTIME-complete description logics with concrete domains. In *IJ-CAR'01*, volume 2083 of *LNCS*, pages 46–60. Springer, 2001.
21. M. Minsky. *Computation, Finite and Infinite Machines*. Prentice Hall, 1967.
22. D. Peled and T. Wilke. Stutter-invariant temporal properties are expressible without the next-time operator. *IPL*, 63:243–246, 1997.
23. C. Rackoff. The covering and boundedness problems for vector addition systems. *TCS*, 6(2):223–231, 1978.

A Proofs of Section 3 on the Stuttering Theorem

The proof by structural induction is by an easy verification once (Claim 1)–(Claim 4) are proved, see e.g. the case when the outermost connective is U presented in the paper. Below, we recall the main definitions and statements, possibly illustrated by figures, and then we prove the four claims.

We recall the definition of the relation $\approx_N$ over pairs of words and positions in $\Sigma^\omega \times \mathbb{N}$. Given $w = w_1 u^M w_2, w' = w_1 u^{M'} w_2 \in \Sigma^\omega$ and $i, i' \in \mathbb{N}$, $\langle w, i\rangle \approx_N \langle w', i'\rangle \stackrel{\text{def}}{\Leftrightarrow} M \approx_{2N} M'$ and one of the conditions holds true:

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

Figure 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 $\langle w, i\rangle \approx_3 \langle w', i'\rangle$.

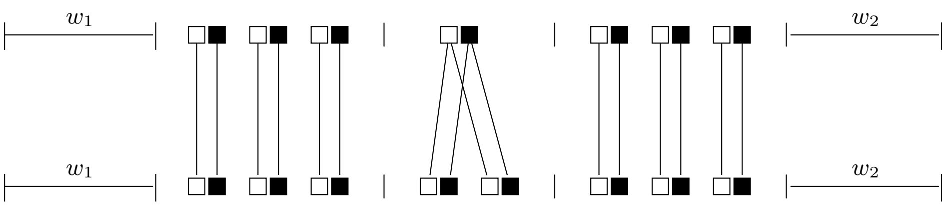


Fig. 2. Two words w, w' with $u = \square\blacksquare$ and the relation $\approx_3$

A.1 A Zone Classification for Proving (Claim 1) – (Claim 4)

For the proofs of (Claim 1) – (Claim 4), the positions of each word w of the form $w = w_1 u^M w_2 \in \Sigma^\omega$ ($w_1 \in \Sigma^*$, $u \in \Sigma^+$ and $w_2 \in \Sigma^\omega$) with $M > 2N$ are partitionned into five zones (A, B, C, D and E). We also assume that $N \geq 2$. Indeed, 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. The definition of zones is illustrated on Figure 3 and here is the formal characterization:

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

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

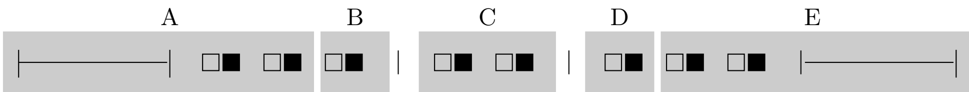


Fig. 3. The five zones for $w_1(\square\blacksquare)^8 w_2$ with $N = 3$ and $u = \square\blacksquare$

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 (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 (A'_N, B'_N etc.) to refer to zones for w'. So, the relation $\approx_N$ can be redefined as follows when $M, M' > 2N$: $\langle w, i \rangle \approx_N \langle w', i' \rangle \overset{\text{def}}{\Leftrightarrow}$ ($M \approx_{2N} M'$ and) one of the conditions holds true:

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

A.2 Proof of (Claim 1)

Before the proof, 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. Let us first prove that $\langle w, i \rangle \approx_{N-1} \langle w', i' \rangle$. Since $N > N - 1$, it is obvious that $M \approx_{2(N-1)} M'$.

– If $i < \text{len}(w_1) + (N - 1) \cdot \text{len}(u)$ **[i is Zone $\mathbf{A}_N$]**, then $i = i'$. Hence either ($i \in \text{A}_{N-1}$, $i' \in \text{A}'_{N-1}$ and $i = i'$) or ($i \in \text{B}_{N-1}$, $i' \in \text{B}'_{N-1}$ and $i = i'$). Hence, $\langle w, i \rangle \approx_{N-1} \langle w', i' \rangle$.
– If $i \geq \text{len}(w_1) + (M - (N - 1)) \cdot \text{len}(u)$ **[i is in zone $\mathbf{E}_N$]** then $i = i' + (M - M') \cdot \text{len}(u)$ and $i' \geq \text{len}(w_1) + (M' - (N - 1)) \cdot \text{len}(u)$ **[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 $i = i' + (M - M') \cdot \text{len}(u)$, we conclude that $\langle w, i \rangle \approx_{N-1} \langle w', i' \rangle$.

– If $\text{len}(w_1) + (N-1) \cdot \text{len}(u) \leq i < \text{len}(w_1) + N \cdot \text{len}(u)$ **[i is in Zone $\mathbf{B}_N$]** then $i = i'$. Hence, $i \in \text{C}_{N-1}$, $i' \in \text{C}'_{N-1}$ and $|i - i'| = 0 \mod \text{len}(u)$. Hence, $\langle w, i\rangle \approx_{N-1} \langle w', i'\rangle$.
– If $\text{len}(w_1)+N\cdot\text{len}(u) \leq i < \text{len}(w_1)+(M-N)\cdot\text{len}(u)$ **[i in Zone $\mathbf{C}_N$]**, then $\text{len}(w_1) + N \cdot \text{len}(u) \leq i' < \text{len}(w_1) + (M' - N) \cdot \text{len}(u)$ **[i' is in Zone $\mathbf{C}'_N$]** and $|i - i'| = 0 \mod \text{len}(u)$. Consequently, i is in Zone C_{N-1}, i' is in Zone C'_{N-1} and $|i - i'| = 0 \mod \text{len}(u)$. This entails that $\langle w, i\rangle \approx_{N-1} \langle w', i'\rangle$.
– If $\text{len}(w_1)+(M-N)\cdot\text{len}(u) \leq i < \text{len}(w_1)+(M-(N-1))\cdot\text{len}(u)$ **[i in Zone $\mathbf{D}_N$]**, then i is in Zone D'_N and $i = i' + (M - M') \cdot \text{len}(u)$. Consequently, i is in Zone C_{N-1}, i' is in Zone C'_{N-1} and $|i - i'| = 0 \mod \text{len}(u)$. 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 also clear that $w(i) = w'(i')$, because either i and i' are at the same position in the word w_1 or w_2 either they are pointing some positions in the portions of the word which belong to u^+ and since their difference will be such that $|i - i'| = 0 \mod \text{len}(u)$, it is easy to see that i and i' will point at the same position in u. □

A.3 Proof of (Claim 2)

Before the proof, 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) $\langle w, i{+}1\rangle \approx_{N-1} \langle w', i'{+}1\rangle$; $i, i' > 0$ implies $\langle w, i{-}1\rangle \approx_{N-1} \langle w', i'{-}1\rangle$.

Proof. Let us first prove that $\langle w, i + 1\rangle \approx_{N-1} \langle w', i' + 1\rangle$. Since $N > N - 1$, it is obvious that $M \approx_{2(N-1)} M'$.

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

– If $\text{len}(w_1)+(M-N)\cdot\text{len}(u) \leq i < \text{len}(w_1)+(M-(N-1))\cdot\text{len}(u)$ **[i in Zone D_N]**, then i is in Zone D'_N and $i = i' + (M - M')\cdot\text{len}(u)$. Consequently, either ($i+1$ is in Zone C_{N-1}, $i'+1$ is in Zone C'_{N-1} and $|(i+1)-(i'+1)| = 0 \mod \text{len}(u)$) or ($i+1$ is in Zone D_{N-1}, $i'+1$ is in Zone D'_{N-1} and $i+1 = i'+1+(M-M')\cdot\text{len}(u)$). This also entails that $\langle w, i+1\rangle \approx_{N-1} \langle w', i'+1\rangle$.

Now, let us prove that $i, i' > 0$ implies $\langle w, i-1\rangle \approx_{N-1} \langle w', i'-1\rangle$. Since $N > N-1$, it is obvious that $M \approx_{2(N-1)} M'$.

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

□

A.4 Proof of (Claim 3)

Before providing the detailed proof, we give a concrete example on Figure 4. 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 3) 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 on the bottom of Figure 4, 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 $i' \leq k \leq j'$, $\langle w, k\rangle \approx_2 \langle w', k\rangle$.

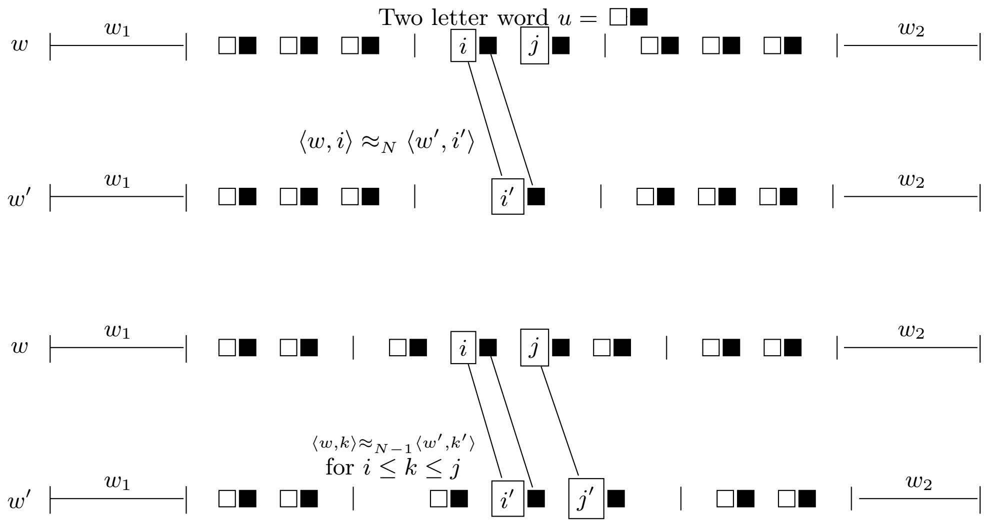


Fig. 4. Relation between $\approx_N$ and $\approx_{N-1}$

Before the proof, 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 $\langle w, i\rangle \approx_N \langle w', i'\rangle$. We can show the following properties:

(Claim 3) 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:

- If $i \geq \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ **[i is in Zone D or E]** then $j \geq \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ **[j is in Zone D or E]** and $i' \geq \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[i' is Zone D or E]** and $i = i' + (M-M')\cdot\text{len}(u)$. We define $j' = j-(M-M')\cdot\text{len}(u)$. 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 $k = k'+(M-M')\cdot\text{len}(u)$, 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 $i < \text{len}(w_1)+N\cdot\text{len}(u)$ **[i is in Zone A or B]** then $i' < \text{len}(w_1)+N\cdot\text{len}(u)$ **[i' is in Zone A or B]** and $i = i'$ and we have the following possibilities for the position $j \geq i$:
  - If $j < \text{len}(w_1) + N\cdot\text{len}(u)$ **[j is in Zone A or B]**, 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 $\text{len}(w_1) + N\cdot\text{len}(u) \leq j < \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ **[j is in Zone C]**, then let $\ell = (j - (\text{len}(w_1) + N\cdot\text{len}(u))) \mod \text{len}(u)$ (ℓ the relative

position of j in the word u it belongs to). Consequently $0 \leq \ell < \text{len}(u)$. Let $j' = \text{len}(w_1)+N\cdot\text{len}(u)+\ell$ (we choose j' at the same relative position of j in the first word u of the Zone C). Then $\text{len}(w_1) + N \cdot \text{len}(u) \leq j' < \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[j' is in Zone C]** (because $(M'-N) > 0$) and $|j-j'| = 0 \mod \text{len}(u)$. 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 $k' < \text{len}(w_1)+N\cdot\text{len}(u)$ **[k' is in Zone A or B]** 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 $\text{len}(w_1) + N \cdot \text{len}(u) \leq k'$ **[k' is in Zone C]** then k is in Zone C and $|k - k'| = 0 \mod \text{len}(u)$ 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 $\text{len}(w_1)+(M-N)\cdot\text{len}(u) \leq j$ **[j is in Zone E or D]**, let $j' = j-(M-M')\cdot\text{len}(u)$. Then, we have $\text{len}(w_1)+(M'-N)\cdot\text{len}(u) \leq j'$ **[j' is in Zone D or E]** 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 $k' < \text{len}(w_1) + N \cdot \text{len}(u)$ **[k' is in Zone A or B]**, 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 $k' \geq \text{len}(w_1) + (M' - N) \cdot \text{len}(u)$ **[k' is in Zone D or E]**, we choose $k = k' + (M - M') \cdot \text{len}(u)$ 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 $w_1 + N \cdot \text{len}(u) \leq k' < \text{len}(w_1) + (M' - N) \cdot \text{len}(u)$ **[k' is in Zone C]**, let $\ell = (k' - (\text{len}(w_1) + N \cdot \text{len}(u))) \mod \text{len}(u)$ (ℓ is the relative position of k' in the word u it belongs to) and let $k = \text{len}(w_1) + N \cdot \text{len}(u) + \ell$ (k is placed at the same relative position of k' in the first word u of the Zone C). Then we have $w_1 + N \cdot \text{len}(u) \leq k < \text{len}(w_1) + (M - N) \cdot \text{len}(u)$ and $|k - k'| = 0 \mod \text{len}(u)$ 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 $\text{len}(w_1) + N \cdot \text{len}(u) \leq i < \text{len}(w_1) + (M - N) \cdot \text{len}(u)$ **[i is Zone C]** then $\text{len}(w_1) + N \cdot \text{len}(u) \leq i' < \text{len}(w_1) + (M' - N) \cdot \text{len}(u)$ **[i' is in Zone C]** and $|i - i'| = 0 \mod \text{len}(u)$. Let $\ell = (i - (\text{len}(w_1) + N \cdot \text{len}(u))) \mod \text{len}(u)$ (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 $j < \text{len}(w_1) + (M - (N - 1)) \cdot \text{len}(u)$ **[j is in Zone C or D]**. We define $j' = i' + (j - i)$ and we have that $\text{len}(w_1) + N \cdot \text{len}(u) \leq j' < \text{len}(w_1) + (M' - (N - 1)) \cdot \text{len}(u)$ **[j' is in Zone C or D]** and since $|i-i'| = 0 \mod \text{len}(u)$, we deduce $|j-j'| = 0 \mod \text{len}(u)$. 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 $\text{len}(w_1)+N\cdot\text{len}(u) \leq k' < \text{len}(w_1)+(M'-(N-1))\cdot\text{len}(u)$ and $\text{len}(w_1)+N\cdot\text{len}(u) \leq k < \text{len}(w_1)+(M-(N-1))\cdot\text{len}(u)$. Since $|i - i'| = 0 \mod \text{len}(u)$, we also have $|k - k'| = 0 \mod \text{len}(u)$. Consequently $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.
  - 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 $j \geq \text{len}(w_1) + (M - N) \cdot \text{len}(u)$ **[j is in Zone E or D]**. Let $j' = j - (M - M') \cdot \text{len}(u)$ then $j' \geq \text{len}(w_1) + (M' - N) \cdot \text{len}(u)$ **[j' is in Zone E or D]** 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 $k' \geq$

$\text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[k' is in Zone D or E]**, then let $k = k'+(M-M')\cdot\text{len}(u)$; we have in this case that $k \geq \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ and this allows us to deduce that $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. Now assume $k' < \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[k' is in Zone C]** and $k'-i' < \text{len}(u)-\ell$ (k' and i' are in the same word u), then let $k = i + k' - i'$. In this case we have $k < \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ **[k is in Zone C]** and since $|i-i'| = 0 \mod \text{len}(u)$, we also have $|k-k'| = 0 \mod \text{len}(u)$, whence $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. Now assume $k' < \text{len}(w_1) + (M'-N)\cdot\text{len}(u)$ **[k' is in Zone C]** and $k'-i' \geq \text{len}(u)-\ell$ (k' and i' are not in the same word u). We denote by $\ell' = (k' - (\text{len}(w_1)+N\cdot\text{len}(u))) \mod \text{len}(u)$ the relative position of k' in u and let $k = i + (\text{len}(u)-\ell) + \ell'$ (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 $\ell' < \text{len}(u)$ and $j - i \geq \text{len}(u)-\ell+\text{len}(u)$) and $k < \text{len}(w_1)+(M-(N-1))\cdot\text{len}(u)$ (because $i + (\text{len}(u)-\ell) < \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ and $\ell' < \text{len}(u)$) and $|k-k'| = 0 \mod \text{len}(u)$ (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 $j - i \geq \text{len}(u)-\ell+\text{len}(u)$ (j is neither in the same word u as i nor in the next word u) and $j < \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ **[j is in Zone C]**. Then let $\ell' = (j-(\text{len}(w_1)+N\cdot\text{len}(u))) \mod \text{len}(u)$ the relative position of j in u. We choose $j' = i' + (\text{len}(u)-\ell)+\ell'$ (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 $j' < \text{len}(w_1)+(M'-(N-1))\cdot\text{len}(u)$ **[j' is in Zone C or D]** (because $i'+(\text{len}(u)-\ell) < \text{len}(w_1)+(M-N)\cdot\text{len}(u)$ and $\ell' < \text{len}(u)$) and $|j-j'| = 0 \mod \text{len}(u)$ (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 $k' \in [i', j'-1]$. If $k'-i' < \text{len}(u)-\ell$ (k' and i' are in the same word u), then let $k = i+k'-i'$. In this case we have $k < \text{len}(w_1)+(M-N)\cdot\text{len}(u)$ **[k is in Zone C]** and since $|i-i'| = 0 \mod \text{len}(u)$, we also have $|k-k'| = 0 \mod \text{len}(u)$, hence $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$. If $k'-i' \geq \text{len}(u)-\ell$ (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 $k < \text{len}(w_1) + (M-N)\cdot\text{len}(u)$ **[k is in Zone C]** and since $|j-j'| = 0 \mod \text{len}(u)$, we also have $|k-k'| = 0 \mod \text{len}(u)$, hence $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.

□

A.5 Proof of (Claim 4)

Before the proof, 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 $j \leq i$, 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 3) 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 $i < \text{len}(w_1)+N\cdot\text{len}(u)$ **[i is in Zone A or B]** then $j < \text{len}(w_1)+N\cdot\text{len}(u)$ **[j is in Zone A or B]** and $i' < \text{len}(w_1)+N\cdot\text{len}(u)$ **[i' is in Zone A or B]** 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 $i \geq \text{len}(w_1)+(M-N)\cdot\text{len}(u)$ **[i is Zone D or E]** then $i' \geq \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[i' is in Zone D or E]** and $i = i'+(M-M')\cdot\text{len}(u)$ and we have the following possibilities for the position $j \leq i$:
  - If $j \geq \text{len}(w_1)+(M-N)\cdot\text{len}(u)$ **[j is in Zone D or E]**, then let $j' = j-(M-M')\cdot\text{len}(u)$. 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 $k = k'+(M-M')\cdot\text{len}(u)$. 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 $\text{len}(w_1)+N\cdot\text{len}(u) \leq j < \text{len}(w_1)+(M-N)\cdot\text{len}(u)$ **[j is in Zone C]**, then let $\ell = (j-(\text{len}(w_1)+N\cdot\text{len}(u))) \mod \text{len}(u)$ (ℓ is the relative position of j in the word u it belongs to). Consequently $0 \leq \ell < \text{len}(u)$. Let $j' = \text{len}(w_1)+(M'-N)\cdot\text{len}(u)-(\text{len}(u)-\ell)$ (j' is at the same position as j in the last word u of the Zone C). Then $\text{len}(w_1)+N\cdot\text{len}(u) \leq j' < \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[j' is in Zone C]** (because $(M' \geq 2N+1)$ and $|j-j'| = 0 \mod \text{len}(u)$ (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 $k = k'+(M-M')\cdot\text{len}(u)$. Then we have that $k \in [j-1, i]$. Furthermore, if $k' \geq \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[k' is in Zone D or E]** then $k \geq \text{len}(w_1)+(M-N)\cdot\text{len}(u)$ **[k is in Zone D or E]** 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 $k' < \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ then necessarily $\text{len}(w_1)+N\cdot\text{len}(u) \leq k'$ **[k' is in Zone C]** (because $j' < k'$) and $|k-k'| = 0 \mod \text{len}(u)$ (because $k = k'+(M-M')\cdot\text{len}(u)$). Whence, k is in Zone C 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 $j < \text{len}(w_1)+N\cdot\text{len}(u)$ **[j is in Zone A or B]**, let $j' = j$. We have then $j' < \text{len}(w_1)+N\cdot\text{len}(u)$ **[j' is in Zone A or B]**. 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 $k' < \text{len}(w_1)+N\cdot\text{len}(u)$ **[k' is in Zone A]**, 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 $k' \geq \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[k' is in Zone D or E]**, we choose $k = k'+(M-M')\cdot\text{len}(u)$ 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 $w_1+N\cdot\text{len}(u) \leq k' < \text{len}(w_1)+(M'-N)\cdot\text{len}(u)$ **[k' is in Zone C]**, let $\ell = (k'-(\text{len}(w_1)+N\cdot\text{len}(u))) \mod \text{len}(u)$ (ℓ is the relative position of k' in the word u it belongs to) and let $k = \text{len}(w_1)+N\cdot\text{len}(u)+\ell$ (k is at the same position of k' in the first word of the zone C). Then we have $w_1+N\cdot\text{len}(u) \leq$

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

and $|j - j'| = 0 \mod \text{len}(u)$ (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 $k \geq \text{len}(w_1) + N \cdot \text{len}(u)$ **[k is in Zone C]** and since $|i - i'| = 0 \mod \text{len}(u)$, we also have $|k - k'| = 0 \mod \text{len}(u)$, 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 $\text{len}(w_1) + N \cdot \text{len}(u) \leq k < \text{len}(w_1) + (M - N) \cdot \text{len}(u)$ **[k is in Zone C]** and since $|j - j'| = 0 \mod \text{len}(u)$, we also have $|k - k'| = 0 \mod \text{len}(u)$, hence $\langle w, k\rangle \approx_{N-1} \langle w', k'\rangle$.

□

B Proof of Lemma 8

Proof. The proof is by reduction from the SAT problem. Let ϕ be a Boolean formula built over the propositional variables $AP = \{p_1, \cdots, p_n\}$. 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 Figure 5 so that the truth of the propositional variable 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

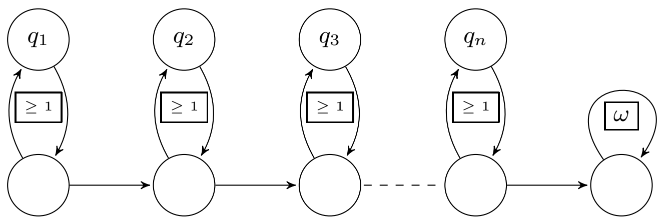


Fig. 5. A simple path schema P

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 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 p_i by $\mathtt{F}(q_i \wedge \mathtt{XX}q_i)$.

Let us check the correctness of the reduction. Let $v : AP \to \{\top, \bot\}$ be a valuation satisfying ϕ. Let us consider the run ρ respecting P such that $iter_P(\rho)[i] \stackrel{\text{def}}{=} 2$ if $v(p_i) = \top$, 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 $\rho, 0 \models \phi$, the valuation v satisfies ϕ where for all $i \in [1, n]$, we have $v(p_i) = \top \stackrel{\text{def}}{\Leftrightarrow} iter_P(\rho)[i] = 2$. □

C Proof of Lemma 11

Proof. The proof is by reduction from the problem SAT. Let ϕ be a Boolean formula built over the propositional variables in $\{p_1, \cdots, p_n\}$. We build a path schema $P \in \mathcal{CPS}(2)$, an initial configuration (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 Figure 6; 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 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 bit. Note that the internal loop has to be visited at least once since P is a path schema.

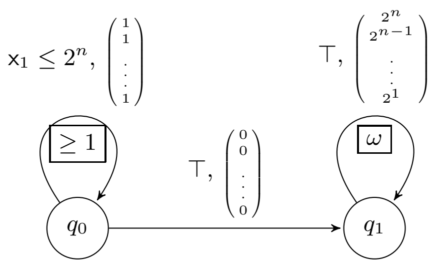


Fig. 6. Path schema P

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_u^i to denote the value in $[0, 2^i - 1]$ corresponding to the $i-1$ first bits of $\alpha - 1$. Similarly, we write α_d^i to denote the value in $[0, 2^{n+1-i} - 1]$ corresponding to the $(n+1-i)$ last bits of $\alpha - 1$. Observe that $\alpha - 1 = \alpha_u^i \times 2^{n-i+1} + \alpha_d^i$. One can show that (1.) the ith bit of $\alpha - 1$ is 1 iff (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 $2^{i-1} - \alpha_u^i$. Before showing the equivalence between (1.) and (2.), we can observe that condition (2.) can be expressed by the formula $\mathsf{F}(q_1 \wedge ((\mathsf{x}_i - 1) \geq 2^n + 2^{n-i}) \wedge ((\mathsf{x}_i - 1) \leq 2^n + 2^{n-i+1} - 1))$.

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, ith bit of $\alpha - 1$ is equal to 1 iff $\alpha_d^i \in [2^{n-i}, 2^{n+1-i} - 1]$. Now, observe that $(2^{i-1} - \alpha_u^i)2^{n+1-i} + (\alpha - 1) = 2^n + \alpha_d^i$. So, if (1.), then $\alpha_d^i \in [2^{n-i}, 2^{n+1-i} - 1]$ and consequently $2^n + \alpha_d^i \in [2^n + 2^{n-i}, 2^n + 2^{n+1-i} - 1]$. 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 $k = 2^{i-1} - \alpha_u^i$).

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, $k \times 2^{n+1-i} + (\alpha - 1) - 2^n \in [2^{n-i}, 2^{n+1-i} - 1]$ 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 p_i by $\mathtt{F}(q_1 \wedge ((\mathsf{x}_i - 1) \geq 2^n + 2^{n-i}) \wedge ((\mathsf{x}_i - 1) \leq 2^n + 2^{n-i+1} - 1))$. 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 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 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 $[2^n + 2^{n-i}, 2^n + 2^{n-i+1} - 1]$. This is explained Table 1 with $n = 4$.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
p_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
p_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
p_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
p_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table 1. Table showing the effect of last loop for 4 variables

Let us check the correctness of the reduction. Let $v : \{p_1, \ldots, p_n\} \rightarrow \{\top, \bot\}$. be a valuation satisfying ϕ. Let us consider the run ρ respecting P such that the first loop is taken $\alpha = (v(p_1)v(p_2) \cdots v(p_n))_2 + 1$ times and the initial counter values are all equal to zero. $\top$ is read as 1, $\bot$ as 0 and $(v(p_1)v(p_2) \cdots v(p_n))_2$ denotes the value of the natural number made of n bits in binary encoding. Hence, for every $i \in [1, n]$, the counter x_i contains the value α after the first loop. As noted earlier, $v(p_i) = 1$ implies that adding 2^{n-i+1} repeatedly to x_i in the last loop, we will hit $[2^n + 2^{n-i}, 2^n + 2^{n-i+1} - 1]$. Hence, the formula $\mathtt{F}(q_1 \wedge ((\mathsf{x}_i - 1) \geq 2^n + 2^{n-i}) \wedge ((\mathsf{x}_i - 1) \leq 2^n + 2^{n-i+1} - 1))$ will be satisfied by ρ iff $v(p_i) = 1$. It is easy to check thus, that $\rho, 0 \models \psi$. Conversely, if there is a run ρ respecting P such that $\rho, 0 \models \phi$ and the initial counter values are all equal to zero, the valuation v satisfies ϕ where for all $i \in [1, n]$, we have $v(p_i)$ iff the i^{th} bit in the binary encoding of $\alpha - 1$ is 1, where α is the number of times the first loop is taken. $\square$

D How to unfold path schemas to get rid of disjunctions or how to prove Theorem 14

D.1 Preliminary results on term maps

Definition 18. *Given a loop effect* $\boldsymbol{u} \in \mathbb{Z}^n$*, we define the relation* $\preceq_{\boldsymbol{u}}$ *on term maps such that* $\boldsymbol{m} \preceq_{\boldsymbol{u}} \boldsymbol{m}' \stackrel{\text{def}}{\Leftrightarrow}$ *for every term* $\mathtt{t} = \sum_i a_i \mathtt{x}_i \in T$*, we have* $\boldsymbol{m}(\mathtt{t}) \leq \boldsymbol{m}'(\mathtt{t})$ *if* $\sum_i a_i \boldsymbol{u}[i] \geq 0$*,* $\boldsymbol{m}(\mathtt{t}) \geq \boldsymbol{m}'(\mathtt{t})$ *if* $\sum_i a_i \boldsymbol{u}[i] \leq 0$ *and* $\boldsymbol{m}(\mathtt{t}) = \boldsymbol{m}'(\mathtt{t})$ *if* $\sum_i a_i \boldsymbol{u}[i] = 0$*. We write* $\boldsymbol{m} \prec_{\boldsymbol{u}} \boldsymbol{m}'$ *whenever* $\boldsymbol{m} \preceq_{\boldsymbol{u}} \boldsymbol{m}'$ *and* $\boldsymbol{m} \neq \boldsymbol{m}'$*.*

Sequences of strictly increasing term maps have bounded length.

Lemma 19. *Let* $\boldsymbol{u} \in \mathbb{Z}^n$ *and* $\boldsymbol{m}_1 \prec_{\boldsymbol{u}} \boldsymbol{m}_2 \prec_{\boldsymbol{u}} \cdots \prec_{\boldsymbol{u}} \boldsymbol{m}_L$*. Then,* $L \leq 2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)$*.*

Proof. For any loop effect $\mathbf{u} \neq 0$, each term $\mathtt{t}$ can be mapped by a $\boldsymbol{m}$ to $\mathrm{card}(I)$ different intervals as by definition for any two maps $\boldsymbol{m}$ and $\boldsymbol{m}'$, it can be either $\boldsymbol{m}(\mathtt{t}) \leq \boldsymbol{m}'(\mathtt{t})$ or $\boldsymbol{m}(\mathtt{t}) \geq \boldsymbol{m}'(\mathtt{t})$ but not both. Also, there are $\mathrm{card}(T)$ number of terms. 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 $\mathrm{card}(I) = 2 \times \mathrm{card}(B) + 1$. Hence, L, the number of different term maps in a sequence which is either increasing or decreasing, can be at most $2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)$. □

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

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

Lemma 20(I) below states that the relation $\vdash$ is easy to check whereas Lemma 20(II) states that $\vdash$ is complete with respect to the adequate notion of validity.

Lemma 20. *(I)* $\boldsymbol{m} \vdash \mathtt{g}$ *can be checked in* PTime *in* $\mathrm{size}(\boldsymbol{m}) + \mathrm{size}(\mathtt{g})$*.*
(II) $\boldsymbol{m} \vdash \mathtt{g}$ *iff for all* $v : \{\mathtt{x}_1, \mathtt{x}_2, \cdots, \mathtt{x}_n\} \to \mathbb{N}$*, (for all* $\mathtt{t} \in T$*,* $v(\mathtt{t}) \in \boldsymbol{m}(\mathtt{t})$*) implies* $v \models \mathtt{g}$*.*

It is worth noting that $\mathrm{size}(\boldsymbol{m})$ is in $\mathcal{O}(\mathrm{card}(I) \times \mathrm{card}(T))$.

Proof. (I) For the PTime algorithm we follow the following steps. First, for each constraint $\mathtt{t} \sim b$ appearing in $\mathtt{g}$, we replace it either $\top$ (true) or $\perp$ (false) depending whether $\boldsymbol{m} \vdash \mathtt{t} \sim b$ or not. After replacing all constraints, we are left with a positive Boolean formula whose atomic formulae are either $\top$ or $\perp$. It can be evaluated in logarithmic space in the size of the resulting formula (less than $\mathrm{size}(\mathtt{g})$).
Note that given a term map $\boldsymbol{m}$ and a constraint $\mathtt{t} \sim b$, checking $\boldsymbol{m} \vdash \mathtt{t} \sim b$ amounts to checking the containement of interval $\boldsymbol{m}(\mathtt{t})$ 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 size($\mathtt{t}$) + size($\boldsymbol{m}$). As the number of constraints is also bounded by size($\mathtt{g}$), the replacement of atomic constraints can be performed in polynomial time in size($\boldsymbol{m}$) + size($\mathtt{g}$). Thus, the procedure completes in time polynomial in size($\boldsymbol{m}$) + size($\mathtt{g}$).

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

  On the other hand, consider some valuation v with $v(\mathtt{t}) \in \boldsymbol{m}(\mathtt{t})$ for each $\mathtt{t} \in T$ and $v \models \mathtt{g}$. Similar to above, we will use inductive argument to show that $\boldsymbol{m} \vdash \mathtt{g}$
  - **Base Case:** Again consider arithmetical constraints of the guard. Specifically, we consider constraints of the form $\mathtt{t} \geq b$. As $v \models \mathtt{t} \geq b$, we know that $v(\mathtt{t}) \in [b, \infty)$. Since, $v(\mathtt{t}) \in \boldsymbol{m}(\mathtt{t})$, we have that, $\boldsymbol{m}(\mathtt{t}) \subseteq [b, \infty)$. Hence, $\boldsymbol{m} \vdash \mathtt{t} \geq b$. 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, $\boldsymbol{m} \vdash \mathtt{t} \sim b$.
  - **Inductive step:** Again, the induction step for $\wedge$ and $\vee$ follows easily.

□

D.2 Proof of Lemma 13

Proof. The proof is by structural induction.

- Base Case 1 ($p \in X$): we have the following equivalences:
  - $\rho, i \models p$,
  - $p \in \mathbf{l}(q_i)$ (by definition of $\models$),
  - $p \in \pi_1(\mathsf{ft}(i))$ (by definition of $\mathsf{ft}(\rho)$),
  - $\mathsf{ft}(\rho), i \models_{\mathsf{symb}} p$ (by definition of $\models_{\mathsf{symb}}$).
- Base Case 2 ($\sum_j a_j \mathsf{x}_j \leq b$ with $\sum_j a_j \mathsf{x}_j \in T$ and $b \in B$): we have the following equivalences:
  - $\rho, i \models \sum_j a_j \mathsf{x}_j \leq b$,
  - $\sum_j a_j \boldsymbol{v_i}[j] \leq b$ (by definition of $\models$),
  - $\pi_2(\mathsf{ft}(i))(\boldsymbol{m}_i(\sum_j a_j \mathsf{x}_j)) \subseteq (-\infty, b]$ (by definition of $\mathsf{ft}(\rho)$),
  - $\mathsf{ft}(\rho), i \models_{\mathsf{symb}} \sum_j a_j \mathsf{x}_j \leq b$ (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 $\models_{\mathsf{symb}}$).

Then, it is immediate that if ρ' is an infinite run such that $\mathsf{ft}(\rho) = \mathsf{ft}(\rho')$, then $\rho, i \models \phi$ iff $\rho', i \models \phi$. □

D.3 Building the set Y_P

Unfolding a path schema Let $\mathsf{R} = \langle X, T, B \rangle$ be a resource and

$$P = p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^\omega$$

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

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

We write $\mathtt{G}^\star(T, B, U)$ to denote the set of guards of the form $\psi(\mathtt{t}, \mathbf{u}, \boldsymbol{m}(\mathtt{t}))$ where $\mathtt{t} \in T$, U is the finite set of updates from P and $\boldsymbol{m} : T \to I$. Each guard in $\mathtt{G}^\star(T, B, U)$ is of linear 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 $\langle q, \boldsymbol{m} \rangle \xrightarrow{\delta, \langle \mathtt{g}_{\boldsymbol{m}'}, \mathbf{u} \rangle} \langle q', \boldsymbol{m}' \rangle \in \Delta'$, the conditions below are satisfied: $\stackrel{\text{def}}{\Leftrightarrow}$

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

We extend the definition of $\mathit{source}(\delta)$ to $\delta' = \langle q, \boldsymbol{m} \rangle \xrightarrow{\delta, \langle \mathtt{g}_{\boldsymbol{m}'}, \mathbf{u} \rangle} \langle q', \boldsymbol{m}' \rangle \in \Delta'$. We define $\mathit{source}(\delta') = \langle q, \boldsymbol{m} \rangle$ and $\mathit{target}(\delta') = \langle q', \boldsymbol{m}' \rangle$. Similarly, for a finite word $w \in (\Delta')^*$, we define $\mathit{source}(w) = \mathit{source}(w(1))$ and $\mathit{target}(w) = \mathit{target}(w(\mathrm{len}(w)))$.

A *skeleton* (compatible with P and $\langle q_0, \boldsymbol{v_0} \rangle$) $\mathtt{sk}$, say $\langle q_1, \boldsymbol{m}_1 \rangle \xrightarrow{\delta_1, \langle \mathtt{g}^1_{\boldsymbol{m}'}, \mathbf{u}_1 \rangle} \langle q_2, \boldsymbol{m}_2 \rangle \xrightarrow{\delta_2, \langle \mathtt{g}^2_{\boldsymbol{m}'}, \mathbf{u}_2 \rangle} \langle q_3, \boldsymbol{m}_3 \rangle \cdots \xrightarrow{\delta_K, \langle \mathtt{g}^K_{\boldsymbol{m}'}, \mathbf{u}_K \rangle} \langle q_{K+1}, \boldsymbol{m}_{K+1} \rangle$, is a finite word over Δ' such that

(init) For every term $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$, we have $\sum_j a_j \boldsymbol{v_0}[j] \in \boldsymbol{m}_1(\mathtt{t})$ where $\boldsymbol{v_0}$ is the initial vector.

(schema) Let $f : (\Delta')^* \to \Delta^*$ be the map such that $f(\varepsilon) = \varepsilon$, $f(w \cdot w') = f(w) \cdot f(w')$ and $f(\langle q, \boldsymbol{m} \rangle \xrightarrow{\delta, \langle \mathtt{g}_{\boldsymbol{m}'}, \mathbf{u} \rangle} \langle q', \boldsymbol{m}' \rangle) = \delta$. We require that $f(\mathtt{sk}) \in p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^+$.

(minimality) For every factor

$$w = \langle q_I, \boldsymbol{m_I} \rangle \xrightarrow{\delta_I, \langle \mathtt{g}^{I}_{\boldsymbol{m}'}, \mathbf{u}_I \rangle} \langle q_{I+1}, \boldsymbol{m}_{I+1} \rangle \cdots \xrightarrow{\delta_{J-1}, \langle \mathtt{g}^{J-1}_{\boldsymbol{m}'}, \mathbf{u}_{J-1} \rangle} \langle q_J, \boldsymbol{m}_J \rangle$$

of $\mathtt{sk}$ such that $f(w) = (l)^3$ for some loop l of P (therefore $J = I+3\times\mathrm{len}(l)$), there is $\alpha \in [1, \mathrm{len}(l)]$ such that $\boldsymbol{m}_{I+\alpha} \prec_{\mathit{effect}(l)} \boldsymbol{m}_{I+\alpha+2\times\mathrm{len}(l)}$.

(last-loop) For the unique suffix w of $\mathtt{sk}$ of length $\mathrm{len}(l_k)$, we have $f(w) = l_k$ and $\mathit{source}(w) = \mathit{target}(w)$.

Lemma 21. *For a skeleton* $\mathtt{sk}$*,* $\mathrm{len}(\mathtt{sk}) \leq (\mathrm{len}(p_1) + \cdots + \mathrm{len}(p_k)) + 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)) \times (\mathrm{len}(l_1) + \cdots + \mathrm{len}(l_k))$

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 $\mathrm{len}(\mathtt{sk}) \leq (\mathrm{len}(p_1)+\cdots+\mathrm{len}(p_k)) + \mathit{max}(n_i) \times (\mathrm{len}(l_1) + \cdots + \mathrm{len}(l_k))$. It remains to bound the values among $n_1, \dots, n_k$. 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 19, we conclude that $n_i \leq 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T))$. Consequently, $\mathrm{len}(\mathtt{sk}) \leq (\mathrm{len}(p_1)+\cdots+\mathrm{len}(p_k)) + 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)) \times (\mathrm{len}(l_1) + \cdots + \mathrm{len}(l_k))$. ☐

We have furthermore the following Lemma concerning skeletons.

Lemma 22. *Checking whether a word* $w \in (Q' \times \Delta \times \mathtt{G}^\star(T,B,U) \times U \times Q')^*$ *is a skeleton compatible with* P *and* $\langle q_0, \boldsymbol{v_0} \rangle$ *assuming* $\mathrm{len}(w) \leq (\mathrm{len}(p_1) + \cdots + \mathrm{len}(p_k)) + 2(2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)) \times (\mathrm{len}(l_1) + \cdots + \mathrm{len}(l_k))$ *can be done in polynomial time in the size of* $\langle q_0, \boldsymbol{v_0} \rangle$*,* P*,* T *and* B*.*

Proof. Let w be a word over $Q' \times \Delta_P \times \mathtt{G}^\star(T,B,U) \times U \times Q'$ 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 the respective sizes of $\langle q_0, \boldsymbol{v_0} \rangle$, P, T and K. Since the length of w is bounded, its size is also polynomial in N.

Checking whether an element in $Q' \times \Delta_P \times \mathtt{G}^\star(T,B,U) \times U \times Q'$ belongs to Δ' can be done in polynomial time in N thanks to Lemma 20(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 $f(w)$ (requires linear time in N) and then by checking whether it belongs to $p_1 l_1^+ p_2 l_2^+ \dots p_k l_k^+$ (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.

☐

From skeletons, we shall define path schemas built over the alphabet $\tilde{\Delta} = Q' \times \mathtt{G}^\star(T, B, U) \times U \times Q'$ (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 $h : (\Delta')^* \to (\tilde{\Delta})^*$ be the map such that $h(\varepsilon) = \varepsilon$, $h(w \cdot w') = h(w) \cdot h(w')$ and $h(\langle q, \boldsymbol{m}\rangle \xrightarrow{\delta, \langle \mathtt{g}_{\boldsymbol{m}'}, \mathbf{u}\rangle} \langle q', \boldsymbol{m}'\rangle) = \langle q, \boldsymbol{m}\rangle \xrightarrow{\langle \mathtt{g}_{\boldsymbol{m}'}, \mathbf{u}\rangle} \langle q', \boldsymbol{m}'\rangle$. 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.

For every factor $w = \langle q_I, \boldsymbol{m}_I\rangle \xrightarrow{\delta_I, \langle \mathtt{g}^I_{\boldsymbol{m}'}, \mathbf{u}_I\rangle} \langle q_{I+1}, \boldsymbol{m}_{I+1}\rangle \cdots \xrightarrow{\delta_{J-1}, \langle \mathtt{g}^{J-1}_{\boldsymbol{m}'}, \mathbf{u}_{J-1}\rangle} \langle q_J, \boldsymbol{m}_J\rangle$ 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 $\mathrm{len}(l_k)$,
3. the sequence of the $\mathrm{len}(l)$ next elements after w is also equal to w,

we replace w^2 (the two consecutive repetitions of w) by $(h(w))^+$. Finally, $l'_{k'}$ is equal to $h(w)$ where w is the unique suffix of $\mathtt{sk}$ of length $\mathrm{len}(l_k)$. Note that 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 labelling function is concerned, the labels of q and $\langle q, \boldsymbol{m}\rangle$ are identical as far as X is concerned, i.e. $\mathbf{l}'(\langle q, \boldsymbol{m}\rangle) \stackrel{\text{def}}{=} \mathbf{l}(q) \cap X$. Hence,

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

We define the set Y_P as the following set of path schemas over the transitions $\tilde{\Delta}$: $Y_P = \{P_{\mathtt{sk}} \mid \mathtt{sk}$ is a skeleton compatible with P and $\langle q_0, \boldsymbol{v_0}\rangle\}$; it corresponds to the set of unfolded path schemas obtained from P.

As an example in Figure 7, we consider the path schema P and two of its unfoldings P' and P'' such that both belong to Y_P (the initial counter value is zero). Even though both P' and P'' are path schemas, there is no valid run respecting P' whereas there is one respecting P''.

D.4 Properties of the set Y_P

Note that in the sequel we assume that the labelling function $\mathbf{l}$ associated to a run respecting a path schema in Y_P is such that for any $\langle q, \boldsymbol{m}\rangle \in Q'$, $\mathbf{l}(q, \boldsymbol{m})$ is equal to $\mathbf{l}(q)$. This allows us to compare the footprints of the runs respecting P with the footprints of runs respecting a path schema in Y_P.

The main property about Y_P is stated below.

Proposition 23. (I) *Let ρ be an infinite run respecting P and starting at $\langle q_0, \boldsymbol{v_0}\rangle$. Then, there are a path schema P' in Y_P and an infinite run ρ' respecting P' such that $\mathsf{ft}(\rho) = \mathsf{ft}(\rho')$.*

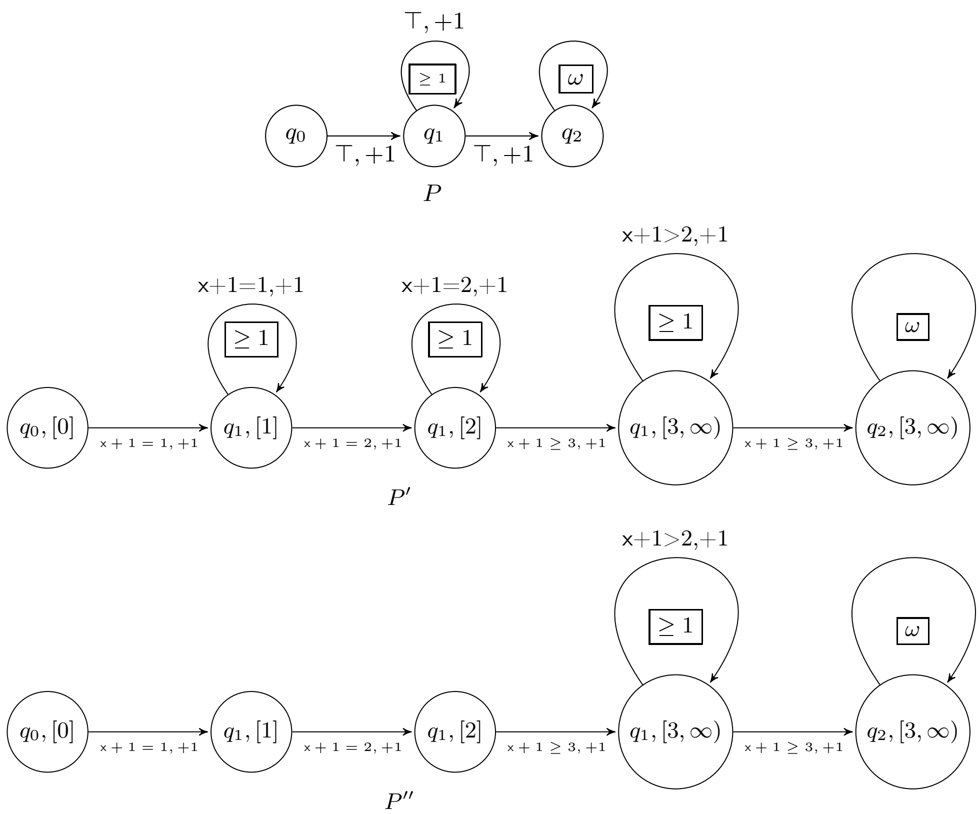


Fig. 7. Apath schema P with two path schemas in Y_P

(II) *Let ρ be an infinite run respecting P' for some $P' \in Y_P$. Then, there is an infinite run ρ' respecting P such that* $\mathsf{ft}(\rho) = \mathsf{ft}(\rho')$.

Proof.

(I) Let $\rho = \langle q_0, \boldsymbol{v_0} \rangle \xrightarrow{\delta_0} \langle q_1, \boldsymbol{v_1} \rangle \xrightarrow{\delta_1} \cdots$ be an infinite run respecting P with footprint $\mathsf{ft}(\rho) : \mathbb{N} \to 2^{AT} \times I^T$. We write $\langle Z_i, \boldsymbol{m}_i \rangle$ 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 $w = \langle q_0, \boldsymbol{m}_0 \rangle \xrightarrow{\delta_0, \langle \mathtt{g}_{\boldsymbol{m}_1}, update(\delta_0) \rangle} \langle q_1, \boldsymbol{m}_1 \rangle \xrightarrow{\delta_1, \langle \mathtt{g}_{\boldsymbol{m}_2}, update(\delta_1) \rangle} \cdots$. 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 $Q' \times \Delta_P \times \mathtt{G}^\star(T, B, U) \times U \times Q'$ 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 $\langle q_0, \boldsymbol{v_0} \rangle$. $\mathtt{sk}$ 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 $i \in [1, k-1]$ and $1 \leq n'_k \leq 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T))$. There exists $I \geq 1$ such that $w = w' \cdot w_0 \cdot w_0 \cdot (w_0)^\omega$ with $f(w_0) = l_k$. 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 thay $\mathtt{sk}$ is indeed a skeleton compatible with P and $\langle q_0, \boldsymbol{v_0} \rangle$. Let us consider that $\mathtt{sk}$ can be written as

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

Considering the path schema $P_{\mathtt{sk}}$ built from $\mathtt{sk}$, one can show that the sequence ρ' below is an infinite run respecting $P_{\mathtt{sk}}$:

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

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 $\mathrm{len}(l_k)$ last 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 labellings, 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 but this is guaranteed by the way guards are defined and by the completeness result in Lemma 20(II).

(II) Let ρ be some run respecting some $P' \in Y_P$ of the form below:

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

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 verification, one can show that $\rho = \langle q_0, \boldsymbol{v_0} \rangle \xrightarrow{\delta_0} \langle q_1, \boldsymbol{v_1} \rangle \xrightarrow{\delta_1} \cdots$ 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_P can be viewed as specific instances of P that differ in the way the term maps evolve. Details are omitted.

□

We define the function proj over infinite words of the alphabet of transitions $\tilde{\Delta} = Q' \times \mathtt{G}^\star(T, B, U) \times U \times Q'$ as follows: for each $w \in \tilde{\Delta}^\omega$, $\mathsf{proj}(w) : \mathbb{N} \to 2^{\mathrm{AT}} \times I^T$ satisfies that for all $i \in \mathbb{N}$, if $w(i) = \langle \langle q, \boldsymbol{m} \rangle, \mathtt{g}, \mathbf{u}, \langle q', \boldsymbol{m'} \rangle \rangle$ and $\mathbf{l}(q) \cap X = L$ then $\mathsf{proj}(w)(i) \stackrel{\text{def}}{=} \langle L, \boldsymbol{m} \rangle$. We have then the following lemma:

Lemma 24. *If $P' = p'_1(l'_1)^+ p'_2(l'_2)^+ \dots p'_{k'}(l'_{k'})^\omega$ is a path schema in Y_P and ρ is a run $\langle \langle q_0, \boldsymbol{m}_0 \rangle, \boldsymbol{v_0} \rangle \xrightarrow{\langle \mathtt{g}_{\boldsymbol{m}_1}, update(\delta_0) \rangle} \langle \langle q_1, \boldsymbol{m}_1 \rangle, \boldsymbol{v_1} \rangle \xrightarrow{\langle \mathtt{g}_{\boldsymbol{m}_2}, update(\delta_1) \rangle} \langle \langle q_2, \boldsymbol{m}_2 \rangle, \boldsymbol{v_2} \rangle \cdots$ respecting P' we have that $\mathsf{proj}(lab(\rho)) = \mathsf{ft}(\rho)$.*

Proof. We will prove that for all $i \in \mathbb{N}$, we have $\mathsf{ft}(\rho)(i) = \mathsf{proj}(lab(\rho))(i)$. For $i = 0$, we have $\mathsf{ft}(\rho)(0) = \langle \mathbf{l}(q_0) \cap X, \boldsymbol{m} \rangle$ and using the definition of a skeleton, the **(init)** case tells us that necessarily, $\boldsymbol{m} = \boldsymbol{m}_0$. Hence we have $\mathsf{ft}(\rho)(0) = \mathsf{proj}(lab(\rho))(0)$. We will now prove the property holds for $i+1$ with $i \in \mathbb{N}$. By definition of the function $\mathsf{ft}(\cdot)$, we have $\mathsf{ft}(\rho)(i+1) = \langle \mathbf{l}(q_i) \cap X, \boldsymbol{m} \rangle$ and in the run ρ we have $\langle \langle q_i, \boldsymbol{m}_i \rangle, \boldsymbol{v_i} \rangle \xrightarrow{\langle \mathtt{g}_{\boldsymbol{m}_i}, update(\delta_i) \rangle} \langle \langle q_{i+1}, \boldsymbol{m}_{i+1} \rangle, \boldsymbol{v_{i+1}} \rangle$. We know than from how we build Y_P that there exists a transition in Δ' of the form $\langle q_i, \boldsymbol{m}_i \rangle \xrightarrow{\delta, \langle \mathtt{g}_{\boldsymbol{m}_i}, update(\delta_i) \rangle} \langle q_{i+1}, \boldsymbol{m}_{i+1} \rangle$, and by definition of the set Δ', $\mathtt{g}_{\boldsymbol{m}_i}$ is a guard that states that after the update $update(\delta_i)$, for each $\mathtt{t} \in T$, its value belongs to $\boldsymbol{m}_{i+1}(\mathtt{t})$. Hence we can deduce that $\boldsymbol{m} = \boldsymbol{m}_{i+1}$ and consequently $\mathsf{ft}(\rho)(i+1) = \mathsf{proj}(lab(\rho))(i+1)$. □

D.5 Proof of Theorem 14

Proof. Let Y_P be the set of path schemas defined from the minimal path schema P.

1. For every path schema in Y_P, the guards on transitions are of the form $\bigwedge_{\mathtt{t} \in T} \psi(\mathtt{t}, \mathbf{u}, \boldsymbol{m}(\mathtt{t}))$ and each guard $\psi(\mathtt{t}, \mathbf{u}, \boldsymbol{m}(\mathtt{t}))$ is itself an atomic guard and a conjunction of two atomic guards. Hence, no path schema in Y_P contains guards with disjunctions in it.
2. By Lemma 21, every skeleton defining a path schema in Y_P has polynomial length in $\mathrm{len}(P) + \mathrm{card}(T) + \mathrm{card}(B)$. Each path schema in Y_P has a linear length in the length of its corresponding skeleton. Consequently, for $P' \in Y_P$, 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_P, 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 22, checking whether $\mathtt{sk}$ is compatible with P and $\langle q_0, \boldsymbol{v_0} \rangle$ can be done in polynomial time in $\mathrm{size}(P) + \mathrm{card}(T) + \mathrm{card}(B)$. In particular, if $\mathtt{sk}$ is too long, this can be checked in polynomial time too.

4. By Proposition 23(I), for every run ρ respecting P and starting at $\langle q_0, \boldsymbol{v_0} \rangle$, there are $P' \in Y_P$ and a run ρ' respecting P' such that $\mathsf{ft}(\rho) = \mathsf{ft}(\rho')$. By Lemma 13, $\rho \models \phi$ iff $\rho' \models \phi$.
5. Similar to (4.) by using Proposition 23(II).
6. We consider an ultimately periodic word $w \cdot u^\omega \in \mathcal{L}(P')$. From it 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 [17], 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.

□

E Proof of Lemma 12

Proof. Let us build a constraint system $\mathcal{E}$ defined from P that characterizes the set $iter_P(c_0)$ included in $\mathbb{N}^{k-1}$ for some initial configuration $c_0 = \langle q_0, \boldsymbol{v_0} \rangle$. For $\alpha \in [1,k]$ and $i \in [1,n]$, we write $\mathit{effect}^<(l_\alpha)[i]$ to denote the term below:

$$\boldsymbol{v_0}[i] + (\mathit{effect}(p_1) + \cdots + \mathit{effect}(p_\alpha))[i] + \mathit{effect}(l_1)[i]\mathsf{y}_1 + \ldots + \mathit{effect}(l_{\alpha-1})[i]\mathsf{y}_{\alpha-1}$$

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

$$\boldsymbol{v_0}[i] + (\mathit{effect}(p_1) + \cdots + \mathit{effect}(p_{\alpha-1}))[i] + \mathit{effect}(l_1)[i]\mathsf{y}_1 + \ldots + \mathit{effect}(l_{\alpha-1})[i]\mathsf{y}_{\alpha-1}$$

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

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

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

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

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

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

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

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

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

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

- For each $\alpha \in [1, k - 1]$ and each $\beta \in [0, \text{len}(l_\alpha) - 1]$, the value of counter i just before entering for the last time in the $(\beta + 1)$th transition of l_α is non-negative:

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

Convexity guarantees that this is sufficient to non-negativity.

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

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 S.

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

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

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

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

Let K be the maximal absolute value of constants occurring in S and $\boldsymbol{v_0}$. Let us 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 $\text{len}(P)$ transitions and therefore the maximal absolute value occurring in $\mathit{effect}(p)$ is at most $K \times \text{len}(P)$.

– The maximal absolute value occurring in $\mathit{effect}^{<}(p)$ is at most $(K \times \mathrm{len}(P)) \times (K + k \times K)$.

Consequently, the maximal absolute value of constants in $\mathcal{E}$ is bounded by $n \times (K \times (K \times \mathrm{len}(P)) \times (K + k \times K))$, which is bounded by $n \times k \times K^4 \times \mathrm{len}(P)^3$. 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 $k \leq \mathrm{card}(Q) \leq \mathrm{size}(S)$.

($\star$) Let $\rho = \langle q_0, \boldsymbol{v_0} \rangle \langle q_1, \boldsymbol{v_1} \rangle \langle q_2, \boldsymbol{v_2} \rangle \cdots$ be an infinite run respecting the path schema P with $c_0 = \langle q_0, \boldsymbol{v_0} \rangle$. We write $V : \{\mathsf{y}_1, \ldots, \mathsf{y}_{k-1}\} \to \mathbb{N}$ to denote the valuation such that for $\alpha \in [1, k-1]$, we have $V(\mathsf{y}_\alpha) = \mathit{iter}_P(\rho)[\alpha]$. V is extended naturally to terms built over variables in $\{\mathsf{y}_1, \ldots, \mathsf{y}_{k-1}\}$, 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 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 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, each $\beta \in [0, \mathrm{len}(p) - 1]$ and each atomic guard $\sum_i a_i \mathsf{x}_i \sim b \in \mathit{guard}(p(j))$, 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 $\alpha \in [1, k - 1]$, each $\beta \in [0, \mathrm{len}(l_\alpha) - 1]$ and 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 $V : \{\mathsf{y}_1, \ldots, \mathsf{y}_{k-1}\} \to \mathbb{N}$ 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 $\rho' = \langle q_0, \boldsymbol{x_0} \rangle \langle q_1, \boldsymbol{x_1} \rangle \langle q_2, \boldsymbol{x_2} \rangle \cdots \in (Q \times \mathbb{Z}^n)^{\omega}$, that will be later shown to be an infinite run respecting the path schema P with $c_0 = \langle q_0, \boldsymbol{v_0} \rangle$. Here is how ρ' is defined:

– For every $i \geq 0$, $q_i \stackrel{\text{def}}{=} \mathit{source}(w(i))$,
– $\boldsymbol{x_0} \stackrel{\text{def}}{=} \boldsymbol{v_0}$ and for every $i \geq 1$, we have $\boldsymbol{x_i} \stackrel{\text{def}}{=} \boldsymbol{x_{i-1}} + \mathit{update}(w(i))$.

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 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 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 $\boldsymbol{z}, \boldsymbol{z} + V(\mathsf{y}_\alpha)\mathit{effect}(l_\alpha) \geq 0$, then for $j \in [2, V(\mathsf{y}_\alpha) - 1]$, we have $\boldsymbol{z} + j \times \mathit{effect}(l_\alpha) \geq 0$ (convexity). Consequently, for $i \geq 0$, we have $\boldsymbol{x_i} \geq \mathbf{0}$.
2. Similarly, counter values should satisfy the guards for each fired transition. Since $V \models \mathcal{E}_3$, for each segment p in P, each $\beta \in [0, \mathrm{len}(p) - 1]$ and each atomic guard $\sum_i a_i \mathsf{x}_i \sim b \in \mathit{guard}(p(j))$, counter values satisfy it the first time the transition is visited. Moreover, since $V \models \mathcal{E}_3$, for each $\alpha \in [1, k-1]$, each $\beta \in [0, \mathrm{len}(l_\alpha) - 1]$ and 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 $\sum_i a_i \boldsymbol{z}[i] \sim b$ and $\sum_i a_i(\boldsymbol{z} + V(\mathsf{y}_\alpha)\mathit{effect}(l_\alpha))[i] \sim b$, then for $j \in [2, V(\mathsf{y}_\alpha) - 1]$, we have $\sum_i a_i(\boldsymbol{z} + j\mathit{effect}(l_\alpha))[i] \sim b$ (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 $V \models \mathcal{E}_1$. Indeed, by definition, each loop has to be visited at least once.

□

F How $p^\star(\cdot)$ is defined

Let us explain below how $p^\star(\cdot)$ is defined. Let S be a flat counter system, $c_0 = \langle q_0, \boldsymbol{v_0} \rangle$ be an initial configuration and $\phi \in \mathrm{PLTL}[\mathtt{C}]$. Let $N = \mathrm{size}(S) + \mathrm{size}(\langle q_0, \boldsymbol{v_0} \rangle) + \mathrm{size}(\phi)$. 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 S and 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.

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

Now, let P' be a path schema in Y_P with $P' = p_1(l_1)^+ p_2(l_2)^+ \cdots p_k(l_k)^\omega$. Since $\mathrm{len}(P') \leq (\mathrm{len}(p_1)+\cdots+\mathrm{len}(p_k)) + 2 \times (2 \times \mathrm{card}(T) \times \mathrm{card}(B) + \mathrm{card}(T)) \times (\mathrm{len}(l_1)+\cdots+\mathrm{len}(l_k))$, 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 MAX' occurring in P' is bounded by $MAX + n \times MAX^2$ which is bounded by $N \times 2^{2 \times N}$. Let $\mathcal{E}$ be the constraint system defined from P'.

- 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 $n \times \mathrm{nbloops}(P') \times (MAX')^4 \times \mathrm{len}(P')^3$, which is bounded by $N(5N^3)(N \times 2^{2 \times N})^4 \times 5^3 N^9 \leq 625 \times N^{17} \times 2^{8 \times N}$.

Let us show that $\mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$ admits a small solution using the theorem below for any $\psi_1 \wedge \cdots \wedge \psi_{k-1}$ built from Algorithm 1.

Theorem 25. *[2] Let $\mathcal{M} \in [-M, M]^{U \times V}$ and $\boldsymbol{b} \in [-M, M]^U$, where $U, V, M \in \mathbb{N}$. If there is $\boldsymbol{x} \in \mathbb{N}^V$ such that $\mathcal{M}\boldsymbol{x} \geq \boldsymbol{b}$, then there is $\boldsymbol{y} \in [0, (\max\{V, M\})^{\mathsf{C}U}]^V$ such that $\mathcal{M}\boldsymbol{y} \geq \boldsymbol{b}$, where C is some constant.*

By Theorem 25, $\mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$ has a solution iff $\mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$ 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 $\sum_i a_i \mathsf{y}_i \geq b$.

G Proof of Lemma 15

Proof. First, let us 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 14(2)).
- $\boldsymbol{y}$ and $\boldsymbol{y'}$ are obviously of polynomial size since their components have values bounded by some exponential expression (values in $\boldsymbol{y}$ can be much smaller than the values in $\boldsymbol{y'}$).

Now, let us 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 Lemma 22).
- Checking whether $\mathsf{proj}(p_1 l_1^{\boldsymbol{y}[1]} p_2 l_2^{\boldsymbol{y}[2]} \ldots l_{k-1}^{\boldsymbol{y}[k-1]} p_k l_k^{\omega}), 0 \models_{\mathsf{symb}} \phi$ can be done in polynomial time using Theorem 14(6) since $p_1 l_1^{\boldsymbol{y}(1)} p_2 l_2^{\boldsymbol{y}(2)} \ldots l_{k-1}^{\boldsymbol{y}(k-1)} p_k l_k$ is of polynomial size with respect to the size of P' and ϕ.
- Building $\mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$ can be done in polynomial time since $\mathcal{E}$ can be built in polynomial time with respect to the size of P' and $\psi_1 \wedge \cdots \wedge \psi_{k-1}$ can be built in polynomial time with respect to the size of ϕ ($td(\phi) \leq \mathrm{size}(\phi)$).
- $\boldsymbol{y'} \models \mathcal{E} \wedge \psi_1 \wedge \cdots \wedge \psi_{k-1}$ can be finally checked in polynomial time since the values in $\boldsymbol{y'}$ are of exponential magnitude and the combined constraint system is of polynomial size.

□

H MC(PLTL[C], $\mathcal{CPS}(1)$) is in PTime

Proof. Consider a path schema $P = p.l^{\omega}$ 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. $\mathsf{ft}(\rho)$ (defined in Section 5.2) 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 size of the input, but $\mathrm{len}(u)$ can be exponential. But, note that $lab(\rho(0)\rho(1)\cdots\rho(\mathrm{len}(u))) \in p \cdot l^{+}$ where the number of repetitions of l may be an exponential number of times. The algorithm computes the number of different possible sets of term maps (defined in Section 5.2), that the nodes of l can have. At most, this can be polynomially many times due to the monotonocity of guards and counter constraints. Next, for each such assignment i of term maps to the nodes of l, the algorithm calculates the number of iterations nl_i of l, for which the terms remain in their respective term map. Note that each of these nl_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 defining the algorithm formally, we need to define some notions to be used in the algorithm. For a path segment $p = \delta_1 \delta_2 \cdots \delta_{\mathrm{len}(p)}$, we define $p[i,j] = \delta_i \delta_{i+1} \cdots \delta_j$ for $1 \leq i \leq j \leq |p|$. Also, for a loop segment l, we say a tuple of term maps $(\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_{\mathrm{len}(l)})$ is *final* iff for every term $\mathtt{t} = \sum_j a_j \mathsf{x}_j \in T$ and for all $1 \leq i \leq \mathrm{len}(l)$,

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

where $\mathit{effect}(l)$ is as defined in Section 4.

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 $f_{\mathrm{init}} : \{0, 1, 2, \ldots, \mathrm{len}(w)\} \rightarrow I^T$ for a given configuration $c = \langle q_0, \mathbf{v}_0 \rangle$, defined as:

- $f_{\text{init}}(0) = \boldsymbol{m}_0$ iff for each term $\mathtt{t} = \sum_j a_j \mathtt{x}_j \in T$, we have that, $\sum_j a_j.\mathbf{v}_0[j] \in \boldsymbol{m}_0(\mathtt{t})$ and $\boldsymbol{m}_0 \vdash guard(w(0))$.
- for $1 \leq i \leq \text{len}(w)$ as $f_{\text{init}}(i) = \boldsymbol{m}_i$ iff, for each term $\mathtt{t} = \sum_j a_j \mathtt{x}_j \in T$, we have that, $\sum_j a_j.(effect(w[1,i])[j] + \mathbf{v}_0[j]) \in \boldsymbol{m}_i(\mathtt{t})$ and $\boldsymbol{m}_i \vdash guard(w(i))$.
- 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.

Also, we consider the function $curr : T \to \mathbb{Z}$ which, in the algorithm, gives the value of the terms at specific positions of the run. The function $val_{curr} : \Delta^+ \to I^T$, is defined as $val_{curr}(w) = \boldsymbol{m}$ where for all $\mathtt{t} = \sum_j a_j \mathtt{x}_j \in T, curr(\mathtt{t}) + \sum_j a_j.(effect(w)[j]) \in \boldsymbol{m}(\mathtt{t})$. For a path segment $p = \delta_1 \delta_2 \cdots \delta_{\text{len}(p)}$ with $\delta_i = (q_i, \mathtt{g}_i, \mathbf{u}_i, q_{i+1}) \in \Delta$ for $i \in [1, \text{len}(p)]$ and a tuple of term maps $a = (\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_{\text{len}(p)})$, we define $p \times a = \delta'_1 \delta'_2 \cdots \delta'_{\text{len}(p)}$ where $\delta'_i = (\langle q_i, \boldsymbol{m}_i \rangle, \mathtt{g}_i, \mathbf{u}_i, \langle q_{i+1}, \boldsymbol{m}_{i+1} \rangle)$.

Given an initial configuration c, we calculate the term maps for each position of p and the first iteration of l, using f_{init}. Subsequently, we calculate new tuples of term maps $(\boldsymbol{m}_1, \boldsymbol{m}_2 \cdots \boldsymbol{m}_{\text{len}(l)})$ for l and the number of iterations nl of l for which the terms remain in their respective term map from the tuple. We store the tuple of term maps in an array A and the number of iterations corresponding to tuple i in nl_i. In case, at any position, we reach some term maps that does not satisfy some guard, the procedure is aborted as it signifies that there does not exist any run. Note that there are polynomially many entries in A but each of the nl_i can be exponential. We perform symbolic model checking over a path schema augmented with the calculated term maps. The augmented path schema is obtained by performing $l \times A[i]$ for each i. But the number of times $l \times A[i]$ is repeated, nl_i can be exponential. Thus, instead of taking $l \times A[i]$, nl_i times, we take it $\mathtt{Min}(nl_i, 2td(\phi) + 5)$ times. By Theorem 3, 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 completes in PTime and is correct.

Lemma 26. *Algorithm 2 terminates in time which is at most a polynomial in the size of the input.*

Proof. We will verify 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.
- Calculation 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 polynomial number of bits. Thus, this can be performed in polynomial time.
- The maximum possible value for h is bounded by a polynomial given by Lemma 21. Indeed, the process described in the while loop is the same as the creation of unfolded path schema set Y_P. The only difference being that there exists only one possible run, if any and hence Y_P is a singleton set.

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

```
Build a resource R = ⟨X, T, B⟩ and a set of intervals I coherent with P and φ.
Compute f_init(i) for all i ∈ [0, len(p.l) − 1].
if for some i ∈ [0, len(p · l) − 1], f_init(i) is undefined then abort
For each term t = Σ_j a_j x_j ∈ T, curr(t) := Σ_j a_j.(effect(p · l)[j] + v_0[j]).
h := 1; A[1] := (f_init(len(p)), f_init(len(p) + 1) ··· f_init(len(p.l) − 1))
while A[h] is not final do
    Compute, nl_h = min{nl | i ∈ [1, len(l)], t ∈ T, val_curr(l^nl · l[1, i])(t) ≠ A[h](i)(t)}.
    h := h + 1
    A[h] := (m_1, m_2 ··· m_len(l)), such that at all positions i in l we have that
        val_curr(l^{nl_h} · l[1, i]) = m_i.
    For every term t = Σ_j a_j · x_j ∈ T, set curr(t) = curr(t) +
        Σ_j a_j.(nl_h.effect(l)[j]).
    if there is i ∈ [1, len(l)] such that A[h](i) ⊭ guard(l(i)) then abort
end while
For j ∈ [1, h − 1], T[j] := Min(nl_j, 2td(φ) + 5)
Check that proj((p × (f_init(0), ..., f_init(len(p) − 1)).(l × A[1])^{T[1]}.(l × A[2])^{T[2]} ... (l ×
    A[h − 1])^{T[h−1]}(l × A[h])^ω), 0 ⊨_symb φ
```

– Calculation of each nl_h requires computing val_{curr} which again involves arithmetical operations on polynomially many bits. Thus, this requires polynomial time only.
– Checking $(p \times (f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p) - 1)).(l \times A[1])^{T[1]}(l \times A[2])^{T[2]} \ldots (l \times A[h-1])^{T[h-1]}(l \times A[h])^\omega, 0 \models_{\mathsf{symb}} \phi$ can be done in polynomial time for the following reasons.
  - By definition of $T[h]$, size of $(p \times (f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p) - 1)).(l \times A[1])^{T[1]}(l \times A[2])^{T[2]} \ldots (l \times A[h-1])^{T[h-1]}(l \times A[h])^\omega$ is polynomial in the size of the input.
  - By [17], $(p \times (f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p) - 1)).(l \times A[1])^{T[1]}(l \times A[2])^{T[2]} \ldots (l \times A[h-1])^{T[h-1]}(l \times A[h])^\omega, 0 \models_{\mathsf{symb}} \phi$ can be checked in time $\mathcal{O}(\text{size}(\phi)^2 \times \text{len}(p \cdot l^{T[1]} l^{T[2]} \cdots l^{T[h-1]} l))$. Indeed, $\models_{\mathsf{symb}}$ is analogous to the satisfaction relation for plain Past LTL.

□

Lemma 27. *$P, c \models \phi$ iff Algorithm 2 on inputs P, c, ϕ has an accepting run.*

Proof. Let us first assume that $P, c \models \phi$. We will show that there exists a vector of positive integers $\boldsymbol{nL} = (nl_1, nl_2 \ldots nl_h)$ for some $h \in \mathbb{N}$ such that Algorithm 2 has an accepting run. Clearly, the transitions taken by a run ρ respecting P and satisfying ϕ is of the form, pl^ω. This can be decomposed in the form $pl^{nl_1} l^{nl_2} \ldots l^{nl_h} l^\omega$, depending on the portion of P traversed, 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 same as the one calculated by the algorithm. Now, the elements of $\boldsymbol{nL}$ can be exponential. But due to Lemma 13 and Stuttering theorem (Theorem 3), we know that, $(p \times (f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p) - 1)).(l \times A[1])^{nl_1}(l \times A[2])^{nl_2} \ldots (l \times A[h-1])^{nl_{h-1}}(l \times A[h])^\omega, 0 \models_{\mathsf{symb}} \phi$ iff $(p \times$

$(f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p)-1)).(l \times A[1])^{T[1]}(l \times A[2])^{T[2]} \ldots (l \times A[h-1])^{T[h-1]}$ $(l \times A[h])^{\omega}, 0 \models_{\mathsf{symb}} \phi$. Hence, the algorithm has an accepting run.

Now, we suppose that the algorithm has an accepting run on inputs P, c and ϕ. We will prove that $P, c \models \phi$. Since the algorithm has an accepting run, we assume the integers calculated by it are $nl_1, nl_2, \cdots, nl_h$. Let $w = pl^{nl_1}l^{nl_2} \ldots l^{nl_h}l^{\omega}$ and $\rho = \langle\langle q_0, \boldsymbol{m}_0\rangle, \boldsymbol{x_0}\rangle\langle\langle q_1, \boldsymbol{m}_1\rangle, \boldsymbol{x_1}\rangle\langle\langle q_2, \boldsymbol{m}_2\rangle, \boldsymbol{x_2}\rangle \cdots \in (Q' \times \mathbb{Z}^n)^{\omega}$ be defined as follows: for every $i \geq 0$, $q_i \stackrel{\text{def}}{=} \pi_1(\mathit{source}(w(i)))$, $\boldsymbol{x_0} \stackrel{\text{def}}{=} \boldsymbol{v_0}$ and for every $i \geq 1$, we have $\boldsymbol{x_i} \stackrel{\text{def}}{=} \boldsymbol{x_{i-1}} + \mathit{update}(w(i))$. By the calculation of l_j, $1 \leq j \leq n$, in the algorithm, it is easy to check that $\langle q_0, \boldsymbol{x_0}\rangle\langle q_1, \boldsymbol{x_1}\rangle\langle q_2, \boldsymbol{x_2}\rangle \cdots \in (Q \times \mathbb{Z}^n)^{\omega}$ is a run respecting P. Algorithm 2 guarantees that $(p \times (f_{\text{init}}(0), \ldots, f_{\text{init}}(\text{len}(p)-1)).(l \times A[1])^{T[1]}(l \times A[2])^{T[2]} \ldots (l \times A[h-1])^{T[h-1]}(l \times A[h])^{\omega}, 0 \models_{\mathsf{symb}} \phi$. And thus, by Lemma 13 and Theorem 3, we have, $\langle q_0, \boldsymbol{x_0}\rangle\langle q_1, \boldsymbol{x_1}\rangle\langle q_2, \boldsymbol{x_2}\rangle \cdots, 0 \models_{\mathsf{symb}} \phi$. □

I NP-hardness of reachability problem for $\mathcal{CPS}$

Proof. First we note that, a path schema in $\mathcal{CPS}$ can also be seen as a flat counter system with the additional condition of taking each loop at least once. 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: Given $S \in \mathcal{C}$ and two states q_0 and q_f, does there exist a finite run from $conf_0(q_0)$ to $conf_0(q_f)$? Below, we prove the NP-hardness for both REACH($\mathcal{CPS}$) and REACH($\mathcal{CFS}$). The proofs are by reduction from the SAT problem. Using the fact that $\mathcal{CPS}$ is a special and constrained $\mathcal{CFS}$, we will only prove NP-hardness of REACH($\mathcal{CPS}$) and hence, as a corollary, have the result for REACH($\mathcal{CFS}$). Let ϕ be a Boolean formula built over the propositional variables $AP = \{p_1, \cdots, p_n\}$. We build a path schema P such that ϕ is satisfiable iff there is a run respecting P starting with the configuration $conf_0(q_0)$ visits the configuration $conf_0(q_f)$. The path schema P is the one described in Figure 8 so that the truth of the propositional variable p_i is encoded by the fact that the loop incrementing x_i is visited at least twice. The guard g is defined as a

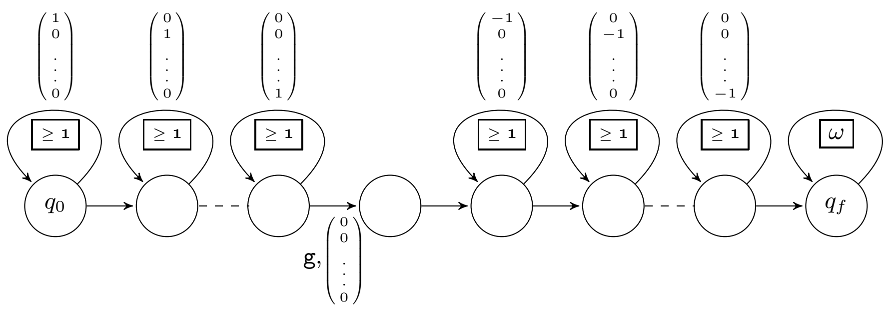


Fig. 8. A simple path schema

formula that establishes the correspondence between the truth value of p_i and the number of times the loop incrementing x_i is visited. It is defined from ϕ by replacing each occurrence of p_i by $\mathsf{x}_i \geq 2$. Note that, since the i^{th} and $(n+i)^{th}$ loops perform the complementary operation on the same counters, both of the loops can be taken equal number of times.

Let us check the correctness of the reduction. Let $v : AP \rightarrow \{\top, \bot\}$ be a valuation satisfying ϕ. Let us 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 $v(p_i) = \top$, 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 g is satisfied by the run and taking i^{th} 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 g ensures that the valuation v satisfies ϕ where for all $i \in [1, n]$, we have $v(p_i) = \top \overset{\text{def}}{\Leftrightarrow} iter_P(\rho)[i] \geq 2$. $\square$